Mathematical Study of Complex Networks: Brain, Internet, and Power Grid

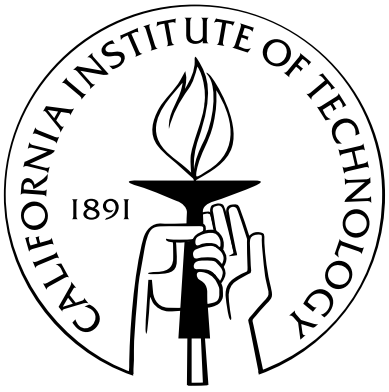
Thesis by

Somayeh Sojoudi

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2013

(Defended May 15, 2013)

To my husband, Javad Lavaei

Acknowledgements

I would like to express my deepest gratitude and appreciation to my PhD advisor, John Doyle, for his constant support and guidance throughout my PhD studies. I am very thankful to John for giving me the unique opportunity of working in his research group. John's passion for groundbreaking research has always inspired me to fearlessly work on fundamental research problems. I would also like to extend my gratitude to Richard Murray for his kind support at all stages of my academic life. My collaboration with him on fault-tolerant controller design was truly fruitful and enjoyable.

I am thankful to Steven Low for introducing me to the world of communication networks and for his generous support during the course of my studies. I sincerely thank Mani Chandi for being on my thesis committee. I am grateful to Jerrold Marsden for being a wonderful teacher and an inspiring role model. I would also like to express my gratitude to my Master's advisor, Amir Aghdam, for his constant encouragement and support.

I am thankful to all my colleagues and friends at Caltech who made my academic life exciting. In particular, I would like to thank my friends Nafiseh Khoram and Pooran Memari for making my time at Caltech much more enjoyable. I would also like to acknowledge the great help of Anissa Scott, Maria Lopez and the rest of administrative staff at CDS and ACM.

I am grateful to my family for their unconditional love and support. I am deeply indebted to my mother for being my best friend, teacher, and supporter, and to my brother for his kindness and encouragement. Finally, I would like to thank my wonderful husband and colleague, Javad Lavaei, whom this thesis is dedicated to. Certainly, this thesis would not have been completed without his generous help and support.

Abstract

The dissertation is concerned with the mathematical study of various network problems. First, three real-world networks are considered: (i) the human brain network (ii) communication networks, (iii) electric power networks. Although these networks perform very different tasks, they share similar mathematical foundations. The high-level goal is to analyze and/or synthesis each of these systems from a "control and optimization" point of view. After studying these three real-world networks, two abstract network problems are also explored, which are motivated by power systems. The first one is "flow optimization over a flow network" and the second one is "nonlinear optimization over a generalized weighted graph". The results derived in this dissertation are summarized below.

Brain Networks: Neuroimaging data reveals the coordinated activity of spatially distinct brain regions, which may be represented mathematically as a network of nodes (brain regions) and links (interdependencies). To obtain the brain connectivity network, the graphs associated with the correlation matrix and the inverse covariance matrix—describing marginal and conditional dependencies between brain regions—have been proposed in the literature. A question arises as to whether any of these graphs provides useful information about the brain connectivity. Due to the electrical properties of the brain, this problem will be investigated in the context of electrical circuits. First, we consider an electric circuit model and show that the inverse covariance matrix of the node voltages reveals the topology of the circuit. Second, we study the problem of finding the topology of the circuit based on only measurement. In this case, by assuming that the circuit is hidden inside a black box and only the nodal signals are available for measurement, the aim is to find the topology of the circuit when a limited number of samples are available. For this purpose, we deploy the graphical lasso technique to estimate a sparse inverse covariance matrix. It is shown that the graphical lasso may find most of the circuit topology if the exact covariance matrix is well-conditioned. However, it may fail to work well when this matrix is ill-conditioned. To

deal with ill-conditioned matrices, we propose a small modification to the graphical lasso algorithm and demonstrate its performance. Finally, the technique developed in this work will be applied to the resting-state fMRI data of a number of healthy subjects.

Communication Networks: Congestion control techniques aim to adjust the transmission rates of competing users in the Internet in such a way that the network resources are shared efficiently. Despite the progress in the analysis and synthesis of the Internet congestion control, almost all existing fluid models of congestion control assume that every link in the path of a flow observes the original source rate. To address this issue, a more accurate model is derived in this work for the behavior of the network under an arbitrary congestion controller, which takes into account of the effect of buffering (queueing) on data flows. Using this model, it is proved that the well-known Internet congestion control algorithms may no longer be stable for the common pricing schemes, unless a sufficient condition is satisfied. It is also shown that these algorithms are guaranteed to be stable if a new pricing mechanism is used.

Electrical Power Networks: Optimal power flow (OPF) has been one of the most studied problems for power systems since its introduction by Carpentier in 1962. This problem is concerned with finding an optimal operating point of a power network minimizing the total power generation cost subject to network and physical constraints. It is well known that OPF is computationally hard to solve due to the nonlinear interrelation among the optimization variables. The objective is to identify a large class of networks over which every OPF problem can be solved in polynomial time. To this end, a convex relaxation is proposed, which solves the OPF problem exactly for every radial network and every meshed network with a sufficient number of phase shifters, provided power over-delivery is allowed. The concept of "power over-delivery" is equivalent to relaxing the power balance equations to inequality constraints.

Flow Networks: In this part of the dissertation, the minimum-cost flow problem over an arbitrary flow network is considered. In this problem, each node is associated with some possibly unknown injection, each line has two unknown flows at its ends related to each other via a nonlinear function, and all injections and flows need to satisfy certain box constraints. This problem, named generalized network flow (GNF), is highly non-convex due to its nonlinear equality constraints. Under the assumption of monotonicity and convexity of the flow and cost functions, a convex relaxation is proposed, which always finds the optimal

injections. A primary application of this work is in the OPF problem. The results of this work on GNF prove that the relaxation on power balance equations (i.e., load over-delivery) is not needed in practice under a very mild angle assumption.

Generalized Weighted Graphs: Motivated by power optimizations, this part aims to find a global optimization technique for a nonlinear optimization defined over a generalized weighted graph. Every edge of this type of graph is associated with a weight set corresponding to the known parameters of the optimization (e.g., the coefficients). The motivation behind this problem is to investigate how the (hidden) structure of a given real/complex-valued optimization makes the problem easy to solve, and indeed the generalized weighted graph is introduced to capture the structure of an optimization. Various sufficient conditions are derived, which relate the polynomial-time solvability of different classes of optimization problems to weak properties of the generalized weighted graph such as its topology and the sign definiteness of its weight sets. As an application, it is proved that a broad class of real and complex optimizations over power networks are polynomial-time solvable due to the passivity of transmission lines and transformers.

Contents

List of Figures

List of Tables

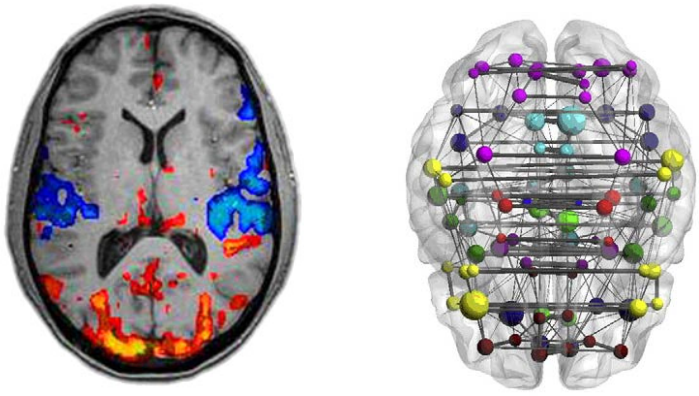
Chapter 1

Introduction

Real-world systems from human brains to the Internet to power systems are complex networks that can all be mathematically represented as abstract graphs. Although these networks perform very different tasks, they share similar mathematical foundations. The main goal of this PhD dissertation is to explore each of these systems from a "control and optimization" perspective. This thesis is composed of five chapters, where the two first chapters study the brain and communication networks, and the remaining chapters are concerned with three problems inspired by electrical power systems. In what follows, each of the problems studied in this work will be spelled out.

1.1 Modeling of Brain Connectivity Networks

Neuroimaging technologies such as structural MRI, functional MRI (fMRI) and EEG/MEG, allow for a non-invasive study of the structure and function of the human brain. This provides a great opportunity for understanding both healthy and disordered states of the brain as one of the most complex systems. Neuroimaging data reveals the coordinated activity of spatially distinct brain regions, which may be represented mathematically as a network of nodes (brain regions) and links (interdependencies). Figure (1.1) illustrates an image of the brain created from fMRI and also a graphical representation of the whole-brain functional connectivity. Various approaches have been proposed for assessing the functional connectivity (statistical dependencies) and effective connectivity (causal interactions) of the brain extracted from noisy and limited data, including general linear model [2], correlation thresholding [3], clustering [4, 5, 6], multivariate auto-regression [7, 8], dynamical causal modelling [9, 10, 11, 12], Bayesian network [13, 14] and sparse regression.



(a) (b)

Figure 1.1: (a) A brain image created from fMRI data (b) A graphical representation of the whole-brain functional network , borrowed from [1].

To assess the brain connectivity network, some methods such as the correlation matrix thresholding are based on the marginal dependencies among the random variables assigned to the brain regions, while some other techniques such as Bayesian networks and sparse regression methods are concerned with the conditional dependencies of Gaussian random variables obtained from the inverse covariance matrix. One may wonder whether the sparsity pattern of a correlation matrix or an inverse covariance matrix can partially or fully reveal the structure of the brain network. In Chapter 2, we study this problem in the context of electrical circuits as it is believed that the brain has certain electrical properties. To this end, we construct an electrical circuit (a resistive circuit) in which the resistors are assumed to be subject to thermal noise. We then show that the sparsity of the inverse covariance matrix (and not the correlation matrix) conforms with the circuit topology.

Assuming that the circuit is hidden inside a black box and only the nodal signals of the circuit are available for measurement, it is desirable to find the topology of the circuit given a limited set of measurements. We use the graphical lasso algorithm to estimate a sparse inverse covariance matrix. A challenge in using the graphical lasso (or other sparse regression techniques) is the choice of the regularization parameters. In this work, we choose this parameter in such a way that the sparsest solution is found while its corresponding graph is still connected. It is shown through some experiments that the graphical lasso may be able to find an estimated inverse covariance matrix revealing most of the circuit topology, provided the exact covariance matrix is well-conditioned. However, it may fail to work well when this matrix is ill-conditioned. To deal with ill-conditioned matrices, we propose a

small modification to the graphical lasso algorithm and show that the small change in the algorithm enables us to find most of the circuit topology. Eventually, we apply both the graphical lasso and the modified algorithms to the resting-state fMRI data acquired from three healthy subjects to estimate a connectivity graph for each subject. A comparison of the graphs of these subjects demonstrates that the modified graphical lasso outperforms the graphical lasso algorithm in the sense that the graphs obtained from the modified graphical lasso are noticeably sparser than the ones obtained from the graphical lasso.

1.2 Buffering Dynamics and Stability of Internet Congestion Control

In computer networks, queues build up when the input rates are larger than the available bandwidth. This causes congestion leading to packet loss and long delays. Congestion control techniques aim to adjust the transmission rates of competing users in such a way that the network resources are shared efficiently. The Internet congestion control has two main components: (i) transmission control protocol (TCP), (ii) active queue management (AQM). TCP adapts the sending rate (or window size) of each user in response to the congestion signal from its route, whereas AQM provides congestion information to the users by manipulating the packets on each router's queue.

Since the seminal works [15] and [16], a great deal of effort has been devoted to the modeling and synthesis of Internet congestion control. This is often performed for a fluid model of the network by solving a proper resource allocation problem in a distributed way. Different resource allocation algorithms, such as primal, dual, and primal-dual algorithms have been proposed in the literature, which enable every user to find its optimal transmission rate asymptotically using local feedback from the network. From a dynamical system perspective, each of these congestion control algorithms corresponds to an autonomous distributed system that is globally asymptotically stable, where its unique equilibrium point is a solution to the resource allocation problem [17, 18].

Despite the progress in the analysis and synthesis of Internet congestion control, an important modeling issue is often neglected for the sake of simplicity. Specifically, most existing fluid models of congestion control assume that all links in the path of a flow see the original source rate. Nonetheless, a fluid flow in practice is modified by the queueing

processes on its path, so that an intermediate link will generally not see the original source rate. Although it is possible to study the buffering effects for any given network through simulations, it is very advantageous to develop a fundamental theory for an arbitrary service discipline relating the buffering effects to various parameters of the network (say the routing matrix or the link capacities). Our goal is to derive a closed-form model for the buffer's dynamics, based on which the stability of congestion control algorithms can be deduced via simple conditions.

Chapter 3 of this dissertation studies the congestion control problem taking the buffering effect into account. To this end, a general model is derived to account for the time evolution of the buffer sizes. This model can be used for different service disciplines such as weighted fair queueing (WFQ) [19, 20] and first-in first-out (FIFO). The dual and primal-dual algorithms are studied, where the pricing mechanism is considered to be based on either queueing delays or queue sizes. It is shown that although these algorithms are stable when the buffering effect is ignored, they can become unstable otherwise. Several issues arising from the precise modeling of buffers are investigated. A new pricing mechanism is also proposed to guarantee the global stability of the dual and primal-dual algorithms.

1.3 Network Topologies with Zero Duality Gap for Optimal Power Flow

The optimal power flow (OPF) problem is concerned with finding an optimal operating point of a power system, which minimizes a certain objective function such as power loss or generation cost subject to network and physical constraints. This optimization problem has been extensively studied since 1962 [21]. Due to the nonlinear interrelation among active power, reactive power and voltage magnitude, OPF is described by nonlinear equations and may have a nonconvex/disconnected feasibility region. Several algorithms have been proposed for solving this highly nonconvex problem, including linear programming, quadratic programming, nonlinear programming, Lagrange relaxation and interior point methods. In order to solve OPF more efficiently, different conic and convex relaxations have been proposed in the past decade [22, 23, 24].

Recently, it has been shown in [25] that the Lagrangian dual of OPF may be used to find a globally optimal solution of OPF for several power networks due to their physical

properties. The primary goal of Chapter 4 is to identify a broad class of networks over which every OPF problem can be solved in polynomial time. To this end, it is first shown that the dual of OPF could be simplified significantly, depending on the number of cycles in the graph of a power network. It is then proved that adding a certain controllable power electronic device, known as phase shifter, to certain lines of the power network has a noticeable effect on reducing the computational complexity of solving OPF. In particular, if a sufficient number of phase shifters are incorporated in the topology of the network, the OPF problem is guaranteed to be solvable in polynomial time, provided power over-delivery is allowed. This result implies that every network topology can be modified by the integration of phase shifters to make OPF solvable in polynomial time for all possible values of loads, physical limits and convex cost functions.

1.4 Convexification of Generalized Network Flow Problem

The minimum-cost flow problem aims to optimize the flows over a flow network that is used to carry some commodity from suppliers to consumers. In a flow network, there is an injection of some commodity at every node, which leads to two flows over each line (arc) at its endpoints. The injection—depending on whether it is positive or negative, corresponds to supply or demand at the node. The minimum-cost flow problem has been studied thoroughly for a lossless network, where the amount of flow entering a line equals the amount of flow leaving the line. However, since many real-world flow networks are lossy, the minimum-cost flow problem has also attracted much attention for generalized networks, also known as networks with gain [26, 27, 28]. In this type of network, each line is associated with a constant gain relating the two flows of the line through a linear function. From the optimization perspective, network flow problems are convex and can be solved efficiently unless there are discrete variables involved [29].

There are several real-world network flows that are lossy, where the loss is a nonlinear function of the flows. An important example is power distribution networks for which the loss over a transmission line (with fixed voltage magnitudes at both ends) is given by a parabolic function due to Kirchhoff's circuit laws [30]. The loss function could be much more complicated depending on the power electronic devices installed on the transmission line. To the best of our knowledge, there is no theoretical result in the literature on the

polynomial-time solvability of network flow problems with nonlinear flow functions, except in very special cases. Chapter 5 is concerned with this general problem, named generalized network flow (GNF). Note that the term "GNF" has already been used in the literature for networks with linear losses, but it corresponds to arbitrary lossy networks in this work.

GNF aims to optimize the nodal injections subject to flow constraints for each line and box constraints for both injections and flows. A flow constraint is a nonlinear equality relating the flows at both ends of a line. To solve GNF, we make the practical assumption that the cost and flow functions are all monotonic and convex. The GNF problem is still highly non-convex due to its equality constraints. Relaxing the nonlinear equalities to convex inequalities gives rise to a convex relaxation of GNF. It can be easily observed that solving the relaxed problem may lead to a solution for which the new inequality flow constraints are not binding. One may speculate that this observation implies that the convex relaxation is not tight. However, the objective of this chapter is to show that as long as GNF is feasible, the convex relaxation is tight. More precisely, the convex relaxation always finds the optimal injections (and hence the optimal objective value), but probably produces wrong flows leading to non-binding inequalities. However, once the optimal injections are obtained at the nodes, a feasibility problem can be solved to find a set of feasible flows corresponding to the injections. Note that the reason why the convex relaxation does not necessarily find the correct flows is that the mapping from flows to injections is not invertible. The main contribution of this chapter is to show that although GNF may be NP-hard (since the flow equations can have an exponential number of solutions), the optimal injections can be found in polynomial time.

Energy-related optimizations with embedded power flow equations can be regarded as nonlinear network flow problems, which are analogous to GNF. The results derived in this chapter for a general GNF problem lead to the generalization of the result of Chapter 4 of this dissertation to networks with virtual phase shifters. This proves that in order to use the SDP relaxation for OPF over an arbitrary power network, it is not necessary to relax power balance equalities to inequality constraints under a very mild angle assumption.

1.5 Semidefinite Relaxation for Nonlinear Optimization over Graphs

Several classes of optimization problems, including polynomial optimization and quadratically-constrained quadratic program (QCQP) as a special case, are nonlinear/non-convex and NP-hard in the worst case. Due to the complexity of such problems, several convex relaxations based on linear matrix inequality (LMI), semidefinite programming (SDP) and second-order cone programming (SOCP) have gained popularity [31, 29]. These techniques enlarge the possibly non-convex feasible set into a convex set characterizable via convex functions, and then provide the exact or a lower bound on the optimal objective value.

The SDP relaxation converts an optimization with a vector variable to a convex optimization with a matrix variable, via a lifting technique. The exactness of the relaxation can then be interpreted as the existence of a low-rank (e.g., rank-1) solution for the SDP relaxation. Several papers have studied the existence of a low-rank solution to matrix optimizations with linear and LMI constraints [32, 33]. The papers [34] and [35] provide an upper bound on the lowest rank among all solutions of a feasible LMI problem. A rank-1 matrix decomposition technique is developed in [36] to find a rank-1 solution whenever the number of constraints is small. This technique is extended in [37] to the complex SDP problem. The paper [38] presents a polynomial-time algorithm for finding an approximate low-rank solution.

Cahpter 6 is motivated by the fact that real-world optimization problems are highly structured in many ways and their structures could in principle help reduce the computational complexity. For example, transmission lines and transformers used in power networks are passive devices, and as a result optimizations defined over electrical power networks have certain structures which distinguish them from abstract optimizations with random coefficients. The high-level objective of this chapter is to understand how the computational complexity of a given nonlinear optimization is related to its (hidden) structure.

This chapter is concerned with a broad class of nonlinear real/complex optimization problems, including QCQP. The main feature of this class is that the argument of each objective and constraint function is quadratic (as opposed to linear) in the optimization variable and the goal is to use three conic relaxations (SDP, reduced SDP and SOCP) to convexify the argument of the optimization. In this chapter, the structure of the nonlinear

optimization is mapped into a generalized weighted graph, where each edge is associated with a weight set constructed from the known parameters of the optimization (e.g., the coefficients). First, it is shown that the proposed relaxations are exact for real-valued optimizations, provided a set of conditions is satisfied. These conditions need each weight set to be sign definite and each cycle of the graph has an even number of positive weight sets. It is also shown that if some of these conditions are not satisfied, the SDP relaxation is guaranteed to have a rank-2 solution for weakly cyclic graphs, from which an approximate rank-1 solution may be recovered. To study the complex-valued case, the notion of "sign-definite complex weight sets" is introduced and it is then proved that the relaxations are exact for a complex optimization if the graph is acyclic with sign definite weight sets (with respect to complex numbers). The complex case is further studied for general graphs and it is proved that if the graph can be decomposed as the union of some edge-disjoint subgraphs in such a way that each subgraph possesses one of the four proposed structural properties, then the SDP relaxation is tight. As an application of this work in optimization for power systems, it is also shown that a broad class of energy optimizations can be convexified due to the physics of power networks.

Chapter 2

Modeling of Brain Connectivity Networks

Various neuro-imaging technologies combined with different mathematical tools are available to model the brain functional connectivity. The brain connectivity graph is often constructed based on either a correlation matrix or an inverse covariance (also known as concentration) matrix, which describe the marginal and conditional dependencies between the random variables associated with the brain regions, respectively. A question of interest is: Which of these matrices could reveal the topology of the brain network? Due to the electrical properties of the brain, we investigate this question in the context of circuits. In particular, we construct an electric circuit for which the inverse covariance matrix reveals the topology of the circuit. Assuming that the circuit is hidden inside a black box and only the nodal signals are available for measurement, the aim is to find the topology of the circuit when a limited number of samples are available. For this purpose, we deploy the graphical lasso technique to estimate a sparse inverse covariance matrix. It is shown that the graphical lasso might be able to find an estimated inverse covariance matrix revealing most of the circuit topology, provided the exact covariance matrix (not the sample covariance) is well-conditioned. However, it may fail to work well when the exact covariance matrix is ill-conditioned. To deal with ill-conditioned matrices, we modify the graphical lasso algorithm and show that the small change in the algorithm enables us to find the topology of the circuit even when the number of samples is very low. Finally, we apply both the graphical lasso and the modified algorithm to the resting-state fMRI data of three different healthy subjects and show that the graphs obtained from the modified graphical lasso are sparser than the ones obtained from the graphical lasso algorithm.

2.1 Introduction

A variety of different imaging technologies such as structural MRI, functional MRI (fMRI) and EEG/MEG, are available to study brain functional/effective connectivity. Neuroimaging data reveals the coordinated activity of spatially distinct brain regions, which may be represented mathematically as a network/graph of nodes (brain regions) and links (interdependencies). Various approaches have been proposed for assessing functional and effective connectivity of the brain using noisy and limited data, including general linear model [2], correlation thresholding [3], clustering [4, 5, 6], multivariate auto-regression [7, 8], dynamical causal modelling [9, 10, 11, 12], Bayesian networks [13, 14], and sparse regression.

In the *general linear model*, data is assumed to be a linear combination of explanatory variables (also known as predictors) plus error or noise which is assumed to follow a Gaussian distribution. The explanatory variables are assumed to have known shapes but their weights (coefficients) are unknown and need to be estimated (see [39] for a review of the general linear model and its application to fMRI data). *Correlation thresholding* directly examines the correlation of image values between pairs of voxels. These correlations are then thresholded to some pre-specified level to reveal the statistically significant connections. *Clustering technique* attempts to form clusters of voxels whose values over time (or over subject) are similar. This is closely related to the correlation thresholding method, since thresholding correlations simply clusters together all voxels whose similarity exceeds a threshold value.

Given multiple time series data, consecutive measurements contain information about the process that has generated it. *Multivariate auto-regression modeling* is a technique that can describe this underlying order by modeling the vector of current values of all variables as a linear sum of previous values. *Dynamical causal modelling* is based upon a bilinear approximation to neuronal dynamics. In this method, each brain region is assumed to have at least one state variable which is considered as a summary of neuronal activity in that region. This activity induces a hemodynamic response which is described by an extended Balloon model [40]. Unlike other techniques such as auto-regressive that assume inputs are unknown and stochastic, dynamic causal modelling assumes the inputs to be known.

A *Bayesian network* is a graphical model for stochastic processes, encoding the conditional independence/dependence relationships among some random variables with a directed acyclic graph. A Bayesian network can be visualized as a graph whose nodes denote brain

regions and whose directed edges denote connections between the regions. Lasso [41] is one of the most popular *sparse regression* techniques that exploits the sparsity enforcing property of l_1 regularization to shrink the most irrelevant or redundant features to zero. Graphical lasso is another method proposed by [42] to estimate sparse undirected graphical models using the lasso penalty (l_1 regularization). For a given sample covariance matrix which is computed from data that is assumed to follow a Gaussian distribution, graphical lasso estimates a sparse inverse covariance matrix by minimizing the negative log-likelihood of the data distribution over the space of positive definite matrices while imposing an l_1 penalty on the covariance matrix.

To assess the brain connectivity network, some methods such as correlation thresholding are proposed for encoding the marginal independence/dependence relationships among random variables while some other techniques such as Bayesian networks and sparse regression techniques aim to show the conditional dependencies of Gaussian random variables using inverse covariance. A question that may arise is: which of these methods provides better information about the structure of the brain network? We study this problem in the context of electrical circuits as it is believed that the brain has certain electrical properties. To this end, we construct an electrical circuit (a resistive circuit) in which the resistors are assumed to be subject to thermal noise. We then show that the sparsity of the inverse covariance matrix (and not the correlation matrix) conforms with the circuit topology.

Assuming that the circuit is hidden inside a black box and only the nodal signals of the circuit are available for measurement, the problem of finding the circuit topology given a limited set of measurements is studied next. We use the graphical lasso algorithm to estimate a sparse inverse covariance matrix. A challenge in using graphical lasso (or other sparse regression techniques) is the choice of the regularization parameters. In this work, we choose this parameter in such a way that the sparsest solution is found while its corresponding graph is still connected. It will be shown through some experiments that graphical lasso may be able to find an estimated inverse covariance matrix which reveals most of the circuit topology, provided the exact covariance matrix (not the sample covariance) is well-conditioned. However, it may fail to work well when the exact covariance matrix is ill-conditioned. To deal with ill-conditioned matrices, we propose a small modification to the graphical lasso algorithm and show that the small change in the algorithm enables us to find the topology of the circuit even when only a small number of samples is available.

We apply both the graphical lasso and the modified algorithm to the resting-state fMRI data of three different healthy subjects. A comparison of the graphs of these subjects shows that the modified graphical lasso outperforms the graphical lasso algorithm in the sense that the graphs obtained from the modified graphical lasso are noticeably sparser than the ones obtained from graphical lasso.

2.2 Mapping of Data into Graphs

Let $y_1, y_2, ..., y_n$ denote n scalar random variables representing the brain activity in n disparate regions. One can regard the brain as an interconnected system S composed of n interacting subsystems $S_1, S_2, ..., S_n$, where y_i represents the output of S_i for $i = 1, 2, ..., n$. Let R and Σ denote the correlation matrix and covariance matrix of these random variables, respectively. The graph associated with the matrices R and Σ is identical, and is called the *correlation graph*. Moreover, the graph associated with the inverse covariance matrix Σ^{-1} (or alternatively R^{-1}) is called the *concentration (partial correlation) graph*.

Assume that $y_1, ..., y_n$ have been sampled N times. The objective is to discover the interrelationship between subsystems $S_1, S_2, ..., S_n$ (i.e., the n brain regions) from the given data. This problem can be tackled from two different perspectives: (i) control theory, (ii) statistics. Control theory regards this problem as "system identification", where a static or dynamic n-channel system is designed whose output matches the measurements up to an acceptable level of error. This reverse engineering problem is challenging because no *a priori* knowledge of the structure of the brain system is available and in addition the system is subject to unknown noise and disturbances. Instead of fully characterizing S, statistics deals with the weaker, yet very important, problem of studying which subsystems in S affect each other directly. In other words, the goal is to identify the graph topology of the interconnected system modeling the brain.

To solve the latter problem, one can easily compute a sample covariance matrix Σ_s for the n-dimensional random variable $\begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T$. As an attempt to visualize the brain connectivity network, some methods (e.g., correlation thresholding) investigate the graph corresponding to Σ_s, while some other techniques (e.g., Bayesian networks or sparse regression) explore Σ_s^{-1} instead of Σ_s. The graphs obtained from Σ_s and Σ_s^{-1} approximate the correlation and concentration graphs, respectively.

Since any given data can be mapped into multiple graphs, a question arises: What graph preserves the structural properties of the brain? Alternatively, it is desirable to discover whether the topology of the interconnected system S modeling the brain can be fully or partially recovered from the sparsity pattern of either Σ_s or Σ_s^{-1}. In Section 2.3, we answer this question in the context of electrical circuits, where it will be shown that the sparsity pattern of the concentration graph has rich information. Before preceding to the circuit model section, we briefly review the graphical lasso algorithm that can be used to estimate a sparse concentration graph from a given sample covariance matrix.

2.2.1 Concentration Graph

The concentration graph is based on the non-zero entries of Σ^{-1}. The main motivation behind the introduction of this graph is that the entries of the inverse covariance matrix (known as concentration matrix) show the conditional (as opposed to marginal) dependencies of Gaussian random variables. For a given sample covariance matrix Σ_s, the graphical lasso estimates a sparse inverse covariance matrix by minimizing the negative log-likelihood of the data distribution over the space of positive definite matrices while imposing an l_1 penalty on the matrix solution. This optimization is as follows:

$$\begin{aligned} \min_{S} \quad & \text{trace}(S\Sigma_s) - \log(\det(S)) + \alpha\|S\|_1 \\ \text{subject to:} \quad & S \succeq 0 \end{aligned} \tag{2.1}$$

where α is the regularization parameter and S is a matrix variable that plays the role of Σ^{-1}. The optimization variable S is a symmetric matrix, $\|.\|_1$ denotes the element wise l_1-norm and $\succeq$ denotes the matrix positive semi-definite sign. The graph associated with S is an estimate of the concentration graph, which depends on the regularization parameters α. In the next section, this technique will be applied to some synthetic data derived from an electrical circuit.

2.3 Circuit Model

Consider a resistive circuit (network) composed of n nodes, $m+n$ resistors and the ground, where each node of the circuit is connected to the ground via a resistor and there are m

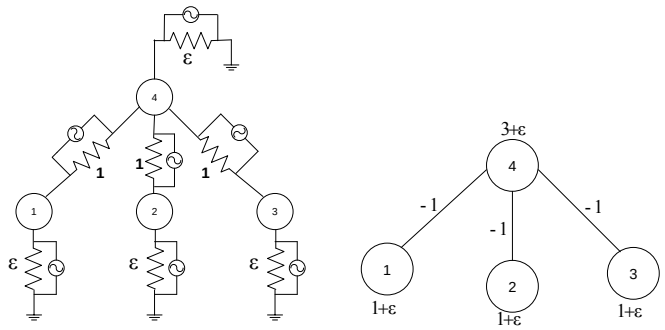
resistors connecting the nodes of the network. Suppose that every node of the network is connected to an external device, which is able to exchange electrical current with the network. Assume that every resistor is subject to thermal noise, namely Johnson-Nyquist (J-N) noise. One common method for modeling the J-N noise is to replace a non-ideal resistor with an ideal resistor in parallel with a current source whose value is white noise. Figure 2.1(a) exemplifies the model of a noisy circuit for $n = 4$ and $m = 3$. Let V denote the vector of the voltages seen at nodes $1, ..., n$ of the circuit, which can be regarded as a random variable. Consider the admittance matrix of this circuit, denoted as Y (see the appendix for the definition of this matrix). The matrix Y has the property that its sparsity pattern is the same as the topology of the circuit. On the other hand, as shown in the appendix, the covariance of the voltage vector V, denoted as Σ, is equal to Y^{-1}. By assuming that the circuit under study has a sparse structure, it can be concluded that:

- Σ is generically a dense matrix, being the inverse of the sparse matrix Y.
- Σ^{-1} is sparse and more importantly its sparsity conforms with the circuit topology.

This example illustrates the fact that the topology of a system may have been encoded in the inverse covariance matrix.

Now, assume that the circuit under study is inside a black box hiding the topology of the circuit, while the nodes of the circuit are available for measuring nodal signals. The question of interest is: Can measuring the node voltages help recover the circuit topology? To address this problem, one can sample the vector V multiple times and construct a sample covariance matrix Σ_s. Due to the error $\Sigma_s - \Sigma$, the inverse of Σ_s may not reveal the circuit topology. Another challenge is that Σ_s may not be invertible due to the lack of enough samples. This is usually true for some neuroimaging data such as fMRI data as a result of limited acquisition time. Hence, the question of interest is how to estimate a *sparse* inverse covariance matrix from Σ_s (note that Σ_s^{-1}, if it exists, is normally non-sparse due to the error $\Sigma_s - \Sigma$). To address this problem, a powerful technique is to use the graphical lasso algorithm. We have done extensive simulations on this algorithm in the context of the circuit problem posed above and made the following observations:

- Graphical lasso may find a sparse covariance matrix that reveals most of the topology of the circuit provided the exact covariance matrix Σ is well-conditioned. Note that the well conditioning of Σ highly depends on the values of the resistors in the circuit.



(a) (b)

Figure 2.1: (a) The resistive circuit studied in Example 1. (b) The concentration graph representing the inverse covariance matrix.

- If the exact covariance matrix Σ is ill-conditioned, graphical lasso may fail to find a sparse inverse covariance matrix. This issue can be fixed by modifying the graphical lasso algorithm via an extra term.

The above results will be elaborated here in a simple example. Consider the star circuit depicted in Figure 2.1(a). In this circuit, nodes 1, 2 and 3 are connected to node 4 (named central node) via three resistors with values of 1 . In addition, each node is grounded via a resistor with value ϵ (this will allow the exact covariance matrix Σ to be invertible). Note that since each resistor is subject to thermal noise by assumption, it has been replaced by an ideal resistor in parallel with a current source whose value is white noise. The admittance matrix Y of this circuit is:

$$Y = \begin{pmatrix} 1+\epsilon & 0 & 0 & -1 \\ 0 & 1+\epsilon & 0 & -1 \\ 0 & 0 & 1+\epsilon & -1 \\ -1 & -1 & -1 & 3+\epsilon \end{pmatrix}, \tag{2.2}$$

The concentration graph representing the inverse covariance matrix $\Sigma^{-1} = Y$ is depicted in Figure 2.1(b) (the nonzero entries of Σ^{-1} are shown on the corresponding nodes and edges of the graph). It is easy to verify that $\Sigma = Y^{-1}$ is dense even though $\Sigma^{-1} = Y$ is sparse. Now, consider the problem of finding the topology of the circuit through voltage measurements. Given ϵ and a sample covariance matrix Σ_s, the graphical lasso algorithm (2.1) will be used to find an estimate of Σ^{-1}. The condition number (i.e., the ratio of the largest singular value of the matrix to the smallest one) of Σ is equal to $\frac{4+\epsilon}{\epsilon}$. Therefore, Σ

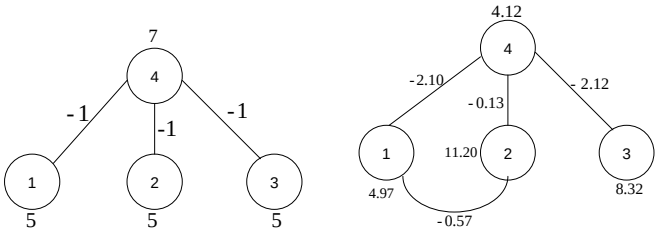


Figure 2.2: (a) The concentration graph obtained from the exact inverse covariance matrix. (b) The estimated concentration graph from 4 samples using graphical lasso algorithm.

will be ill-conditioned for a small ϵ (say $\epsilon = 0.01$), while for a large ϵ (say $\epsilon = 4$) this matrix is well-conditioned. Given a sample covariance matrix Σ_s, the goal is to understand how much of the topology could be revealed using optimization (2.1) in both ill-conditioned and well-conditioned cases.

I) Well-conditioned Σ

Consider ϵ as 4 and assume that Σ_s is constructed based on only 4 samples of the voltage vector V. Using optimization (2.1), a sparse invariance covariance can be estimated for an appropriate choice of the regularization parameter α. The exact concentration graph (the graph obtained from the exact covariance matrix Σ) and the graph which is obtained by solving optimization (2.1) for $\alpha = 0.05$ are depicted in Figure 2.2. The regularization parameter α is chosen in such a way that the obtained graph has the sparsest structure while the graph is still connected (i.e., there is a path from any point to any other point in the graph). To check the connectivity of the graph, we simply check the eigenvalues of the Laplacian matrix of the graph. The graph is not connected if its Laplacian matrix has more than one zero eigenvalue.

Comparing the exact graph with the estimated one in Figure 2.2, one can conclude that the estimated graph reveals most of the topology. More precisely, the graphical lasso algorithm (2.1) is able to detect all of the connections and has given only one extra (undesired) connection. By inspecting the numbers on the edges of the graph in Figure 2.2(a), it can be observed that the signs of the connections (corresponding to the nonzero off-diagonal elements of the estimated invariance covariance matrix) are also found correctly. This suggests that although the number of samples is small and hence taking the inverse of Σ_s directly does not provide useful information about the topology of the circuit (due to the error

$\Sigma - \Sigma_s$ being large), optimization (2.1) is able to reveal most of the topology. For $\epsilon = 4$ the condition number of Σ is equal to 2 and therefore it is a well-conditioned matrix. For a well-conditioned matrix, more accurate models can be obtained by increasing the number of samples.

II) Ill-conditioned Σ

Consider again the circuit depicted in Figure 2.1(a) and assume that $\epsilon = 0.01$. As mentioned earlier, a small ϵ results in an ill-conditioned matrix Σ. For instance, the condition number of Σ is equal to 401 for $\epsilon = 0.01$. In this case, the solution of optimization (2.1) is extremely sensitive to the value of the regularization parameter α. To explore this property, first consider the case where the error $\Sigma - \Sigma_s$ is zero. In this case, Σ_s is invertible and its inverse has a sparse structure due to the relation $\Sigma_s^{-1} = \Sigma^{-1} = Y$. Not surprisingly, the solution of optimization (2.1) becomes $\Sigma^{-1} = Y$ if the regularization parameter α is equal to zero. However, this solution quickly becomes dense for a small nonzero regularization parameter α. This issue is demonstrated in Figure 2.3. For $\epsilon = 0.01$ and $\Sigma_s = \Sigma$, Figure 2.3(a) shows the sparse concentration graph obtained from $\Sigma_s^{-1} = \Sigma^{-1} = Y$. Figures 2.3(b) and (c) show that increasing the regularization parameter α makes the sparse matrix Σ_s completely dense. In fact, increasing α makes the weights of the redundant (wrong) connections comparable to the weights of the true connections. This implies that the l_1 penalty term in the graphical lasso algorithm fails to enforce sparsity on the solution. This issue seems to be related to the ill conditioning of Σ. Similar issues with graphical lasso algorithm have also been observed in [43].

So far, it was assumed that $\Sigma - \Sigma_s = 0$. Obviously, the above-mentioned issue becomes worse when only a very limited number of samples are available. Since Σ_s can be rank deficient and optimization (2.1) may fail to find a sparse solution, the question of interest is how to find a sparse graph estimating the topology of the desired network in such cases. In the next subsection, we answer this question by modifying optimization (2.1).

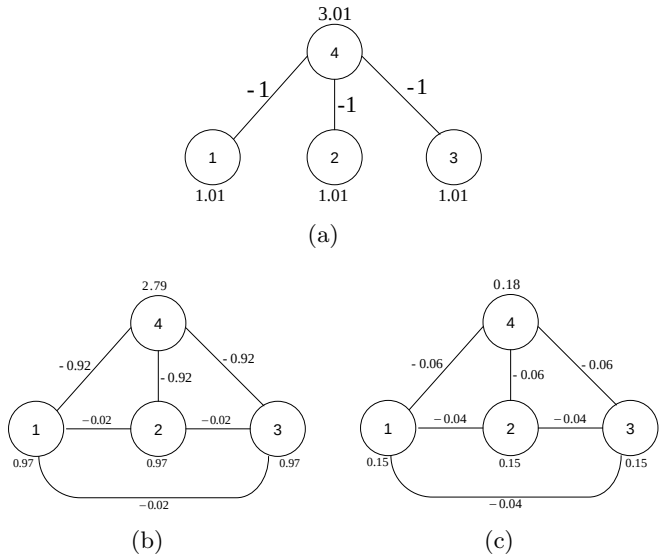


Figure 2.3: (a) The graph for Σ_s^{-1} in the case $\Sigma - \Sigma_s = 0$. (b) The estimated concentration graph obtained from optimization (2.1) for $\alpha = 0.01$. (c) The estimated concentration graph obtained from optimization (2.1) for $\alpha = 2$.

2.3.1 Modified Graphical Lasso

Consider the graphical lasso algorithm given in (2.1) and make a small modification to it as follows:

$$\begin{aligned} \min_{S} \quad & \text{trace}(S\Sigma_s) - \log(\det(S)) + \alpha\|S - \beta I\|_1 \\ \text{subject to:} \quad & S \succeq 0 \end{aligned} \tag{2.3}$$

where β is a positive scalar and I is an $n \times n$ identity matrix. Since $S - \beta I$ and S have the same off-diagonal entries, the last penalty term still aims to sparsify S. It is easy to verify that optimization (2.3) is equivalent to:

$$\begin{aligned} \min_{S} \quad & \text{trace}((S + \beta I)\Sigma_s) - \log(\det(S + \beta I)) + \alpha\|S\|_1 \\ \text{subject to:} \quad & S + \beta I \succeq 0 \end{aligned} \tag{2.4}$$

This implies that the modification of graphical lasso algorithm is based on adding a positive-definite matrix βI to each of the two terms in the log-likelihood function and the positivity constraint. As verified in extensive simulations, this modification reduces the sensitivity of the solution to the regularization parameter α and makes it possible to find a sparse solution independent of the conditioning of Σ.

To understand how well the modified algorithm (2.3) behaves, consider again the exam-

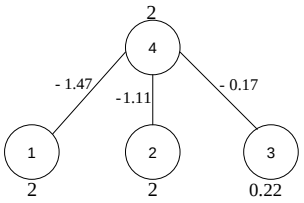


Figure 2.4: The estimated concentration graph obtained from the modified graphical lasso for $\alpha = 5.4$ and $\beta = 2$.

ple studied before in the ill-conditioned case for $\epsilon = 0.01$. Recall that the graph associated with the exact inverse covariance matrix Σ^{-1} is given Figure 2.3(a). Given 4 samples of the voltage vector V, it is desirable to estimate the topology of the circuit. Since Y is ill-conditioned, optimization (2.1) fails to find a sparse solution as discussed before. Therefore, we use the modified optimization (2.3) to estimate the structure of the circuit. The graph depicted in Figure 2.4 is obtained from optimization (2.3) for $\alpha = 5.4$ and $\beta = 2$. Comparing this graph with the exact graph given in Figure 2.3(a), the modified graphical lasso was clearly able to fully detect the topology of the circuit (note also the signs of the connections weights). An interesting observation is that although the exact covariance matrix Σ is ill-conditioned (remember that its condition number is equal to 401) and there is only a very small number of samples available, optimization (2.3) detects the right structure of the circuit. In the next section, the graphical lasso algorithm and its modified version will be applied to fMRI data for comparison.

2.4 FMRI data: Graphical Lasso vs. Modified Graphical Lasso

Consider the data set available in [44] in which resting state fMRI data was acquired for a group of 20 healthy subjects. 134 samples of the low frequency neurophysiological oscillations were taken at 140 cortical brain regions in the right hemisphere. The 140×140 sample covariance matrix Σ_s can be computed for each subject from this data set. Note that the number of samples is smaller than the number of variables, and therefore Σ_s is ill-conditioned and non-invertible. Figure 2.5 shows the 2-D picture of the 140 nodes (brain regions).

The aim of this section is to model the brain connectivity network using the graphical

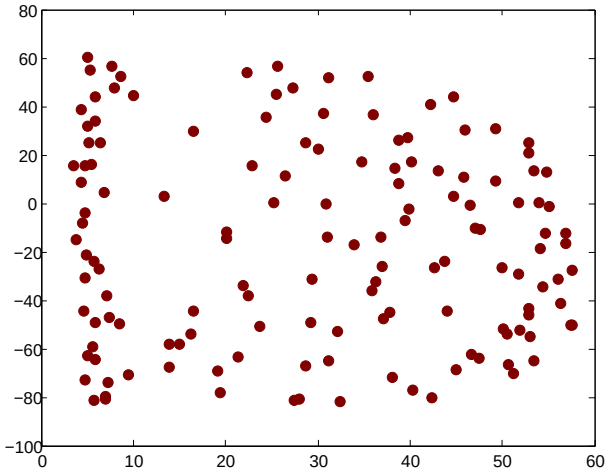


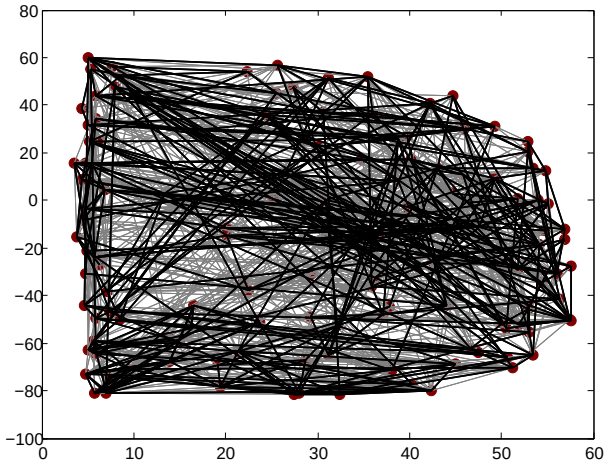
Figure 2.5: 2-D picture of the 140 brain regions.

lasso algorithm (2.1) and the modified graphical lasso (2.3). For both of these two optimization problems, the regularization parameter α is to be chosen in such a way that the resulting graph will have the sparsest possible structure and yet be a connected graph. We will solve optimizations (2.1) and (2.3) for resting-state fMRI data of three different subjects, called Subject 1, Subject 2 and Subject 3. The connectivity (concentration) graph of each subject will then be plotted as follows:

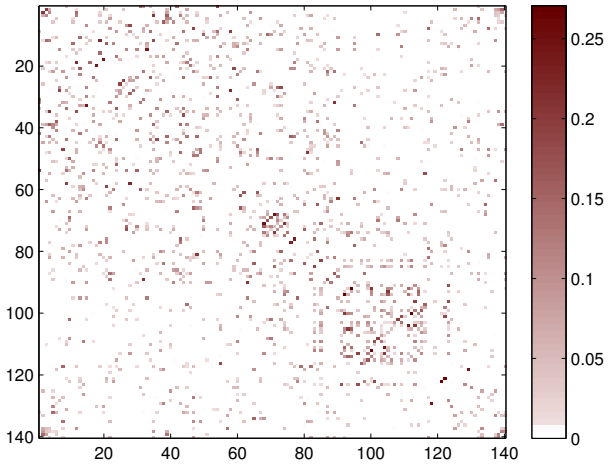
- The strong connections are plotted in black (we consider a connection strong if its weight is at least 10 times larger than average of the absolute values of all weights). Furthermore, the width of each black line in the graph represents the strength of the connection, which means stronger connections are shown with thicker lines.

- The rest of the edges (weak connections) are shown in gray.

As mentioned before, the graphs of the inverse correlation matrix (R^{-1}) and the inverse covariance matrix (Σ^{-1}) have the same sparsity pattern. Therefore, one may feed the sample correlation matrix Σ_s instead of the sample covariance matrix into the graphical lasso algorithm (2.1) and the modified optimization (2.3). Simulations on the fMRI data show that the graphs based on the sample correlation matrix are much sparser (by a factor of 2) than the ones based on the sample covariance matrix. Therefore, we substitute the sample covariance matrix Σ_s in optimizations (2.1) and (2.3) with the sample correlation matrix of the resting state fMRI data of each subject. Note that each of graphs obtained from these algorithms is an *estimated* concentration graph.



(a)



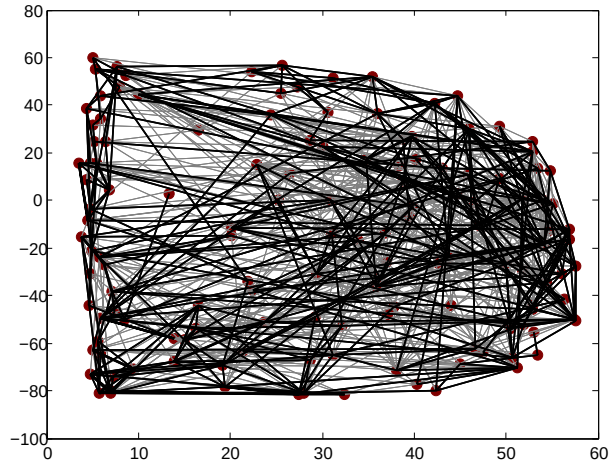
(b)

Figure 2.6: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 1 obtained from optimization (2.1) for $\alpha = 0.315$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.1) for Subject 1 after taking the absolute value of its elements.

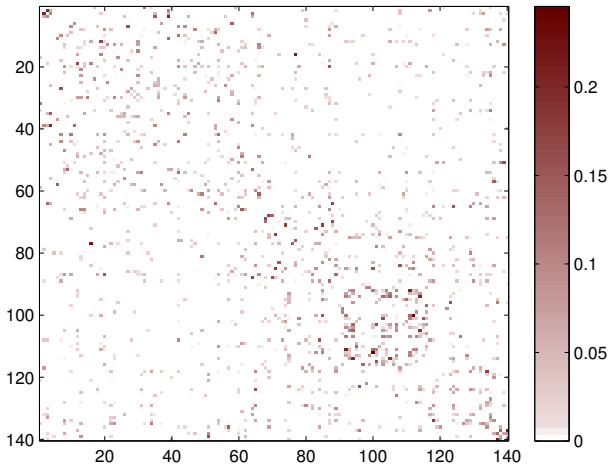
I) Graphical lasso and brain connectivity

Subject 1: For the resting-state fMRI data acquired from Subject 1, the sparsest connected graph that can be obtained from optimization (2.1) is depicted in Figure 2.6(a). This graph, corresponding to $\alpha = 0.315$, has 987 edges connecting the 140 spatially disjoint brain regions. The color map depicted in Figure 2.6(b) illustrates the sparseness of the off-diagonal entries of the matrix solution obtained from optimization (2.1) after taking the absolute value of the matrix elements.

Subject 2: For the resting-state fMRI data acquired from Subject 2, the sparsest connected graph that can be obtained from optimization (2.1) is depicted in Figure 2.7(a). This



(a)

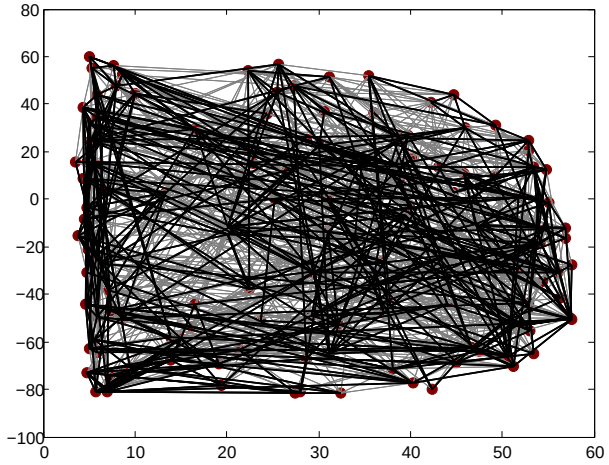


(b)

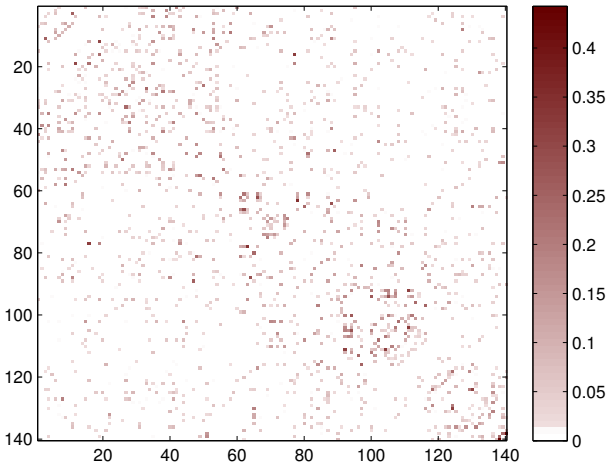
Figure 2.7: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 2 obtained from optimization (2.1) for $\alpha = 0.355$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.1) for Subject 2 after taking the absolute value of its elements.

graph, corresponding to $\alpha = 0.355$, has 764 edges. Figure 2.7(b) illustrates the sparseness of off-diagonal entries of the solution of optimization (2.1) after taking the absolute value of the matrix elements.

Subject 3: For the resting-state fMRI data of Subject 3, the sparsest connected graph that can be obtained from optimization (2.1) is depicted in Figure 2.8(a). This graph, corresponding to $\alpha = 0.275$, has 998 edges. Figure 2.8(b) illustrates the sparseness of off-diagonal entries of the solution of optimization (2.1) after taking the absolute value of the matrix elements.

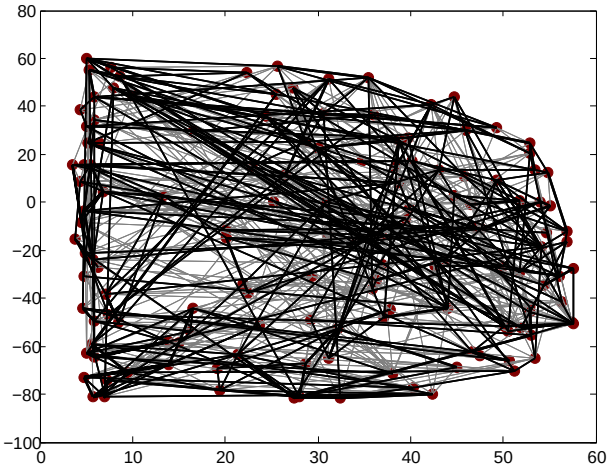


(a)

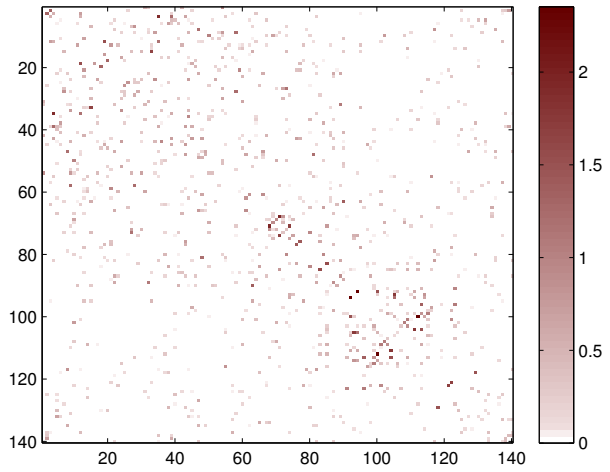


(b)

Figure 2.8: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 3 obtained from optimization (2.1) for $\alpha = 0.275$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.1) for Subject 3 after taking the absolute value of its elements.



(a)

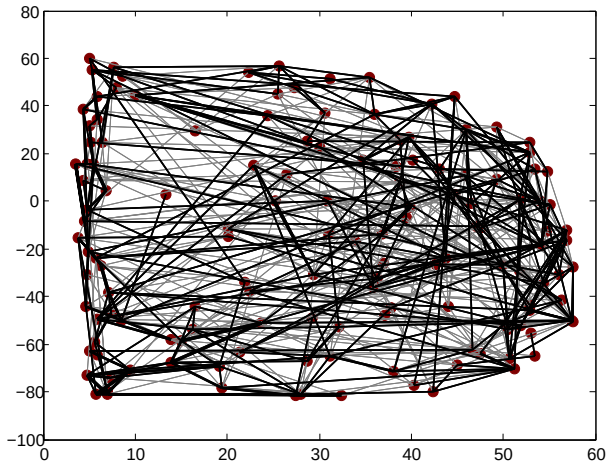


(b)

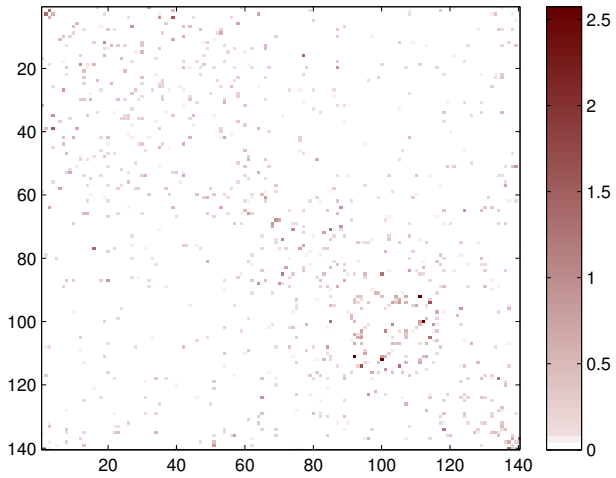
Figure 2.9: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 1 obtained from optimization (2.3) for $\alpha = 0.445$ and $\beta = 5$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.3) for Subject 1 after taking the absolute value of its elements.

II) Modified graphical lasso and brain connectivity

Subject 1: For the resting-state fMRI data acquired from Subject 1, the sparsest connected graph that can be obtained from optimization (2.3) is depicted in Figure 2.9(a). This graph, obtained for $\alpha = 0.315$ and $\beta = 5$, has 608 edges. This graph has 595 edges in common with the graph depicted in Figure 2.6(a) for the same subject but obtained by solving the graphical lasso algorithm (2.1). Figure 2.9(b) illustrates the sparseness of the off-diagonal entries of the solution of optimization (2.3) after taking the absolute value of its elements.



(a)

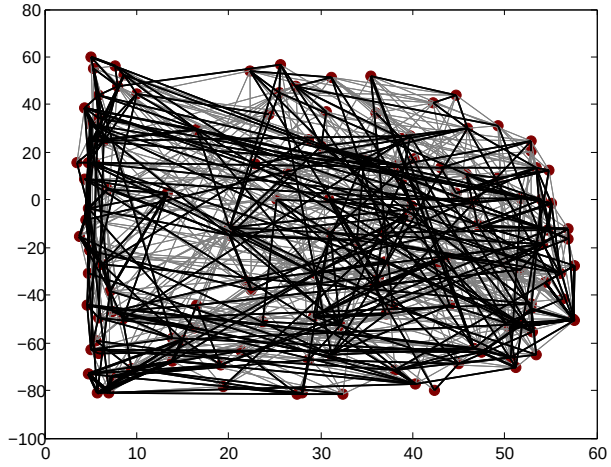


(b)

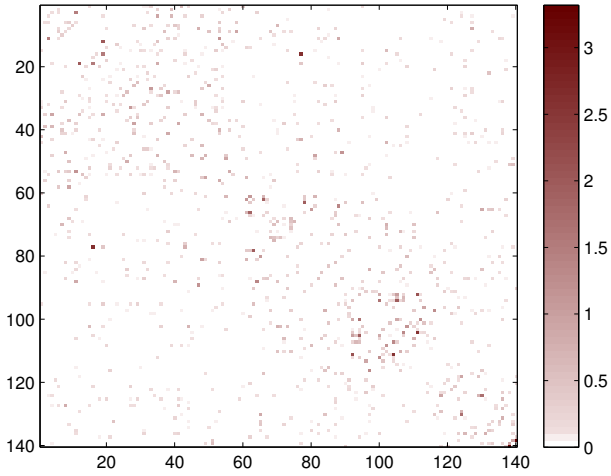
Figure 2.10: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 2 obtained from optimization (2.3) for $\alpha = 0.356$ and $\beta = 5$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.3) for Subject 2 after taking the absolute value of its elements.

Subject 2: For the resting-state fMRI data acquired from Subject 2, the sparsest connected graph that can be obtained from optimization (2.3) is depicted in Figure 2.10(a). This graph, obtained for $\alpha = 0.356$ and $\beta = 5$, has 540 edges and is a subset of the graph depicted in Figure 2.7(a) for the same subject but obtained by solving the graphical lasso algorithm (2.1). Figure 2.10(b) illustrates the sparseness of the off-diagonal entries of the solution of optimization (2.3) after taking the absolute value of its elements.

Subject 3: For the resting-state fMRI data acquired from Subject 3, the sparsest connected graph that can be obtained from optimization (2.3) is depicted in Figure 2.11(a). This graph, obtained for $\alpha = 0.275$ and $\beta = 5$, has 688 edges out of which 680 edges are



(a)



(b)

Figure 2.11: (a) The sparsest connected graph for the resting-state-fMRI data of Subject 3 obtained from optimization (2.3) for $\alpha = 0.275$ and $\beta = 5$. (b) The sparseness of the off-diagonal entries of the solution of optimization (2.3) for Subject 3 after taking the absolute value of its elements.

in common with the graph depicted in Figure 2.8(a) for the same subject but obtained by solving the graphical lasso algorithm (2.1). Figure 2.11(b) illustrates the sparseness of the off-diagonal entries of the solution of optimization (2.3) after taking the absolute value of its elements.

The above simulations are summarized in Table 2.1. In summary, the graph obtained from the modified graphical lasso for each subject is not only sparser than but also mostly a subgraph of the one obtained from the graphical lasso. It can be verified that the graphs of Subjects 1-3 obtained from the modified graphical lasso have 62 common edges. The common subgraph of these three graphs is shown in Figure 2.12.

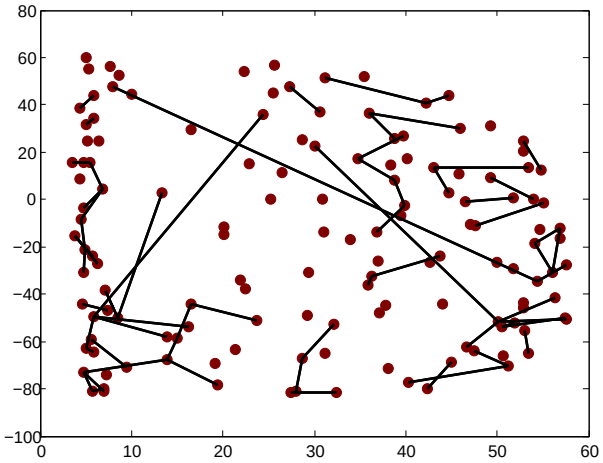


Figure 2.12: The 62 edges that are in common among the graphs of Subjects 1-3 obtained from optimization (2.3).

Subject	Optimization	α	β	Edges
1	(2.1)	0.315	0	987
1	(2.3)	0.315	5	608
2	(2.1)	0.355	0	764
2	(2.3)	0.356	5	540
3	(2.1)	0.275	0	998
3	(2.3)	0.275	5	688

Table 2.1: This table shows the number of edges for the graphs obtained from optimization (2.1) and optimization (2.3) for Subjects 1-3.

2.5 Summary

Two popular methods for assessing the brain functional connectivity are: (i) mapping the thresholded correlation matrix into a graph, which shows the marginal independence/dependence relationships among random variables, (ii) mapping the inverse covariance matrix into a graph, which shows the conditional dependencies of Gaussian random variables. The latter method is based on Bayesian networks and sparse regression. An important question arises as to which of these methods provides better information about the structure (topology) of the brain network. Due to the electrical properties of the brain, we study this problem in the context of circuits and show that the inverse covariance matrix reveals the topology of a circuit subject to thermal noise. We then use the graphical lasso technique to estimate a sparse inverse covariance matrix from the measurements taken from the circuit. It is shown that the graphical lasso algorithm may find an estimated inverse covariance matrix revealing most of the circuit topology, provided that the exact covariance matrix (not the sample covariance) is well-conditioned. It is also shown that this algorithm may fail to work satisfactorily when the exact covariance matrix is ill-conditioned. To deal with ill-conditioned matrices, we modify the graphical lasso algorithm and then show that the modified algorithm is able to find most of the topology of the circuit even in the case when a very limited number of samples are available. Finally, the graphical lasso algorithm and the modified algorithm are both applied to the resting-state fMRI data of three different healthy subjects. Simulations show that the graphs obtained from the modified graphical lasso are sparser than the ones obtained from the graphical lasso.

2.6 Appendix

To find the covariance of the voltage vector V corresponding to the circuit introduced in Section 2.3, we need to define the admittance matrix of the circuit. This matrix, denoted as Y, is an $n \times n$ matrix whose $(i,j)^{\text{th}}$ entry is given as follows:

- This entry is zero provided that $i \neq j$ and that nodes i and j are not directly connected (via a resistor) in the circuit.
- If $i \neq j$ and nodes i and j are directly connected in the circuit, then the $(i,j)^{\text{th}}$ entry of Y is equal to $-\frac{1}{z_{ij}}$, where z_{ij} denotes the value of the resistor between nodes i and

j.

- If $i = j$, then the $(i,j)^{\text{th}}$ entry of Y is equal to $\frac{1}{z_{ii}} + \sum_{k \in \mathcal{N}(i)} \frac{1}{z_{ik}}$, where z_{ii} denotes the value of the resistor connected to the ground at node i and $\mathcal{N}(i)$ denotes the set of the neighboring nodes of node i in the circuit.

Let N denote the vector of the currents injected to the nodes from the external devices. In order to relate N to V, we need to define some matrices:

- B: $n \times m$ incidence matrix associated with the circuit,
- Y_e: $m \times m$ diagonal edge admittance matrix,
- Y_g: $n \times n$ diagonal node-to-ground admittance matrix,
- W_e: m dimensional edge current unit white noise vector (corresponding to the series resistors in the circuit),
- W_g: n dimensional node-to-ground current unit white noise vector (corresponding to the shunt resistors in the circuit).

It turns out that $Y = BY_eB^T + Y_g$ and

$$N = B\sqrt{Y_e}W_e + \sqrt{Y_g}W_g \tag{2.5}$$

(a constant $2KT$ has been removed from the above modeling to simplify the presentation). In the circuit, N and V are related to one another through the relation $V = Y^{-1}N$. The vector N can be regarded as the noisy input of the network, which makes V a random variable. It follows from the equation (2.5) and $V = Y^{-1}N$ that the covariance of V is equal to Y^{-1}.

Chapter 3

Buffering Dynamics and Stability of Internet Congestion Control

Many existing fluid-flow models of the Internet congestion control algorithms ignore the effects of buffers on the data flows for simplicity. In particular, they assume that all links in the path of a flow are able to see the original source rate. However, a fluid flow in practice is modified by the queueing processes on its path, so that an intermediate link will generally not see the original source rate. In this chapter, a more accurate model is derived for the behavior of the network under a congestion controller, which takes into account of the effect of buffering on output flows. It is shown how this model can be deployed for some well-known service disciplines such as first-in first-out and generalized weighted fair queueing. Based on the derived model, the dual and primal-dual algorithms are studied under the common pricing mechanisms, and it is shown that these algorithms can become unstable. Sufficient conditions are provided to guarantee the stability of the dual and primal-dual algorithms. Finally, a new pricing mechanism is proposed under which these congestion control algorithms are both stable.

3.1 Introduction

In computer networks, queues build up when the input rates are larger than the available bandwidth. This causes congestion leading to packet loss and long delays. Congestion control techniques aim to adjust the transmission rates of competing users in such a way that the network resources are shared efficiently. Internet congestion control has two main components: (i) transmission control protocol (TCP) and (ii) active queue management

(AQM). TCP adapts the sending rate (or window size) of each user in response to the congestion signal from its route, whereas AQM provides congestion information to the users by manipulating the packets on each router's queue. DropTail and random early detection (RED) are two examples of AQM schemes [45]. TCP-Reno [46, 47], TCP-NewReno [48] and SACK TCP [49] are different versions of TCP congestion control protocols that have been deployed in the Internet. Mathematical models of these algorithms can be found in [50, 51, 52]. Congestion control protocols are either based on explicit feedback (which requires explicit communications between sources and links) or implicit feedback (which only requires end-to-end communications). For instance, packet loss in TCP Reno and queueing delay in TCP Vegas [53] are two congestion signals that can be provided to the users without needing explicit communication. However, the explicit congestion notification (ECN), as an extension of TCP, allows that each router writes some congestion information in the IP header of a packet and then the congestion signal is explicitly communicated to the users [54].

Since the seminal works [15, 16], a great deal of effort has been devoted to the modeling and synthesis of Internet congestion control. This is often performed for a fluid model of the network by solving a proper resource allocation problem in a distributed way. Different resource allocation algorithms, such as primal, dual and primal-dual algorithms, have been proposed in the literature, which enable every user to find its optimal transmission rate asymptotically using local feedback from the network. From a dynamical system perspective, each of these congestion control algorithms corresponds to an autonomous distributed system that is globally asymptotically stable, where its unique equilibrium point is a solution to the resource allocation problem [17, 18].

Despite the progress in the analysis and synthesis of Internet congestion control, an important modeling issue is often neglected for the sake of simplicity. Specifically, most existing fluid models of congestion control assume that all links in the path of a flow see the original source rate. Nonetheless, a fluid flow in practice is modified by the queueing processes on its path, so that an intermediate link will generally not see the original source rate. Reference [55] acknowledges such buffering effects on TCP/AQM networks, incorporating the model in [56] to account for the nature in which competing flows pass through congested links. In [57] and [58], the linear stability of such networks was analyzed for RED and PI AQM. Reference [59] proposes a form of deterministic nonlinear dynamic queue model and

studies how instability of deterministic fluid flow models for congestion control analysis leads to a significant increase in the variance of the flow in stochastic networks. Although it is possible to study the buffering effects for any given network through simulations, it is very advantageous to develop a fundamental theory for an arbitrary service discipline relating the buffering effects to various parameters of the network (say the routing matrix or the link capacities). Our goal is to derive a closed-form model for the buffers dynamics, based on which the stability of congestion control algorithms can be deduced via simple conditions.

The main objective of this chapter is to study congestion control taking this buffering effect into account. To this end, a general model is derived to account for the time evolution of the buffer sizes. This model can be used for different service disciplines such as weighted fair queueing (WFQ) [19, 20] and first-in first-out (FIFO). Then, the dual and primal-dual algorithms are studied, where the pricing mechanism is considered to be based on either queueing delays or queue sizes. It is shown that although these algorithms are stable when the buffering effect is ignored, they can become unstable otherwise. Several issues arising from the precise modeling of buffers are investigated here. A new pricing mechanism is also proposed to guarantee the global stability of dual and primal-dual algorithms.[1]

3.2 Preliminaries and Existing Models

Consider a network with the set $\mathcal{L} := \{1, ..., L\}$ of unidirectional links, where each link $l \in \mathcal{L}$ has a finite capacity c_l. Assume that the network is shared by a set $\mathcal{S} := \{1, ..., S\}$ of sources such that each source $s \in \mathcal{S}$ is identified by an origin, a destination and a fixed route. Let x_s denote the transmission rate associated with source $s \in \mathcal{S}$, $\mathcal{L}(s)$ denote the collection of the links belonging to the route of source $s \in \mathcal{S}$, and $\mathcal{S}(l)$ denote the set of those sources whose route passes through link $l \in \mathcal{L}$. Moreover, let R be the *routing matrix* of the network, defined as an $L \times S$ matrix whose (l, s) entry ($l \in \mathcal{L}$ and $s \in \mathcal{S}$) is 1 if link l belongs to the route of source s and is 0 otherwise. In addition, define $\mathbf{c}$ as the vector of the link capacities $c_1, ..., c_L$. Suppose that each source $s \in \mathcal{S}$ is accompanied by a utility function $U_s : \mathbf{R} \to \mathbf{R}$, which is assumed to be continuously differentiable, strictly concave and increasing. The utility function $U_s(x_s)$ specifies the benefit that source s gains for data

[1] See [60] for the preliminary results of this work.

transmission at an arbitrary rate x_s.

From the mathematical perspective, the objective of the congestion control is to allocate network resources among the users in an optimal way:

$$\max_{x_s \geq 0} \quad \sum_{s \in \mathcal{S}} U_s(x_s) \tag{3.1}$$

subject to the network constraints

$$\sum_{s \in \mathcal{S}(l)} x_s \leq c_l, \quad \forall l \in \mathcal{L}. \tag{3.2}$$

Define the aggregate flow rate y_l, the route price q_s and the Lagrangian $\mathbf{L}(\mathbf{x}, \mathbf{p})$ as

$$y_l := \sum_{s \in \mathcal{S}(l)} x_s, \quad l \in \mathcal{L}, \tag{3.3a}$$

$$q_s := \sum_{l \in \mathcal{L}(s)} p_l, \quad s \in \mathcal{S}, \tag{3.3b}$$

$$\mathbf{L}(\mathbf{x}, \mathbf{p}) := \sum_{s \in \mathcal{S}} U_s(x_s) - \sum_{l \in \mathcal{L}} p_l(y_l - c_l), \tag{3.3c}$$

where $\mathbf{p}$ is the vector of the Lagrange multipliers $p_1, ..., p_L$ and $\mathbf{x}$ is the vector of the transmission rates $x_1, ..., x_S$. The Karush-Kuhn-Tucker (KKT) optimality conditions for the resource allocation problem can be written as

$$U'(x_s) - q_s \leq 0 \quad \text{with equality if } x_s > 0, \tag{3.4a}$$

$$p_l(y_l - c_l) = 0, \tag{3.4b}$$

$$y_l - c_l \leq 0, \tag{3.4c}$$

$$x_s \geq 0, \quad p_l \geq 0, \tag{3.4d}$$

for all $s \in \mathcal{S}$ and $l \in \mathcal{L}$. Suppose that the matrix R has full row rank. Under this assumption, the KKT conditions (3.4) have a unique solution $(\mathbf{x}^*, \mathbf{p}^*)$. Solving (3.4) in a centralized way is not feasible because the routing matrix R and the utility functions $U_1(x_1), ..., U_S(x_S)$ may not be known globally. Hence, different distributed/decentralized protocols have been proposed in the literature to solve the KKT conditions, each of which enables every user to obtain its optimal transmission rate asymptotically by receiving local feedback from

the network. Mathematically, a typical distributed congestion control algorithm can be expressed as

$$\dot{x}_s(t) = f_s(x_s(t), q_s(t)) \quad \forall s \in \mathcal{S}, \tag{3.5a}$$

$$\dot{p}_l(t) = g_l(p_l(t), y_l(t)) \quad \forall l \in \mathcal{L}, \tag{3.5b}$$

for some functions $f_1, ..., f_S$ and $g_1, ..., g_L$. Since $x_s(t), q_s(t)$ are local information for user s and $p_l(t), y_l(t)$ are local information for link l, this algorithm is naturally distributed. Note that although it is assumed that all users and links employ dynamic local controllers in the above algorithms to update their corresponding transmission rates and link prices, one can replace some of them with static local controllers. A special type of the controller (3.5) is referred to as the "primal-dual controller", which has the control law

$$\dot{x}_s(t) = k_s(x_s(t))(U_s'(x_s(t)) - q_s(t)), \qquad \forall s \in \mathcal{S}, \tag{3.6a}$$

$$\dot{p}_l(t) = h_l(p_l(t))(y_l(t) - c_l)^+_{p_l(t)}, \qquad \forall l \in \mathcal{L}, \tag{3.6b}$$

where k_s, h_l are some non-decreasing continuous functions and $(\cdot)^+_a$ is the positive projection operator, i.e.

$$(y_l(t) - c_l)^+_{p_l(t)} = \begin{cases} y_l(t) - c_l & p_l(t) > 0 \\ \max(y_l(t) - c_l, 0) & p_l(t) = 0. \end{cases} \tag{3.7}$$

Note that a projection operator can be incorporated into the dynamics of $x_s(t)$ in (3.6a) to ensure the nonnegativity of the transmission rates; however, we do not consider such an operator here for simplicity. Under mild assumptions, the above dynamical control system has the unique equilibrium point $(\mathbf{x}^*, \mathbf{p}^*)$ that is globally asymptotically stable [51]. This implies that if each user $s \in \mathcal{S}$ updates its rate based on (3.6a) by starting from an arbitrary initial rate $x_s(0)$ and each link $l \in \mathcal{L}$ adjusts its price using (3.6b) by commencing from any initial price $p_l(0)$, then the transmission rate $x_s(t)$ converges to the optimal value x_s^* as t goes to infinity. This interesting property also holds for the dual congestion control algorithm, which is as follows [51]:

$$x_s(t) = U_s'^{-1}\big(q_s(t)\big), \qquad \forall s \in \mathcal{S}, \tag{3.8a}$$

$$\dot{p}_l(t) = h_l(p_l(t))(y_l(t) - c_l)^+_{p_l(t)}, \qquad \forall l \in \mathcal{L}. \tag{3.8b}$$

In order for the dual and primal-dual algorithms to work, each source should know the sums of the link prices in its route. This, in general, requires communication from routers to sources. Similarly, communication from sources to links might also be necessary for the above algorithms. Due to the overheads of such communications, it is desirable to choose $h_l(p_l(t))$ in such a way that the communication between links and sources is obviated. Two particular values for $h_l(p_l(t))$ which achieve this goal have been studied in this literature, 1 and $\frac{1}{c_l}$. These values for the weighting parameter $h_l(p_l(t))$ make the price $p_l(t)$ have very interesting properties under the simplifying assumption that the dynamics of the buffer of link l is governed by:

$$\dot{b}_l(t) = (y_l(t) - c_l)^+_{b_l(t)} \tag{3.9}$$

where $b_l(t)$ denotes the queue length at time t. More precisely, under the above assumption, two important scenarios can be considered as follows:

- *First scenario:* [53, 61, 62]

  Let $h_l(p_l(t))$ be $\frac{1}{c_l}$ for every $l \in \mathcal{L}$. In this case, the price $p_l(t)$ is the same as the queueing delay at link l, and therefore $q_s(t)$ is equal to the aggregative queueing delay for source s. As a result, $q_s(t)$ can be approximated from the round-trip time without having to receive explicit feedback from links.

- *Second scenario:* [45, 63, 64]

  Let $h_l(p_l(t))$ be equal to K for every $l \in \mathcal{L}$, where K is a given positive constant. In this case, the price $p_l(t)$ can be interpreted as $p_l(t) = Kb_l(t)$, where $b_l(t)$ denotes the queue length of link l at time t. To connect this case to the existing protocols, consider an AQM algorithm (e.g. RED), which drops or marks every packet at a router's queue with a certain probability depending on the queue length (say $Kb_l(t)$). The price $p_l(t)$ represents the marking/dropping probability for RED and interestingly $q_s(t)$ can be reported to source s using a binary feedback.

As can be observed from the above scenarios, two important pricing mechanisms for congestion control algorithms are queue sizes and queueing delays. There is a common belief that the dual and primal-dual algorithms based on these pricing mechanisms are stable for a fluid model of the network because the systems (3.6) and (3.8) are globally asymptotically stable. However, it is essential to note that the unrealistic modeling assumption (3.9) has

been used to draw this conclusion. These algorithms would be stable if the rate of a fluid flow at the input of every link on its path were the same as its (source) rate at the input of its first link. To be more precise, an exact model for the buffer size $b_l(t)$ is as follows;

$$\dot{b}_l(t) = (\tilde{y}_l(t) - c_l)^+_{b_l(t)}, \tag{3.10}$$

rather than the one given in (3.9), where $\tilde{y}_l(t)$ denotes the incoming flow rate of buffer l. This is due to the buffering process at each link.

Most existing fluid models do not consider the above-mentioned fact that the flow rate of every user changes along its path due to the presence of buffers. Motivated by this shortcoming, the objective of this work is to study the stability of the dual and primal-dual algorithms with buffer-size and queueing-delay pricing mechanisms, where the dynamics of buffers are taken explicitly into account. In particular, we show that congestion control based on a realistic model of buffers leads to a hard problem with unexpected results on the stability of the network.

3.3 Modeling of Buffer Occupancies

For simplicity and with no loss of generality, assume that every link of the network is unidirectional so that all flows over each link go in the same direction. Under this assumption, every link has only one buffer as opposed to two buffers at its endpoints. For every $s \in \mathcal{S}, l \in \mathcal{L}, t \geq 0$, let the following notations be introduced:

- $x_{ls}(t)$: Input rate associated with source s at the buffer of link l at time t.
- $b_{ls}(t)$: Backlog associated with source s at link l at time t.
- $b_l(t) = \sum_{s \in S(l)} b_{ls}(t)$: Aggregate backlog at link l at time t.
- $\tilde{y}(t) = \sum_{s \in S(l)} x_{ls}(t)$: Aggregate flow rate at the input of the buffer of link l.
- $g(s, j)$: The j^{th} link in the path of source s, for every $j \in \{1, ..., |\mathcal{L}(s)|\}$.
- $q(s, l)$: A natural number showing the position of link l in the path of source s if $l \in \mathcal{L}(s)$.

Note that $x_{ls}(t)$ and $b_{ls}(t)$ are both zero if $s \notin \mathcal{S}(l)$. Given $l \in \mathcal{L}$, since the buffer of link l is shared by its incoming flows, it is useful to specify at what *relative* rates different flows leave the buffer. To this end, define the parameter $\theta_{ls}(t)$ via the equation

$$\dot{b}_{ls}(t) = \theta_{ls}(t)\dot{b}_l(t), \quad \forall l \in \mathcal{L},\ s \in \mathcal{S},\ t \geq 0. \tag{3.11}$$

Notice that $\theta_{ls}(t)$ is contingent upon the queueing strategy deployed by the routers. There are two special, nonetheless important, cases in which $\theta_{ls}(t)$ is a constant or a function of $\mathbf{x}(t)$, where $\mathbf{x}(t)$ denotes the set of the source transmission rates $x_s(t)$, $\forall s \in \mathcal{S}$. To emphasize the type of $\theta_{ls}(t)$, the notations θ_{ls} and $\theta_{ls}(\mathbf{x}(t))$ will be used throughout this chapter for these two cases. $\theta_{ls}(t)$ plays a significant role in modeling the time evolution of buffer sizes.

3.3.1 Parameter $\theta_{ls}(t)$ for Different Service Disciplines

Before proceeding with the dynamics of buffers, let $\theta_{ls}(t)$ be calculated for some well-known service disciplines, namely WFQ and FIFO. To simplify the derivation of $\theta_{ls}(t)$, it will be implicitly assumed in this subsection that $b_{ls}(t)$ and $x_{ls}(t)$ are both nonzero at a given time t for all $s \in \mathcal{S}(l)$.

- *WFQ*:

  Recall that $x_{ls}(t)$, $\forall s \in \mathcal{S}(l)$, are $|\mathcal{S}(l)|$ active flows entering the buffer of link l with the capacity c_l at time t. Let $w_{ls}(t)$, $\forall s \in \mathcal{S}(l)$, denote some arbitrary weights associated with each of these flows. The output flow rate at the link l corresponding to source s is equal to

  $$\frac{w_{ls}(t)}{\sum_{s' \in \mathcal{S}(l)} w_{ls'}(t)} c_l \tag{3.12}$$

  This implies that the output flow rate associated with each source is proportional to the link rate c_l. To derive the equation for $\theta_{ls}(t)$, we consider a special, nonetheless interesting, case where $w_{ls}(t) = x_{ls}(t), \forall s \in \mathcal{S}, l \in \mathcal{L}$. We call this scheme special WFQ (S-WFQ). To find $\theta_{li}(\mathbf{x}(t))$ corresponding to the S-WFQ scheduling, one can

write

$$\begin{aligned}\dot{b}_{ls}(t) &= x_{ls}(t) - \frac{w_{ls}(t)}{\sum_{s'\in\mathcal{S}(l)} w_{ls'}(t)} c_l \\ &= x_{ls}(t) - \frac{x_{ls}(t)}{\sum_{s'\in\mathcal{S}(l)} x_{ls'}(t)} c_l.\end{aligned} \tag{3.13}$$

On the other hand, the time evolution of the aggregate backlog at link l is given by

$$\dot{b}_l(t) = \sum_{s'\in\mathcal{S}(l)} x_{ls'}(t) - c_l. \tag{3.14}$$

It follows from (3.13) and (3.14) that

$$\theta_{ls}(t) = \frac{\dot{b}_{ls}(t)}{\dot{b}_l(t)} = \frac{x_{ls}(t)}{\sum_{s'\in\mathcal{S}(l)} x_{ls'}(t)}, \quad \forall s \in \mathcal{S}(l). \tag{3.15}$$

Now, note that the output flow at each link l corresponding to any arbitrary flow is a function of the input flows of the link (due to (3.12) and $w_{ls}(t) = x_{ls}(t)$), and moreover the input flow rates at a link are the same as the output rates of its previous links. This implies that one can write recursive algebraic equations to obtain each $x_{ls'}(t)$ as a function $\mathbf{x}(t)$ (although this function can be complicated). As a result of this fact and (3.15), $\theta_{ls}(t)$ is indeed state dependent and therefore the notation $\theta_{ls}(\mathbf{x}(t))$ can be used for its representation. The scheme S-WFQ will be studied in detail later in this chapter. Note that the same methodology used above can be deployed to derive $\theta_{ls}(t)$ for a more general WFQ scheme.

- *FIFO*:

  In the FIFO scheduling, the data flows are served in a first-come, first-served basis. In other words, whatever comes in first is handled first, and what comes in next waits until the first is finished. Consider again the flows $x_{ls}(t)$, $\forall s \in \mathcal{S}(l)$, entering the buffer of link l with the capacity c_l. To simplify the formulation, assume that the buffer has always been nonempty up to the time t, which implies that

$$\int_0^t \Big(\sum_{s'\in\mathcal{S}(l)} x_{ls'}(\sigma) \Big) d\sigma \geq t \cdot c_l. \tag{3.16}$$

  Note that the left side of the above inequality shows the amount of data that entered the buffer of link l in the time duration $[0, t]$, whereas its right side indicates the

amount of data that left the buffer in that period. Define $\tau(t)$ as a nonnegative number satisfying the equation

$$\int_0^{t-\tau(t)} \Big(\sum_{s' \in \mathcal{S}(l)} x_{ls'}(\sigma) \Big) d\sigma = t \cdot c_l. \tag{3.17}$$

The interpretation behind the parameter $\tau(t)$ is as follows: the data leaving the buffer of link l at time t had arrived at the buffer at time $t - \tau(t)$ and therefore was subject to the delay $\tau(t)$. Taking the time derivatives of both sides of (3.17) yields

$$\frac{d(t-\tau(t))}{dt} \times \Big(\sum_{s' \in \mathcal{S}(l)} x_{ls'}(t-\tau(t)) \Big) = c_l. \tag{3.18}$$

On the other hand, the backlog at link l from source s can be obtained as

$$b_{ls}(t) = \int_{t-\tau(t)}^{t} x_{ls}(\sigma) d\sigma, \qquad \forall s \in \mathcal{S}(l). \tag{3.19}$$

The time evolution of the buffer at link l corresponding to source s can be found by taking the time derivative of the above equation and combining it with (3.18). This leads to the relation

$$\begin{aligned} \dot{b}_{ls}(t) &= x_{ls}(t) - x_{ls}(t-\tau(t)) \times \frac{d(t-\tau(t))}{dt} \\ &= x_{ls}(t) - x_{ls}(t-\tau(t)) \times \frac{c_l}{\sum_{s' \in \mathcal{S}(l)} x_{ls'}(t-\tau(t))}. \end{aligned} \tag{3.20}$$

Moreover, the aggregate backlog at link l is given by

$$\dot{b}_l(t) = \sum_{s' \in \mathcal{S}(l)} x_{ls'}(t) - c_l. \tag{3.21}$$

It follows immediately from (3.22) and (3.21) that $\theta_{ls}(t)$ is equal to

$$\theta_{ls}(t) = \frac{x_{ls}(t) - \frac{x_{ls}(t-\tau(t))}{\sum_{s' \in \mathcal{S}(l)} x_{ls'}(t-\tau(t))} \cdot c_l}{\sum_{s' \in \mathcal{S}(l)} x_{ls'}(t) - c_l}. \tag{3.22}$$

It can be seen from (3.13) and (3.20) that $\dot{b}_{ls}(t)$ had the same expression for both S-WFQ and FIFO if the delay term $\tau(t)$ were zero. However, this term has a significant role, as witnessed by the non-trivial definition (3.17), which makes $\theta_{ls}(t)$ very different

for FIFO and S-WFQ.

3.3.2 Dynamics of Buffer Sizes

In the preceding subsection, it was explained how $\theta_{ls}(t)$ can be obtained for some well-known service disciplines. Given $\theta_{ls}(t)$ associated with some arbitrary service discipline, the objective of this subsection is to study the evolution of the buffer sizes in time. More precisely, the goal is to relate the buffer sizes directly to the original source rates $x_s(t)$, $\forall s \in \mathcal{S}$, rather than the intermediate rates $x_{ls}(t)$, $\forall l \in \mathcal{L},\ s \in \mathcal{S}$.

Definition 1. *For every $t \geq 0$, define $R(\Theta(t))$ as an $L \times S$ matrix whose (l, s) entry is equal to $\theta_{ls}(t)$ for every $l \in \mathcal{L}$ and $s \in \mathcal{S}$.*

Definition 2. *For every $t \geq 0$, define $\Phi(t)$ as an $L \times L$ matrix with the (l_1, l_2) entry equal to $\phi_{l_1 l_2} = \sum \theta_{l_2 s}(t)$ (for every $l_1, l_2 \in \mathcal{L}$), where the sum is taken over all sources s that pass first through link l_2 and then through link l_1 not necessary immediately. In particular, all diagonal entries of $\Phi(t)$ are equal to 1.*

Some remarks regarding the above definitions:

- The matrix $R(\Theta(t))$ inherits its structure from the routing matrix R, meaning that if an entry of R is zero, the corresponding entry of $R(\Theta(t))$ is also zero.
- The matrix $\Phi(t)$ specifies the effect of each buffer on the remaining buffers. Indeed, the (l_1, l_2) entry of $\Phi(t)$ shows what portion in the rate of change of buffer l_2 corresponds to the flows passing first through l_2 and then through l_1.

In this work, we assume that the matrix $\Phi(t)$ is nonsingular for every $t \geq 0$. As will be shown later in Remark 2, this assumption is always satisfied for an important subclass of routing matrices.

Theorem 1. *The buffer occupancies satisfy the differential equation*

$$\dot{\mathbf{b}}(t) = \Big(\big(I - \Phi(t)\big)\dot{\mathbf{b}}(t) + R\mathbf{x}(t) - \mathbf{c} \Big)^{+}_{\mathbf{b}(t)}, \quad \forall t \geq 0 \tag{3.23}$$

where $\mathbf{b}(t)$ denotes the vector of queue sizes $b_1(t), ..., b_L(t)$. In particular, if the vector $\mathbf{b}(t)$ is strictly positive, then

$$\dot{\mathbf{b}}(t) = \Phi(t)^{-1} \left(R\mathbf{x}(t) - \mathbf{c} \right). \tag{3.24}$$

Proof: For every $s \in \mathcal{S}$ and $j \in \{1, 2, ..., |\mathcal{L}(s)| - 1\}$, one can write

$$\dot{b}_{g(s,j)s}(t) = x_{g(s,j)s}(t) - x_{g(s,j+1)s}(t). \tag{3.25}$$

Adding up the above equations over j yields that

$$x_{g(s,k)s}(t) = x_s(t) - \sum_{j=1}^{k-1} \dot{b}_{g(s,j)s}(t), \quad k \in \{2, ..., |\mathcal{L}(s)|\},$$

where, by convention, the sum in the right side of the above equation is considered as zero if $k = 1$. Hence, given $l \in \mathcal{L}$, it holds that

$$\begin{aligned}
\tilde{y}_l(t) &= \sum_{s \in \mathcal{S}(l)} x_{ls}(t) \\
&= \sum_{s \in \mathcal{S}(l)} \left(x_s(t) - \sum_{j=1}^{q(s,l)-1} \dot{b}_{g(s,j)s}(t) \right) \\
&= \sum_{s \in \mathcal{S}(l)} \left(x_s(t) - \sum_{j=1}^{q(s,l)-1} \theta_{g(s,j)s}(t) \dot{b}_{g(s,j)}(t) \right) \\
&= \sum_{s \in \mathcal{S}(l)} x_s(t) - \sum_{l' \in \mathcal{L} \setminus \{l\}} \phi_{ll'}(t) \dot{b}_{l'}(t).
\end{aligned} \tag{3.26}$$

By defining $\tilde{\mathbf{y}}(t)$ as the vector of the link rates $\tilde{y}_1(t), ..., \tilde{y}_l(t)$, it can be concluded from the above equation that

$$\tilde{\mathbf{y}}(t) = R\mathbf{x}(t) + (I - \Phi(t))\dot{\mathbf{b}}(t). \tag{3.27}$$

As a result,

$$\begin{aligned}
\dot{\mathbf{b}}(t) &= \big(\tilde{\mathbf{y}}(t) - \mathbf{c}\big)^+_{\mathbf{b}(t)} \\
&= \left(\big(I - \Phi(t)\big)\dot{\mathbf{b}}(t) + R\mathbf{x}(t) - \mathbf{c} \right)^+_{\mathbf{b}(t)}.
\end{aligned} \tag{3.28}$$

It follows immediately from the above nonlinear differential equation that if $\mathbf{b}(t)$ is strictly positive, then $\dot{\mathbf{b}}(t)$ can be obtained from the equation (3.24). This completes the proof. ■

Theorem 1 describes how all buffer occupancies evolve in time. As explained in Section II, the existing results for a fluid model of the network simplify the model of a buffer size from $\dot{b}_l(t) = (\tilde{y}_l(t) - c_l)^+_{b_l(t)}$ to $\dot{b}_l(t) = (y_l(t) - c_l)^+_{b_l(t)}$. It can be shown that this approximation amounts to replacing the matrix $\Phi(t)$ in (3.24) with the identity matrix, which

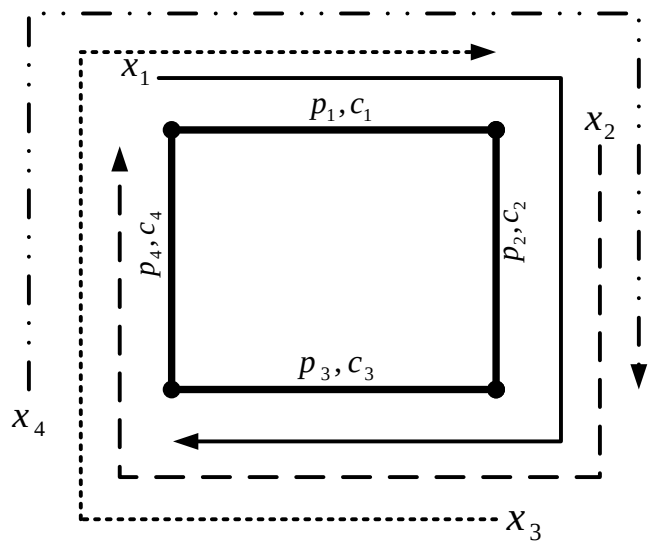


Figure 3.1: Network studied in Example 1.

might be a very poor approximation at times. Before studying the underlying differential equation for $\mathbf{b}(t)$, we illustrate Theorem 1 with an example.

Example 1: Consider the 4-edge ring network depicted in Figure 3.1 consisting of 4 flows, where each flow passes through 3 consecutive links. For every $l \in \{1, 2, 3, 4\}$, assume that $\theta_{ls}(t)$, $\forall s \in \mathcal{S}(l)$, are all constant and equal to each other. Assume also that the capacity of each link is normalized to 1. The matrix $\Phi(t)$ for this network can be obtained as

$$\Phi(t) = \begin{bmatrix} 1 & 0 & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & 1 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & 1 & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix}. \tag{3.29}$$

It can be concluded from Theorem 1 that if the vector $\mathbf{b}(t)$ is strictly positive at a time $t \geq 0$, then $b_1(t)$ evolves according to the following differential equation

$$\dot{b}_1(t) = \frac{9}{8}x_1(t) - \frac{3}{8}x_2(t) + \frac{3}{8}x_3(t) + \frac{3}{8}x_4(t) - \frac{1}{2}. \tag{3.30}$$

This equation implies that flows $1, 2, 3, 4$ contribute to the buffer size at link 1 with the factors $\frac{9}{8}, -\frac{3}{8}, \frac{3}{8}, \frac{3}{8}$, respectively. Interestingly, the contribution of flow 2 to the buffer size of link 1 is negative, meaning that the size of buffer 1 decreases if the flow rate of source 2

increases.

The above example demonstrates the power of Theorem 1 in formulating how the buffer size of each link reacts to any change in the transmission rates at the edges of the network. Now, when $\mathbf{b}(t)$ is strictly positive, the differential equation (3.24) describes how $b_1(t), ..., b_L(t)$ evolve. However, if some entries of $\mathbf{b}(t)$ turn out to be zero, the non-conventional differential equation (3.23) must be solved. Since the term $\dot{\mathbf{b}}(t)$ appears nonlinearly in the right side of this equation, the equation (3.23) can have no solution or multiple solutions for $\dot{\mathbf{b}}(t)$. In light of (3.24), one may use a continuity argument and speculate that $\dot{\mathbf{b}}(t)$ can be obtained as follows:

$$\dot{\mathbf{b}}(t) = \left(\Phi(t)^{-1} \left(R\mathbf{x}(t) - \mathbf{c} \right) \right)^{+}_{\mathbf{b}(t)}. \tag{3.31}$$

Unfortunately, the equations (3.23) and (3.31) are not identical as the next example shows.

Example 2: Consider the network shown in Figure 3.2, which comprises three sources and two links with the routing matrix

$$R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{3.32}$$

The matrix $\Phi(t)$ for this network turns out to be:

$$\Phi(t) = \begin{bmatrix} 1 & 0 \\ \theta_{13}(t) & 1 \end{bmatrix}. \tag{3.33}$$

If $b_1(t)$ and $b_2(t)$ are strictly positive at a time $t \geq 0$, then it follows from Theorem 1 that the governing differential equations for the buffer sizes are

$$\begin{aligned} \dot{b}_1(t) &= x_1(t) + x_3(t) - c_1 \\ \dot{b}_2(t) &= -\theta_{13}(t)(x_1(t) + x_3(t) - c_1) + x_2(t) + x_3(t) - c_2. \end{aligned}$$

Now, consider the case when some buffers are empty and therefore the above equations

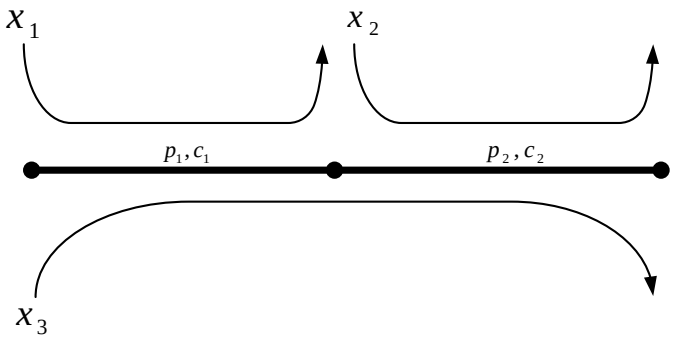


Figure 3.2: Network studied in Example 2.

cannot be used. To this end, assume that $\mathbf{x}(0)$ and $\mathbf{c}$ are such that

$$\begin{aligned} &b_1(0) = b_2(0) = 0, \quad \theta_{13}(0) = 0.5, \\ &R\mathbf{x}(0) - \mathbf{c} = \begin{bmatrix} -4 & -1 \end{bmatrix}^T. \end{aligned} \tag{3.34}$$

As can be argued intuitively, the time evolutions of $b_1(t)$ and $b_2(t)$ at $t = 0$ are

$$\dot{b}_1(0) = \dot{b}_2(0) = 0, \tag{3.35}$$

which also satisfy the nonlinear equation (3.23). However, the solution of (3.31) is as follows

$$\dot{b}_1(0) = 0, \quad \dot{b}_2(0) = 1, \tag{3.36}$$

which is different from the true solution (3.35).

Example 2 demonstrates that the nonlinear equation (3.23) cannot be simplified into (3.31). Now, why could the equation (3.23) describing the dynamics of buffer sizes have multiple or no solutions while the time evolution of buffer sizes must be unique in a real network? This is because (3.23) is an approximation that ignores delays between links, i.e., when a source changes its rate at its first link l, its effect is felt immediately at all the links it affects (all links l', s.t. $\phi_{ll'} \neq 0$). With this approximation, $\dot{\mathbf{b}}(t)$ appears in both sides of the equation (3.23) in a nonlinear way, causing ambiguity. However, due to the existence of delays in practice, the term $\dot{\mathbf{b}}(t)$ in the right side of the equation must be replaced by some delayed version of this vector, which resolves the source of ambiguity and makes the equation have a unique solution (because $\dot{\mathbf{b}}(t)$ appears only once in that case). To bypass

this issue, we make the following assumption:

Assumption 1. *Assume that the vector* $\mathbf{b}(t)$ *either is always positive or is allowed to be negative so that its dynamic can be described with the equation (3.24).*

Under this assumption, none of the buffers becomes empty at a transient time t so that the projection operator can be eliminated from the equations characterizing the dynamics of the buffer sizes. As will be explained later, this is a reasonable assumption for studying the local behavior of congestion control algorithms as long as all links of the network are bottleneck links at the equilibrium point where $\mathbf{p}^* > 0$. Hence, assume hereafter that $\mathbf{p}^* > 0$.

Remark 1. *Since the primary goal of this chapter is to study the instability as well as the local stability of the network, the assumption* $\mathbf{p}^* > 0$ *is made with no loss of generality. The reason is that if some links are not bottlenecked at the equilibrium, they will not affect instability or local stability of the network and therefore such links can be simply ignored. The same argument applies to Assumption 1 as well. Note that if it turns out that the congestion control algorithm is locally stable, then studying the global stability of the network necessitates the consideration of model (3.28) precisely without making Assumption 1.*

Remark 2. *Recall that the matrix* $R(\Theta(t))$ *inherits its structure from the routing matrix* R*. Nonetheless, it is not obvious whether* $\Phi(t)$ *can be constructed from* R *and* $R(\Theta(t))$ *using common matrix operations. To investigate this problem, consider a special, but important, case where there exist no two distinct sources* $s_1, s_2 \in \mathcal{S}$ *and two distinct links* l_1, l_2 *such that* $l_1, l_2 \in \mathcal{L}(s_1) \cap \mathcal{L}(s_2)$ *and*

- l_1 *appears before* l_2 *in the flow of source* s_1*,*
- l_2 *appears before* l_1 *in the flow of source* s_2*.*

In this case, the network is acyclic from a flow perspective, meaning that there are not two links whose buffers are mutually dependent. Now, it is possible to renumber the links of the network such that every flow passes through links in an ascending order, i.e. $g(s,1) < g(s,2) < \cdots < g(s,|\mathcal{L}(s)|)$ *for every* $s \in \mathcal{S}$*. By assuming that the links have been already arranged this way, it can be shown that* $\Phi(t)$ *is a lower triangular matrix satisfying the equality*

$$\Phi(t) = Lower\left\{R \times R(\Theta(t))^T\right\}, \tag{3.37}$$

where Lower$\{\cdot\}$ is a matrix operator that returns the lower part of its matrix argument (including its diagonal).

3.4 Congestion Control and Buffering Effect

The focus of the section is to show that while the congestion control algorithms have been proved to be stable using popular models that ignore the effect of buffering on output process, they can be unstable in a more realistic model that explicitly models the buffering effect. To this end, the primal-dual and dual algorithms are studied separately in the following subsections.

3.4.1 Instability of Primal-Dual Algorithm

Consider the primal-dual algorithm

$$\dot{x}_s(t) = k_s(U_s'(x_s(t)) - q_s(t)) \quad \forall s \in \mathcal{S} \tag{3.38a}$$

$$p_l(t) = h_l b_l(t) \quad \forall l \in \mathcal{L}, \tag{3.38b}$$

for some positive constants k_s and h_l, where $b_l(t)$ denotes the buffer size at link l whose dynamics was studied in the preceding section. In the pricing update (3.38b), if h_l is equal to 1, $p_l(t)$ is the l^{th} buffer size, and if h_l is equal to $\frac{1}{c_l}$, $p_l(t)$ is the queuing delay at the l^{th} buffer. In this section, with no loss of generality, assume that k_s and h_l are both equal to 1. The goal of this subsection is to show that the primal-dual algorithm can become unstable. Since the stability analysis provided here is based on linearizing the nonlinear primal-dual algorithm, the projection operator can be removed from the nonlinear differential equation (3.23) describing the dynamics of $b_l(t)$ (see Section V-A for more details). Hence, with no loss of generality, assume in the rest of this chapter that the buffer dynamics can be modeled as

$$\dot{\mathbf{b}}(t) = \Phi(t)^{-1}\left(R\mathbf{x}(t) - \mathbf{c}\right). \tag{3.39}$$

Note that the dynamical system corresponding to the primal-dual algorithm has two components (3.38) and (3.39), where (3.38) is time-invariant but (3.39) could be time-varying due to the term $\Phi(t)$. However, as long as $\Phi(t)$ is constant (denoted by Φ) or state-dependent (denoted by $\Phi(\mathbf{x}(t))$, then the primal-dual algorithm becomes time-invariant and there-

fore both time domain and frequency domain analyses can be performed to study its local behavior. These two important cases for $\Phi(t)$ will be studied in the sequel.

3.4.1.1 Constant Buffer Partitioning

Assume that $\Phi(t)$ is constant, i.e. $\Phi(t) = \Phi$. One can linearize the dynamical system corresponding to the primal-dual algorithm around its unique equilibrium point $(\mathbf{x}^*, \mathbf{p}^*)$ to obtain the linearized system

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} -\text{Diag}\left\{U_1''(x_1^*), \ldots, U_S''(x_S^*)\right\} & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) - \mathbf{x}^* \\ \mathbf{p}(t) - \mathbf{p}^* \end{bmatrix}, \tag{3.40}$$

where the operator Diag$\{\cdot\}$ makes a diagonal matrix from its arguments. If the above linearized system has unstable modes, then the primal-dual algorithm must be unstable. As stated after Theorem 1, if Φ is the identity matrix, the above system becomes the standard primal-dual algorithm which is known to be stable (in the absence of feedback delay). However, it will be shown next that the above system might become unstable for a general buffer-partitioning matrix Φ.

Example 3: Consider the network shown in Figure 3.3 consisting of seven sources and six links, with the routing matrix

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{3.41}$$

Assume that the matrix $R(\theta)$ describing the buffer partitions is as follows:

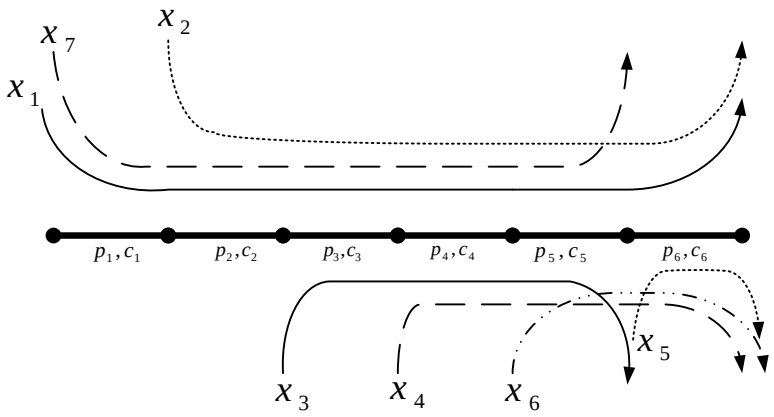


Figure 3.3: Network studied in Examples 3 and 4.

$$R(\theta) = \begin{bmatrix} 0.8 & 0 & 0 & 0 & 0 & 0 & 0.2 \\ 0.3 & 0.5 & 0 & 0 & 0 & 0 & 0.2 \\ 0.4 & 0.2 & 0.3 & 0 & 0 & 0 & 0.1 \\ 0.1 & 0.3 & 0.3 & 0.2 & 0 & 0 & 0.1 \\ 0.1 & 0.3 & 0.3 & 0.1 & 0 & 0.1 & 0.1 \\ 0.2 & 0.4 & 0 & 0.1 & 0.2 & 0.1 & 0 \end{bmatrix}. \tag{3.42}$$

In light of Remark 2, the matrix Φ can be obtained as

$$\Phi = \text{Lower}\left\{RR(\theta)^T\right\} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0.8 & 0.8 & 0.6 & 0.6 & 0.6 & 1 \end{bmatrix}.$$

Let $\mathbf{c}$ be equal to

$$\begin{bmatrix} 10 & 20 & 30 & 40 & 50 & 60 \end{bmatrix}^T \tag{3.43}$$

and the utility functions be taken as $U_s(x_s) = w_s \log x_s,\ s \in 1, ..., 7$, where

$$\begin{aligned} &w_1 = 15, \quad w_2 = 12.5, \quad w_3 = 7.5, \quad w_4 = 7.5, \\ &w_5 = 5, \quad w_6 = 5, \quad w_7 = 5. \end{aligned} \tag{3.44}$$

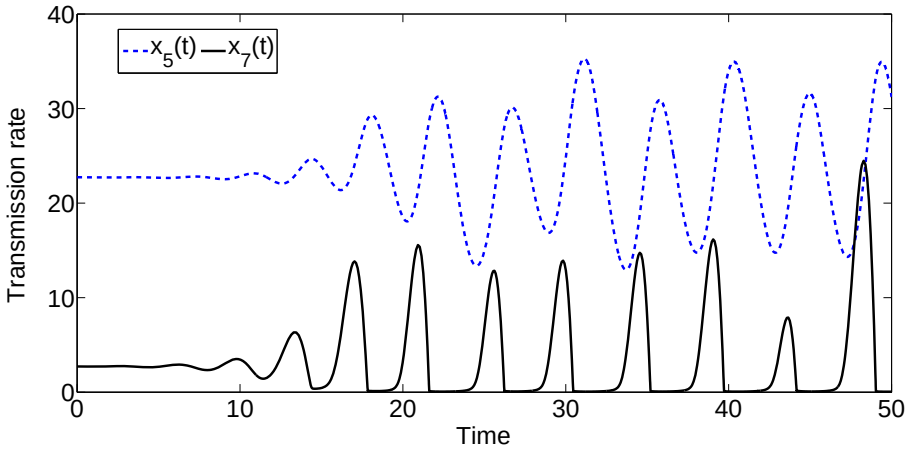


Figure 3.4: This figure illustrates the instability of the primal-dual algorithm with the buffer-size pricing mechanism for Example 3.

The linearized system (3.40) in this case has six eigenvalues $0.4060 \pm 1.7721i$, 0.2590 ± 1.0134 and 0.0279 ± 0.6940 whose real parts are positive. Thus, the linearization of the primal-dual algorithm around its equilibrium point leads to an unstable system. Therefore, the nonlinear system corresponding to the primal-dual algorithm must be unstable. To observe this instability phenomenon in simulation, assume that the initial value of every buffer size is equal to its optimal value (i.e. $\mathbf{b}(0) = \mathbf{p}^*$), whereas the initial value of every transmission rate is perturbed from its optimal value by a small amount 0.001 (i.e. $\mathbf{x}(0) = 1.001\ \mathbf{x}^*$). The signals $x_5(t)$ and $x_7(t)$ are plotted in Figure 3.4 to show how this small deviation from the equilibrium point makes the transmission rates oscillate.

Example 3 demonstrates that the primal-dual algorithm can experience instability even in the very simple case of constant $\Phi(t)$. The origin of this issue is studied mathematically in the next theorem.

Theorem 2. *For a constant $\Phi(t) = \Phi$, the following two cases can occur for the primal-dual algorithm given by (3.38) and (3.39):*

i) All eigenvalues of $R^T\Phi^{-1}R$ are real and nonnegative. In this case, the congestion algorithm is globally stable if Φ is symmetric.

ii) At least one eigenvalue of $R^T\Phi^{-1}R$ is complex or negative real. In this case, there exists a strictly positive number α such that the primal-dual algorithm becomes unstable if $U_s(x_s)$ is taken as $\alpha\log(x_s)$ for every $s \in \mathcal{S}$.

Proof of Part (i): By assumption, R has full row rank, and in addition $R^T\Phi^{-1}R$ is symmetric and has nonnegative eigenvalues. Hence, Φ is a positive definite matrix. The global stability of the primal-dual algorithm can be shown using the Lyapunov function $V(\mathbf{x},\mathbf{p}) = (\mathbf{x}(t)-\mathbf{x}^*)^T(\mathbf{x}(t)-\mathbf{x}^*) + (\mathbf{P}(t)-\mathbf{P}^*)^T\Phi(\mathbf{P}(t)-\mathbf{P}^*)$ (see Theorem 4 for a similar proof). The details are omitted for brevity.

Proof of Part (ii): It follows from (3.40) that the primal-dual algorithm is unstable if the matrix A defined as

$$\begin{bmatrix} \alpha D & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix} \tag{3.45}$$

has some unstable eigenvalues, where D is a negative definite diagonal matrix with the (s,s) entry equal to $-\frac{1}{(x_s^*)^2}$. Decompose the matrix A as follows:

$$A = \begin{bmatrix} -\alpha D & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix}. \tag{3.46}$$

Let λ denote an arbitrary eigenvalue of the second matrix in the right side of the above equation, i.e.

$$\begin{bmatrix} 0 & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix} \tag{3.47}$$

The parameter λ should satisfy the relation

$$\begin{aligned} 0 &= \text{determinant}\left(\lambda\mathbf{I} - \begin{bmatrix} 0 & -R^T \\ \Phi^{-1}R & 0 \end{bmatrix}\right) \\ &= \text{determinant}\left(\lambda^2\mathbf{I} + R^T\Phi^{-1}R\right). \end{aligned} \tag{3.48}$$

Hence, λ^2 is an eigenvalue of the matrix $-R^T\Phi^{-1}R$. Conversely, if $\bar{\lambda}$ denotes any eigenvalue of $R^T\Phi^{-1}R$ that is either complex or negative real, then $\pm\sqrt{-\bar{\lambda}}$ are both eigenvalues of the matrix (3.47) and at least one of them lies on the open right-half complex plane. This implies that the matrix (3.47) is unstable. Therefore, if the positive number α is chosen to be sufficiently small, the matrix A given in (3.46) will definitely have unstable eigenvalues. This completes the proof. ■

The matrix Φ in practice is very likely to be non-symmetric, and it can even be lower triangular (see Remark 2). Thus, the matrix $R^T\Phi^{-1}R$ might generically have complex

eigenvalues. Now, Theorem 2 states that under this circumstance, the users can choose their utility functions in such a way that the primal-dual algorithm will not be able to solve the resource allocation problem given in (3.1) and (3.2) asymptotically, due to the buffering effects.

3.4.1.2 State-Dependent Buffer Partitioning

As shown earlier, the primal-dual algorithm is not always stable under constant buffer partitioning coefficients. A question arises as whether this instability issue can be resolved if $\Phi(t)$ depends on the state of the system. To address this problem, let $\Phi(t)$ be state-dependent and denote it as $\Phi(\mathbf{x}(t))$. The linearization of the primal-dual algorithm given in (3.38) and (3.39) leads to

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} \text{Diag}\{U_1''(x_1^*), \ldots, U_S''(x_S^*)\} & -R^T \\ \Phi(\mathbf{x}^*)^{-1}R & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) - \mathbf{x}^* \\ \mathbf{p}(t) - \mathbf{p}^* \end{bmatrix}., \tag{3.49}$$

By comparing the above system with (3.40), it can be observed that in order to analyze local stability of the primal-dual algorithm with a state-dependent $\Phi(\mathbf{x}(t))$, one can replace $\Phi(\mathbf{x}(t))$ with the constant matrix $\Phi(\mathbf{x}^*)$. In other words, as far as the local stability is concerned, state-dependent buffer partitioning reduces to constant buffer partitioning, where the constant matrix Φ is obtained by evaluating $\Phi(\mathbf{x}(t))$ at the equilibrium point. Having observed this fact, the next example shows that the primal-dual algorithm can still become unstable.

Example 4: Consider the network given in the Example 3 and let the coefficients $\theta_{ls}(\mathbf{x})$ be taken according to the S-WFQ scheduling technique explained in Section III-A. Recall that $\Phi(t)$ corresponding to this service discipline is state-dependent and time-invariant. It is straightforward to justify that all rates $x_{ls}(t)$, $l \in \mathcal{L}(s)$, become equal to x_s^* at the equilibrium point, for every $s \in \mathcal{S}$. Now, if $\Phi(\mathbf{x}^*)$ is calculated for this example, one can observe that the linearized system (3.49) has six unstable modes with the eigenvalues $0.3838 \pm 1.7096i$, $0.2427 \pm 1.0384i$ and $0.0517 \pm 0.7018i$. Hence, the nonlinear primal-dual algorithm is not locally stable under the S-WFQ scheduling.

One can generalize Theorem 2 to a non-constant matrix $\Phi(t)$.

Theorem 3. *Let* $\mathbf{x}^*$ *denote the solution of the resource allocation problem introduced*

in (3.1) and (3.2), associated with the utility functions $U_1(x_1), ..., U_S(x_S)$. *For a state-dependent* $\Phi(t) = \Phi(\mathbf{x}(t))$, *the following two cases can occur for the primal-dual algorithm given by (3.38) and (3.39):*

i) All eigenvalues of $R^T\Phi(\mathbf{x}^*)^{-1}R$ *are real and nonnegative. In this case, the congestion algorithm is locally stable if* $\Phi(\mathbf{x}^*)$ *is symmetric.*

ii) At least one eigenvalue of $R^T\Phi(\mathbf{x}^*)^{-1}R$ *is complex or negative real. In this case, there exists a strictly positive number* α *such that the primal-dual algorithm becomes unstable if each user* $s \in \mathcal{S}$ *takes its utility function as* $\alpha U_s(x_s)$ *rather than* $U_s(x_s)$.

Proof: If the utility function of each user $s \in \mathcal{S}$ changes from $U_s(x_s)$ to $\alpha U_s(x_s)$, the optimal resource allocation vector will be still $\mathbf{x}^*$. Hence, this uniform change in all of the utility functions yields the linearized system associated with the primal-dual algorithm as follows

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} \alpha \mathrm{Diag}\left\{U_1''(x_1^*), \ldots, U_S''(x_S^*)\right\} & -R^T \\ \Phi(\mathbf{x}^*)^{-1}R & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) - \mathbf{x}^* \\ \mathbf{p}(t) - \mathbf{p}^* \end{bmatrix}, \tag{3.50}$$

where $\mathbf{x}^*$ is the optimal solution to the resource allocation problem for $\alpha = 1$. After noticing this fact, one can prove this theorem using the same arguments as in the proof of Theorem 2. ■

Remark 3. *As discussed in Section III-A, the matrix* $\Phi(t)$ *corresponding to the S-WFQ scheme is state-dependent and can be expressed in terms of* $\mathbf{x}(t)$. *Hence, the results of this part are applicable to this service discipline. However, as explained in Section III-A,* $\Phi(\mathbf{x}(t))$ *will have a complicated form, which makes the global analysis of the primal-dual algorithm very difficult. To mitigate this issue, let the parameter* $\theta_{ls}(\mathbf{x}(t))$ *for the S-WFQ scheme be given by*

$$\theta_{ls}(\mathbf{x}(t)) = \frac{x_{ls}(t)}{\sum_{s' \in \mathcal{S}(l)} x_{ls'}(t)}, \quad \forall s \in \mathcal{S}(l), \tag{3.51}$$

approximated as

$$\theta_{ls}(\mathbf{x}(t)) = \frac{x_s(t)}{\sum_{s' \in \mathcal{S}(l)} x_{s'}(t)}, \quad \forall s \in \mathcal{S}(l), \tag{3.52}$$

which allows for expressing $\Phi(t)$ *in terms of* $\mathbf{x}(t)$ *in a much easier way. A question arises as to whether this new service discipline is a good approximation for the S-WFQ. To answer this*

question, notice that $\Phi(\mathbf{x}^*)$ is identical for both (3.51) and (3.52), because $x^*_{ls} = x^*_s$ (meaning that the intermediate flow rates are the same as the original source rates at equilibrium). As a result of (3.49), the S-WFQ scheduling can be approximated in such a way that its local behavior is unchanged, while its global behavior can be approximately analyzed with a much lower complexity.

3.4.2 Stability of Dual Algorithm

Consider the dual algorithm

$$x_s(t) = {U'_s}^{-1}(q_s(t)) \quad \forall s \in \mathcal{S}, \tag{3.53a}$$

$$p_l(t) = h_l b_l(t) \quad \forall l \in \mathcal{L}, \tag{3.53b}$$

where $b_l(t)$ is governed by the equation (3.39). Unlike the primal-dual algorithm which can become unstable in presence of buffers, we have not been able to find any unstable example for the dual algorithm. Hence, it seems that the dual algorithm is almost always stable. This conjecture might be true because of two reasons: (i) the matrix Φ is highly structured and (ii) since the transmission rates of users are updated in a static way via the dual algorithm, this algorithm has far fewer dynamics involved compared to the primal-dual algorithm. Although it is hard to prove or disprove this conjecture, a sufficient condition will be provided in the sequel to guarantee the stability of this algorithm for a state-dependent $\Phi(t)$ (note that constant $\Phi(t)$ is a special case of state-dependent $\Phi(t)$).

Theorem 4. *Given $\Phi(t) = \Phi(\mathbf{x}(t))$, assume that $\Phi(\mathbf{x}(t))$ is a continuous function at the point $\mathbf{x}(t) = \mathbf{x}^*$. The dual algorithm (3.53) is locally stable, provided the matrix $\Phi(\mathbf{x}^*) + \Phi(\mathbf{x}^*)^T$ is positive definite. In addition, if $\Phi(t)$ is constant, the algorithm is globally asymptotically stable.*

Proof: Assume that the matrix $\Phi(\mathbf{x}^*) + \Phi(\mathbf{x}^*)^T$ is positive definite. Consider the candidate Lyapunov function

$$V(\mathbf{p}) = \sum_{s \in \mathcal{S}} \int_{q^*_s}^{q_s} (x^*_s - (U'_s)^{-1}(\sigma)) d\sigma. \tag{3.54}$$

This function is nonnegative, radially unbounded, and equal to zero only at $\mathbf{p} = \mathbf{p}^*$. One can write

$$\begin{aligned}
\frac{dV}{dt} &= \sum_{s\in\mathcal{S}}(x_s^* - (U_s')^{-1}(q_s))\dot{q}_s \\
&= (\mathbf{x}^* - \mathbf{x})^T\dot{\mathbf{q}} = (\mathbf{x}^* - \mathbf{x})^T R^T\dot{\mathbf{p}} \\
&= -(R\mathbf{x} - \mathbf{c})^T\Phi(\mathbf{x}(t))^{-1}(R\mathbf{x} - \mathbf{c}) \\
&= -\frac{1}{2}(R\mathbf{x} - \mathbf{c})^T\Big(\Phi(\mathbf{x}(t))^{-1} + \Phi(\mathbf{x}(t))^{-T}\Big)(R\mathbf{x} - \mathbf{c}) \\
&= -\frac{1}{2}(R\mathbf{x} - \mathbf{c})^T\Phi(\mathbf{x}(t))^{-1}\Big(\Phi(\mathbf{x}(t)) + \Phi(\mathbf{x}(t))^T\Big)\times \\
&\quad \Phi(\mathbf{x}(t))^{-T}(R\mathbf{x} - \mathbf{c}).
\end{aligned} \tag{3.55}$$

Hence, due to the continuity of $\Phi(\mathbf{x}(t))$ at $\mathbf{x}(t) = \mathbf{x}^*$, positive definiteness of $\Phi(\mathbf{x}^*)+\Phi(\mathbf{x}^*)^T$, and radially unboundedness of $V(\mathbf{p})$, there exists an invariant set in $\Re^L$ containing $\mathbf{p}^*$ such that $\dot{V}(\mathbf{p})$ is non-positive for every $\mathbf{p}$ in this invariant set. Now, it follows from the Lyapunov theorem that the unique equilibrium point $(\mathbf{x}^*, \mathbf{p}^*)$ of the dual algorithm is locally stable. In the case when $\Phi(\mathbf{x}(t))$ is constant, it can be concluded from (3.55) that this equilibrium point is globally stable. ■

3.5 Discussions

3.5.1 Alternative Congestion Feedback

So far, it is proved that distributed congestion control algorithms (e.g. the primal-dual algorithm) might become unstable when buffers are modeled explicitly. To remedy this issue, a slightly different pricing mechanism will be proposed below to ensure the stability under both the primal-dual and dual algorithms. For simplicity, the result will be explained for a constant Φ. Here, we again assume that the buffer dynamics can be modeled as (3.39).

Theorem 5. *The dual algorithm (3.53) and the primal-dual algorithm (3.38) are both globally asymptotically stable with the unique equilibrium point* $(\mathbf{x}^*, \Phi^{-1}\mathbf{p}^*)$*, provided the source price vector* $\mathbf{q}(t)$ *is taken as* $R^T\Phi\mathbf{p}(t)$ *as opposed to* $R^T\mathbf{p}(t)$*.*

Proof: The proof will be provided here only for the primal-dual algorithm because the proof for the dual algorithm is similar. By defining $\tilde{\mathbf{p}}(t)$ as $\Phi\mathbf{p}(t)$ and considering the source price $\mathbf{q}(t)$ as $R^T\tilde{\mathbf{p}}(t)$, it can be shown that the modified primal-dual algorithm turns out to

be

$$\dot{x}_s(t) = k_s \left(U_s'(x_s(t)) - q_s(t) \right), \qquad \forall s \in \mathcal{S} \tag{3.56a}$$

$$\dot{\tilde{p}}_l(t) = h_l \left(y_l(t) - c_l \right), \qquad \forall l \in \mathcal{L}, \tag{3.56b}$$

where $\tilde{p}_1, ..., \tilde{p}_L$ denote the entries of $\tilde{\mathbf{p}}(t)$. Notice that the above algorithm is a special type of the standard primal-dual algorithm given in (3.6), for which it is known that $x_s(t) \to x_s^*$ and $\tilde{p}_l(t) \to p_l^*$ as t goes to infinity. This result implies that $\mathbf{x}(t) \to \mathbf{x}^*$ and $\tilde{\mathbf{p}}(t) = \Phi \mathbf{p}(t) \to \mathbf{p}^*$ as t increases. Consequently, the states $\mathbf{x}(t)$ and $\mathbf{p}(t)$ of the modified primal-dual algorithm with $\mathbf{q}(t) = R^T \Phi \mathbf{p}(t)$ converge to $\mathbf{x}^*$ and $\Phi^{-1}\mathbf{p}^*$, respectively. This completes the proof. ■

As illustrated in Examples 3 and 4, if each source $s \in \mathcal{S}$ updates its transmission rate $x_s(t)$ based on the price $q_s(t)$ obtained by just adding up the link prices along the route of source s, the corresponding congestion control algorithm may not be stable. Instead, Theorem 5 suggests taking the source price $q_s(t)$ as the s^{th} entry of the vector $R^T \Phi \mathbf{p}(t)$. This can be implemented using the following scheme.

For every $s \in \mathcal{S}$, let a zero price value be initially assigned to the flow of source s. As this flow passes through every link, say link l, it reports its price value to the link and then increases its price by $\theta_{ls} b_l(t)$. On the other hand, each link maintains a price for itself (say $\tilde{p}_l(t)$) by adding up the source prices reported to the link by its incoming flows. Now, the acknowledgment of the flow in the return path accumulates the new link prices in its route and reports it back to the source.

It can be verified that the above strategy corresponds to the new price updating $\mathbf{q}(s) = R^T \Phi \mathbf{p}(t)$ that is needed in the strategy proposed by Theorem 5.

To illustrate the aforementioned idea, let the new pricing mechanism proposed in Theorem 5 be deployed to fix the instability of the primal-dual algorithm for the network studied in Example 3. To this end, consider the initial parameters

$$\begin{aligned} \mathbf{x}(0) &= \begin{bmatrix} 10 & 10 & \cdots & 10 \end{bmatrix}, \\ \mathbf{b}(0) = \mathbf{v} &= \begin{bmatrix} 20 & 20 & \cdots & 20 \end{bmatrix}. \end{aligned} \tag{3.57}$$

The transmission rates and buffer sizes associated with the modified primal-dual algorithm

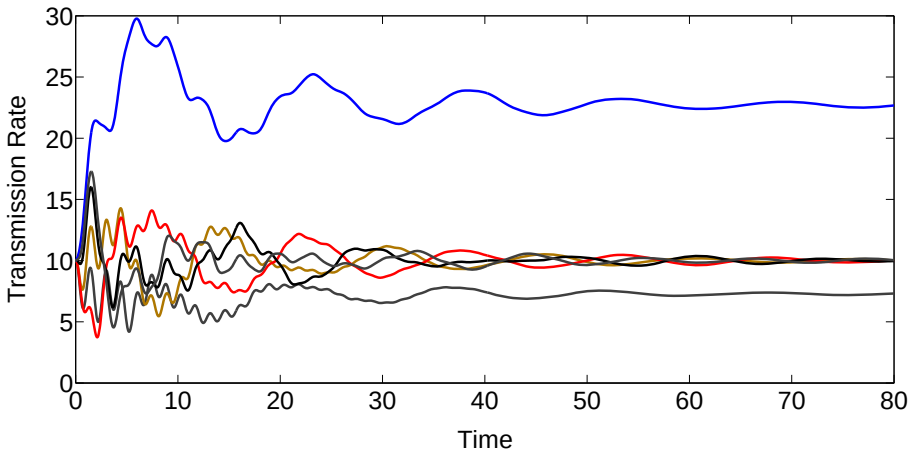


Figure 3.5: This figure illustrates the stability of the primal-dual algorithm using the modified buffer-size pricing mechanism for Example 3.

are plotted in Figures 3.5 and 3.6 to demonstrate that all these signals converge and therefore the resulting congestion control algorithm is stable.

3.5.2 Nonzero Buffer Assumption

Despite the fact that Theorem 1 derives a mathematical model for buffer sizes in a general setting, the main results developed in this work (say in Sections IV-A and IV-B) rely on the assumption that the buffer sizes $b_1(t), ..., b_L(t)$ never become zero. In what follows, we elaborate on the validity of this assumption and justify why the removal of this assumption does not change the conclusions drawn in this chapter.

First, consider the constant-buffer-partitioning case studied in Subsection IV-A-1. A mild assumption used in this part was that all links of the network would be bottleneck links under the optimal transmission rates of all users. This implies that $\mathbf{p}^*$ is a strictly positive vector and as a result $\mathbf{b}^* > 0$. Hence, if the initial buffer sizes $b_1(0), ..., b_L(0)$ as well as the initial transmission rates $x_1(0), ..., x_S(0)$ are all in the neighborhood of their optimal values, then the assumption of positivity of $b_1(t), ..., b_L(t)$ at all times $t \geq 0$ would be met as long as the algorithm is stable. The same argument also holds for the case with state-dependent buffer partitioning coefficients studied in Subsection IV-A-2.

Recall that Examples 3 and 4 demonstrate that the primal-dual algorithm may not be stable for both constant and state-dependent partitioning coefficients. Since the assumption of strict positivity of all buffer sizes is used in these examples, it could be conjectured that

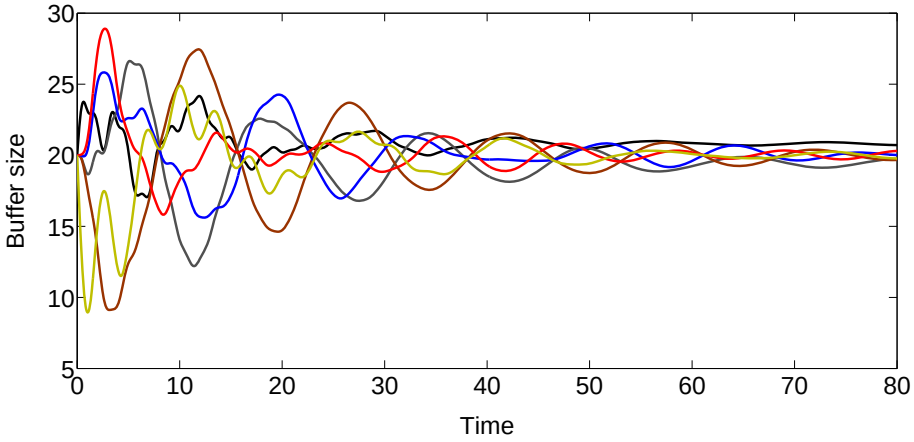


Figure 3.6: This figure illustrates the stability of the primal-dual algorithm using the modified buffer-size pricing mechanism for Example 3.

this instability phenomenon may not occur in practice due to the underlying assumption not being valid. Nonetheless, it can be argued that the primal-dual algorithm is still unstable for Examples 3 and 4 even if the buffer sizes are allowed to become zero. Indeed, it follows from the definition of stability that if the network in either Example 3 or Example 4 is stable (in the absence of the aforementioned assumption), then $b_1(t), ..., b_L(t)$ always stay positive and bounded for the initial values $b_1(0), ..., b_L(0), x_1(0), ..., x_S(0)$ sufficiently close to their optimal values. Now that the buffer sizes are always positive in the transient time, the underlying assumption is automatically satisfied, under which the instability of the network was already proved.

As far as the primal-dual algorithm with the new pricing mechanism $\mathbf{q}(t) = R^T \Phi \mathbf{p}(t)$ given in Theorem 5 is concerned, the relation $\mathbf{b}^* = \Phi^{-1}\mathbf{p}^*$ is satisfied. Now, notice that although $\mathbf{p}^*$ is a positive vector, $\Phi^{-1}\mathbf{p}^*$ might have some negative entries. This yields the contradictory result that the optimal buffer sizes $b_1^*, ..., b_L^*$ may not be all nonnegative, which implies that some of the buffers $1, ..., L$ must become empty during transient time. To remedy this problem, note that if the pricing mechanism $\mathbf{q}(t) = R^T \Phi(\mathbf{p}(t) - \boldsymbol{\nu})$ is used instead, for some positive vector $\boldsymbol{\nu}$, then the steady-state vector $\mathbf{b}^*$ becomes equal to $\Phi^{-1}\mathbf{p}^* + \boldsymbol{\nu}$. Hence, a proper choice of $\boldsymbol{\nu}$ makes the buffer sizes at the equilibrium point strictly positive and, therefore, it resolves the problem. Note that the interpretation of this technique is simply as follows: the buffer size that each link reports to the corresponding users should be less than the true value so that users transmit data at higher rates to keep

the buffers nonempty. This idea of reporting some virtual values for buffer sizes is also applicable to the primal-dual algorithm with the standard pricing mechanism.

3.6 Summary

Congestion control algorithms aim to allocate resources to demands in a network in such a way that the total utilization is optimized. The existing stability results for congestion control algorithms are derived for a fluid model of the network under the assumption that the same flow appears on an end-to-end basis in the network. However, buffers in the network might cause the flows to be thinned as they pass through. In this chapter, the buffers are first modeled for well-known service disciplines such as first-in first-out and weighted fair queuing, and then the effect of buffers on the stability of the dual and primal-dual congestion control algorithms is studied accordingly. It is shown that these algorithms might no longer be stable if the effect of buffering is taken explicitly into account. Sufficient conditions are also provided to guarantee the stability of these algorithms under the common pricing techniques. Finally, a new pricing mechanism is introduced, which makes the dual and primal-dual algorithms stable.

Chapter 4

Network Topologies with Zero Duality Gap for Optimal Power Flow

It has been recently observed and justified that the optimal power flow (OPF) problem with a quadratic cost function may be solved in polynomial time for a large class of power networks, including IEEE benchmark systems. In this work, this result is extended to OPF with arbitrary convex cost functions and then a more rigorous theoretical foundation is provided accordingly. First, a necessary and sufficient condition is derived to guarantee the solvability of OPF in polynomial time through its Lagrangian dual. Since solving the dual of OPF is expensive for a large-scale network, a far more scalable algorithm is designed by utilizing the sparsity in the graph of a power network. The computational complexity of this algorithm is related to the number of cycles of the network. Furthermore, it is proved that due to the physics of a power network, the polynomial-time algorithm proposed here always solves every full AC OPF problem precisely or after two mild modifications.

4.1 Introduction

The optimal power flow (OPF) problem is concerned with finding an optimal operating point of a power system, which minimizes a certain objective function (e.g., power loss or generation cost) subject to network and physical constraints [65, 66]. This optimization problem has been extensively studied since 1962 [21]. Due to the nonlinear interrelation among active power, reactive power and voltage magnitude, OPF is described by nonlinear equations and may have a nonconvex/disconnected feasibility region [67]. Several algorithms

have been proposed for solving this highly nonconvex problem, including linear programming, quadratic programming, nonlinear programming, Lagrange relaxation, interior point methods, artificial intelligence, artificial neural network, fuzzy logic, genetic algorithms, evolutionary programming and particle swarm optimization [68, 69, 65, 66, 70, 71, 72]. In order to solve OPF more efficiently, different conic and convex relaxations have been proposed in the past decade [22, 23, 24]. An efficient algorithm for solving OPF should possess two properties: (i) polynomial-time complexity, (ii) ability to find a global solution. As will be demonstrated later in Section I-A, the second property is highly desirable because the cost for a local solution could be much larger than the cost for a globally optimal solution.

The classical OPF problem with a quadratic cost function has been recently studied in [25, 73, 74]. The Lagrangian dual of OPF is obtained in [25] as a semidefinite program (SDP), from which a globally optimal solution of OPF can be found (in polynomial time) if the duality gap between OPF and its dual is zero (i.e., if strong duality holds). It is then shown that the duality gap is zero for IEEE benchmark systems with 14, 30, 57, 118 and 300 buses, in addition to several randomly generated power networks. The paper [25] proves that the duality gap is expected to be zero for a large class of power networks due to the passivity of transmission lines and transformers. In particular, there exists an unbounded set of network topologies (admittance matrices) that make the duality gap zero for all possible values of loads, provided load over-satisfaction (power over delivery) is allowed. Note that allowing load over-satisfaction means that the power balance equations are expressed as inequality constraints rather than equality constraints [71, 75, 25]. In [73], the results were extended to the case when there are more sources of non-convexity in OPF, such as variable transformer ratios, variable shunt elements and contingency constraints. Note that the convexification of OPF through its Lagrangian dual, if possible, has two main advantages. In terms of operation planning, this means that a global solution can be found efficiently. In terms of electricity market, this means that the Lagrange multipliers used for selling or buying power are meaningful pricing signals.

The above-mentioned convex relaxation approach has been further developed in a number of recent papers to handle other energy-related optimization problems such as multi-period optimal dispatch [76], state estimation in power systems [77], optimal power flow with distributed energy storage dynamics [78], transmission system planning [79], optimal charging of plug-in hybrid electric vehicles [80], distributed control for power networks [81],

and optimal power balance under uncertainty [82]. Moreover, the SDP program proposed in [25] has also been explored by various researchers in the context of OPF to both enhance the underlying theory and resolve practical issues. For instance, the paper [83] proposes a distributed algorithm for solving this SDP problem. [84] and [85] prove that the SDP relaxation is guaranteed to work for tree networks under certain assumptions. A similar result is also derived in [86] using a different formulation for the OPF problem. Moreover, the work [87] studies possible issues associated with the SDP formulation and also demonstrates the application of this method in finding multiple solutions of a power flow problem.

Before spelling out the contribution of the current work, we first present a motivating example in the sequel.

4.1.1 Motivating Example

Consider the three-bus network depicted in Figure 4.1 with the following line impedances (z) and total shunt susceptances (b):

$$z_{12} = 0.42 + 0.9\mathrm{i}, \quad z_{23} = 0.25 + 0.75\mathrm{i}, \quad z_{13} = 0.55 + 0.9\mathrm{i},$$
$$b_{12} = 0.3, \quad b_{23} = 0.7, \quad b_{13} = 0.45,$$

where every bus is associated with a constant-active-power load, while buses 1 and 2 are also associated with two generators with the active-power outputs P_{G_1} and P_{G_2}. This network is adopted from [87] with a 100 MVA base. Assume that reactive power can be compensated arbitrarily at every bus. The goal is to find the optimal values of P_{G_1} and P_{G_2} in such a way that the generation cost $5P_{G_1} + P_{G_2}$ is minimized and that the load demand is satisfied at every bus. Denote the complex voltages at buses 1, 2 and 3 as V_1, V_2 and V_3, respectively. We consider two cases in the following:

- *Case 1:* Assume that the voltage magnitudes $|V_1|, |V_2|, |V_3|$ must all be equal to the nominal value 0.8. The MATLAB interior point solver called by the toolbox MATPOWER yields the local solution

$$V_1^{\mathrm{opt}} = 0.8\angle 0, \; V_2^{\mathrm{opt}} = 0.8\angle -67.50^o,$$
$$V_3^{\mathrm{opt}} = 0.8\angle -115.72^o, \; P_{G_1}^{\mathrm{opt}} = 272.79, \; P_{G_2}^{\mathrm{opt}} = 138.70,$$

associated with the generation cost 1502.64. In contrast, using the method proposed in [25], one can show that the global solution to the above-mentioned optimization is given by

$$V_1^{\text{opt}} = 0.8\angle -2.94^o,\ V_2^{\text{opt}} = 0.8\angle 56.86^o,$$
$$V_3^{\text{opt}} = 0.8\angle 0,\ P_{G_1}^{\text{opt}} = 70.45,\ P_{G_2}^{\text{opt}} = 249.93,$$

corresponding to the optimal generation cost 602.20. Observe that the local solution found by MATPOWER is physically meaningless, and more importantly the generation cost associated with this local solution is at least twice more than the cost for the global solution.

- *Case 2:* Assume that the voltage magnitudes $|V_1|, |V_2|, |V_3|$ are confined to the interval $[0.8, 1.2]$. The toolbox MATPOWER gives the same local solution as before with the generation cost 1502.64, whereas the generation cost corresponding to the global solution (obtained using [25]) is equal to 338.

So far, it has been demonstrated that a global solution can be far better than a local solution. Note that more than one local solution may be attained in practice depending on the type of the algorithm used together with its initialization. Now, assume that there is a limit on the active flow transferred on the line $(2, 3)$, say $P_{23}, P_{32} \leq P_{23}^{\max} = P_{32}^{\max}$. It can be shown that the duality gap is always zero in both Case 1 and Case 2 for all nonnegative values of $P_{23}^{\max}$. Note that the duality gap being zero does not imply that the OPF problem is feasible. For instance, the OPF problem in Case 1 becomes infeasible if $P_{23}^{\max} \leq 23.11$, but zero duality gap implies that this infeasibility can be detected using a polynomial-time algorithm by obtaining an unbounded optimal objective value for the dual of OPF. On the other hand, as observed in [87], the duality gap might become nonzero if the apparent power (as opposed to active power) on the line $(2, 3)$ is upper bounded by a small number. However, as will be shown later in this work, if there is a phase shifter in this network, the duality gap will always be zero. It is worth mentioning that even if the line $(2, 3)$ is removed to make the network radial, the original OPF problem is still nonconvex.

Motivated by this example, the goal of this chapter is to perform a deeper study of the duality gap for the OPF problem.

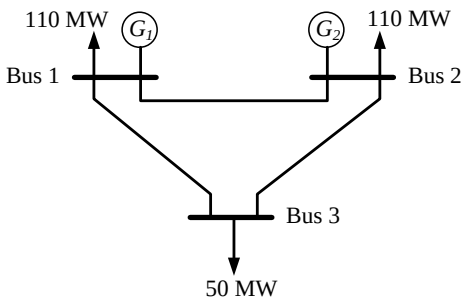


Figure 4.1: The three-bus power network studied in Section I-A.

4.1.2 Contributions

This chapter extends the previous results on the zero duality gap of OPF in several important directions. The first goal is to generalize the results to an arbitrary convex cost function. To this end, the dual of OPF in the general case is derived as the maximization of a concave function subject to a linear matrix inequality (LMI) (see [29] or [25] for the definition of LMI). In the case when the cost function is quadratic, this dual optimization simply becomes an SDP. Necessary and sufficient conditions are derived to guarantee zero duality gap. Since solving the dual of OPF is hard for a very large-scale network, the second goal is to design a more scalable algorithm by exploiting the sparsity in the topology of the network. The third goal of this chapter is to understand what reasonable approximation on the OPF problem guarantees its solvability in polynomial time. To address this problem, it is shown that adding phase shifters to certain lines of the network reduces the computational complexity of OPF. Moreover, if every cycle of the network contains a line with a controllable phase shifter, then the duality gap can be verified by solving a simple generalized SOCP problem. It is also shown that if load over-satisfaction is allowed, then the duality gap is always zero for all possible values of loads, physical limits and cost functions, due to the physics of a power network. This result implies that an OPF problem can always be solved efficiently after two modifications: (i) expressing power balance equations as inequality constraints, (ii) adding (virtual) phase shifters to the network. As stated in [71, 75, 25], modification (i) often has a negligible effect (see Chapter 15 of [75] for more details). We will also show that only a few phase shifters are needed in practice through modification (ii) (for instance, 1 or 2 phase shifters are enough for IEEE systems with 30 and 118 buses). Note that phase shifters are used in practice to improve controllability of

the network for relieving congestion and routing active power, but this work substantiates their role in convexifying the OPF problem.

4.1.3 Notations

The following notations will be used throughout this chapter.

- i : Imaginary unit.
- $\mathbf{R}$: Set of real numbers.
- $\mathbf{H}^{n\times n}$: Set of $n \times n$ Hermitian matrices.
- $\mathrm{Re}\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$: Real and imaginary parts of a complex matrix.
- $*$: Conjugate transpose operator.
- T : Transpose operator.
- $(\cdot)^{\mathrm{opt}}$: Notation used to denote a **globally** optimal solution.
- $\succeq$: Matrix inequality sign in the positive semidefinite sense [29].

Moreover, given complex values a_1 and a_2, the inequality $a_1 \geq a_2$ used in this chapter means $\mathrm{Re}\{a_1\} \geq \mathrm{Re}\{a_2\}$ and $\mathrm{Im}\{a_1\} \geq \mathrm{Im}\{a_2\}$.

4.2 Problem Formulation

Consider a power network with the set of buses $\mathcal{N} := \{1, 2, ..., n\}$ and the set of flow lines $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$. Define:

- $P_{D_k} + Q_{D_k}$i: Complex power of the load connected to bus $k \in \mathcal{N}$.
- $P_{G_k} + Q_{G_k}$i: Output complex power of the generator connected to bus $k \in \mathcal{N}$.
- V_k: Complex voltage at bus $k \in \mathcal{N}$.
- P_{lm}: Active power transferred from bus $l \in \mathcal{N}$ to bus $m \in \mathcal{N}$ through the line $(l, m) \in \mathcal{L}$.
- S_{lm}: Complex power transferred from bus $l \in \mathcal{N}$ to bus $m \in \mathcal{N}$ through the line $(l, m) \in \mathcal{L}$.

- $f_k(P_{G_k})$: A convex function representing the cost associated with generator $k \in \mathcal{G}$.

Define $\mathbf{V}$, $\mathbf{P}_G$, $\mathbf{Q}_G$, $\mathbf{P}_D$ and $\mathbf{Q}_D$ as the vectors $\{V_k\}_{k\in\mathcal{N}}$, $\{P_{G_k}\}_{k\in\mathcal{N}}$, $\{Q_{G_k}\}_{k\in\mathcal{N}}$, $\{P_{D_k}\}_{k\in\mathcal{N}}$ and $\{Q_{D_k}\}_{k\in\mathcal{N}}$, respectively. The power network has some controllable parameters which can all be recovered from $\mathbf{V}$, $\mathbf{P}_G$ and $\mathbf{Q}_G$. In order to optimize these controllable parameters, the optimal power flow (OPF) problem can be solved. Given the known vectors $\mathbf{P}_D$ and $\mathbf{Q}_D$, OPF minimizes the cost $\sum_{k\in\mathcal{N}} f_k(P_{G_k})$ over the unknown parameters $\mathbf{V}$, $\mathbf{P}_G$ and $\mathbf{Q}_G$ subject to the power balance equations at all buses as well as the physical constraints

$$P_k^{\min} \le P_{G_k} \le P_k^{\max}, \qquad \forall\ k \in \mathcal{N} \tag{4.1a}$$

$$Q_k^{\min} \le Q_{G_k} \le Q_k^{\max}, \qquad \forall\ k \in \mathcal{N} \tag{4.1b}$$

$$V_k^{\min} \le |V_k| \le V_k^{\max}, \qquad \forall\ k \in \mathcal{N} \tag{4.1c}$$

$$P_{lm} \le P_{lm}^{\max}, \qquad \forall\ (l,m) \in \mathcal{L} \tag{4.1d}$$

where the limits $P_k^{\min}, P_k^{\max}, Q_k^{\min}, Q_k^{\max}, V_k^{\min}, V_k^{\max}$, $P_{lm}^{\max} = P_{ml}^{\max}$ are given. Instead of the flow limit constraint (4.1d), one may impose a restriction on the value of $|V_l - V_m|$ or S_{lm}. The results to be presented in this work can be easily generalized to handle these constraints. Define the following notations:

- Let y_{lm} denote the mutual admittance between buses l and m, and y_{kk} denote the admittance-to-ground at bus k, for every $k \in \mathcal{N}$ and $(l,m) \in \mathcal{L}$.
- Let $\mathbf{Y}$ represent the admittance matrix of the power network (see [25] for an explicit expression of $\mathbf{Y}$).
- Define the current vector $\mathbf{I} := \begin{bmatrix} I_1 & I_2 & \cdots & I_n \end{bmatrix}^T$ as $\mathbf{YV}$, where I_k is the net current injected to bus $k \in \mathcal{N}$.
- Define $e_1, e_2, ..., e_n$ as the standard basis vectors in $\mathbf{R}^n$.

One can write:

$$\begin{aligned} &(P_{G_k} - P_{D_k}) + (Q_{G_k} - Q_{D_k})\mathrm{i} = \mathbf{V}_k \mathbf{I}_k^* \\ &= (e_k^* \mathbf{V})(e_k^* \mathbf{I})^* = \mathrm{trace}\{\mathbf{V}\mathbf{V}^*\mathbf{Y}^* e_k e_k^*\}, \quad k \in \mathcal{N}. \end{aligned} \tag{4.2}$$

Since the above equality constraint is nonlinear in $\mathbf{V}$, one may replace $\mathbf{VV}^*$ with a new matrix variable $\mathbf{W} \in \mathbf{H}^{n\times n}$ to make this constraint linear. However, in order to make the

map from $\mathbf{V}$ to $\mathbf{W}$ invertible, $\mathbf{W}$ must be constrained to be both positive semidefinite and rank-one. Hence, OPF can be formulated as:

OPF: Minimize the function

$$\sum_{k\in\mathcal{G}} f_k(P_{G_k}) \tag{4.3}$$

over $\mathbf{W} \in \mathbf{H}^{n\times n}$, $\mathbf{P}_G \in \mathbf{R}^n$ and $\mathbf{Q}_G \in \mathbf{R}^n$, subject to

$$P_k^{\min} \le P_{G_k} \le P_k^{\max}, \tag{4.4a}$$

$$Q_k^{\min} \le Q_{G_k} \le Q_k^{\max}, \tag{4.4b}$$

$$(V_k^{\min})^2 \le W_{kk} \le (V_k^{\max})^2, \tag{4.4c}$$

$$\mathrm{Re}\,\{(W_{ll} - W_{lm})y_{lm}^*\} \le P_{lm}^{\max}, \tag{4.4d}$$

$$\mathrm{trace}\{\mathbf{W}\mathbf{Y}^* e_k e_k^*\} = P_{G_k} - P_{D_k} + (Q_{G_k} - Q_{D_k})\mathrm{i}, \tag{4.4e}$$

$$\mathbf{W} = \mathbf{W}^* \succeq 0, \tag{4.4f}$$

$$\mathrm{rank}\,\{\mathbf{W}\} = 1 \tag{4.4g}$$

for every $k \in \mathcal{N}$ and $(l, m) \in \mathcal{L}$.

The details of the above formulation can be found in [25]. Note that if there is a transformer on the line (l, m), an extra term may be required in the left side of (4.4d), depending on how the transformer is modeled. Recall that transmission lines and transformers are passive (dissipative) devices. This implies that

$$\mathrm{Re}\{\mathbf{Y}\} \succeq 0, \qquad y_{lm}^* \ge 0, \quad \forall\ (l, m) \in \mathcal{L}. \tag{4.5}$$

It has been shown in [25] that although OPF is NP-hard in the worst case, it may be solved in polynomial-time for a large class of admittance matrices $\mathbf{Y}$ due to the above-mentioned physical properties of a power circuit, provided the cost function f_k is quadratic. Under the assumption (4.5), the objective of this chapter is threefold:

- The first goal is to extend the results of [25] to every convex function f_k.
- The second goal is to design a scalable algorithm for solving a large-scale OPF by exploiting the topology of the power network.
- The third goal is to show that due to the physics of a power network, every OPF

problem can be solved in polynomial time after the following approximations: (i) write power balance equations as inequalities, and (ii) place virtual (fictitious) phase shifters in certain loops of the network. As will be discussed later, approximation (i) does not change the solution to the OPF problem in many practical situations, and moreover a few virtual phase shifters are often enough in approximation (ii).

4.3 Main Results

Given an index $k \in \mathcal{N}$, define the convex conjugate function $\bar{f}_k(x) : \mathbf{R} \to \mathbf{R}$ as:

$$\bar{f}_k(x) = -\min_{P_{G_k}} \left(f_k(P_{G_k}) - xP_{G_k} \right), \qquad \forall\ x \in \mathbf{R}. \tag{4.6}$$

Let $\underline{\lambda}_k$, $\overline{\lambda}_k$ and λ_k denote the Lagrange multipliers for the power constraints $P_k^{\min} \leq P_{G_k}$, $P_{G_k} \leq P_k^{\max}$ and $\mathrm{Re}\{\mathrm{trace}(\mathbf{W}\mathbf{Y}^* e_k e_k^*)\} = P_{G_k} - P_{D_k}$, respectively. Define $\mathbf{\Theta}$ as the vector of all Lagrange mutipliers associated with OPF. In line with the technique used in [25], the Lagrangian dual of OPF can be written as:

$$\max_{\mathbf{\Theta}} \left\{ -\sum_{k \in \mathcal{N}} \bar{f}_k(\underline{\lambda}_k - \overline{\lambda}_k + \lambda_k) + h(\mathbf{\Theta}) \right\} \tag{4.7a}$$

$$\text{subject to} \qquad A(\mathbf{\Theta}) \succeq 0 \tag{4.7b}$$

where $h(\mathbf{\Theta}) \in \mathbf{R}$ is a linear function and $A(\mathbf{\Theta}) \in \mathbf{H}^{n \times n}$ is a linear matrix function. The above convex optimization, referred to as *Dual OPF*, has a concave objective and an LMI constraint. Hence, this optimization can be solved efficiently in polynomial time [29]. However, its optimal objective value is only a lower bound on the optimal objective value of OPF. Whenever OPF and Dual OPF have the same optimal values, it is said that *strong duality holds* or *duality gap is zero for OPF*. The duality gap will be studied in the subsequent subsections.

4.3.1 Various SDP Relaxations and Zero Duality Gap

Define $\mathcal{G} := (\mathcal{N}, \mathcal{L})$ as the graph corresponding to the power network. With no loss of generality, assume that $\mathcal{G}$ is a connected graph (otherwise, it can be partitioned into a set of disconnected sub-networks). This graph may have several cycles, which all together

establish a cycle space of dimension $|\mathcal{L}|-|\mathcal{N}|+1$. Let $\{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}\}$ be an arbitrary basis for this cycle space, meaning that $\mathcal{C}_1, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}$ are all cycles of $\mathcal{G}$ from which every other cycle of $\mathcal{G}$ can be constructed.

Definition 1. *Define the subgraph set $\mathcal{S}$ as*

$$\mathcal{S} := \mathcal{L} \cup \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}\} \tag{4.8}$$

(note that since each edge of $\mathcal{G}$ can be regarded as a two-vertex subgraph, $\mathcal{L}$ is indeed a set consisting of $|\mathcal{L}|$ subgraphs).

Definition 2. *Given a Hermitian matrix $\mathbf{W} \in \mathbf{H}^{n\times n}$ and an arbitrary subgraph $\mathcal{G}_s \in \mathcal{S}$, define $\mathbf{W}(\mathcal{G}_s)$ as a matrix obtained from $\mathbf{W}$ by removing those columns and rows of $\mathbf{W}$ whose indices do not appear in the vertex set of $\mathcal{G}_s$. Note that $\mathbf{W}(\mathcal{G}_s)$ is a submatrix of $\mathbf{W}$ corresponding to the subgraph $\mathcal{G}_s$ of $\mathcal{G}$.*

To clarify Definition 2, let $\mathcal{G}_s$ be a single edge $(l, m) \in \mathcal{L}$. Since the vertex set of this subgraph has only two elements l and m, one can write:

$$\mathbf{W}(\mathcal{G}_s) = \begin{bmatrix} W_{ll} & W_{lm} \\ W_{ml} & W_{mm} \end{bmatrix} \tag{4.9}$$

where W_{lm} denotes the (l, m) entry of $\mathbf{W}$. Three convex relaxations of OPF will be introduced below.

Relaxed OPF 1 (ROPF 1): This optimization is obtained from the OPF problem formulated in (4.3)–(4.4) by removing its rank constraint (4.4g).

Relaxed OPF 2 (ROPF 2): This optimization is obtained from ROPF 1 by replacing its constraint $\mathbf{W} \succeq 0$ with the set of constraints

$$\mathbf{W}(\mathcal{G}_s) \succeq 0, \qquad \forall\ \mathcal{G}_s \in \mathcal{S}. \tag{4.10}$$

Relaxed OPF 3 (ROPF 3): This optimization is obtained from ROPF 2 by replacing its constraint (4.10) with

$$\mathbf{W}(\mathcal{G}_s) \succeq 0, \qquad \forall\ \mathcal{G}_s \in \mathcal{L} \tag{4.11}$$

or equivalently

$$W_{11}, W_{22}, ..., W_{nn} \geq 0, \tag{4.12a}$$

$$W_{ll} W_{mm} \geq |W_{lm}|^2, \qquad \forall\ (l,m) \in \mathcal{L}. \tag{4.12b}$$

Note that ROPF 1 and ROPF 2 have convex objectives with SDP constraints, whereas ROPF 3 has a convex objective with SOCP constraints. ROPF 2 has a lower computational complexity compared to ROPF 1 because only submatrices of $\mathbf{W}$ corresponding to edges and certain cycles of the network are required to be positive semidefinite. ROPF 3 is even simpler than ROPF 2 because it imposes constraints only on the submatrices of $\mathbf{W}$ corresponding to the edges of the network. The relations among OPF, Dual OPF and ROPF 1-3 will be studied in the sequel.

Theorem 1. *The following statements hold:*

- *The duality gap is zero for OPF if and only if ROPF 1 has a solution* $(\mathbf{W}^{opt}, \mathbf{P}_G^{opt}, \mathbf{Q}_G^{opt})$ *with the property rank*$\{\mathbf{W}^{opt}\} = 1$*, in which case a global solution* $\mathbf{V}^{opt}$ *can be found using the equation* $\mathbf{W}^{opt} = \mathbf{V}^{opt}(\mathbf{V}^{opt})^T$.

- *The duality gap is zero if* $A(\mathbf{\Theta}^{opt})$ *in Dual OPF has rank* $n-1$*, in which case a global solution* $\mathbf{V}^{opt}$ *can be found using the equation* $A(\mathbf{\Theta}^{opt})\mathbf{V}^{opt} = 0$.

Proof: This theorem is a natural extension of the results of [25], which were developed for quadratic cost functions and real-valued Dual OPF (rather than complex Dual OPF). The techniques used in [25] can be used to prove this theorem. ■

Although ROPF 1 can be solved in polynomial time, it has a matrix variable $\mathbf{W}$ with n^2 unknown entries. Since the number of scalar variables of ROPF 1 is on the order of $O(n^2)$, this optimization may not be solved efficiently for a large value of n. The same argument is valid for Dual OPF. Due to this drawback, the goal is to reduce the computational complexity of these optimizations.

Lemma 1. *Given a Hermitian matrix* $\mathbf{W} \in \mathbf{H}^{n \times n}$*, the following two statements are equivalent:*

i) There exists a matrix $\mathbf{W}^{(1)} \in \mathbf{H}^{n\times n}$ *such that*

$$W^{(1)}_{lm} = W_{lm},\ \forall\ (l,m) \in \mathcal{L} \cup \{(1,1),...,(n,n)\} \tag{4.13a}$$

$$\mathbf{W}^{(1)} \succeq 0, \tag{4.13b}$$

$$rank\{\mathbf{W}^{(1)}\} = 1. \tag{4.13c}$$

ii) There exists a matrix $\mathbf{W}^{(2)} \in \mathbf{H}^{n\times n}$ *such that*

$$W^{(2)}_{lm} = W_{lm},\ \forall\ (l,m) \in \mathcal{L} \cup \{(1,1),...,(n,n)\} \tag{4.14a}$$

$$\mathbf{W}^{(2)}(\mathcal{G}_s) \succeq 0, \qquad \forall\ \mathcal{G}_s \in \mathcal{S} \tag{4.14b}$$

$$rank\{\mathbf{W}^{(2)}(\mathcal{G}_s)\} = 1, \qquad \forall\ \mathcal{G}_s \in \mathcal{S}. \tag{4.14c}$$

Proof: The proof has been moved to the appendix of Chapter 4. ■

Lemma 1 will be exploited next to propose a simpler method for checking the duality gap of OPF, as an alternative to solving ROPF 1,

Theorem 2. *The duality gap is zero for OPF if ROPF 2 has a solution* $(\mathbf{W}^{opt}, \mathbf{P}_G^{opt}, \mathbf{Q}_G^{opt})$ *with the property that* $rank\{\mathbf{W}^{opt}(\mathcal{G}_s)\} = 1$ *for every* $\mathcal{G}_s \in \mathcal{S}$.

Proof: Given an arbitrary matrix $\mathbf{W}$, notice that an entry W_{lm} of the matrix $\mathbf{W}$ does not appear in the constraints (4.4a)–(4.4e) of OPF unless $l = m$ or $(l,m) \in \mathcal{L}$, which means that some entries of $\mathbf{W}$ may not be important. The proof follows from this fact, Lemma 1 and Theorem 1. ■

4.3.2 Acyclic Networks

Throughout this subsection, assume that the power network is radial so that the graph $\mathcal{G}$ has no cycle. Note that this network does not necessarily represent a distribution network with a single feeder (generator), and indeed it can have an arbitrary number of generators.

Theorem 3. *The duality gap is zero for OPF if and only if ROPF 3 has an optimal solution for which every inequality in (4.12b) becomes an equality.*

Proof: Since a cyclic graph is chordal, it can be inferred from the matrix completion theorem that ROPF 1 and ROPF 3 have the same optimal objective value. The proof follows from this fact, Theorem 1 and Lemma 1. ■

As pointed out earlier, the SDP constraint $\mathbf{W} \succeq 0$ in ROPF 1 makes it hard to solve the problem numerically for a large value of n. Nonetheless, Corollary 3 states that this constraint can be replaced by the SOCP constraint (4.12), and yet the zero duality gap of OPF can be verified from this optimization.

Remark 1. *The paper [22] proposes an SOCP relaxation for the power flow problem in the radial case, by formulating the problem in terms of the variables* $|V_k|^2$, $|V_l||V_m|\cos(\measuredangle V_l - \measuredangle V_m)$ *and* $|V_l||V_m|\sin(\measuredangle V_l - \measuredangle V_m)$ *for every* $k \in \mathcal{N}$ *and* $(l,m) \in \mathcal{L}$. *It can be shown that this relaxation is tantamount to ROPF 3. As a result, the SOCP relaxation provided in [22] for solving the power flow problem in radial networks works correctly if and only if the duality gap is zero for the corresponding power flow problem.*

The power balance equations

$$\text{trace}\{\mathbf{W}\mathbf{Y}^* e_k e_k^*\} = P_{G_k} - P_{D_k} + (Q_{G_k} - Q_{D_k})\text{i} \tag{4.15}$$

$\forall\ k \in \mathcal{N}$, appear in OPF and ROPF 1–3. It is said that *load over-satisfaction (power over-delivery) is allowed* if these equality constraints in the aforementioned optimizations are permitted to be replaced by

$$\text{trace}\{\mathbf{W}\mathbf{Y}^* e_k e_k^*\} \leq P_{G_k} - P_{D_k} + (Q_{G_k} - Q_{D_k})\text{i}. \tag{4.16}$$

The main idea behind this notion is that power over delivery to a bus is allowed, in which case the excess power should be thrown away (wasted/stored). However, it is generally true that even when load over-satisfaction is permitted, a practical power network is maintained in a normal condition so that almost all nodes of the network receive no extra power for free. This is due to two properties: (i) transmission lines are lossy, and (ii) cost funcntions are monotonically increasing. Note that the notion of load over-satisfaction has already been used by other researchers [71, 75]. This notion has also been studied in the recent work [25] via the name *modified OPF (MOPF)*. In the case when some of the power balance inequalities do not bind at an optimal solution of MOPF due to highly congested transmission lines, the obtained solution can still be used as an approximation of the globally optimal solution of OPF. In the rest of this chapter, MOPF and RMOPF 1–3 will refer to the optimizations OPF and ROPF 1–3 in the load over-satisfaction case. It can be easily

shown that all of the results derived so far hold true when load over-satisfaction is allowed.

The goal of this part is to show that the duality gap is zero for acyclic networks, when load over-satisfaction is allowed. To this end, the following lemma is needed, which holds for both acyclic and cyclic networks. This lemma is primarily based on the physical properties of a power network (i.e., passivity of transmission lines).

Lemma 2. *RMOPF 3 has an optimal solution for which every inequality in (4.12b) becomes an equality.*

Proof: Consider an arbitrary solution $(\mathbf{W}^{\text{opt}}, \mathbf{P}_G^{\text{opt}}, \mathbf{Q}_G^{\text{opt}})$ of RMOPF 3. Define $\widehat{\mathbf{W}}^{\text{opt}}$ as a matrix whose $(l,m) \in \mathcal{N} \times \mathcal{N}$ entry, denoted by $\widehat{W}_{lm}^{\text{opt}}$, is equal to W_{lm}^{opt} if $(l,m) \notin \mathcal{L}$ and

$$\widehat{W}_{lm}^{\text{opt}} = \sqrt{W_{ll}^{\text{opt}} W_{mm}^{\text{opt}} - \Big(\text{Im}\{W_{lm}^{\text{opt}}\}\Big)^2} + \text{Im}\{W_{lm}^{\text{opt}}\}\text{i}$$

otherwise. It is evident that inequality (4.12b) becomes an equality for $\mathbf{W} = \widehat{\mathbf{W}}^{\text{opt}}$. Furthermore, given an index $(l,m) \in \mathcal{L}$, since (4.12b) is satisfied for $\mathbf{W} = \mathbf{W}^{\text{opt}}$, one can write

$$\begin{aligned} &(\widehat{W}_{ll}^{\text{opt}} - \widehat{W}_{lm}^{\text{opt}}) y_{lm}^* - \Big(W_{ll}^{\text{opt}} - W_{lm}^{\text{opt}}\Big) y_{lm}^* \\ &= \left(\text{Re}\{W_{lm}^{\text{opt}}\} - \sqrt{W_{ll}^{\text{opt}} W_{mm}^{\text{opt}} - \text{Im}\{W_{lm}^{\text{opt}}\}^2}\right) y_{lm}^* \leq 0. \end{aligned} \tag{4.17}$$

Note that the above inequality is inferred from the non-negativity of the real and imaginary parts of y_{lm}^*. Besides,

$$\begin{aligned} &\text{trace}\{\widehat{\mathbf{W}}^{\text{opt}} \mathbf{Y}^* e_k e_k^*\} - \text{trace}\{\mathbf{W}^{\text{opt}} \mathbf{Y}^* e_k e_k^*\} \\ &= \sum_{l \in \mathcal{N}(k)} \left(\text{Re}\{W_{kl}^{\text{opt}}\} - \sqrt{W_{kk}^{\text{opt}} W_{ll}^{\text{opt}} - \text{Im}\{W_{kl}^{\text{opt}}\}^2}\right) y_{kl}^* \leq 0 \end{aligned}$$

for every $k \in \mathcal{N}$. This inequality, together with (4.17), yields that $(\widehat{\mathbf{W}}^{\text{opt}}, \mathbf{P}_G^{\text{opt}}, \mathbf{Q}_G^{\text{opt}})$ is another solution of RMOPF 3 for which every inequality in (4.12b) becomes an equality. ■

Theorem 4. *The duality gap is zero for OPF if load over-satisfaction is allowed.*

Proof: The proof follows immediately from Theorem 3 and Lemma 2. ■

4.3.3 General Networks

In this part, the results of the preceding subsection will be generalized to the case when the graph $\mathcal{G}$ has at least one cycle (loop). As proved earlier, RMOPF 3 always has a desirable solution because of the physical properties of a power network. On the other hand, RMOPF 1 is the exact formulation of MOPF in the case of zero duality gap. Nonetheless, RMOPF 3 may not be equivalent to RMOPF 1 unless there is no cycle in the network. To fix this, we aim to study what minor modifications are needed in the topology of the network so that RMOPFs 1-3 all give the same solution in the absence of a duality gap.

Define *controllable phase shifter* as an ideal (lossless) phase-shifting transformer with the ratio $e^{\gamma \mathrm{i}}$, where the phase shift γ is a variable of the OPF problem. If there are some controllable phase shifters in the network, the term *OPF* refers to the optimal power flow problem with the variables $\mathbf{V}, \mathbf{P}_G, \mathbf{Q}_G$ and the phase shifts of these transformers. Note that adding a controllable phase shifter to a transmission line may require reformulating OPF to incorporate the unknown phase shift of the transformer. The objective is to investigate the role of controllable phase shifters in diminishing the duality gap of OPF.

A bridge of the graph $\mathcal{G}$ is an edge of this graph whose removal makes $\mathcal{G}$ disconnected. The next theorem studies the importance of a phase shifter installed on a bridge line.

Lemma 3. *Assume that $(l,m) \in \mathcal{L}$ is a bridge of the graph $\mathcal{G}$ and that the line (l,m) of the power network has a controllable phase shifter. The OPF problem has a globally optimal solution for which the optimal phase shift of the phase shifter is 0.*

Proof: Define $\mathcal{N}_1$ and $\mathcal{N}_2$ as the sets of vertices of the two disconnected subgraphs obtained by removing the edge (l,m) from the graph $\mathcal{G}$. Assume that $l \in \mathcal{N}_1$ and that the phase shifter of the line (l,m) is located on the side of bus l. Let $(\mathbf{V}^{\mathrm{opt}}, \mathbf{P}_G^{\mathrm{opt}}, \mathbf{Q}_G^{\mathrm{opt}}, \gamma^{\mathrm{opt}})$ denote an optimal solution of OPF, where γ represents the phase of the phase-shifting transformer on the line (l,m). Define $\tilde{\mathbf{V}}^{\mathrm{opt}} \in \mathbf{R}^n$ with the entries

$$\tilde{V}_j^{\mathrm{opt}} = \begin{cases} V_j^{\mathrm{opt}} e^{\gamma^{\mathrm{opt}} \mathrm{i}} & \text{if } j \in \mathcal{N}_1 \\ V_j^{\mathrm{opt}} & \text{if } j \in \mathcal{N}_2. \end{cases}$$

Note that $\tilde{V}_j^{\mathrm{opt}}$ is uniquely defined above, because of the relations $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$ and $\mathcal{N}_1 \cap \mathcal{N}_2 = \phi$. It is straightforward to observe that $(\tilde{\mathbf{V}}^{\mathrm{opt}}, \mathbf{P}_G^{\mathrm{opt}}, \mathbf{Q}_G^{\mathrm{opt}}, 0)$ is another solution of OPF. This completes the proof. ■

Lemma 3 states that a controllable phase shifter on a bridge line of the power network has no effect on the optimal value of OPF. In particular, since every line of a radial network is a bridge, as far as the OPF problem is concerned, phase shifters are not useful for this type of network. In contrast, adding a phase shifter to a non-bridge line of a cyclic network may improve the performance of the network. From the optimization perspective, this addition may require the modification of the formulation of OPF by introducing new variables.

Given a natural number t, assume that t phase shifters are added to the lines (l_j, m_j) for $j = 1, ..., t$. Let each phase shifter j be located on the side of bus l_j, with the variable phase shift γ_j. One can write:

$$S_{l_j m_j} = |V_{l_j}|^2 \left(y^*_{l_j m_j} + \frac{1}{2} b_{l_j m_j} \mathrm{i} \right) - V_{l_j} V^*_{m_j} y^*_{l_j m_j} e^{\gamma_j \mathrm{i}} \tag{4.18}$$

where $b_{l_j m_j}$ denotes the charging susceptance of the line (l_j, m_j). In the previous formulation of OPF, $W_{l_j m_j} = W^*_{m_j l_j}$ represented the parameter $V_{l_j} V^*_{m_j}$. In order to account for the inclusion of the phase shifters, the equation (4.18) suggests two modifications:

- Use the new notations $\overline{W}_{l_j m_j} = \overline{W}^*_{l_j m_j}$ for $V_{l_j} V^*_{m_j}$.
- Use the previous notations $W_{l_j m_j} = W^*_{m_j l_j}$ for $V_{l_j} V^*_{m_j} e^{\gamma_j \mathrm{i}}$.

This implies that the following modifications must be made to the OPF problem formulated in (4.3) and (4.4):

- Introduce a new matrix variable $\overline{\mathbf{W}} \in \mathbf{H}^{n \times n}$.
- Replace the constraints $\mathbf{W} \succeq 0$ and $\mathrm{rank}\{\mathbf{W}\} = 1$ with $\overline{\mathbf{W}} \succeq 0$ and $\mathrm{rank}\{\overline{\mathbf{W}}\} = 1$.
- Add the new constraints

$$\mathbf{W}(\mathcal{G}_s) \succeq 0, \quad \mathrm{rank}\{\mathbf{W}(\mathcal{G}_s)\} = 1, \qquad \forall\, \mathcal{G}_s \in \mathcal{L}.$$

- Impose the constraint that the corresponding entries of $\mathbf{W}$ and $\overline{\mathbf{W}}$ are equal to each other, with the exception of the entries $(l_1, m_1), ..., (l_t, m_t)$ and $(m_1, l_1), ..., (m_t, l_t)$.

One can drop the rank constraints in the above formulation of OPF to obtain a convexified problem. We prove in the next theorem that if the phase shifters are added to the network

in a certain way, the formulation of OPF becomes even simpler than the case with no phase shifters and indeed the new variables $\overline{W}_{l_j m_j}$'s need not be introduced.

Theorem 5. *Consider a subset of the cycle basis $\{\mathcal{C}_1, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}\}$, say $\mathcal{C}_1, ..., \mathcal{C}_t$ for a given number $t \leq |\mathcal{L}| - |\mathcal{N}| + 1$. For every $j \in \{1, 2, ..., t\}$, assume that a controllable phase shifter is added to a line (l_j, m_j) of the cycle $\mathcal{C}_j$ such that*

i) The graph $\mathcal{G}$ remains connected after removing the edges $(l_1, m_1), ..., (l_t, m_t)$.

ii) The set $\{l_j, m_j\}$ is not a subset of the vertex set of any of the reminding cycles $\mathcal{C}_{t+1}, , ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}$.

Consider the optimization obtained from ROPF 2 by replacing its constraint (4.10) with the reduced set of constraints

$$\mathbf{W}(\mathcal{G}_s) \succeq 0, \qquad \forall\ \mathcal{G}_s \in \mathcal{S} \backslash \{\mathcal{C}_1, ..., \mathcal{C}_t\}. \tag{4.19}$$

The duality gap is zero for OPF if every submatrix $\mathbf{W}(\mathcal{G}_s)$ in the above inequality becomes rank-one at an optimal solution.

Proof: Let $(\mathbf{W}^{\text{opt}}, \mathbf{P}_G, \mathbf{Q}_G)$ denote an optimal solution of the optimization in Theorem 5 for which every matrix $\mathbf{W}(\mathcal{G}_s)$ becomes rank-one. In line with the argument made in the proof of Lemma 1, it can be shown that

$$\sum_{(l,m) \in \vec{\mathcal{C}_j}} \measuredangle W_{lm}^{\text{opt}} = 0, \quad \forall\ j \in \{t+1, ..., |\mathcal{L}| - |\mathcal{N}| + 1\}. \tag{4.20}$$

By Assumption (i) of the theorem, if the edges (l_1, m_1), ..., (l_t, m_t) are removed from $\mathcal{G}$, then $\{\mathcal{C}_{t+1}, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}\}$ forms a cycle basis for the resulting subgraph. In light of (4.20), this implies that there exist angles $\theta_1^{\text{opt}}, ..., \theta_n^{\text{opt}}$ such that $\theta_l - \theta_m = \measuredangle W_{lm}^{\text{opt}}$ for every edge (l, m) of this subgraph. Now, define the following phase shifts and voltage parameters:

$$\begin{aligned} \gamma_j^{\text{opt}} &= \measuredangle W_{l_j m_j}^{\text{opt}} - \theta_{l_j}^{\text{opt}} + \theta_{m_j}^{\text{opt}}, \quad \forall\ j \in \{1, ..., t\} \\ V_k^{\text{opt}} &= \sqrt{W_{kk}^{\text{opt}}}\ \measuredangle \theta_k^{\text{opt}}, \qquad\qquad \forall\ k \in \mathcal{N}. \end{aligned}$$

It can be verified that the above parameters correspond to a global solution of OPF and that the duality gap is zero. ■

An implication of Theorem 5 is that adding phase shifters to the network in a certain way simplifies the formulation of OPF, instead of increasing the number of variables and/or constraints. This theorem shows that the phase shifters added to the cycles $\mathcal{C}_1, ..., \mathcal{C}_t$ give rise to the exclusion of the t constraints $\mathbf{W}(\mathcal{C}_s) \succeq 0$, $s = 1, ..., t$, from ROPF 2.

A spanning tree of the connected graph $\mathcal{G}$ is an acyclic subgraph of $\mathcal{G}$ with $|\mathcal{N}|$ vertices and $|\mathcal{N}| - 1$ edges. Note that $\mathcal{G}$ might have an exponential number of spanning trees.

Corollary 1. *Given a spanning tree $\mathcal{T}$ of the graph $\mathcal{G}$, assume that a controllable phase shifter is added to every line of the network not belonging to this tree. The duality gap is zero for OPF if and only if ROPF 3 has an optimal solution for which every inequality in (4.12b) becomes an equality.*

Proof: Let $(l_1, m_1), ..., (l_t, m_t)$ denote those edges of the graph $\mathcal{G}$ that do not belong to $\mathcal{T}$, where $t = |\mathcal{L}| - |\mathcal{N}| + 1$. Adding each edge (l_j, m_j), $j = 1, ..., t$, to the tree $\mathcal{T}$ creates a cycle. With a slight abuse of notation, let $\mathcal{C}_j$ denote this cycle. Theorem 5 can be applied to the power network with the phase shifters installed on its lines $(l_1, m_1), ..., (l_t, m_t)$. The proof is completed by noting that constraint (4.19) is equivalent to (4.12), i.e., the optimization introduced in Theorem 5 and ROPF 3 are identical for this set of phase shifters. ■

In the case of OPF with no controllable phase shifters, one needs to solve Dual OPF, ROPF 1 or ROPF 2, which have SDP constraints. Nonetheless, Corollary 1 states that if a sufficient number of phase shifters is added to the network, then it suffices to solve ROPF 3 with simple SOCP constraints. Hence, phase shifters can significantly reduce the computational complexity of OPF if they are formulated properly. Note that ROPF 3 is independent of the choice of the non-unique spanning tree $\mathcal{T}$. This introduces some flexibility in the locations of the phase shifters.

Theorem 6. *Given an arbitrary spanning tree $\mathcal{T}$ of the graph $\mathcal{G}$, assume that a controllable phase shifter is added to every line of the network that does not belong to this tree. The duality gap is zero for OPF if load over-satisfaction is allowed.*

Proof: The proof can be deduced from Lemma 2 and Corollary 1. The details are omitted for brevity. ■

It can be inferred from Corollary 1 and Theorem 6 that adding a sufficient number of phase shifters guarantees that a global solution of OPF can be found by solving an SOCP

optimization. Note that this zero-duality-gap property is due to the (weighted) topology of the network and holds for all possible values of loads, physical limits and cost functions. The application of this result is twofold:

- Every OPF problem can be solved in polynomial time after two approximations: (i) convert the power balance equations to inequality constraints, (ii) assume the presence of a sufficient number of virtual phase shifters in the network.

- Every network topology can be turned into a "good" one by adding phase shifters to achieve two goals: improve the performance of the network (e.g., to relieve congestion), (ii) make every OPF problem defined on the modified network solvable in polynomial time. This result could be used for transmission system planning.

As will be shown in simulations, only a few actual or virtual phase shifters may be sufficient in practice.

4.4 Examples

Example 1: To illustrate the efficacy of Theorem 2, consider the network depicted in Figure 4.2. This network consists of three acyclic (radial) subnetworks $1-10$, $11-20$ and $21-30$, which are interconnected via the cycle (transmission network) $\{1, 11, 21\}$. Zero duality gap can be verified from both ROPF 1 and ROPF 2. In fact, the solutions of ROPF 1 and ROPF 2 are identical as the graph under study is chordal. Note that ROPF 2 exploits the sparsity of the network and therefore is much easier to solve. More precisely,

- ROPF 1 has a 30×30 matrix constraint $\mathbf{W} \succeq 0$ in which $\frac{30\times 31}{2} = 465$ scalar complex variables are involved.

- The graph $\mathcal{G}$ has 30 edges (i.e. $|\mathcal{L}| = 30$) and a single cycle $\{1, 11, 21\}$. Thus, ROPF 2 has 31 matrix constraints

$$\begin{bmatrix} W_{ll} & W_{lm} \\ W_{ml} & W_{mm} \end{bmatrix} \succeq 0, \quad \forall\ (l,m) \in \mathcal{L},\ l < m \tag{4.21}$$

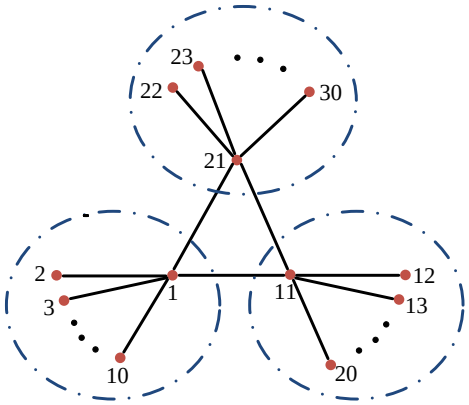


Figure 4.2: Power network used to illustrate Theorem 2.

and

$$\begin{bmatrix} W_{1,1} & W_{1,11} & W_{1,21} \\ W_{11,1} & W_{11,11} & W_{11,21} \\ W_{21,1} & W_{21,11} & W_{21,21} \end{bmatrix} \succeq 0 \tag{4.22}$$

(note that $W_{l,m}$ stands for W_{lm}). In light of (4.22), constraint (4.21) need not be written for the edges $(1, 11)$, $(1, 21)$ and $(11, 21)$. Hence, ROPF 2 is nearly an SOCP optimization with 28 non-redundant 2×2 and 3×3 matrix constraints in which only 60 scalar variables are involved. Therefore, several entries of $\mathbf{W}$ never appear in the constraints of ROPF 2 and can be simply ignored.

Example 2: Let the results of this chapter be illustrated on IEEE systems [88]. Similar to [25], a small amount of resistance (10^{-5} per unit) is added to a few purely-inductive lines of these networks. The simulations performed here are run on a computer with a Pentium IV 3.0 GHz and 3.62 GB of memory. The toolbox "YALMIP", together with the solvers "SEDUMI" and "SDPT3", is used to solve different LMI problems, where the numerical tolerance is chosen as 10^{-15}.

Consider the IEEE 30-bus system with $f_k(P_{G_k}) = P_{G_k}$, $\forall k \in \mathcal{G}$, which has 30 buses and 41 lines. Dual OPF can be solved in 1.2 seconds for this power network to detect the zero duality gap and attain the optimal generation cost 191.09. Alternatively, one can solve ROPF 1 in 9.3 seconds to verify the zero duality gap. Assume now that every line of the network has a controllable phase shifter. Due to the results developed here, at most $41 - 30 + 1$ of the 41 phase shifters are important and the remaining ones can be simply ignored. The duality gap is zero for OPF with 12 controllable phase shifters. Indeed,

ROPF 3 can be solved in only 0.4 second, leading to the optimal value 190.66. Note that (i) most of the phase shifters have negligible effect, and (ii) even if load over-satisfaction is allowed, it will never occur. As another experiment, suppose that $f_k(P_{G_k})$ is the quadratic cost function specified in [88]. The solutions of ROPF 1 and ROPF 3 turn out to be 576.90 and 573.59, respectively. To substantiate that most of the 12 controllable phase shifters are not important, notice that

- OPF with a single variable phase shifter on the line $(25, 27)$ has the optimal cost 573.92, corresponding to the phase 7.82°.
- OPF with two variable phase shifters on the lines $(25, 27)$ and $(8, 28)$ gives the optimal cost 573.67, corresponding to the optimal phases 5.70° and -0.30°.

We have repeated the above experiment for several random cost functions and observed that OPF with 1-2 phase shifters is a good approximation of ROPF 3.

Consider now the IEEE 118-bus system with $f_k(P_{G_k}) = P_{G_k}$, $\forall k \in \mathcal{G}$. A global solution of OPF is found by solving Dual OPF in 11.2 seconds, leading to the optimal cost 4251.9. ROPF 1 can also detect the zero duality gap, but its running time is more than 1 minute. In contrast, ROPF 3 is solved in 0.9 seconds to attain the optimal value 4251.9. Note that (i) OPF and ROPF 3 have the same optimal value, and (ii) the duality gap is zero for OPF without phase shifters. Hence, the total generation $\sum_{k\in\mathcal{G}} P_{G_k}$ is never reduced by adding phase shifters to the lines of the network. However, phase shifters may have an important role for other types of cost functions.

4.5 Summary

It has been recently shown that several practical instances of the optimal power flow (OPF) problem can be solved in polynomial time as long as the objective function is quadratic. The present work first generalizes this result to arbitrary convex functions and then studies how the presence of phase shifters in the network guarantees solvability of OPF in polynomial time. A global solution of OPF can be found from the dual of OPF if the duality gap is zero, or alternatively if a linear matrix inequality (LMI) optimization derived here has a specific solution. It is shown that the computational complexity of verifying the duality gap may be reduced significantly by exploiting the sparsity of the power network's topology. In

particular, if the network has no cycle, the LMI problem can be equivalently converted to a generalized second-order cone program. It is also proved that the integration of controllable phase shifters with variable phases into the cycles of the network makes the verification of the duality gap easier. More importantly, if every cycle (loop) of the network has a line with a controllable phase shifter, then OPF with variable phase shifters is guaranteed to be solvable in polynomial time, provided load over-satisfaction is allowed. This result implies that every OPF problem is guaranteed to be solvable in polynomial time after two modifications.

4.6 Appendix

Proof of Lemma 1: In order to prove that Condition (i) implies Condition (ii), consider a matrix $\mathbf{W}^{(1)}$ satisfying the relations given in (4.13). Since $\mathbf{W}^{(1)}$ is both rank-one and positive semidefinite, its principal minors $\mathbf{W}^{(1)}(\mathcal{G}_s)$, $\mathcal{G}_s \in \mathcal{S}$, are also rank-one and positive semidefinite. Hence, Condition (ii) holds if $\mathbf{W}^{(2)}$ is taken as $\mathbf{W}^{(1)}$.

Now, assume that Condition (ii) is satisfied for a matrix $\mathbf{W}^{(2)}$. The goal is to find a matrix $\mathbf{W}^{(1)}$ for which Condition (i) holds. To this end, notice that

$$\sum_{(l,m)\in\vec{\mathcal{C}}_j} \measuredangle W^{(2)}_{lm} = 0, \qquad j = 1, ..., |\mathcal{L}| - |\mathcal{N}| + 1 \tag{4.23}$$

where $\vec{\mathcal{C}}_j$ denotes a directed cycle obtained from $\mathcal{C}_j$ by giving an appropriate orientation to each edge of the cycle and $\measuredangle$ represents the phase of a complex number. Regard the graph $\mathcal{G}$ as a weighted directed graph, where the weights $\measuredangle W^{(2)}_{lm}$ and $\measuredangle W^{(2)}_{ml}$ are assigned to each edge $(l, m) \in \mathcal{L}$ in the forward and backward directions, respectively. Equation (4.23) can be interpreted as the directed sum of the edge weights around every cycle $\vec{\mathcal{C}}_j$ being zero. Since the set $\{\mathcal{C}_1, ..., \mathcal{C}_{|\mathcal{L}|-|\mathcal{N}|+1}\}$ constitutes a basis for the cycle space of the graph $\mathcal{G}$, it can be concluded that the relation (4.23) holds even if the cycle $\vec{\mathcal{C}}_j$ is replaced by an arbitrary directed cycle of the graph $\mathcal{G}$. Therefore, it is straightforward to show that the n vertices of the graph $\mathcal{G}$ can be labeled by some angles $\theta_1, ..., \theta_n$ such that

$$\measuredangle W^{(2)}_{lm} = \theta_l - \theta_m, \qquad \forall\ (l, m) \in \mathcal{L}. \tag{4.24}$$

Define $\mathbf{W}^{(1)}$ to be a matrix with the $(l,m) \in \mathcal{N} \times \mathcal{N}$ entry

$$W_{lm}^{(1)} = \sqrt{W_{ll}^{(2)}} \sqrt{W_{mm}^{(2)}} \measuredangle(\theta_l - \theta_m). \tag{4.25}$$

It is easy to observe that (4.13b) and (4.13c) are satisfied for this choice of $\mathbf{W}^{(1)}$. On the other hand, (4.13a) obviously holds for an index $(l,m) \in \{(1,1),...,(n,n)\}$. It remains to show the validity of (4.13a) for an edge $(l,m) \in \mathcal{L}$. One can write (4.14c) for the subgraph $\mathcal{G}_s = (l,m)$ to obtain

$$W_{ll}^{(2)} W_{mm}^{(2)} = \left| W_{lm}^{(2)} \right|^2.$$

Thus, it follows from (4.14a), (4.24) and (4.25) that

$$\begin{aligned} W_{lm} = W_{lm}^{(2)} &= \sqrt{W_{ll}^{(2)} W_{mm}^{(2)}} \measuredangle W_{lm}^{(2)} \\ &= \sqrt{W_{ll}^{(2)} W_{mm}^{(2)}} \measuredangle(\theta_l - \theta_m) = W_{lm}^{(1)}. \end{aligned} \tag{4.26}$$

This completes the proof. ■

Chapter 5

Convexification of Generalized Network Flow Problem

This chapter is concerned with the minimum-cost flow problem over an arbitrary flow network. In this problem, each node is associated with some possibly unknown injection, each line has two unknown flows at its ends related to each other via a nonlinear function, and all injections and flows need to satisfy certain box constraints. This problem, named generalized network flow (GNF), is highly non-convex due to its nonlinear equality constraints. Under the assumption of monotonicity and convexity of the flow and cost functions, a convex relaxation is proposed, which always finds the optimal injections. This relaxation may fail to find optimal flows because the mapping from injections to flows may not be unique. A primary application of this work is in optimization over power networks. Recent work on the optimal power flow (OPF) problem has shown that this non-convex problem can be solved efficiently using semidefinite programming (SDP) after two approximations: relaxing angle constraints (by adding virtual phase shifters) and relaxing power balance equations to inequality constraints. The results of this work on GNF prove that the second relaxation (on balance equations) is not needed in practice under a very mild angle assumption.

5.1 Introduction

The area of "network flows" plays a central role in operations research, computer science and engineering [89, 26]. This area is motivated by many real-word applications in assignment, transportation, communication networks, electrical power distribution, production scheduling, financial budgeting, and aircraft routing, to name only a few. Started

by the classical book [90] in 1962, network flow problems have been studied extensively [91, 92, 93, 94, 95, 96, 97, 98, 99].

The minimum-cost flow problem aims to optimize the flows over a flow network that is used to carry some commodity from suppliers to consumers. In a flow network, there is an injection of some commodity at every node, which leads to two flows over each line (arc) at its endpoints. The injection—depending on being positive or negative, corresponds to supply or demand at the node. The minimum-cost flow problem has been studied thoroughly for a lossless network, where the amount of flow entering a line equals the amount of flow leaving the line. However, since many real-world flow networks are lossy, the minimum-cost flow problem has also attracted much attention for generalized networks, also known as networks with gain [26, 27, 28]. In this type of network, each line is associated with a constant gain relating the two flows of the line through a linear function. From the optimization perspective, network flow problems are convex and can be solved efficiently unless there are discrete variables involved [29].

There are several real-world network flows that are lossy, where the loss is a nonlinear function of the flows. An important example is power distribution networks for which the loss over a transmission line (with fixed voltage magnitudes at both ends) is given by a parabolic function due to Kirchhoff's circuit laws [30]. The loss function could be much more complicated depending on the power electronic devices installed on the transmission line. To the best of our knowledge, there is no theoretical result in the literature on the polynomial-time solvability of network flow problems with nonlinear flow functions, except in very special cases. This chapter is concerned with this general problem, named generalized network flow (GNF). Note that the term "GNF" has already been used in the literature for networks with linear losses, but it corresponds to arbitrary lossy networks in this work.

GNF aims to optimize the nodal injections subject to flow constraints for each line and box constraints for both injections and flows. A flow constraint is a nonlinear equality relating the flows at both ends of a line. To solve GNF, this work makes the practical assumption that the cost and flow functions are all monotonic and convex. The GNF problem is still highly non-convex due to its equality constraints. Relaxing the nonlinear equalities to convex inequalities gives rise to a convex relaxation of GNF. It can be easily observed that solving the relaxed problem may lead to a solution for which the new inequality flow constraints are not binding. One may speculate that this observation implies that the con-

vex relaxation is not tight. However, the objective of this work is to show that as long as GNF is feasible, the convex relaxation is tight. More precisely, the convex relaxation always finds the optimal injections (and hence the optimal objective value), but probably produces wrong flows leading to non-binding inequalities. However, once the optimal injections are obtained at the nodes, a feasibility problem can be solved to find a set of feasible flows corresponding to the injections. Note that the reason why the convex relaxation does not necessarily find the correct flows is that the mapping from flows to injections is not invertible. For example, it is known in the context of power systems that the power flow equations may not have a unique solution [100]. The main contribution of this work is to show that although GNF may be NP-hard (since the flow equations can have an exponential number of solutions), the optimal injections can be found in polynomial time.

5.1.1 Application of GNF in Power Systems

The operation of a power network depends heavily on various large-scale optimization problems such as state estimation, optimal power flow (OPF), contingency constrained OPF, unit commitment, sizing of capacitor banks and network reconfiguration. These problems are highly non-convex due to the nonlinearities imposed by the laws of physics [67, 101]. For example, each of the above problems has the power flow equations embedded in it, which are nonlinear equality constraints. The nonlinearity of OPF, as the most fundamental optimization problem for power systems, has been studied since 1962, leading to various heuristic and local-search algorithms [21, 102, 65, 66, 103, 75, 70, 23, 104]. These algorithms suffer from sensitivity and convergence issues, and more importantly they may converge to a local optimum that is noticeably far from a global solution.

Recently, it has been shown in [25] that the semidefinite programming (SDP) relaxation is able to find the global solution of the OPF problem under a sufficient condition, which is satisfied for IEEE benchmark systems with 14, 30, 57, 118 and 300 buses and many randomly generated power networks. The papers [25] and [101] show that this condition holds widely in practice due to the passivity of transmission lines and transformers. In particular, [101] shows that in the case when this condition is not satisfied (see [87] for counterexamples), OPF can always be solved globally in polynomial time after two approximations: (i) relaxing angle constraints by adding a sufficient number of actual/virtual phase shifters to the network, (ii) relaxing power balance equalities at the buses to inequality constraints.

OPF under Approximation (ii) was also studied in [85] and [105] for distribution networks. The paper [106] studies the optimization of active power flows over distribution networks under fixed voltage magnitudes and shows that the SDP relaxation works without Approximation (i) as long as a very practical angle condition is satisfied. The idea of convex relaxation developed in [107] and [25] can be applied to many other power problems, such as voltage regulation [108], state estimation [109], calculation of voltage stability margin [110], charging of electric vehicles [80], SCOPF with variable tap-changers and capacitor banks [73], dynamic energy management [30], and electricity market [111].

Energy-related optimizations with embedded power flow equations can be regarded as nonlinear network flow problems, which are analogous to GNF. The results derived in this work for a general GNF problem lead to the generalization of the result of [106] to networks with virtual phase shifters. This proves that in order to use SDP relaxation for OPF over an arbitrary power network, it is not needed to relax power balance equalities to inequality constraints under a very mild angle assumption.

5.1.2 Notations

The following notations will be used throughout this chapter:

- $\mathcal{R}$ and $\mathcal{R}_+$ denote the sets of real numbers and nonnegative numbers, respectively.
- Given two matrices M and N, the inequality $M \leq N$ means that M is less than or equal to N element-wise.
- Given a set $\mathcal{T}$, its cardinality is shown as $|\mathcal{T}|$.
- Lowercase, bold lowercase and uppercase letters are used for scalars, vectors and matrices (say x, $\mathbf{x}$ and X).

5.2 Problem Statement and Contributions

Consider an undirected graph (network) $\mathcal{G}$ with the vertex set $\mathcal{N} := \{1, 2, ..., m\}$ and the edge set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$. For every $i \in \mathcal{N}$, let $\mathcal{N}(i)$ denote the set of the neighboring vertices of node i. Assume that every edge $(i, j) \in \mathcal{E}$ is associated with two unknown flows p_{ij} and p_{ji} belonging to $\mathcal{R}$. The parameters p_{ij} and p_{ji} can be regarded as the flows entering the

edge (i, j) from the endpoints i and j, respectively. Define

$$p_i = \sum_{j \in \mathcal{N}(i)} p_{ij}, \qquad \forall i \in \mathcal{N}. \tag{5.1}$$

The parameter p_i is called "nodal injection at vertex i" or simply "injection", which is equal to the sum of the flows leaving vertex i through the edges connected to this vertex. Given an edge $(i, j) \in \mathcal{E}$, we assume that the flows p_{ij} and p_{ji} are related to each other via a function $f_{ij}(\cdot)$ to be introduced later. To specify which of the flows p_{ij} and p_{ji} is a function of the other, we give an arbitrary orientation to every edge of the graph $\mathcal{G}$ and denote the resulting graph as $\vec{\mathcal{G}}$. Denote the directed edge set of $\vec{\mathcal{G}}$ as $\vec{\mathcal{E}}$. If an edge $(i, j) \in \mathcal{E}$ belongs to $\vec{\mathcal{E}}$, we then express p_{ji} as a function of p_{ij}.

Definition 1. *Define the vectors* $\mathbf{p}_n$, $\mathbf{p}_e$ *and* $\mathbf{p}_d$ *as follows:*

$$\mathbf{p}_n = \{p_i \mid \forall i \in \mathcal{N}\}, \tag{5.2a}$$

$$\mathbf{p}_e = \{p_{ij} \mid \forall (i, j) \in \mathcal{E}\}, \tag{5.2b}$$

$$\mathbf{p}_d = \{p_{ij} \mid \forall (i, j) \in \vec{\mathcal{E}}\} \tag{5.2c}$$

(the subscripts "n", "e" and "d" stand for nodes, edges and directed edges). The terms $\mathbf{p}_n$, $\mathbf{p}_e$ *and* $\mathbf{p}_d$ *are referred to as injection vector, flow vector and semi-flow vector, respectively (note that* $\mathbf{p}_e$ *contains two flows per each line, while* $\mathbf{p}_d$ *has only one flow per line).*

Definition 2. *Given two arbitrary points* $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$, *the box* $\mathcal{B}(\mathbf{x}, \mathbf{y})$ *is defined as follows:*

$$B(\mathbf{x}, \mathbf{y}) = \{\mathbf{z} \in \mathcal{R}^n \mid \mathbf{x} \leq \mathbf{z} \leq \mathbf{y}\} \tag{5.3}$$

(note that $B(\mathbf{x}, \mathbf{y})$ *is non-empty only if* $\mathbf{x} \leq \mathbf{y}$*).*

Assume that each nodal injection p_i must be within the given interval $[p_i^{\min}, p_i^{\max}]$ for every $i \in \mathcal{N}$. We use the shorthand notation $\mathcal{B}$ for the box $\mathcal{B}(\mathbf{p}_{\mathrm{n}}^{\min}, \mathbf{p}_{\mathrm{n}}^{\max})$, where $\mathbf{p}_{\mathrm{n}}^{\min}$ and $\mathbf{p}_{\mathrm{n}}^{\max}$ are the vectors of the lower bounds $p_i^{\min}$'s and the upper bounds $p_i^{\max}$'s, respectively. This chapter is concerned with the following problem.

Generalized network flow (GNF):

$$\min_{\mathbf{P}_{\mathrm{n}}\in\mathcal{B},\mathbf{P}_{\mathrm{e}}\in\mathcal{R}^{|\mathcal{E}|}} \quad \sum_{i\in\mathcal{N}} f_i(p_i) \tag{5.4a}$$

$$\text{subject to} \quad p_i = \sum_{j\in\mathcal{N}(i)} p_{ij}, \qquad \forall i \in \mathcal{N} \tag{5.4b}$$

$$p_{ji} = f_{ij}(p_{ij}), \qquad \forall (i,j) \in \vec{\mathcal{E}} \tag{5.4c}$$

$$p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}], \qquad \forall (i,j) \in \vec{\mathcal{E}} \tag{5.4d}$$

where

1) $f_i(\cdot)$ is convex and monotonically increasing for every $i \in \mathcal{N}$.

2) $f_{ij}(\cdot)$ is convex and monotonically decreasing for every $(i,j) \in \vec{\mathcal{E}}$.

3) The limits $p_{ij}^{\min}$ and $p_{ij}^{\max}$ are given for every $(i,j) \in \vec{\mathcal{E}}$.

In the case when $f_{ij}(p_{ij})$ is equal to $-p_{ij}$ for all $(i,j) \in \vec{\mathcal{E}}$, the GNF problem reduces to the network flow problem for which every line is lossless. A few remarks can be made here:

- Given an edge $(i,j) \in \vec{\mathcal{E}}$, there is no explicit limit on p_{ji} in the formulation of the GNF problem because restricting p_{ji} is equivalent to limiting p_{ij}.
- Given a node $i \in \mathcal{N}$, the assumption of $f_i(p_i)$ being monotonically increasing is motivated by the fact that increasing the injection p_i normally elevates the cost in practice.
- Given an edge $(i,j) \in \vec{\mathcal{E}}$, p_{ij} and $-p_{ji}$ can be regarded as the input and output flows of the line (i,j), which travel in the same direction. The assumption of $f_{ij}(p_{ij})$ being monotonically decreasing is motivated by the fact that increasing the input flow normally makes the output flow higher in practice (note that $-p_{ji} = -f_{ij}(p_{ij})$).

Definition 3. *Define $\mathcal{P}$ as the set of all vectors $\mathbf{p}_n$ for which there exists a vector $\mathbf{p}_e$ such that $(\mathbf{p}_n, \mathbf{p}_e)$ satisfies equations (5.4b), (5.4c) and (5.4d). The set $\mathcal{P}$ and $\mathcal{P} \cap \mathcal{B}$ are referred to as* injection region *and* box-constrained injection region*, respectively.*

Regarding Definition 3, the box-constrained injection region is indeed the projection of the feasible set of GNF onto the space of the injection vector $\mathbf{p}_{\mathrm{n}}$. Now, one can express

GNF geometrically as follows:

$$\text{Geometric GNF}: \quad \min_{\mathbf{p}_\text{n} \in \mathcal{P} \cap \mathcal{B}} \sum_{i \in \mathcal{N}} f_i(p_i) \tag{5.5}$$

Note that $\mathbf{p}_\text{e}$ has been eliminated in Geometric GNF. It is hard to solve this problem directly because the injection region $\mathcal{P}$ is non-convex in general. This non-convexity can be observed in Figure 5.2(a), which shows $\mathcal{P}$ for the two-node graph drawn in Figure 5.1. To address this non-convexity issue, the GNF problem will be convexified naturally next.

Convexified generalized network flow (CGNF):

$$\min_{\mathbf{P}_\text{n} \in \mathcal{B}, \mathbf{P}_\text{e} \in \mathcal{R}^{|\mathcal{E}|}} \quad \sum_{i \in \mathcal{N}} f_i(p_i) \tag{5.6a}$$

$$\text{subject to} \quad p_i = \sum_{j \in \mathcal{N}(i)} p_{ij}, \qquad \forall i \in \mathcal{N} \tag{5.6b}$$

$$p_{ji} \geq f_{ij}(p_{ij}), \qquad \forall (i,j) \in \vec{\mathcal{E}} \tag{5.6c}$$

$$p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}], \qquad \forall (i,j) \in \mathcal{E} \tag{5.6d}$$

where $(p_{ij}^{\min}, p_{ij}^{\max}) = (f_{ji}(p_{ji}^{\max}), f_{ji}(p_{ji}^{\min}))$ for every $(i,j) \in \mathcal{E}$ such that $(j,i) \in \vec{\mathcal{E}}$. Note that CGNF has been obtained from GNF by relaxing equality (5.4c) to inequality (5.6c) and adding limits to p_{ij} for every $(j,i) \in \vec{\mathcal{E}}$. One can write:

$$\text{Geometric CGNF}: \quad \min_{\mathbf{p}_\text{n} \in \mathcal{P}_\text{c} \cap \mathcal{B}} \sum_{i \in \mathcal{N}} f_i(p_i) \tag{5.7}$$

where $\mathcal{P}_\text{c}$ denotes the set of all vectors $\mathbf{p}_\text{n}$ for which there exists a vector $\mathbf{p}_\text{e}$ such that $(\mathbf{p}_\text{n}, \mathbf{p}_\text{e})$ satisfies equations (5.6b), (5.6c) and (5.6d).

Two main results to be proved in this chapter are:

- **Geometry of injection region:** Given any two points $\mathbf{p}_\text{n}$ and $\tilde{\mathbf{p}}_\text{n}$ in the injection region, the box $\mathcal{B}(\mathbf{p}_\text{n}, \tilde{\mathbf{p}}_\text{n})$ is entirely contained in the injection region. Similar result holds true for the box-constrained injection region.

- **Relationship between GNF and CGNF:** If $(\mathbf{p}_\text{n}^*, \mathbf{p}_\text{e}^*)$ and $(\bar{\mathbf{p}}_\text{n}^*, \bar{\mathbf{p}}_\text{e}^*)$ denote two arbitrary solutions of GNF and CGNF, then $\mathbf{p}_\text{n}^* = \bar{\mathbf{p}}_\text{n}^*$. Hence, although CGNF may not be able to find a feasible flow vector for GNF, it always finds the correct optimal

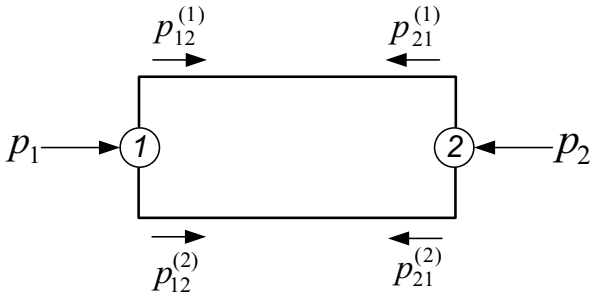
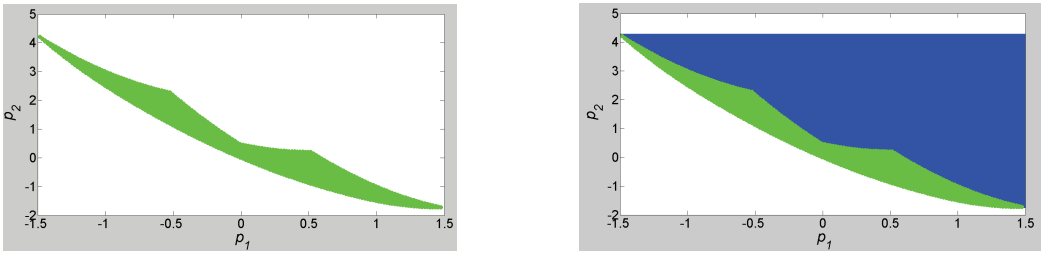


Figure 5.1: The graph $\mathcal{G}$ studied in Section 5.3.1.



(a) (b)

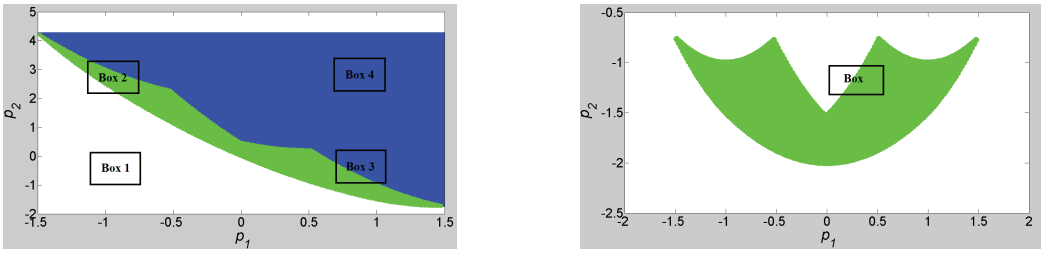
Figure 5.2: (a) Injection region $\mathcal{P}$ for the GNF problem given in (5.8). (b) The set $\mathcal{P}_c$ corresponding to the GNF problem given in (5.8).

injection vector for GNF.

The application of these results in power systems will also be discussed. Note that this work implicitly assumes that every two nodes of $\mathcal{G}$ are connected via, at most, one edge. However, the results to be derived later are all valid in the presence of multiple edges between two nodes. To avoid complicated notations, the proof will not be provided for this case. However, Section 5.3.1 studies a simple example with parallel lines.

5.3 Main Results

In this section, a detailed illustrative example will first be provided to clarify the issues and highlight the contribution of this work. In Subsections 5.3.2 and 5.3.3, the main results for GNF will be derived, whose application in power systems will be later discussed in Subsection 5.3.4.



(a) (b)

Figure 5.3: (a) This figure shows the set $\mathcal{P}_c$ corresponding to the GNF problem given in (5.8) together with a box constraint $(p_1, p_2) \in \mathcal{B}$ for four different positions of $\mathcal{B}$. (b) This figure shows the injection region $\mathcal{P}$ for the GNF problem given in (5.8) but after changing (5.8b) to (5.10).

5.3.1 Illustrative Example

In this subsection, we study the particular graph $\mathcal{G}$ depicted in Figure 5.1. This graph has two vertices and two parallel edges. Let $(p_{12}^{(1)}, p_{21}^{(1)})$ and $(p_{12}^{(2)}, p_{21}^{(2)})$ denote the flows associated with the first and second edges of the graph, respectively. Consider the following GNF problem:

$$\min \quad f_1(p_1) + f_2(p_2) \tag{5.8a}$$

$$\text{subject to} \quad p_{21}^{(i)} = \left(p_{12}^{(i)} - 1\right)^2 - 1, \quad i = 1, 2 \tag{5.8b}$$

$$-0.5 \leq p_{12}^{(1)} \leq 0.5, \quad -1 \leq p_{12}^{(2)} \leq 1, \tag{5.8c}$$

$$p_1 = p_{12}^{(1)} + p_{12}^{(2)}, \quad p_2 = p_{21}^{(1)} + p_{21}^{(2)} \tag{5.8d}$$

with the variables $p_1, p_2, p_{12}^{(1)}, p_{21}^{(1)}, p_{12}^{(2)}, p_{21}^{(2)}$, where $f_1(\cdot)$ and $f_2(\cdot)$ are both convex and monotonically increasing. The CGNF problem corresponding to this problem can be obtained by replacing (5.8b) with $p_{21}^{(i)} \geq (p_{12}^{(i)} - 1)^2 - 1$ and adding the limits $p_{21}^{(1)} \leq 1.5^2 - 1$ and $p_{21}^{(2)} \leq 2^2 - 1$. One can write:

$$\text{Geometric GNF:} \quad \min_{(p_1,p_2)\in\mathcal{P}} f_1(p_1) + f_2(p_2), \tag{5.9a}$$

$$\text{Geometric CGNF:} \quad \min_{(p_1,p_2)\in\mathcal{P}_c} f_1(p_1) + f_2(p_2), \tag{5.9b}$$

where $\mathcal{P}$ and $\mathcal{P}_c$ are indeed the projections of the feasible sets of GNF and CGNF over the injection space for (p_1, p_2) (note that there is no box constraint on (p_1, p_2) at this point).

The green area in Figure 5.2(a) shows the injection region $\mathcal{P}$. As expected, this set is non-convex. In contrast, the set $\mathcal{P}_c$ is a convex set containing $\mathcal{P}$. This set is shown in Figure 5.2(b), which includes two parts: (i) the green area is the same as $\mathcal{P}$, (ii) the blue area is the part of $\mathcal{P}_c$ that does not exist in $\mathcal{P}$. Thus, the transition from GNF to CGNF extends the injection region $\mathcal{P}$ to a convex set by adding the blue area. Notice that $\mathcal{P}_c$ has three boundaries: (i) a straight line on the top, (ii) a straight line on the right side, (iii) a lower curvy boundary. Since $f_1(\cdot)$ and $f_2(\cdot)$ are both monotonically increasing, the unique solution of Geometric CGNF must lie on the lower curvy boundary of $\mathcal{P}_c$. Since this lower boundary is in the green area, it is contained in $\mathcal{P}$. As a result, the unique solution of Geometric CGNF is a feasible point of $\mathcal{P}$ and therefore it is a solution of Geometric GNF. This means that CGNF finds the optimal injection vector for GNF.

To make the problem more interesting, we add the box constraint $(p_1, p_2) \in \mathcal{B}$ to GNF (and correspondingly to CGNF), where $\mathcal{B}$ is an arbitrary rectangular convex set in $\mathcal{R}^2$. The effect of this box constraint will be investigated in four different scenarios:

- Assume that $\mathcal{B}$ corresponds to Box 1 (including its interior) in Figure 5.3(a). In this case, $\mathcal{P} \cap \mathcal{B} = \mathcal{P}_c \cap \mathcal{B} = \phi$, meaning that Geometric GNF and Geometric CGNF are both infeasible.

- Assume that $\mathcal{B}$ corresponds to Box 2 (including its interior) in Figure 5.3(a). In this case, the solution of Geometric CGNF lies on the lower boundary of $\mathcal{P}_c$ and therefore it is also a solution of Geometric GNF.

- Assume that $\mathcal{B}$ corresponds to Box 3 (including its interior) in Figure 5.3(a). In this case, the solutions of Geometric GNF and Geometric CGNF are identical and both correspond to the lower left corner of the box $\mathcal{B}$.

- Assume that $\mathcal{B}$ corresponds to Box 4 (including its interior) in Figure 5.3(a). In this case, $\mathcal{P} \cap \mathcal{B} = \phi$ but $\mathcal{P}_c \cap \mathcal{B} \neq \phi$. Hence, Geometric GNF is infeasible while Geometric CGNF has an optimal solution.

In summary, it can be argued that independent of the position of the box $\mathcal{B}$ in $\mathcal{R}^2$, CGNF finds the optimal injection vector for GNF as long as GNF is feasible.

Assume now that the relationship between $P_{21}^{(i)}$ and $P_{12}^{(i)}$ is given by

$$p_{21}^{(i)} = \left(p_{12}^{(i)}\right)^2 - 1, \qquad i = 1, 2 \tag{5.10}$$

instead of (5.8b). The injection region $\mathcal{P}$ in the case is depicted in Figure 5.3(b). As before, we impose a box constraint $(p_1, p_2) \in \mathcal{B}$ on GNF, where $\mathcal{B}$ is shown as "Box" in the figure. It is easy to show that the lower left corner of this box belongs to $\mathcal{P}_{\rm c}$ and hence it is a solution of Geometric CGNF. However, this corner point does not belong to Geometric GNF. More precisely, Geometric GNF is feasible in this case, while its solution does not coincide with that of Geometric CGNF. Hence, Geometric GNF and Geometric CGNF are no longer equivalent after changing (5.8b) to (5.10). This is a consequence of the fact that the function $(p-1)^2 - 1$ is decreasing in p over the interval $[-1, 1]$ while the function $p^2 - 1$ is not. This explains the necessity of the assumption of the monotonicity of $f_{ij}(\cdot)$'s made earlier.

5.3.2 Geometry of Injection Region

In order to study the relationship between GNF and CGNF, it is beneficial to explore the geometry of the feasible set of GNF. Hence, we investigate the geometry of the injection region $\mathcal{P}$ and the box-constrained injection region $\mathcal{P} \cap \mathcal{B}$ in this part.

GNF-Theorem 1. *Consider two arbitrary points $\hat{\mathbf{p}}_n$ and $\tilde{\mathbf{p}}_n$ in the injection region $\mathcal{P}$. The box $\mathcal{B}(\hat{\mathbf{p}}_n, \tilde{\mathbf{p}}_n)$ is contained in $\mathcal{P}$.* ■

The proof of this theorem is based on four lemmas, and will be provided later in this subsection. To understand this theorem, consider the injection region $\mathcal{P}$ depicted in Figure 5.2(a) corresponding to the illustrative example given in Section 5.3.1. If any arbitrary box is drawn in $\mathcal{R}^2$ in such a way that its upper right corner and lower left corner both lie in the green area, then the entire box must lie in the green area completely. This can be easily proved in this special case and is true in general due to Theorem 1. However, this result does not hold for the injection region given in Figure 5.3(b) because the assumption of monotonicity of $f_{ij}(\cdot)$'s is violated in this case. The result of Theorem 1 can be generalized to the box-constrained injection region, as stated below.

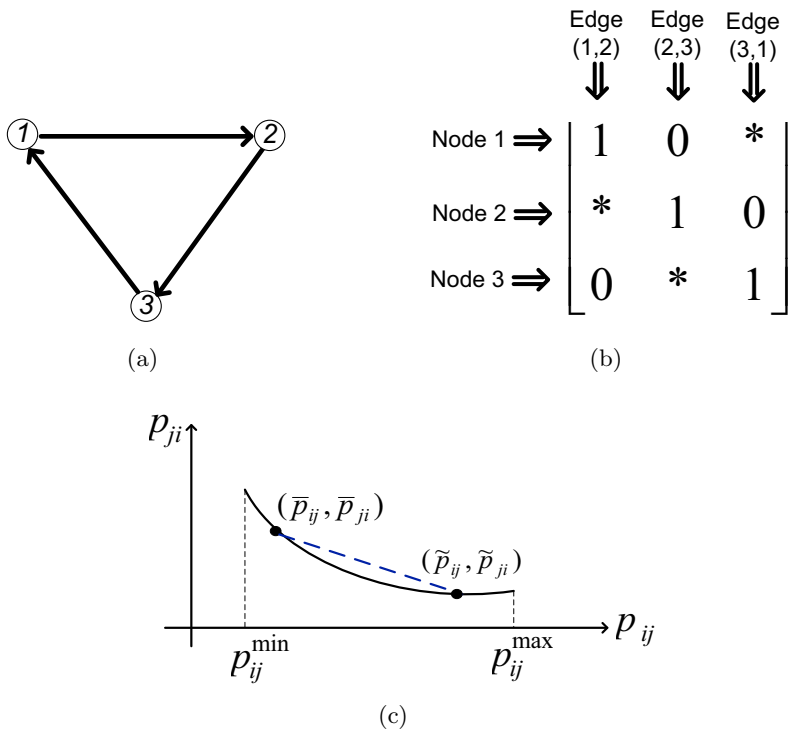


Figure 5.4: (a) A particular graph $\vec{\mathcal{G}}$. (b) The matrix $M(\bar{\mathbf{p}}_{\mathrm{d}}, \tilde{\mathbf{p}}_{\mathrm{d}})$ corresponding to the graph $\vec{\mathcal{G}}$ in Figure (a). (c) The $(j,(i,j))^{\text{th}}$ entry of $M(\bar{\mathbf{p}}_{\mathrm{d}}, \tilde{\mathbf{p}}_{\mathrm{d}})$ (shown as "*") is equal to the slope of the line connecting the point $(\bar{p}_{ij}, \bar{p}_{ji})$ to $(\tilde{p}_{ij}, \tilde{p}_{ji})$.

Corollary 1. *Consider two arbitrary points $\hat{\mathbf{p}}_n$ and $\tilde{\mathbf{p}}_n$ belonging to the box-constrained injection region $\mathcal{P} \cap \mathcal{B}$. The box $\mathcal{B}(\hat{\mathbf{p}}_n, \tilde{\mathbf{p}}_n)$ is contained in $\mathcal{P} \cap \mathcal{B}$.*

Proof: The proof follows immediately from Theorem 1. ■

The rest of this subsection is dedicated to the proof of Theorem 1, which is based on a series of definitions and lemmas.

Definition 4. *Define $\mathcal{B}_d$ as the box containing all vectors $\mathbf{p}_d$ introduced in (5.2c) satisfying the condition $p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}]$ for every $(i,j) \in \vec{\mathcal{E}}$.*

Definition 5. *Given two arbitrary points $\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d \in \mathcal{B}_d$, define $M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d)$ as follows:*

- *Let $M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d)$ be a matrix with $|\mathcal{N}|$ rows indexed by the vertices of $\mathcal{G}$ and with $|\vec{\mathcal{E}}|$ columns indexed by the edges in $\vec{\mathcal{E}}$.*
- *For every vertex $k \in \mathcal{N}$ and edge $(i,j) \in \vec{\mathcal{E}}$, set the $(k,(i,j))^{th}$ entry of $M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d)$ (the*

one in the intersection of row k and column (i,j)) as

$$\begin{cases} 1 & \text{if} \quad k=i \\ \frac{f_{ij}(\bar{p}_{ij})-f_{ij}(\tilde{p}_{ij})}{\bar{p}_{ij}-\tilde{p}_{ij}} & \text{if} \quad k=j \text{ and } \bar{p}_{ij} \neq \tilde{p}_{ij} \\ f'_{ij}(\bar{p}_{ij}) & \text{if} \quad k=j \text{ and } \bar{p}_{ij} = \tilde{p}_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

where $f'_{ij}(\bar{p}_{ij})$ denotes the right derivative of $f_{ij}(\bar{p}_{ij})$ if $\bar{p}_{ij} < p_{ij}^{max}$ and the left derivative of $f_{ij}(\bar{p}_{ij})$ if $\bar{p}_{ij} = p_{ij}^{max}$.

To illustrate Definition 5, consider the three-node graph $\vec{\mathcal{G}}$ depicted in Figure 5.4(a). The matrix $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ associated with this graph has the structure shown in Figure 5.4(b), where the "*" entries depend on the specific values of $\bar{\mathbf{p}}_{\rm d}$ and $\tilde{\mathbf{p}}_{\rm d}$. Consider an edge $(i,j) \in \vec{\mathcal{E}}$. The $(j,(i,j))^{\rm th}$ entry of $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ is equal to

$$\frac{f_{ij}(\bar{p}_{ij}) - f_{ij}(\tilde{p}_{ij})}{\bar{p}_{ij} - \tilde{p}_{ij}}, \tag{5.12}$$

provided $\bar{p}_{ij} \neq \tilde{p}_{ij}$. As can be seen in Figure 5.4(c), this is equal to the slope of the line connecting the point $(\bar{p}_{ij}, \bar{p}_{ji})$ to the point $(\tilde{p}_{ij}, \tilde{p}_{ji})$ on the parameterized curve (p_{ij}, p_{ji}), where $p_{ji} = f_{ij}(p_{ij})$. Moreover, $f'_{ij}(\bar{p}_{ij})$ is the limit of this slope as the point $(\tilde{p}_{ij}, \tilde{p}_{ji})$ approaches $(\bar{p}_{ij}, \bar{p}_{ji})$. It is also interesting to note that $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ has one positive entry, one negative entry and $m-2$ zero entries in each column (note that the slope of the line connecting $(\bar{p}_{ij}, \bar{p}_{ji})$ to $(\tilde{p}_{ij}, \tilde{p}_{ji})$ is always negative). The next lemma explains how the matrix $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ can be used to relate the semi-flow vector to the injection vector.

Lemma 1. *Consider two arbitrary injection vectors $\bar{\mathbf{p}}_n$ and $\tilde{\mathbf{p}}_n$ in $\mathcal{P}$, associated with the semi-flow vectors $\bar{\mathbf{p}}_d$ and $\tilde{\mathbf{p}}_d$ (defined in (5.2)). The relation*

$$\bar{\mathbf{p}}_n - \tilde{\mathbf{p}}_n = M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d) \times (\bar{\mathbf{P}}_d - \tilde{\mathbf{P}}_d) \tag{5.13}$$

holds.

Proof: One can write

$$\bar{p}_i - \tilde{p}_i = \sum_{j \in \mathcal{N}(i)} (\bar{p}_{ij} - \tilde{p}_{ij}), \qquad \forall i \in \mathcal{N}. \tag{5.14}$$

By using the relations

$$\bar{p}_{ji} = f_{ij}(\bar{p}_{ij}), \quad \tilde{p}_{ji} = f_{ij}(\tilde{p}_{ij}), \quad \forall (i,j) \in \vec{\mathcal{E}}, \tag{5.15}$$

it is straightforward to verify that (5.13) and (5.14) are equivalent. ■

The next lemma investigates an important property of the matrix $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$.

Lemma 2. *Given two arbitrary points $\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d \in \mathcal{B}_d$, assume that there exists a nonzero vector $\mathbf{x} \in \mathcal{R}^m$ such that $\mathbf{x}^T M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d) \geq 0$. If $\mathbf{x}$ has at least one strictly positive entry, then there exists a nonzero vector $\mathbf{y} \in \mathcal{R}^m_+$ such that $\mathbf{y}^T M(\bar{\mathbf{p}}_d, \tilde{\mathbf{p}}_d) \geq 0$.*

Proof: Consider an index $i_0 \in \mathcal{N}$ such that $x_{i_0} > 0$. Define $\mathcal{V}(i_0)$ as the set of all vertices $i \in \mathcal{N}$ from which there exists a directed path to vertex i_0 in the graph $\vec{\mathcal{G}}$. Note that $\mathcal{V}(i_0)$ includes vertex i_0 itself. The first goal is to show that

$$x_i \geq 0, \qquad \forall i \in \mathcal{V}(i_0). \tag{5.16}$$

To this end, consider an arbitrary set of vertices $i_1, ..., i_k$ in $\mathcal{V}(i_0)$ such that $\{i_0, i_1 ..., i_k\}$ forms a direct path in $\vec{\mathcal{G}}$ as

$$i_k \to i_{k-1} \to \cdots i_1 \to i_0. \tag{5.17}$$

To prove (5.16), it suffices to show that $x_{i_1}, ..., x_{i_k} \geq 0$. For this purpose, one can expand the product $\mathbf{x}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ and use the fact that each column of $M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ has $m-2$ zero entries to conclude that

$$x_{i_1} + \frac{f_{i_1 i_0}(\bar{p}_{i_1 i_0}) - f_{i_1 i_0}(\tilde{p}_{i_1 i_0})}{\bar{p}_{i_1 i_0} - \tilde{p}_{i_1 i_0}} x_{i_0} \geq 0. \tag{5.18}$$

Since x_{i_0} is positive and $f_{i_1 i_0}(\cdot)$ is a decreasing function, x_{i_1} turns out to be positive. Now, repeating the above argument for i_1 instead of i_0 yields that $x_{i_2} \geq 0$. Continuing this reasoning leads to $x_{i_1}, ..., x_{i_k} \geq 0$. Hence, inequality (5.16) holds. Now, define $\mathbf{y}$ as

$$y_i = \begin{cases} x_i & \text{if} \quad i \in \mathcal{V}(i_0) \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in \mathcal{N}. \tag{5.19}$$

In light of (5.16), $\mathbf{y}$ is a nonzero vector in $\mathcal{R}^m_+$. To complete the proof, it suffices to show that $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d}) \geq 0$. Similar to the indexing procedure used for the columns of the matrix

$M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$, we index the entries of the $|\vec{\mathcal{E}}|$ dimensional vector $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ according to the edges of $\vec{\mathcal{G}}$. Now, given an arbitrary edge $(\alpha, \beta) \in \vec{\mathcal{E}}$, the following statements hold true:

- If $\alpha, \beta \in \mathcal{V}(i_0)$, then the $(\alpha, \beta)^{\rm th}$ entries of $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ and $\mathbf{x}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ (i.e., the entries corresponding to the edge (α, β)) are identical.
- If $\alpha \in \mathcal{V}(i_0)$ and $\beta \notin \mathcal{V}(i_0)$, then the $(\alpha, \beta)^{\rm th}$ entry of $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ is equal to y_α.
- If $\alpha \notin \mathcal{V}(i_0)$ and $\beta \notin \mathcal{V}(i_0)$, then the $(\alpha, \beta)^{\rm th}$ entry of $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d})$ is equal to zero.

Note that the case $\alpha \notin \mathcal{V}(i_0)$ and $\beta \in \mathcal{V}(i_0)$ cannot happen, because if $\beta \in \mathcal{V}(i_0)$ and $(\alpha, \beta) \in \vec{\mathcal{E}}$, then $\alpha \in \mathcal{V}(i_0)$ by the definition of $\mathcal{V}(i_0)$. It follows from the above results and the inequality $\mathbf{x}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d}) \geq 0$ that $\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \tilde{\mathbf{p}}_{\rm d}) \geq 0$. ■

Definition 6. *Consider the graph $\mathcal{G}$ and an arbitrary flow vector $\mathbf{p}_e$. Given a subgraph $\mathcal{G}_s$ of the graph $\mathcal{G}$, define $\mathbf{p}_e(\mathcal{G}_s)$ as the flow vector associated with the edges of $\mathcal{G}_s$, which has been induced by $\mathbf{p}_e$. Define $\mathbf{p}_d(\mathcal{G}_s)$, $\mathbf{p}_n(\mathcal{G}_s)$ and $p_i(\mathcal{G}_s)$ as the semi-flow vector, injection vector and injection at node $i \in \mathcal{G}_s$ corresponding to $\mathbf{p}_e(\mathcal{G}_s)$, respectively. Define also $\mathcal{P}(\mathcal{G}_s)$ as the injection region associated with $\mathcal{G}_s$.*

The next lemma studies the injection region $\mathcal{P}$ in the case when $f_{ij}(\cdot)$'s are all piecewise linear.

Lemma 3. *Assume that the function $f_{ij}(\cdot)$ is piecewise linear for every $(i, j) \in \vec{\mathcal{E}}$. Consider two arbitrary points $\hat{\mathbf{p}}_n, \bar{\mathbf{p}}_n \in \mathcal{P}$ and a vector $\Delta\bar{\mathbf{p}}_n \in \mathcal{R}^m$ satisfying the relations*

$$\hat{\mathbf{p}}_n \leq \bar{\mathbf{p}}_n - \Delta\bar{\mathbf{p}}_n \leq \bar{\mathbf{p}}_n. \tag{5.20}$$

There exists a strictly positive number ϵ^{max} with the property

$$\bar{\mathbf{p}}_n - \varepsilon\Delta\bar{\mathbf{p}}_n \in \mathcal{P}, \qquad \forall \varepsilon \in [0, \epsilon^{max}]. \tag{5.21}$$

Proof: In light of (5.20), we have $\Delta\bar{\mathbf{p}}_{\rm n} \geq 0$. If $\Delta\bar{\mathbf{p}}_{\rm n} = 0$, then the lemma becomes trivial as ε can take any arbitrary value. So, assume that $\Delta\bar{\mathbf{p}}_{\rm n} \neq 0$. Let $\hat{\mathbf{p}}_{\rm e}$ and $\bar{\mathbf{p}}_{\rm e}$ denote two flow vectors associated with the injection vectors $\hat{\mathbf{p}}_{\rm n}$ and $\bar{\mathbf{p}}_{\rm n}$, respectively. Denote the corresponding semi-flow vectors as $\hat{\mathbf{p}}_{\rm d}$ and $\bar{\mathbf{p}}_{\rm d}$. Given an edge $(i, j) \in \vec{\mathcal{E}}$, the curve

$$\left\{(p_{ij}, f_{ij}(p_{ij})) \mid p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}]\right\} \tag{5.22}$$

is a Pareto set in $\mathcal{R}^2$ due to $f_{ij}(\cdot)$ being monotonically decreasing. Since $(\hat{p}_{ij}, \hat{p}_{ji})$ and $(\bar{p}_{ij}, \bar{p}_{ji})$ both lie on the above curve, one of the following cases occurs:

- *Case 1:* $\hat{p}_{ij} \geq \bar{p}_{ij}$ and $\hat{p}_{ji} \leq \bar{p}_{ji}$.
- *Case 2:* $\hat{p}_{ij} \leq \bar{p}_{ij}$ and $\hat{p}_{ji} \geq \bar{p}_{ji}$.

This fact can be observed in Figure 5.4(c) for the points $(\bar{p}_{ij}, \bar{p}_{ji})$ and $(\tilde{p}_{ij}, \tilde{p}_{ji})$ instead of $(\hat{p}_{ij}, \hat{p}_{ji})$ and $(\bar{p}_{ij}, \bar{p}_{ji})$. With no loss of generality, it can be assumed that Case (1) occurs. Indeed, if Case (2) happens, it suffices to make two changes:

- Change the orientation of the edge (i, j) in the graph $\vec{\mathcal{G}}$ so that $(j, i) \in \vec{\mathcal{E}}$ instead of $(i, j) \in \vec{\mathcal{E}}$.
- Replace the constraint $p_{ji} = f_{ij}(p_{ij})$ in (5.4c) with $p_{ij} = f_{ij}^{-1}(p_{ji})$, where the existence and monotonicity of the inverse function $f_{ij}^{-1}(\cdot)$ is guaranteed by the decreasing property of $f_{ij}(\cdot)$.

Therefore, suppose that

$$\hat{p}_{ij} \geq \bar{p}_{ij}, \quad \hat{p}_{ji} \leq \bar{p}_{ji}, \qquad \forall (i, j) \in \vec{\mathcal{E}} \tag{5.23}$$

or

$$\hat{\mathbf{p}}_{\mathrm{d}} \geq \bar{\mathbf{p}}_{\mathrm{d}}. \tag{5.24}$$

First, consider the case $\hat{\mathbf{p}}_{\mathrm{d}} > \bar{\mathbf{p}}_{\mathrm{d}}$. In light of Lemma 1, the assumption $\hat{\mathbf{p}}_{\mathrm{n}} \leq \bar{\mathbf{p}}_{\mathrm{n}}$ can be expressed as

$$M(\hat{\mathbf{p}}_{\mathrm{d}}, \bar{\mathbf{p}}_{\mathrm{d}}) \times (\hat{\mathbf{p}}_{\mathrm{d}} - \bar{\mathbf{p}}_{\mathrm{d}}) = \hat{\mathbf{p}}_{\mathrm{n}} - \bar{\mathbf{p}}_{\mathrm{n}} \leq 0. \tag{5.25}$$

In order to guarantee the relation $\bar{\mathbf{p}}_{\mathrm{n}} - \varepsilon \Delta \bar{\mathbf{p}}_{\mathrm{n}} \in \mathcal{P}$, it suffices to seek a vector $\Delta \bar{\mathbf{p}}_{\mathrm{d}} \in \mathcal{R}^{|\vec{\mathcal{E}}|}$ satisfying

$$\bar{\mathbf{p}}_{\mathrm{d}} - \varepsilon \Delta \bar{\mathbf{p}}_{\mathrm{d}} \in \mathcal{B}_{\mathrm{d}} \tag{5.26}$$

and

$$M(\bar{\mathbf{p}}_{\mathrm{d}}, \bar{\mathbf{p}}_{\mathrm{d}} - \varepsilon \Delta \bar{\mathbf{p}}_{\mathrm{d}}) \times (\bar{\mathbf{p}}_{\mathrm{d}} - (\bar{\mathbf{p}}_{\mathrm{d}} - \varepsilon \Delta \bar{\mathbf{p}}_{\mathrm{d}})) = \bar{\mathbf{p}}_{\mathrm{n}} - (\bar{\mathbf{p}}_{\mathrm{n}} - \varepsilon \Delta \bar{\mathbf{p}}_{\mathrm{n}}) \tag{5.27}$$

(see the proof of Lemma 1), or equivalently

$$\bar{\mathbf{p}}_{\rm d} - \varepsilon \Delta \bar{\mathbf{p}}_{\rm d} \in \mathcal{B}_{\rm d} \tag{5.28a}$$

$$M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d} - \varepsilon \Delta \bar{\mathbf{p}}_{\rm d}) \times \Delta \bar{\mathbf{p}}_{\rm d} = \Delta \bar{\mathbf{p}}_{\rm n}. \tag{5.28b}$$

Consider an arbitrary vector $\Delta \bar{\mathbf{p}}_{\rm d} \in \mathcal{R}^{|\vec{\mathcal{E}}|}$ with all negative entries. In light of Definition 5, the inequality $\hat{\mathbf{p}}_{\rm d} > \bar{\mathbf{p}}_{\rm d}$ and the piecewise linear property of the $f_{ij}(\cdot)$'s, there exists a positive number $\varepsilon^{\max}$ such that

$$\bar{\mathbf{p}}_{\rm d} - \varepsilon \Delta \bar{\mathbf{p}}_{\rm d} \in \mathcal{B}_{\rm d} \tag{5.29a}$$

$$M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d} - \varepsilon \Delta \bar{\mathbf{p}}_{\rm d}) = M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \tag{5.29b}$$

for every $\varepsilon \in [0, \varepsilon^{\max}]$. To prove the lemma, it follows from (5.28) and (5.29) that it is enough to show the existence of a negative vector $\Delta \bar{\mathbf{p}}_{\rm d}$ satisfying the relation

$$M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \times \Delta \bar{\mathbf{p}}_{\rm d} = \Delta \bar{\mathbf{p}}_{\rm n} \tag{5.30}$$

in which ε does not appear. To prove this by contradiction, assume that the above equation does not have a solution. By Farkas' Lemma, there exists a vector $\mathbf{x} \in \mathcal{R}^m$ such that

$$\mathbf{x}^T M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \geq 0, \qquad \mathbf{x}^T \Delta \bar{\mathbf{p}}_{\rm n} > 0. \tag{5.31}$$

Since $\Delta \bar{\mathbf{p}}_{\rm n}$ is nonnegative, the inequality $\mathbf{x}^T \Delta \bar{\mathbf{p}}_{\rm n} > 0$ does not hold unless $\mathbf{x}$ has at least one strictly positive entry. Now, it follows from $\mathbf{x}^T M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \geq 0$ and Lemma 2 that there exists a nonzero vector $\mathbf{y} \in \mathcal{R}^m$ such that

$$\mathbf{y}^T M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \geq 0, \qquad \mathbf{y} \geq 0. \tag{5.32}$$

On the other hand, given an edge $(i, j) \in \vec{\mathcal{E}}$, since $\hat{p}_{ij} \geq \bar{p}_{ij}$ (due to (5.23)), the slope of the line connecting the points $(\hat{p}_{ij}, \hat{p}_{ji})$ and $(\bar{p}_{ij}, \bar{p}_{ji})$ is more than or equal to $f'(\bar{p}_{ij})$. This yields that

$$M(\bar{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}) \leq M(\hat{\mathbf{p}}_{\rm d}, \bar{\mathbf{p}}_{\rm d}). \tag{5.33}$$

Now, it follows from (5.24), (5.25), (5.32), and (5.33) that

$$0 \geq \mathbf{y}^T M(\hat{\mathbf{p}}_{\mathrm{d}}, \bar{\mathbf{p}}_{\mathrm{d}}) \times (\hat{\mathbf{p}}_{\mathrm{d}} - \bar{\mathbf{p}}_{\mathrm{d}}) \geq \mathbf{y}^T M(\bar{\mathbf{p}}_{\mathrm{d}}, \bar{\mathbf{p}}_{\mathrm{d}}) \times (\hat{\mathbf{p}}_{\mathrm{d}} - \bar{\mathbf{p}}_{\mathrm{d}}) \geq 0. \tag{5.34}$$

Thus,

$$0 = \mathbf{y}^T M(\hat{\mathbf{p}}_{\mathrm{d}}, \bar{\mathbf{p}}_{\mathrm{d}}) \times (\hat{\mathbf{p}}_{\mathrm{d}} - \bar{\mathbf{p}}_{\mathrm{d}}) = \mathbf{y}^T (\hat{\mathbf{p}}_{\mathrm{n}} - \bar{\mathbf{p}}_{\mathrm{n}}). \tag{5.35}$$

This is a contradiction because $\hat{\mathbf{p}}_{\mathrm{n}} - \bar{\mathbf{p}}_{\mathrm{n}}$ is strictly negative and the nonzero vector $\mathbf{y}$ is positive.

So far, the lemma has been proven in the case when $\hat{\mathbf{p}}_{\mathrm{d}} > \bar{\mathbf{p}}_{\mathrm{d}}$. To extend the proof to the case $\hat{\mathbf{p}}_{\mathrm{d}} \geq \bar{\mathbf{p}}_{\mathrm{d}}$, define $\mathcal{E}_{\mathrm{r}}$ as the set of every edge $(i, j) \in \mathcal{E}$ such that

$$\hat{p}_{ij} \neq \bar{p}_{ij} \tag{5.36}$$

(note that $\hat{p}_{ij} = \bar{p}_{ij}$ if and only if $\hat{p}_{ji} = \bar{p}_{ji}$). Define also $\mathcal{G}_{\mathrm{r}}$ as the unique subgraph of $\mathcal{G}$ induced by the edge set $\mathcal{E}_{\mathrm{r}}$. Let $\mathcal{N}_{\mathrm{r}}$ denote the vertex set of $\mathcal{G}_{\mathrm{r}}$, which may be different from $\mathcal{N}$. It is easy to verify that

$$\hat{\mathbf{p}}_{\mathrm{d}}(\mathcal{G}_{\mathrm{r}}) > \bar{\mathbf{p}}_{\mathrm{d}}(\mathcal{G}_{\mathrm{r}}), \tag{5.37a}$$

$$\bar{p}_i - \hat{p}_i = \bar{p}_i(\mathcal{G}_{\mathrm{r}}) - \hat{p}_i(\mathcal{G}_{\mathrm{r}}), \qquad \forall i \in \mathcal{N}_{\mathrm{r}}. \tag{5.37b}$$

Therefore,

$$\hat{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) \leq \bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) - \Delta\bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) \leq \bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) \tag{5.38}$$

where the relationship between $\Delta\bar{\mathbf{p}}_{\mathrm{n}}$ and the new vector $\Delta\bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}})$ is as follows:

$$\Delta\bar{p}_i = \begin{cases} \Delta\bar{p}_i(\mathcal{G}_{\mathrm{r}}) & \text{if} \quad i \in \mathcal{N}_{\mathrm{r}} \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{N}. \tag{5.39}$$

In light of (5.37a) and (5.38), one can adopt the proof given earlier for the case $\hat{\mathbf{p}}_{\mathrm{d}} > \bar{\mathbf{p}}_{\mathrm{d}}$ to conclude the existence of a positive number $\epsilon^{\max}$ with the property

$$\bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) - \varepsilon\Delta\bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) \in \mathcal{P}(\mathcal{G}_{\mathrm{r}}), \qquad \forall \varepsilon \in [0, \epsilon^{\max}]. \tag{5.40}$$

Given an arbitrary number $\varepsilon \in [0, \epsilon^{\max}]$, we use the shorthand notation $\mathbf{p}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}})$ for $\bar{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_{\mathrm{r}}) -$

$\varepsilon\Delta\bar{\mathbf{p}}_{\rm n}(\mathcal{G}_{\rm r})$. Let $\mathbf{p}_{\rm e}(\mathcal{G}_{\rm r})$ denote a flow vector corresponding to the injection vector $\mathbf{p}_{\rm n}(\mathcal{G}_{\rm r})$. One can expand the vector $\mathbf{p}_{\rm e}(\mathcal{G}_{\rm r})$ into a flow vector $\mathbf{p}_{\rm e}$ for the graph $\mathcal{G}$ as follows:

- For every $(i,j) \in \mathcal{E}_{\rm r}$, the $(i,j)^{\rm th}$ entries of $\mathbf{p}_{\rm e}$ and $\mathbf{p}_{\rm e}(\mathcal{G}_{\rm r})$ (the ones corresponding to the edge (i,j)) are identical.
- For every $(i,j) \in \mathcal{E}\backslash\mathcal{E}_{\rm r}$, the $(i,j)^{\rm th}$ entry of $\mathbf{p}_{\rm e}$ is equal to $\bar{p}_{ij}$.

It is straightforward to show that $\mathbf{p}_{\rm n} \in \mathcal{P}$, where $\mathbf{p}_{\rm n}$ denotes the injection vector associated with the flow vector $\mathbf{p}_{\rm e}$. The proof is completed by noting that $\mathbf{p}_{\rm n} = \bar{\mathbf{p}}_{\rm n} - \varepsilon\Delta\bar{\mathbf{p}}_{\rm n}$. ■

The next lemma proves Theorem 1 in the case when $f_{ij}(\cdot)$'s are all piecewise linear.

Lemma 4. *Assume that the function $f_{ij}(\cdot)$ is piecewise linear for every $(i,j) \in \vec{\mathcal{E}}$. Given any two arbitrary points $\hat{\mathbf{p}}_n, \tilde{\mathbf{p}}_n \in \mathcal{P}$, the box $\mathcal{B}(\hat{\mathbf{p}}_n, \tilde{\mathbf{p}}_n)$ is a subset of the injection region $\mathcal{P}$.*

Proof: With no loss of generality, assume that $\hat{\mathbf{p}}_{\rm n} \leq \tilde{\mathbf{p}}_{\rm n}$ (because otherwise $\mathcal{B}(\hat{\mathbf{p}}_{\rm n}, \tilde{\mathbf{p}}_{\rm n})$ is empty). To prove the lemma by contradiction, suppose that there exists a point $\mathbf{p}_{\rm n} \in \mathcal{B}(\hat{\mathbf{p}}_{\rm n}, \tilde{\mathbf{p}}_{\rm n})$ such that $\mathbf{p}_{\rm n} \notin \mathcal{P}$. Consider the set

$$\left\{ \gamma \;\middle|\; \gamma \in [0,1],\ \tilde{\mathbf{p}}_{\rm n} + \gamma(\mathbf{p}_{\rm n} - \tilde{\mathbf{p}}_{\rm n}) \in \mathcal{P} \right\} \tag{5.41}$$

and denote its maximum as $\gamma^{\max}$ (the existence of this maximum number is guaranteed by the closedness and compactness of $\mathcal{P}$). Note that $\tilde{\mathbf{p}}_{\rm n} + \gamma(\mathbf{p}_{\rm n} - \tilde{\mathbf{p}}_{\rm n})$ is equal to $\mathbf{p}_{\rm n}$ at $\gamma = 1$. Since $\mathbf{p}_{\rm n} \notin \mathcal{P}$ by assumption, we have $\gamma^{\max} < 1$. Denote $\tilde{\mathbf{p}}_{\rm n} + \gamma^{\max}(\mathbf{p}_{\rm n} - \tilde{\mathbf{p}}_{\rm n})$ as $\bar{\mathbf{p}}_{\rm n}$. Hence, $\bar{\mathbf{p}}_{\rm n} \in \mathcal{P}$ and $\hat{\mathbf{p}}_{\rm n} \leq \mathbf{p}_{\rm n} \leq \bar{\mathbf{p}}_{\rm n}$ (recall that $\gamma^{\max} < 1$). Define $\Delta\bar{\mathbf{p}}_{\rm n}$ as $\bar{\mathbf{p}}_{\rm n} - \mathbf{p}_{\rm n}$. One can write:

$$\hat{\mathbf{p}}_{\rm n} \leq \bar{\mathbf{p}}_{\rm n} - \Delta\bar{\mathbf{p}}_{\rm n} \leq \bar{\mathbf{p}}_{\rm n}, \qquad \hat{\mathbf{p}}_{\rm n}, \bar{\mathbf{p}}_{\rm n} \in \mathcal{P}. \tag{5.42}$$

By Lemma 3, there exists a strictly positive number $\epsilon^{\max}$ with the property

$$\bar{\mathbf{p}}_{\rm n} - \varepsilon\Delta\bar{\mathbf{p}}_{\rm n} \in \mathcal{P}, \qquad \forall \varepsilon \in [0, \epsilon^{\max}] \tag{5.43}$$

or equivalently

$$\tilde{\mathbf{p}}_{\rm n} + (\gamma^{\max} + \varepsilon(1 - \gamma^{\max}))(\mathbf{p}_{\rm n} - \tilde{\mathbf{p}}_{\rm n}) \in \mathcal{P}, \quad \forall \varepsilon \in [0, \epsilon^{\max}]. \tag{5.44}$$

Notice that

$$\gamma^{\max} + \varepsilon(1 - \gamma^{\max}) > \gamma^{\max}, \quad \forall \varepsilon > 0. \tag{5.45}$$

Due to (5.44), this violates the assumption that $\gamma^{\max}$ is the maximum of the set given in (5.41). ■

Lemma 4 will be deployed next to prove Theorem 1 in the general case.

Proof of Theorem 1: Consider an arbitrary approximation of $f_{ij}(\cdot)$ by a piecewise linear function for every $(i,j) \in \vec{\mathcal{E}}$. As a counterpart of $\mathcal{P}$, let $\mathcal{P}_s$ denote the injection region in the piecewise-linear case. By Lemma 4, we have

$$\mathcal{B}(\hat{\mathbf{p}}_{\mathrm{n}}, \tilde{\mathbf{p}}_{\mathrm{n}}) \subseteq \mathcal{P}_s. \tag{5.46}$$

Since the piecewise linear approximation can be made in such a way that the sets $\mathcal{P}$ and $\mathcal{P}_s$ become arbitrarily close to each other, the above relation implies that the interior of $\mathcal{B}(\hat{\mathbf{p}}_{\mathrm{n}}, \tilde{\mathbf{p}}_{\mathrm{n}})$ is a subset of $\mathcal{P}$. On the other hand, $\mathcal{P}$ is a closed set. Hence, the box $\mathcal{B}(\hat{\mathbf{p}}_{\mathrm{n}}, \tilde{\mathbf{p}}_{\mathrm{n}})$ must entirely belong to $\mathcal{P}$. ■

5.3.3 Relationship between GNF and CGNF

In this subsection, the relationship between GNF and CGNF will be explored.

Theorem 2. *Assume that the GNF problem is feasible. Let* $(\mathbf{p}_n^*, \mathbf{p}_e^*)$ *and* $(\bar{\mathbf{p}}_n^*, \bar{\mathbf{p}}_e^*)$ *denote arbitrary solutions of GNF and CGNF, respectively. The relation* $\mathbf{p}_n^* = \bar{\mathbf{p}}_n^*$ *holds.* ■

Before presenting the proof of Theorem 2 in the general case, one special case will be studied for which the proof is much simpler. Observe that since $(\bar{\mathbf{p}}_{\mathrm{n}}^*, \bar{\mathbf{p}}_{\mathrm{e}}^*)$ is a feasible point of CGNF, one can write

$$\bar{p}_i^* \geq p_i^{\min}, \qquad \forall i \in \mathcal{N}. \tag{5.47}$$

The proof of Theorem 2 will be first derived in the special case

$$\bar{p}_i^* = p_i^{\min}, \qquad \forall i \in \mathcal{N}. \tag{5.48}$$

Proof of Theorem 2 under Condition (5.48): $(\mathbf{p}_{\mathrm{n}}^*, \mathbf{p}_{\mathrm{e}}^*)$ being a feasible point of GNF implies that

$$p_i^* \geq p_i^{\min}, \qquad \forall i \in \mathcal{N}. \tag{5.49}$$

Equations (5.48) and (5.49) lead to

$$\bar{\mathbf{p}}_{\mathrm{n}}^{*} \leq \mathbf{p}_{\mathrm{n}}^{*}. \tag{5.50}$$

Define the vector $\tilde{\mathbf{p}}_{\mathrm{n}}$ as

$$\tilde{p}_i = \sum_{(i,j)\in\vec{\mathcal{E}}} \bar{p}_{ij}^{*} + \sum_{(j,i)\in\vec{\mathcal{E}}} f_{ij}(\bar{p}_{ij}^{*}), \qquad \forall i \in \mathcal{N}. \tag{5.51}$$

Notice that $\tilde{\mathbf{p}}_{\mathrm{n}}$ belongs to $\mathcal{P}$. It can be inferred from the definition of CGNF that

$$\tilde{\mathbf{p}}_{\mathrm{n}} \leq \bar{\mathbf{p}}_{\mathrm{n}}^{*}. \tag{5.52}$$

Since $\tilde{\mathbf{p}}_{\mathrm{n}}, \mathbf{p}_{\mathrm{n}}^{*} \in \mathcal{P}$, it follows from Theorem 1, (5.50) and (5.52) that $\bar{\mathbf{p}}_{\mathrm{n}}^{*} \in \mathcal{P}$. On the other hand, $\bar{\mathbf{p}}_{\mathrm{n}}^{*} \in \mathcal{B}$. Therefore, $\bar{\mathbf{p}}_{\mathrm{n}}^{*} \in \mathcal{P} \cap \mathcal{B}$, meaning that $\bar{\mathbf{p}}_{\mathrm{n}}^{*}$ is a feasible point of Geometric GNF. Since the feasible set of Geometric CGNF includes that of Geometric GNF, $\bar{\mathbf{p}}_{\mathrm{n}}^{*}$ must be a solution of Geometric GNF as well. The proof follows from equation (5.50) and the fact that $\mathbf{p}_{\mathrm{n}}^{*}$ is another solution of Geometric GNF. ■

Before deriving the proof of Theorem 2 in the general case, some ideas need to be developed. Since $f_i(p_i)$ can be approximated by a differentiable function arbitrarily precisely, with no loss of generality, assume that $f_i(p_i)$ is differentiable for every $i \in \mathcal{N}$. Since CGNF is convex, one can take its Lagrangian dual. Let $\lambda_i^{\min}$ and $\lambda_i^{\max}$ denote the Lagrange multipliers corresponding to the constraints $p_i^{\min} \leq p_i$ and $p_i \leq p_i^{\max}$. Using the duality theorem, it can be shown that

$$\begin{aligned}
(\bar{\mathbf{P}}_{\mathrm{n}}^{*}, \bar{\mathbf{P}}_{\mathrm{e}}^{*}) = \underset{\mathbf{P}_{\mathrm{n}}\in\mathcal{R}^m, \mathbf{P}_{\mathrm{e}}\in\mathcal{B}_{\mathrm{e}}}{\arg\min} \quad & \sum_{i\in\mathcal{N}} \lambda_i p_i \\
\text{subject to} \quad & p_i = \sum_{j\in\mathcal{N}(i)} p_{ij}, \qquad \forall i \in \mathcal{N}, \\
& f_{ij}(p_{ij}) \leq p_{ji}, \qquad \forall (i,j) \in \vec{\mathcal{E}}, \\
& p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}], \quad \forall (i,j) \in \mathcal{E},
\end{aligned}$$

where

$$\lambda_i = f_i'(\bar{p}_i^{*}) - \lambda_i^{\min} + \lambda_i^{\max}, \qquad \forall i \in \mathcal{N}. \tag{5.53}$$

Hence,

$$(\bar{p}_{ij}^*, \bar{p}_{ji}^*) = \underset{(p_{ij},p_{ji})\in\mathcal{R}^2}{\arg\min} \quad \lambda_i p_{ij} + \lambda_j p_{ji} \tag{5.54a}$$

$$\text{subject to} \quad f_{ij}(p_{ij}) \leq p_{ji}, \tag{5.54b}$$

$$p_{ij} \in [p_{ij}^{\min}, p_{ij}^{\max}], \tag{5.54c}$$

$$p_{ji} \in [p_{ji}^{\min}, p_{ji}^{\max}] \tag{5.54d}$$

for every $(i,j) \in \vec{\mathcal{E}}$.

Definition 7. *Define $\mathcal{V}$ as the set of every index $i \in \mathcal{N}$ for which $\lambda_i \leq 0$. Define $\bar{\mathcal{V}}$ as the set of every index $i \in \mathcal{N}\backslash\mathcal{V}$ for which there exists a vertex $j \in \mathcal{V}$ such that $(i,j) \in \mathcal{G}$ (i.e., $\bar{\mathcal{V}}$ denotes the set of the neighbors of $\mathcal{V}$ in the graph $\mathcal{G}$).*

Since the objective function of optimization (5.54) is linear, it is straightforward to verify that $f_{ij}(\bar{p}_{ij}^*) = \bar{p}_{ji}^*$ as long as $\lambda_i > 0$ or $\lambda_j > 0$. In particular,

$$f_{ij}(\bar{p}_{ij}^*) = \bar{p}_{ji}^*, \qquad \forall (i,j) \in \vec{\mathcal{E}},\ \{i,j\} \not\subseteq \mathcal{V}, \tag{5.55a}$$

$$\bar{p}_{ij}^* = p_{ij}^{\min}, \qquad \forall (i,j) \in \mathcal{E},\ i \in \bar{\mathcal{V}},\ j \in \mathcal{V}. \tag{5.55b}$$

If $f_{ij}(\bar{p}_{ij}^*)$ were equal to $\bar{p}_{ji}^*$ for every $(i,j) \in \vec{\mathcal{E}}$, then the proof of Theorem 2 would be complete. However, the relation $f_{ij}(\bar{p}_{ij}^*) < \bar{p}_{ji}^*$ might hold in theory if $(i,j) \in \vec{\mathcal{E}}$ and $\{i,j\} \subseteq \mathcal{V}$. Hence, is important to study this scenario.

Proof of Theorem 2 in the general case: For every given index $i \in \mathcal{V}$, the term λ_i is negative by definition. On the other hand, $f_i'(\cdot)$ is strictly positive (as $f_i(\cdot)$ is monotonically increasing), and $\lambda_i^{\min}$ and $\lambda_i^{\max}$ are both nonnegative (as they are the Lagrange multipliers for some inequalities). Therefore, it follows from (5.53) that $\lambda_i^{\min} < 0$, implying that

$$\bar{p}_i^* = p_i^{\min}, \qquad \forall i \in \mathcal{V}. \tag{5.56}$$

Thus,

$$p_i^* \geq p_i^{\min} = \bar{p}_i^*, \qquad \forall i \in \mathcal{V}. \tag{5.57}$$

Let $\mathcal{G}_s$ denote a subgraph of $\mathcal{G}$ with the vertex set $\mathcal{V} \cup \bar{\mathcal{V}}$, which includes every edge $(i,j) \in \mathcal{E}$ if

- $\{i, j\} \subseteq \mathcal{V}$, or
- $i \in \mathcal{V}$ and $j \in \bar{\mathcal{V}}$.

Note that $\mathcal{G}_s$ includes all edges of $\mathcal{G}$ within the vertex subset $\mathcal{V}$ and those between the sets $\mathcal{V}$ and $\bar{\mathcal{V}}$, but this subgraph contains no edge between the vertices in $\bar{\mathcal{V}}$. The first objective is to show that

$$p_i^*(\mathcal{G}_s) \geq \bar{p}_i^*(\mathcal{G}_s), \qquad \forall i \in \mathcal{V} \cup \bar{\mathcal{V}}. \tag{5.58}$$

To this end, two possibilities will be investigated:

- *Case 1)* Consider a vertex $i \in \mathcal{V}$. Given any edge $(i, j) \in \mathcal{E}$, vertex j must belong to $\mathcal{V} \cup \bar{\mathcal{V}}$, due to Definition 7. Hence, $p_i^*(\mathcal{G}_s) = p_i^*$ and $\bar{p}_i^*(\mathcal{G}_s) = \bar{p}_i^*$. Combining these equalities with (5.57) gives rise to $p_i^*(\mathcal{G}_s) \geq \bar{p}_i^*(\mathcal{G}_s)$.
- *Case 2)* Consider a vertex $i \in \bar{\mathcal{V}}$. Based on (5.55b), one can write:

$$\bar{p}_i^*(\mathcal{G}_s) = \sum_{j \in \mathcal{V} \cap \mathcal{N}(i)} \bar{p}_{ij}^* = \sum_{j \in \mathcal{V} \cap \mathcal{N}(i)} p_{ij}^{\min}. \tag{5.59}$$

  Similarly,

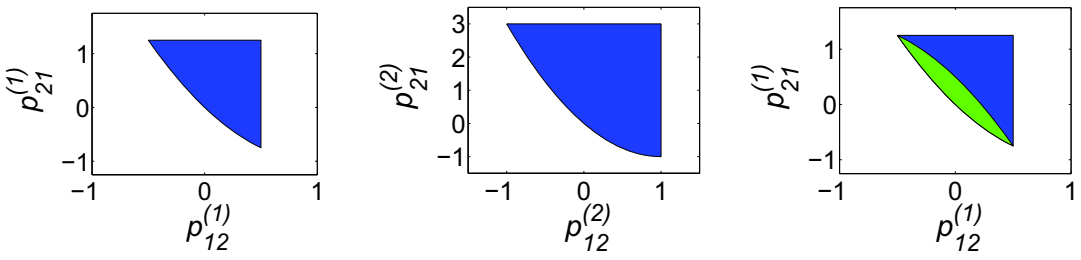
$$p_i^*(\mathcal{G}_s) = \sum_{j \in \mathcal{V} \cap \mathcal{N}(i)} p_{ij}^* \geq \sum_{j \in \mathcal{V} \cap \mathcal{N}(i)} p_{ij}^{\min}. \tag{5.60}$$

  Thus, $p_i^*(\mathcal{G}_s) \geq \bar{p}_i^*(\mathcal{G}_s)$.

So far, inequality (5.58) has been proven. Consider $\tilde{\mathbf{p}}_{\mathrm{n}}$ introduced in (5.51). Similar to (5.52), it is straightforward to show that $\tilde{p}_i(\mathcal{G}_s) \leq \bar{p}_i^*(\mathcal{G}_s)$ for every $i \in \mathcal{V} \cup \bar{\mathcal{V}}$. Hence,

$$\tilde{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_s) \leq \bar{\mathbf{p}}_{\mathrm{n}}^*(\mathcal{G}_s) \leq \mathbf{p}_{\mathrm{n}}^*(\mathcal{G}_s). \tag{5.61}$$

On the other hand, $\tilde{\mathbf{p}}_{\mathrm{n}}(\mathcal{G}_s)$ and $\mathbf{p}_{\mathrm{n}}^*(\mathcal{G}_s)$ are both in $\mathcal{P}(\mathcal{G}_s)$. Using (5.61) and Theorem 2 (but for $\mathcal{G}_s$ as opposed to $\mathcal{G}$) yields that $\bar{\mathbf{p}}_{\mathrm{n}}^*(\mathcal{G}_s) \in \mathcal{P}(\mathcal{G}_s)$. Hence, there exists a flow vector $\hat{\mathbf{p}}_{\mathrm{e}}(\mathcal{G}_s)$



(a) (b) (c)

Figure 5.5: Figures (a) and (b) show the feasible sets $\mathcal{T}_{\rm c}^{(1)}$ and $\mathcal{T}_{\rm c}^{(2)}$ for the example studied in Section 5.3.1, respectively. Figure (c) aims to show that CGNF may have an infinite number of solutions (any point in the yellow area may correspond to a solution of a given GNF).

associated with $\bar{\mathbf{p}}_{\rm n}^*(\mathcal{G}_s)$, meaning that

$$\bar{p}_i^*(\mathcal{G}_s) = \sum_{j\in\mathcal{N}(i)\cap(\mathcal{V}\cup\bar{\mathcal{V}})} \hat{p}_{ij}(\mathcal{G}_s), \qquad \forall i \in \mathcal{V}, \tag{5.62a}$$

$$\bar{p}_i^*(\mathcal{G}_s) = \sum_{j\in\mathcal{N}(i)\cap\mathcal{V}} \hat{p}_{ij}(\mathcal{G}_s), \qquad \forall i \in \bar{\mathcal{V}}, \tag{5.62b}$$

$$\hat{p}_{ji}(\mathcal{G}_s) = f_{ij}(\hat{p}_{ij}(\mathcal{G}_s)), \qquad \forall (i,j) \in \vec{\mathcal{G}}_s. \tag{5.62c}$$

Now, one can expand $\hat{\mathbf{p}}_{\rm e}(\mathcal{G}_s)$ to $\hat{\mathbf{p}}_{\rm e}$ as

$$\hat{p}_{jk} = \begin{cases} \hat{p}_{jk}(\mathcal{G}_s) & \text{if} \quad \{j,k\} \subseteq \mathcal{V}\cup\bar{\mathcal{V}} \\ \bar{p}_{jk}^* & \text{otherwise} \end{cases}, \quad \forall (j,k) \in \mathcal{E}. \tag{5.63}$$

Let $\hat{\mathbf{p}}_{\rm n}$ denote the injection vector associated with the flow vector $\hat{\mathbf{p}}_{\rm e}$. Two observations can be made:

1) $\hat{\mathbf{p}}_{\rm n}$ is equal to $\bar{\mathbf{p}}_{\rm n}^*$.

2) Due to (5.55a), (5.62c) and (5.63), $(\hat{\mathbf{p}}_{\rm n}, \hat{\mathbf{p}}_{\rm e})$ is a feasible point of GNF.

This means that $\bar{\mathbf{p}}_{\rm n}^*$ is the unique optimal solution of Geometric CGNF and yet a feasible point of Geometric GNF. The rest of the proof is the same as the proof of Theorem 2 under Condition (5.48) (given earlier). ■

An optimal solution of CGNF comprises two parts: injection vector and flow vector. Theorem 2 states that CGNF always finds the correct optimal injection vector solving

GNF. Now, the aim is to understand the reason why CGNF may not be able to find the correct optimal flow vector solving GNF. Consider again the illustrative example studied in Section 5.3.1, corresponding to the graph $\mathcal{G}$ depicted in Figure 5.1. Let $\mathcal{T}$ denote the projection of the feasible set of the GNF problem given in (5.8) over the flow space associated with the vector $(p_{12}^{(1)}, p_{21}^{(1)}, p_{12}^{(2)}, p_{21}^{(2)})$. It is easy to verify that $\mathcal{T}$ can be decomposed as the product of $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, where

$$\mathcal{T}^{(1)} = \left\{ (p_{12}^{(1)}, p_{21}^{(1)}) \;\middle|\; p_{12}^{(1)} \in [-0.5, 0.5],\ p_{21}^{(1)} = \left(p_{12}^{(1)} - 1\right)^2 - 1 \right\}$$

and

$$\mathcal{T}^{(2)} = \left\{ (p_{12}^{(2)}, p_{21}^{(2)}) \;\middle|\; p_{12}^{(2)} \in [-1, 1],\ p_{21}^{(2)} = \left(p_{12}^{(2)} - 1\right)^2 - 1 \right\}.$$

Likewise, define $\mathcal{T}_{\mathrm{c}}$ as the projection of the feasible set of the CGNF problem over its flow space. As before, $\mathcal{T}_{\mathrm{c}}$ can be written as $\mathcal{T}_{\mathrm{c}}^{(1)} \times \mathcal{T}_{\mathrm{c}}^{(2)}$, where $\mathcal{T}_{\mathrm{c}}^{(i)}$ is obtained from $\mathcal{T}^{(i)}$ by changing its equality

$$p_{21}^{(i)} = \left(p_{12}^{(i)} - 1\right)^2 - 1 \tag{5.64}$$

to the inequality

$$p_{21}^{(i)} \geq \left(p_{12}^{(i)} - 1\right)^2 - 1 \tag{5.65}$$

for $i = 1, 2$, and adding the limits $p_{21}^{(1)} \leq 1.5^2 - 1$ and $p_{21}^{(2)} \leq 2^2 - 1$. The sets $\mathcal{T}_{\mathrm{c}}^{(1)}$ and $\mathcal{T}_{\mathrm{c}}^{(2)}$ are drawn in Figures 5.5(a) and 5.5(b). Given $i \in \{1, 2\}$, note that $\mathcal{T}_{\mathrm{c}}^{(i)}$ has two flat boundaries and one curvy (lower) boundary that is the same as $\mathcal{T}^{(i)}$. Consider the flow vector $(\bar{p}_{12}^{(1)}, \bar{p}_{21}^{(1)}, \bar{p}_{12}^{(2)}, \bar{p}_{21}^{(2)}) \in \mathcal{T}_{\mathrm{c}}$ defined as:

$$\begin{aligned} \left(\bar{p}_{12}^{(1)}, \bar{p}_{21}^{(1)}\right) &= \left(0.5, (0.5-1)^2 - 1\right), \\ \left(\bar{p}_{12}^{(2)}, \bar{p}_{21}^{(2)}\right) &= \left(-0.5, (-0.5-1)^2 - 1\right). \end{aligned} \tag{5.66}$$

Define $\bar{p}_1 = \bar{p}_{12}^{(1)} + \bar{p}_{12}^{(2)}$ and $\bar{p}_2 = \bar{p}_{21}^{(1)} + \bar{p}_{21}^{(2)}$. It can be verified that for every point $(\tilde{p}_{12}^{(1)}, \tilde{p}_{21}^{(1)})$ in the green area of Figure 5.5(c), there exists a vector $(\tilde{p}_{12}^{(2)}, \tilde{p}_{21}^{(2)}) \in \mathcal{T}_{\mathrm{c}}^{(2)}$ such that

$$\bar{p}_1 = \tilde{p}_{12}^{(1)} + \tilde{p}_{12}^{(2)}, \quad \bar{p}_2 = \tilde{p}_{21}^{(1)} + \tilde{p}_{21}^{(2)}. \tag{5.67}$$

This means that if $(\bar{p}_1, \bar{p}_2, \bar{p}_{12}^{(1)}, \bar{p}_{21}^{(1)}, \bar{p}_{12}^{(2)}, \bar{p}_{21}^{(2)})$ turns out to be an optimal solution of CGNF,

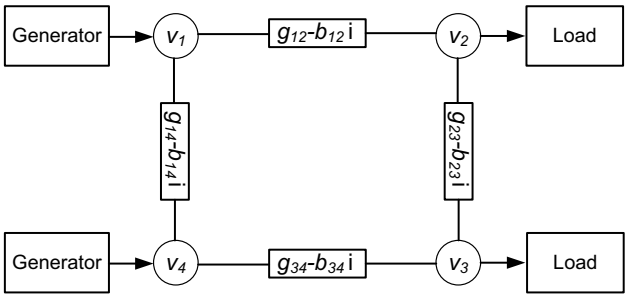


Figure 5.6: An example of electrical power network.

then $(\bar{p}_1, \bar{p}_2, \tilde{p}_{12}^{(1)}, \tilde{p}_{21}^{(1)}, \tilde{p}_{12}^{(2)}, \tilde{p}_{21}^{(2)})$ becomes another solution of CGNF. As a result, although Geometric CGNF has a unique solution, CGNF may have an infinite number of solutions whose corresponding flow vectors do not necessarily satisfy the constraints of GNF.

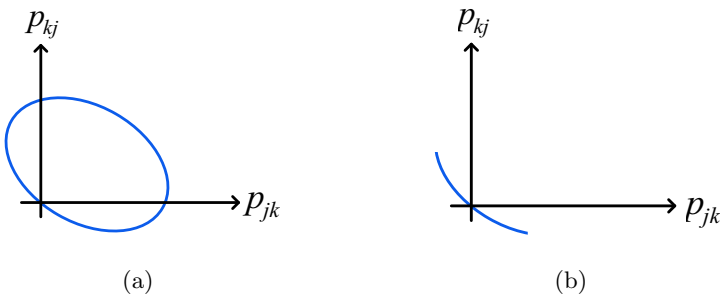
5.3.4 Optimal Power Flow in Electrical Power Networks

In this subsection, the results derived earlier for GNF will be applied to power networks. Consider a group of generators (sources of energy), which are connected to a group of electrical loads (consumers) via an electrical power network (grid). This network comprises a set of transmission lines connecting various nodes to each other (e.g., a generator to a load). Figure 5.6 exemplifies a four-node power network with two generators and two loads. Each load requests certain amount of energy, and the question of interest is to find the most economical power dispatch by the generators so that the demand and network constraints are met. To formulate the problem, let $\mathcal{G}$ denote the flow network corresponding to the electrical power network, where:

- Each injection p_i, $i \in \mathcal{G}$, represents either the active power produced by a generator and injected to the network or the active power absorbed from the network by an electrical load.

- Each p_{ij}, $(i,j) \in \mathcal{E}$, represents the active power entering the transmission line (i,j) from its i endpoint.

The problem of optimizing the flows in a power network is called optimal power flow (OPF). In this part, the goal is to optimize only active power, but most of the results to be developed later can be generalized to reactive power as well.



(a) (b)

Figure 5.7: (a) Feasible set for (p_{jk}, p_{kj}). (b) Feasible set for (p_{jk}, p_{kj}) after imposing lower and upper bounds on θ_{jk}.

Let v_i denote the complex (phasor) voltage at node $i \in \mathcal{N}$ of the power network. Denote the phase of v_i as θ_i. Given an edge $(j,k) \in \mathcal{G}$, we denote the admittance of the transmission line between nodes j and k as $g_{jk} - \mathrm{i}b_{jk}$, where the symbol i denotes the imaginary unit. g_{jk} and b_{jk} are nonnegative numbers due to the passivity of the line. There are two flows entering the transmission line from each of its ends. These flows are given by:

$$\begin{aligned} p_{jk} &= |v_j|^2 g_{jk} + |v_j||v_k| b_{jk} \sin(\theta_{jk}) - |v_j||v_k| g_{jk} \cos(\theta_{jk}), \\ p_{kj} &= |v_k|^2 g_{jk} - |v_j||v_k| b_{jk} \sin(\theta_{jk}) - |v_j||v_k| g_{jk} \cos(\theta_{jk}) \end{aligned}$$

where $\theta_{jk} = \theta_j - \theta_k$. As traditionally done in the power area, assume that $|v_j|$ and $|v_k|$ are fixed at their nominal values, while θ_{jk} is a variable to be designed. If θ_{jk} varies from $-\pi$ to π, then the feasible set of (p_{jk}, p_{kj}) becomes an ellipse, as illustrated in Figure 5.7(a). It can be seen from this figure that p_{kj} cannot be written as a function of p_{jk}. This observation is based on the implicit assumption that there is no limit on θ_{jk}. Suppose that θ_{jk} must belong to an interval $[-\theta_{jk}^{\max}, \theta_{jk}^{\max}]$ for some angle $\theta_{jk}^{\max}$. If the new feasible set for (p_{jk}, p_{kj}) resembles the partial ellipse drawn in Figure 5.7(b), then p_{kj} can be expressed as $f_{jk}(p_{jk})$ for a monotonically decreasing, convex function $f_{jk}(\cdot)$. This happens if

$$\theta_{jk}^{\max} \leq \tan^{-1}\left(\frac{b_{jk}}{g_{jk}}\right). \tag{5.68}$$

It is interesting to note that the right side of the above inequality is equal to 45.0°, 63.4° and 78.6° for $\frac{b_{jk}}{g_{jk}}$ equal to 1, 2 and 5, respectively. Note that $\frac{b_{jk}}{g_{jk}}$ is normally greater than 5 (due to the specifications of transmission lines) and $\theta_{jk}^{\max}$ is normally less than 15° and very rarely as high as 30° due to stability and thermal limits (this angle constraint is forced either

directly or through $p_{jk}^{\min}$ and $p_{jk}^{\max}$ in practice). Hence, Condition (5.68) is very practical. By assuming that this condition is satisfied, there exists a monotonically decreasing, convex function $f_{jk}(\cdot)$ such that

$$p_{kj} = f_{jk}(p_{jk}), \quad \forall p_{jk} \in [p_{jk}^{\min}, p_{jk}^{\max}], \tag{5.69}$$

where $p_{jk}^{\min}$ and $p_{jk}^{\max}$ correspond to $\theta_{jk}^{\max}$ and $-\theta_{jk}^{\max}$, respectively.

Given two disparate edges (j,k) and (j',k'), the phase differences θ_{jk} and $\theta_{j'k'}$ may not be varied independently if the graph $\mathcal{G}$ is cyclic (because the sum of the phase differences over a cycle must be zero). This is not an issue if the graph $\mathcal{G}$ is acyclic (corresponding to distribution networks) or if there is a sufficient number of phase-shifting transformers in the network. If none of these cases is true, then one could add virtual phase shifters to the power network at the cost of approximating the OPF problem. As soon as the flows (or phase differences) on various lines can be varied independently, equation (5.69) yields that the problem of optimizing active flows reduces to GNF. In this case, Theorems 1 and 2 can be used to study the corresponding approximated OPF problem. As a result, the optimal injections for the approximated OPF can be found via the corresponding CGNF problem. This implies two facts about the SDP and SOCP relaxations proposed in [25] and [101] for solving the OPF problem:

- The relaxations are exact without using the concept of load over-satisfaction (i.e., relaxing the flow constraints). This is the generalization of the result derived in [106].
- The relaxations always yield the optimal injections, but the produced flow vector can be wrong (meaning that the flow inequality constraints are not all binding). It is easy to contrive such examples.

In addition to active powers, voltage magnitudes and reactive powers are usually variable in power systems. The following remarks can be made for a general OPF problem:

- Reactive flows can be written as linear functions of active flows (under fixed voltage magnitudes). This implies that the above conclusions on OPF are valid even if the reactive power at each bus is upper bounded by a given number.
- In the case when the voltage magnitudes are variable, the flow constraint $p_{kj} = f_{jk}(p_{jk})$ needs to be replaced by $p_{kj} = f_{jk}(p_{jk}, \mathbf{x})$, where $\mathbf{x}$ is an exogenous input

containing the voltage magnitudes at all buses. The technique proposed in this chapter can be used to show that there is a region for $\mathbf{x}$ over which the above conclusions on OPF are valid. Due to space restrictions, the details are omitted here.

5.4 Summary

Network flow plays a central role in operations research, computer science and engineering. Due to the complexity of this problem, the main focus has been on lossless flow networks and more recently on networks with a linear loss function. This chapter studies the generalized network flow (GNF) problem, which aims to optimize the flows over a lossy flow network. It is assumed that the two flows over a line are related to each other via an arbitrary convex monotonic function. The GNF problem is hard to solve due to the presence of nonlinear equality flow constraints. If the flow constraints are relaxed to convex inequalities, these constraints may not be binding at optimality (as verified in simulations). This implies that a natural convex relaxation of GNF may lead to wrong flows. Nonetheless, this work proves that the nodal injections obtained by solving the convex relaxation are optimal, as long as GNF is feasible. In other words, this work proposes a polynomial-time algorithm for finding the optimal injections. Obtaining a set of flows associated with the optimal injections is a separate problem and has been considered as future work. An immediate application of this work is in power systems, where the goal is to optimize the power flows at buses and over transmission lines. Recent work on the optimal power flow problem has shown that this non-convex problem can be solved via a convex relaxation after two approximations: relaxing angle constraints (by adding virtual phase shifters) and relaxing power balance equations to inequality flow constraints. The results on GNF prove that the second relaxation (on power balance equations) is redundant under a very mild angle assumption.

Chapter 6

Semidefinite Relaxation for Nonlinear Optimization Over Graphs

This chapter is concerned with finding a global optimization technique for a broad class of non-linear optimization problems, including quadratic and polynomial optimizations. The main objective of this chapter is to investigate how the (hidden) structure of a given real/complex-valued optimization makes the problem easy to solve. To this end, three conic relaxations are proposed. Necessary and sufficient conditions are derived for the exactness of each of these relaxations, and it is shown that these conditions are satisfied if the optimization is highly structured. More precisely, the structure of the optimization is mapped into a generalized weighted graph, where each edge is associated with a weight set extracted from the coefficients of the optimization. In the real-valued case, it is shown that the relaxations are all exact if each weight set is sign definite and in addition a condition is satisfied for each cycle of the graph. It is also proved that if some of these conditions are violated, the relaxations still provide a low-rank solution for weakly cyclic graphs. In the complex-valued case, the notion of "sign definite complex sets" is introduced for complex weight sets. It is then shown that the relaxations are exact if each weight set is sign definite (with respect to complex numbers) and the graph is acyclic. Three other structural properties are derived for the generalized weighted graph in the complex case, each of which guarantees the exactness of some of the proposed relaxations. It is also shown that this result holds true if the graph can be decomposed as a union of edge-disjoint subgraphs, where each subgraph has one of the derived structural properties. As an application, it is finally proved that a broad class of real and complex optimizations over power networks are

polynomial-time solvable due to the passivity of transmission lines and transformers.

6.1 Introduction

Several classes of optimization problems, including polynomial optimization and quadratically-constrained quadratic program (QCQP) as a special case, are nonlinear/non-convex and NP-hard in the worst case. The paper [112] provides a survey on the computational complexity of optimizing various classes of continuous functions over some simple constraint sets. Due to the complexity of such problems, several convex relaxations based on linear matrix inequality (LMI), semidefinite programming (SDP) and second-order cone programming (SOCP) have gained popularity [31, 29]. These techniques enlarge the possibly non-convex feasible set into a convex set characterizable via convex functions, and then provide the exact or a lower bound on the optimal objective value. The paper [113] shows how SDP relaxation can be used to find better approximations for maximum cut (MAX CUT) and maximum 2-satisfiability (MAX 2SAT) problems. Another approach is proposed in [114] to solve the max-3-cut problem via complex SDP. The approaches in [113] and [114] have been generalized in several papers, including [115, 116, 117, 118, 119, 120, 121, 122].

The SDP relaxation converts an optimization with a vector variable to a convex optimization with a matrix variable, via a lifting technique. The exactness of the relaxation can then be interpreted as the existence of a low-rank (e.g., rank-1) solution for the SDP relaxation. Several papers have studied the existence of a low-rank solution to matrix optimizations with linear and LMI constraints [32, 33]. The papers [34] and [35] provide an upper bound on the lowest rank among all solutions of a feasible LMI problem. A rank-1 matrix decomposition technique is developed in [36] to find a rank-1 solution whenever the number of constraints is small. This technique is extended in [37] to the complex SDP problem. The paper [38] presents a polynomial-time algorithm for finding an approximate low-rank solution.

This work is motivated by the fact that real-world optimization problems are highly structured in many ways and their structures could in principle help reduce the computational complexity. For example, transmission lines and transformers used in power networks are passive devices, and as a result optimizations defined over electrical power networks have certain structures which distinguish them from abstract optimizations with random coef-

ficients. The high-level objective of this chapter is to understand how the computational complexity of a given nonlinear optimization is related to its (hidden) structure. This chapter is concerned with a broad class of nonlinear real/complex optimization problems, including QCQP. The main feature of this class is that the argument of each objective and constraint function is quadratic (as opposed to linear) in the optimization variable and the goal is to use three conic relaxations (SDP, reduced SDP and SOCP) to convexify the argument of the optimization.

In this chapter, the structure of the nonlinear optimization is mapped into a generalized weighted graph, where each edge is associated with a weight set constructed from the known parameters of the optimization (e.g., the coefficients). This generalized weighted graph captures both the sparsity of the optimization and possible patterns in the coefficients. First, it is shown that the proposed relaxations are exact for real-valued optimizations, provided a set of conditions is satisfied. These conditions need each weight set to be sign definite and each cycle of the graph has an even number of positive weight sets. It is also shown that if some of these conditions are not satisfied, the SDP relaxation is guaranteed to have a rank-2 solution for weakly cyclic graphs, from which an approximate rank-1 solution may be recovered. To study the complex-valued case, the notion of "sign-definite complex weight sets" is introduced and it is then proved that the relaxations are exact for a complex optimization if the graph is acyclic with sign definite weight sets (with respect to complex numbers). The complex case is further studied for general graphs and it is proved that if the graph can be decomposed as the union of some edge-disjoint subgraphs in such a way that each subgraph possesses one of the four proposed structural properties, then the SDP relaxation is tight. As an application of this work in optimization for power systems, it is also shown that a broad class of energy optimizations can be convexified due to the physics of power networks. The results of this chapter extend the recent works on energy optimization [25, 73, 123, 80, 106, 105] and general quadratic optimization [124, 125].

6.2 Problem Statement and Contributions

Before introducing the problem, we need to make several notations and definitions.

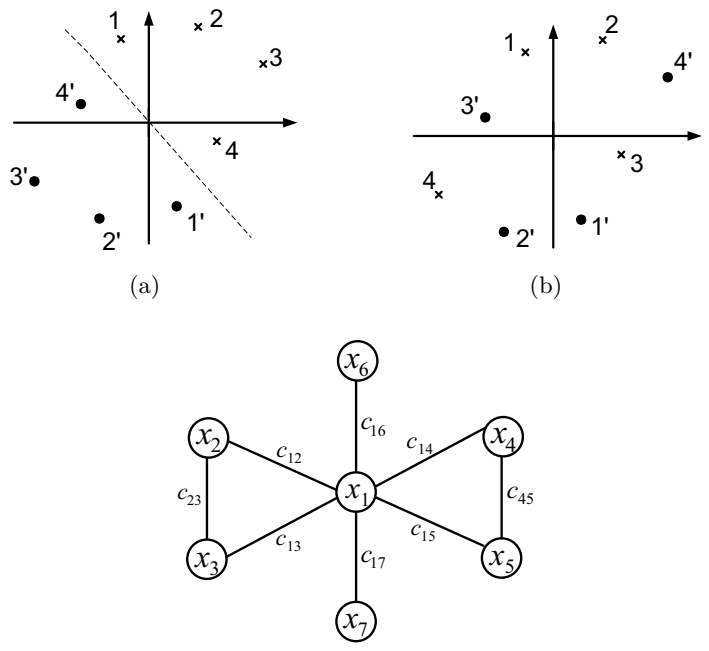


Figure 6.1: In Figure (a), there exists a line separating x's (elements of $\mathcal{T}$) from o's (elements of $-\mathcal{T}$) so the set $\mathcal{T}$ is sign definite. In Figure (b), this is not the case. Figure (c) shows the weighted graph $\mathcal{G}$ studied in Example 2.

6.2.1 Notations

Essential notations and definitions will be provided below.

Notation 1. *In this work, scalars, vectors and matrices will be shown by lowercase, bold lowercase and uppercase letters (e.g., x, $\mathbf{x}$ and X). Furthermore, x_i denotes the i^{th} entry of a vector $\mathbf{x}$, and X_{ij} denotes the $(i,j)^{th}$ entry of a matrix X.*

Notation 2. *$\mathcal{R}$, $\mathcal{C}$, $\mathcal{S}^n$ and $\mathcal{H}^n$ denote the sets of real numbers, complex numbers, $n \times n$ symmetric matrices and $n \times n$ Hermitian matrices, respectively.*

Notation 3. *$Re\{M\}$, $Im\{M\}$, M^H, $Rank\{M\}$ and $Trace\{M\}$ denote the real part, imaginary part, conjugate transpose, rank and trace of a given scalar/matrix M, respectively. The notation $M \succeq 0$ means that M is symmetric/Hermitian and positive semidefinite.*

Notation 4. *The symbol $\measuredangle(x)$ represents the phase of a complex number x. The imaginary unit is denoted as "i", while "i" is used for indexing.*

Notation 5. *Given an undirected graph $\mathcal{G}$, the notation $i \in \mathcal{G}$ means that i is a vertex of $\mathcal{G}$. Moreover, the notation $(i,j) \in \mathcal{G}$ means that (i,j) is an edge of $\mathcal{G}$ and <u>besides, $i < j$</u>.*

Notation 6. *Given a set $\mathcal{T}$, $|\mathcal{T}|$ denotes its cardinality. Given a graph $\mathcal{G}$, $|\mathcal{G}|$ shows the number of its vertices. Given a number (vector) $\mathbf{x}$, $|\mathbf{x}|$ denotes its absolute value (2-norm).*

Definition 1. *A finite set $\mathcal{T} \subset \mathcal{R}$ is said to be sign definite (with respect to $\mathcal{R}$) if its elements are either all negative or all nonnegative. $\mathcal{T}$ is called negative if its elements are negative and is called positive if its elements are nonnegative.*

Definition 2. *A finite set $\mathcal{T} \subset \mathcal{C}$ is said to be sign definite (with respect to $\mathcal{C}$) if when the sets $\mathcal{T}$ and $-\mathcal{T}$ are mapped into two collections of points in $\mathcal{R}^2$, then there exists a line separating the two sets (the elements of the sets are allowed to lie on the line).*

To illustrate Definition 2, consider a complex set $\mathcal{T}$ with four elements, whose corresponding points are labeled as 1, 2, 3 and 4 in Figure 6.1(a). The points corresponding to $-\mathcal{T}$ are labeled as 1', 2', 3' and 4' in the same picture. Since there exists a line separating x's (elements of $\mathcal{T}$) from o's (elements of $-\mathcal{T}$), the set $\mathcal{T}$ is sign definite. In contrast, if the elements of $\mathcal{T}$ are distributed according to Figure 6.1(b), the set will no longer be sign definite. Note that Definition 2 is inspired by the fact that a real set $\mathcal{T}$ is sign definite with respect to $\mathcal{R}$ if $\mathcal{T}$ and $-\mathcal{T}$ are separable via a point (on the horizontal axis).

Definition 3. *Given a graph $\mathcal{G}$, a cycle space is the set of all possible cycles in the graph. An arbitrary basis for this cycle space is called a "cycle basis".*

Definition 4. *In this work, a graph $\mathcal{G}$ is called weakly cyclic if every edge of the graph belongs to at most one cycle in $\mathcal{G}$ (i.e., the cycles of $\mathcal{G}$ are all edge-disjoint).*

Definition 5. *Consider a graph $\mathcal{G}$, a subgraph $\mathcal{G}_s$ of this graph and a matrix $X \in \mathcal{C}^{|\mathcal{G}|\times|\mathcal{G}|}$. Define $X\{\mathcal{G}_s\}$ as a sub-matrix of X obtained by picking every row and column whose index belongs to the vertex set of $\mathcal{G}_s$. For instance, $X\{(i,j)\}$, where $(i,j) \in \mathcal{G}$, has rows i, j and columns i, j of X.*

6.2.2 Problem Statement

Consider an undirected graph $\mathcal{G}$ with n vertices (nodes), where each edge $(i,j) \in \mathcal{G}$ has been assigned a nonzero edge weight set $\{c_{ij}^1, c_{ij}^2, ..., c_{ij}^k\}$ with k real/complex numbers (note that the superscripts in the weights are not exponents). This graph is called a *generalized weighted graph* as every edge is associated with a set of weights as opposed to a single

weight. Consider an unknown vector $\mathbf{x} = \begin{bmatrix} x_1 & \cdots x_n \end{bmatrix}$ belonging to $\mathcal{D}^n$, where $\mathcal{D}$ is either $\mathcal{R}$ or $\mathcal{C}$. For every $i \in \mathcal{G}$, x_i is a variable associated with node i of the graph $\mathcal{G}$. Define:

$$\mathbf{y} = \left\{ |x_i|^2 \mid \forall i \in \mathcal{G} \right\}, \quad \mathbf{z} = \left\{ \mathrm{Re}\{c_{ij}^t x_i x_j^H\} \mid \forall (i,j) \in \mathcal{G},\ t \in \{1, ..., k\} \right\}.$$

Note that according to Notation 5, $(i,j) \in \mathcal{G}$ means that (i,j) is an edge of the graph and that $i < j$. The sets $\mathbf{y}$ and $\mathbf{z}$ can be regarded as two vectors, where

- $\mathbf{y}$ collects the quadratic terms $|x_i|^2$'s (one term for each vertex),
- $\mathbf{z}$ collects the cross terms $\mathrm{Re}\{c_{ij}^t x_i x_j^H\}$'s ($k$ terms for each edge).

Although the above formulation deals with $\mathrm{Re}\{c_{ij}^t x_i x_j^H\}$ whenever $(i,j) \in \mathcal{G}$, it can handle terms of the form $\mathrm{Re}\{\alpha x_j x_i^H\}$ and $\mathrm{Im}\{\alpha x_i x_j^H\}$ for a complex weight α. This can be carried out using the transformations:

$$\mathrm{Re}\{\alpha x_j x_i^H\} = \mathrm{Re}\{(\alpha^H) x_i x_j^H\}, \qquad \mathrm{Im}\{\alpha x_i x_j^H\} = \mathrm{Re}\{(-\alpha \mathrm{i}) x_i x_j^H\}.$$

This work is concerned with the following optimization:

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{D}^n} \quad & f_0(\mathbf{y}, \mathbf{z}) \\ \text{subject to} \quad & f_j(\mathbf{y}, \mathbf{z}) \leq 0, \quad j = 1, 2, ..., m \end{aligned} \tag{6.1}$$

for given functions $f_0, ..., f_m$. The computational complexity of the above optimization depends in part on the structure of the functions f_j's. Regardless of these functions, Optimization (6.1) is intrinsically hard to solve (NP-hard in the worst case) because $\mathbf{y}$ and $\mathbf{z}$ are both nonlinear functions of $\mathbf{x}$. The objective is to convexify the second-order nonlinearity embedded in $\mathbf{y}$ and $\mathbf{z}$. To this end, notice that there exist two linear functions $l_1 : \mathcal{C}^{n \times n} \to \mathcal{R}^n$ and $l_2 : \mathcal{C}^{n \times n} \to \mathcal{R}^{k\tau}$ such that $\mathbf{y} = l_1\left(\mathbf{x}\mathbf{x}^H\right)$ and $\mathbf{z} = l_2\left(\mathbf{x}\mathbf{x}^H\right)$, where τ denotes the number of edges in $\mathcal{G}$. Motivated by the above observation, if $\mathbf{x}\mathbf{x}^H$ is replaced by a new matrix variable X, then $\mathbf{y}$ and $\mathbf{z}$ both become linear in X. This implies that the non-convexity induced by the quadratic terms $\mathrm{Re}\{c_{ij}^t x_i x_j\}$'s and $|x_i|$'s all disappear if Optimization (6.1) is reformulated in terms of X. However, the optimal solution X may not be decomposable as $\mathbf{x}\mathbf{x}^H$ unless some additional constraints are imposed on X. It is

straightforward to verify that Optimization (6.1) is equivalent to

$$\min_{X} \quad f_0(l_1(X), l_2(X)) \tag{6.2a}$$

$$\text{s.t.} \quad f_j(l_1(X), l_2(X)) \leq 0, \quad j = 1, ..., m \tag{6.2b}$$

$$X \succeq 0, \tag{6.2c}$$

$$\text{Rank}\{X\} = 1 \tag{6.2d}$$

where there is an implicit constraint that $X \in \mathcal{S}^n$ if $\mathcal{D} = \mathcal{R}$ and $X \in \mathcal{H}^n$ if $\mathcal{D} = \mathcal{C}$. To reduce the computational complexity of the above problem, two actions can be taken: (i) removing the nonconvex constraint (6.2d), (ii) relaxing the convex, but computationally-expensive, constraint (6.2c) to a set of simpler constraints on certain low-order submatrices of X. Based on this methodology, three relaxations will be proposed for Optimization (6.1) next.

SDP relaxation: This optimization is defined as

$$\min_{X} \quad f_0(l_1(X), l_2(X)) \tag{6.3a}$$

$$\text{s.t.} \quad f_j(l_1(X), l_2(X)) \leq 0, \quad j = 1, ..., m \tag{6.3b}$$

$$X \succeq 0. \tag{6.3c}$$

Reduced SDP relaxation: Choose a set of cycles $\mathcal{O}_1, ..., \mathcal{O}_p$ in the graph $\mathcal{G}$ such that they form a cycle basis. Let Ω denote the set of all subgraphs $\mathcal{O}_1, ..., \mathcal{O}_p$ as well as all edges of $\mathcal{G}$ that do not belong to any cycle in the graph (i.e., bridge edges). The reduced SDP relaxation is defined as

$$\min_{X} \quad f_0(l_1(X), l_2(X)) \tag{6.4a}$$

$$\text{s.t.} \quad f_j(l_1(X), l_2(X)) \leq 0, \qquad j = 1, ..., m \tag{6.4b}$$

$$X\{\mathcal{G}_s\} \succeq 0, \qquad \forall \mathcal{G}_s \in \Omega. \tag{6.4c}$$

SOCP relaxation: This optimization is defined as

$$\min_{X} \quad f_0(l_1(X), l_2(X)), \tag{6.5a}$$

$$\text{s.t.} \quad f_j(l_1(X), l_2(X)) \leq 0, \qquad j = 1, ..., m, \tag{6.5b}$$

$$X\{(i,j)\} \succeq 0, \qquad \forall (i,j) \in \mathcal{G}. \tag{6.5c}$$

The reason why the above optimization is called an SOCP problem is that the condition $X\{(i,j)\} \succeq 0$ can be replaced by the linear and norm constraints

$$X_{ii}, X_{jj} \geq 0, \quad X_{ii} + X_{jj} \geq \left| \begin{bmatrix} X_{ii} & X_{jj} & \sqrt{2}X_{ij} \end{bmatrix} \right|.$$

The above SDP, reduced SDP and SOCP relaxations are targeted at the non-convexity caused by the nonlinear relationship between $\mathbf{x}$ and $(\mathbf{y}, \mathbf{z})$. Note that these optimizations are convex relaxations only when the functions $f_0, ..., f_m$ are convex. If any of these functions is nonconvex, additional relaxations might be needed to convexify the SDP, reduced SDP or SOCP optimization. Define $f^*, f^*_{\text{SDP}}, f^*_{\text{r-SDP}}$ and f^*_{SOCP} as the optimal solutions of Optimizations (6.2), (6.3), (6.4) and (6.5), respectively. By comparing the feasible sets of these optimizations, it can be concluded that

$$f^*_{\text{SOCP}} \leq f^*_{\text{r-SDP}} \leq f^*_{\text{SDP}} \leq f^*. \tag{6.6}$$

Given a particular optimization of the form (6.1), if any of the above inequalities for f^* turns into an equality, the associated relaxation will be able to find the solution of the original optimization. In this case, it is said that the relaxation is "tight" or "exact". The objective of this chapter is to relate the exactness of the proposed relaxations to the topology of the graph $\mathcal{G}$ and its weights sets $\{c^1_{ij}, c^2_{ij}, ..., c^k_{ij}\}$'s.

It is noteworthy that the aforementioned problem formulation can be easily generalized in two directions:

- *Allowance of weight sets with different cardinalities:* The above problem formulation assumes that every edge weight set has k elements. However, if the weight sets have different sizes, the trivial weight 0 can be added to each set multiple times in such a way that all expanded sets reach the same cardinality.

- *Inclusion of linear terms in* $\mathbf{x}$*:* Optimization 6.1 is formulated in $\mathbf{x}\mathbf{x}^H$ with no linear term in $\mathbf{x}$. This issue can be fixed by defining an expanded vector $\tilde{\mathbf{x}}$ as $\begin{bmatrix} 1 & \mathbf{x}^H \end{bmatrix}^H$. Then, the matrix $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H$ needs to be replaced by a new matrix variable $\tilde{X}$ under the constraint $\tilde{X}_{11} = 1$.

6.2.3 Related Work

Consider the QCQP optimization:

$$\min_{\mathbf{x}\in\mathcal{D}^n} \quad \mathbf{x}^H M_1 \mathbf{x} \qquad \text{s.t.} \;\; \mathbf{x}^H M_j \mathbf{x} \leq 0 \quad j = 2, ..., k \tag{6.7a}$$

for given matrices $M_1, ..., M_k \in \mathcal{H}^n$. This problem is a special case of Optimization (6.1), where its generalized weighted graph $\mathcal{G}$ has two properties:

- Given two nodes $i, j \in \{1, ..., n\}$ such that $i < j$, there exists an edge between nodes i and j if and only if the (i, j) off-diagonal entry of at least one of the matrices $M_1, ..., M_k$ is nonzero.

- For every $(i, j) \in \mathcal{G}$, the weight set $\{c_{ij}^1, c_{ij}^2, ..., c_{ij}^k\}$ is the union of the $(i, j)^{\text{th}}$ entries of $M_1, ..., M_k$.

Due to the relation $\mathbf{x}^H M_i \mathbf{x} = \text{Trace}\{M_i \mathbf{x}\mathbf{x}^H\}$ for $i = 1, ..., k$, the SDP relaxation of Optimization (6.7) turns out to be

$$\min_{X} \quad \text{Trace}\{M_1 X\} \quad \text{s.t.} \;\; \text{Trace}\{M_j X\} \leq 0 \quad j = 2, ..., k, \quad X \succeq 0.$$

The SOCP relaxation of Optimization (6.7) is obtained by replacing the constraint $X \succeq 0$ with $X\{(i, j)\} \succeq 0$, $(i, j) \in \mathcal{G}$. The relationship between Optimization (6.7) and its relaxations have been studied in two special cases in the literature:

- Consider the case $D = \mathcal{R}$. It has been shown in [124] that $f_{\text{SOCP}}^* = f_{\text{SDP}}^* = f^*$ if $-M_0, ..., -M_k$ are all Metzler matrices. This implies that the proposed relaxations are all exact, independent of the topology of $\mathcal{G}$, as long as the set $\{c_{ij}^1, c_{ij}^2, ..., c_{ij}^k\}$ is negative for all $(i, j) \in \mathcal{G}$.

- Consider the case $D = \mathcal{C}$. It has been shown in the recent work [125] that $f_{\text{SDP}}^* = f^*$ if three conditions hold:

1. $\mathcal{G}$ is a tree graph.
2. M_1 is a positive semidefinite matrix.
3. For every $(i,j) \in \mathcal{G}$, the origin $(0,0)$ is not an interior point of the convex hull of the 2-d polytope induced by the weight set $\{c_{ij}^1, c_{ij}^2, ..., c_{ij}^k\}$.

It can be shown that Condition (3) implies that the complex set $\{c_{ij}^1, c_{ij}^2, ..., c_{ij}^k\}$ is sign definite (see Definition 2). The above results suggest that the polynomial-time solvability of certain classes of QCQP problems might be inferred from weak properties of their underlying generalized weighted graphs.

6.2.4 Contributions

Throughout this chapter, we assume that $f_j(\mathbf{y}, \mathbf{z})$ is monotonic in every entry of $\mathbf{z}$ for $j = 0, 1, ..., m$ (but possibly nonconvex in $\mathbf{y}$ and $\mathbf{z}$). With no loss of generality, suppose that $f_j(\mathbf{y}, \mathbf{z})$ is an increasing function with respect to all entries of $\mathbf{z}$ (to ensure this property, it may be needed to change the sign of some edge weights and then redefine the functions). A few of the results to be developed in this work do not need this assumption, in which cases the name of the function f_j will be changed to g_j to avoid any confusion in the assumptions.

The objective of this chapter is to study the interrelationship between f_{SOCP}^*, $f_{\text{r-SDP}}^*$, f_{SDP}^*, and f^*. In particular, it is aimed to understand what properties the generalized weighted graph $\mathcal{G}$ should have to guarantee the exactness of some of the proposed relaxations. Another goal is to find out how low rank the solution of the SDP relaxation will be in the case when the relaxation is not exact.

In section 6.3, we derive necessary and sufficient conditions for the exactness of each of the three aforementioned relaxations in both real and complex cases.

In Section 6.4, we consider the real-valued case $\mathcal{D} = \mathcal{R}$ and show that the SOCP, reduced SDP and SDP relaxations are all tight, provided each weight set $\{c_{ij}^1, ..., c_{ij}^k\}$ is sign definite (with respect to $\mathcal{R}$) and

$$\prod_{(i,j)\in\mathcal{O}_r} \sigma_{ij} = (-1)^{|\mathcal{O}_r|}, \qquad \forall r \in \{1, ..., p\}$$

where σ_{ij} shows the sign of the weight set associated with the edge $(i,j) \in \mathcal{G}$. This condition is naturally satisfied in three special cases:

- $\mathcal{G}$ is acyclic with arbitrary sign definite edge sets.
- $\mathcal{G}$ is bipartite with positive weight sets.
- $\mathcal{G}$ is arbitrary with negative weight sets.

It is also shown that if the SDP relaxation is not exact, it still has a low rank (rank-2) solution in two cases:

- $\mathcal{G}$ is acyclic (but with potentially indefinite weight sets).
- $\mathcal{G}$ is a weakly-cyclic bipartite graph with sign definite edge sets.

In section 6.5, we consider the complex-valued case $\mathcal{D} = \mathcal{C}$ under the assumption that each edge set $\{c_{ij}^1, ..., c_{ij}^k\}$ is sign definite with respect to $\mathcal{C}$. This assumption is trivially met if $k \leq 2$ or the weight set contains only real (or imaginary) numbers. Some of the results developed in this section are:

- The SOCP, reduced SDP and SDP relaxations are all tight if $\mathcal{G}$ is acyclic.
- The SOCP, reduced SDP and SDP relaxations are tight if each weight set contains only real or imaginary numbers and

$$\prod_{(i,j)\in\mathcal{O}_r} \sigma_{ij} = (-1)^{|\mathcal{O}_r|}, \qquad \forall r \in \{1, ..., p\}$$

  where $\sigma_{ij} \in \{0, \pm 1, \pm \mathrm{i}\}$ shows the sign of each weight set.
- The reduced SDP and SDP relaxations are exact if $\mathcal{G}$ is bipartite and weakly cyclic with positive or negative real weight sets.
- The reduced SDP and SDP relaxations (and not SOCP relaxation) are exact if $\mathcal{G}$ is a weakly cyclic graph with imaginary weight sets and nonzero signs σ_{ij}'s.

We also show that if the graph $\mathcal{G}$ can be decomposed as a union of edge-disjoint subgraphs in an acyclic way in such a way that each subgraph has one of the above four structural properties, then the SDP relaxation is exact.

In Section 6.6, a detailed discussion is given to demonstrate how the results of this chapter can be used for optimization over power networks. Finally, four illustrative examples are provided in section 6.7.

6.3 SDP, Reduced-SDP and SOCP Relaxations

In this section, the objective is to derive necessary and sufficient conditions for the exactness of the SDP, reduced-SDP and SOCP Relaxations. For every $r \in \{1, 2, ..., p\}$, let $\vec{\mathcal{O}}_r$ denote a directed cycle corresponding to $\mathcal{O}_r$, meaning that all edges of the undirected cycle $\mathcal{O}_r$ have been oriented consistently.

Theorem 1. *The following statements hold true in both real and complex cases $\mathcal{D} = \mathcal{R}$ and $\mathcal{D} = \mathcal{C}$:*

i) The SDP relaxation is exact (i.e., $f^*_{SDP} = f^*$) if and only if it has a rank-1 solution X^*.

ii) The reduced SDP relaxation is exact (i.e., $f^*_{r\text{-}SDP} = f^*$) if and only if it has a solution X^* such that

$$Rank\{X^*\{\mathcal{G}_s\}\} = 1, \qquad \forall \mathcal{G}_s \in \Omega. \tag{6.8}$$

iii) The SOCP relaxation is exact (i.e., $f^*_{SOCP} = f^*$) if and only if it has a solution X^* such that

$$Rank\{X^*\{(i,j)\}\} = 1, \qquad \forall (i,j) \in \mathcal{G}$$

and that

$$\sum \measuredangle X^*_{ij} = 0, \qquad \forall r \in \{1, 2, ..., p\} \tag{6.9}$$

where the sum is taken over all directed edges (i, j) of the oriented cycle $\vec{\mathcal{O}}_r$. Moreover, the same result holds even if the condition (6.9) is replaced by (6.8).

Proof of Part (i): The proof is omitted due to its simplicity.

Proof of Part (ii): To prove the "only if" part, let $\mathbf{x}^*$ denote an arbitrary solution of Optimization (6.1). If $f^*_{\text{r-SDP}} = f^*$, then $X^* = (\mathbf{x}^*)(\mathbf{x}^*)^H$ is a solution of the reduced SDP relaxation, which satisfies the condition (6.8).

To prove the "if" part, consider a matrix X^* satisfying (6.8). For every $r \in \{1, ..., p\}$, since $X\{\mathcal{O}_r\}$ is positive semidefinite and rank-1, it can be written as the product of a vector and its transpose. This yields that

$$\sum \measuredangle X^*_{ij} = 0, \qquad \forall r \in \{1, 2, ..., p\} \tag{6.10}$$

where the sum is taken over all directed edges (i,j) of the oriented cycle $\vec{\mathcal{O}}_r$. Let $\mathcal{T}$ be an arbitrary spanning tree of $\mathcal{G}$. The vertices of $\mathcal{T}$ can be iteratively labeled by some real numbers (angles) $\theta_1,...,\theta_n$ in such a way that $\theta_i - \theta_j = \measuredangle X^*_{ij}$, $\forall (i,j) \in \mathcal{T}$, and that these numbers belong to the set $\{0, 180^0\}$ in the case $\mathcal{C} = \mathcal{R}$. It can be inferred from (6.10) that $\theta_i - \theta_j = \measuredangle X^*_{ij}$ for every $(i,j) \in \mathcal{G}$. Now, define $\mathbf{x}^*$ as

$$\left[\begin{array}{cccc} \sqrt{X_{11}}e^{-\theta_1 \mathrm{i}} & \sqrt{X_{22}}e^{-\theta_2 \mathrm{i}} & \cdots & \sqrt{X_{nn}}e^{-\theta_n \mathrm{i}} \end{array} \right]^H$$

Observe that $(\mathbf{x}^*)(\mathbf{x}^*)^H$ and $\mathbf{X}^*$ are the same on the diagonal and have identical off-diagonal entries $(i,j) \in \mathcal{G}$. This implies that $(\mathbf{x}^*)(\mathbf{x}^*)^H$ is a rank-1 solution of the reduced SDP relaxation. Therefore, the relaxation is exact.

Proof of Part (iii): The proof is omitted due to its similarity to the proof of Part (ii) provided above. ■

Theorem 1 provides necessary and sufficient conditions for the exactness of the SDP, reduced SDP and SOCP relaxations. As mentioned before, one can write $f^*_{\text{SOCP}} \leq f^*_{\text{r-SDP}} \leq f^*_{\text{SDP}} \leq f^*$. Using the matrix completion theorem, two conclusions can be made [126]:

- If $\mathcal{G}$ is an acyclic graph, then the relation $f^*_{\text{SOCP}} = f^*_{\text{r-SDP}} = f^*_{\text{SDP}}$ holds, independent of whether or not $f^*_{\text{SDP}} = f^*$.

- Expand the graph $\mathcal{G}$ by connecting all vertices inside each cycle $\mathcal{O}_r$ to each other for $r = 1,2,...,p$. Then, the relation $f^*_{\text{r-SDP}} = f^*_{\text{SDP}}$ holds (independent of whether or not $f^*_{\text{SDP}} = f^*$) if every maximal clique (complete subgraph) of the expanded graph corresponds to a single edge of $\mathcal{G}$ or one of the cycles $\mathcal{O}_1,...,\mathcal{O}_p$. This mild condition is met for weakly cyclic graphs as well as a broad class of planar graphs.

Part (iii) of Theorem 1 shows that the SOCP relaxation is exact if two conditions are satisfied for an optimal solution X^* of this optimization: (1) every 2×2 edge submatrix $X^*\{(i,j)\}$ loses rank, and (2) if the phase of X^*_{ij} is assigned to the edge (i,j) of the graph $\mathcal{G}$ for every $(i,j) \in \mathcal{G}$, then the sum of the edge phases becomes zero for every cycle in the cycle basis. As will be shown throughout this chapter, Condition (1) is satisfied by imposing a sign definiteness constraint on each edge weight set. In contrast, Condition (2) is strongly related to the graph topology and weakly related to the structure of each edge weight set.

6.4 Real-Valued Optimization

In this section, Optimization (6.1) will be studied in the real-valued case (i.e., $\mathcal{D} = \mathcal{R}$). Since $\mathbf{x} \in \mathcal{R}^n$, one can write $\mathrm{Re}\left\{c_{ij}^t x_i x_j^H\right\} = \mathrm{Re}\left\{\mathrm{Re}\{c_{ij}^t\} x_i x_j^H\right\}$, for all $(i,j) \in \mathcal{G}$ and $t \in \{1,...,k\}$. Hence, changing the complex weight c_{ij}^t to $\mathrm{Re}\{c_{ij}^t\}$ does not affect the optimization. Therefore, with no loss of generality, assume that the edge weights are all real numbers. For every edge $(i,j) \in \mathcal{G}$, define the edge sign σ_{ij} as follows:

$$\sigma_{ij} = \begin{cases} 1 & \text{if} \quad c_{ij}^1, ..., c_{ij}^k \geq 0 \\ -1 & \text{if} \quad c_{ij}^1, ..., c_{ij}^k \leq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.11}$$

By convention, we define $\sigma_{ij} = 1$ if $c_{ij}^1 = \cdots = c_{ij}^k = 0$.

Theorem 2. *The relations $f_{SOCP}^* = f_{r\text{-}SDP}^* = f_{SDP}^* = f^*$ hold for Optimization (6.1) in the real-valued case $\mathcal{D} = \mathcal{R}$ if*

$$\sigma_{ij} \neq 0, \qquad \forall (i,j) \in \mathcal{G} \tag{6.12a}$$

$$\prod_{(i,j)\in\mathcal{O}_r} \sigma_{ij} = (-1)^{|\mathcal{O}_r|}, \qquad \forall r \in \{1,...,p\}. \tag{6.12b}$$

Proof: In light of the relation $f_{\mathrm{SOCP}}^* \leq f_{\mathrm{r\text{-}SDP}}^* \leq f_{\mathrm{SDP}}^* \leq f^*$, it suffices to prove that $f^* \leq f_{\mathrm{SOCP}}^*$. Consider an arbitrary feasible point X of Optimizations (6.5). It is enough to show the existence of a feasible point $\mathbf{x}$ for Optimization (6.1) with the property that the objective value of this optimization at $\mathbf{x}$ is lower than or equal to the objective value of the SOCP relaxation at the point X. For this purpose, choose an arbitrary spanning tree $\mathcal{T}$ of the graph $\mathcal{G}$. A set of ± 1 numbers $\sigma_1, \sigma_2, ..., \sigma_n$ can be iteratively assigned to the vertices of this tree in such a way that $\sigma_i \sigma_j = -\sigma_{ij}$ for every $(i,j) \in \mathcal{T}$ (this is because of (6.12a)). Now, it can be deduced from (6.12b) that

$$\sigma_i \sigma_j = -\sigma_{ij}, \qquad \forall (i,j) \in \mathcal{G}.$$

Corresponding to the feasible point X of the SOCP relaxation, define the vector $\mathbf{x}$ as

$$\left[\begin{array}{cccc} \sigma_1\sqrt{X_{11}} & \sigma_2\sqrt{X_{22}} & \cdots & \sigma_n\sqrt{X_{nn}} \end{array}\right]^H.$$

(note that $X_{11}, ..., X_{nn} \geq 0$ due to the condition $X \succeq 0$). Observe that

$$|x_i|^2 = X_{ii}, \qquad i = 1, ..., n. \tag{6.13}$$

On the other hand, (6.5c) yields

$$X_{ij} \leq \sqrt{X_{ii}}\sqrt{X_{jj}}, \qquad \forall (i,j) \in \mathcal{G},$$

and therefore

$$\begin{aligned} c_{ij}X_{ij} &\geq -|c_{ij}|\sqrt{X_{ii}}\sqrt{X_{jj}} = -c_{ij}\sigma_{ij}\sqrt{X_{ii}}\sqrt{X_{jj}} \\ &= c_{ij}\sigma_i\sigma_j\sqrt{X_{ii}}\sqrt{X_{jj}} = c_{ij}x_ix_j, \quad \forall (i,j) \in \mathcal{G}. \end{aligned} \tag{6.14}$$

It can be concluded from (6.13) and (6.14) that

$$l_1\left(\mathbf{x}\mathbf{x}^H\right) = l_1(X), \qquad l_2\left(\mathbf{x}\mathbf{x}^H\right) \leq l_2(X).$$

Hence, since $f_0(\cdot,\cdot)$ is increasing in its second vector argument, one can write:

$$f_j(\mathbf{y}, \mathbf{z}) \leq f_j(l_1(X), l_2(X))$$

for $j = 0, 1, ..., m$, where $\mathbf{y} = l_1\left(\mathbf{x}\mathbf{x}^H\right)$ and $\mathbf{z} = l_2\left(\mathbf{x}\mathbf{x}^H\right)$. This implies that $\mathbf{x}$ is a feasible point of Optimization (6.1) whose corresponding objective value is smaller than or equal to the objective value for the feasible point X of Optimization (6.5). This proves the claim $f^* \leq f^*_{\text{SOCP}}$ and thus completes the proof. ■

Condition (6.12a) ensures that each edge weight set is sign definite. Theorem 2 states that the SDP, reduced SDP and SOCP relaxations are exact for the original optimization (6.1) under the above sign definite condition, provided that each cycle in the cycle basis has an even number of edges with positive signs. This holds true in three important special cases, as explained below.

Corollary 1. *The relations $f^*_{SOCP} = f^*_{r\text{-}SDP} = f^*_{SDP} = f^*$ hold for Optimization (6.1) in the case $\mathcal{D} = \mathcal{R}$ if one of the following happens:*

1) $\mathcal{G}$ is acyclic with arbitrary sign definite edge sets (with respect to $\mathcal{R}$).

2) $\mathcal{G}$ is bipartite with positive weight sets.

3) $\mathcal{G}$ is arbitrary with negative weight sets.

Proof: The proof follows immediately from Theorem 2 by noting that a bipartite graph has no odd cycle. ■

Assume that the edge sets of the graph $\mathcal{G}$ are all sign definite. Corollary 1 implies a trade-off between the topology and the edge signs σ_{ij}'s. On one extreme, the edge signs could be arbitrary as long as the graph has a very sparse topology. On the other extreme, the graph topology could be arbitrary (sparse or dense) as long as the edge signs are all negative. The following theorem proves that if σ_{ij}'s are zero, Optimization (6.1) becomes NP-hard even for an acyclic graph $\mathcal{G}$.

Theorem 3. *Finding an optimal solution of Optimization (6.1) is an NP-hard problem for an acyclic $\mathcal{G}$ with sign-indefinite weight sets (even if $k = 2$).*

Proof: Given a set of real numbers $\{\omega_1, ..., \omega_t\}$, the number partitioning problem (NPP) aims to find out whether there exists a sign set $\{s_1, ..., s_t\}$ with the property

$$\sum_{i=1}^{t} s_i \omega_i = 0, \qquad s_1, ..., s_t \in \{-1, 1\}. \tag{6.15}$$

This decision problem is known to be NP-complete. NPP can be written as the following quadratic optimization:

$$\min_{s_1,...,s_{t+1}} \quad 0 \qquad \text{s.t.} \quad s_{t+1} \times \sum_{i=1}^{t} s_i \omega_i = 0, \quad s_1^2 = \cdots = s_{t+1}^2 = 1,$$

where s_{t+1} is a new slack variable, which is either -1 or 1 and has been introduced to make the first constraint of the above optimization quadratic. By defining n as $t + 1$ and $\mathbf{x}$ as $\begin{bmatrix} s_1 & s_2 & \cdots & s_{t+1} \end{bmatrix}$, the above optimization reduces to:

$$\min_{\mathbf{x}} \quad 0 \qquad \text{s.t.} \quad \sum_{i=1}^{n-1} x_i x_n \omega_i \leq 0, \quad \sum_{i=1}^{n-1} x_i x_n (-\omega_i) \leq 0, \quad x_1^2 = \cdots = x_n^2 = 1.$$

Since NPP is NP-hard, solving the above optimization is NP-hard as well. On the other hand, the generalized weighted graph for the above optimization has the following form: node n is connected to node i with the weight set $\{\omega_i, -\omega_i\}$ for $i = 1, ..., n - 1$. Hence, optimization over this acyclic graph is NP-hard. ■

Theorem 3 states that optimization over a very sparse generalized weighted graph (acyclic graph with only two elements in each weight set) is still hard unless the weight sets are sign definite. However, it will be shown in the next subsection that the SDP relaxation always has a rank-2 solution for this type of graph, which may be used to find an approximate solution to the original problem.

6.4.1 Low-Rank Solution for SDP Relaxation

Suppose that the conditions stated in Theorem 2 do not hold. The SDP relaxation may still be exact (depending on the coefficients of Optimization (6.1)), in which case the relaxation has a rank-1 solution X^*. A question arises as to whether the rank of X^* is yet small whenever the relaxation is inexact. The objective of this subsection is to address this problem in two important scenarios. Given the graph $\mathcal{G}$ and the parameters $\mathbf{x}, \mathbf{y}, \mathbf{z}$ introduced earlier, consider the optimization

$$\min_{\mathbf{x}\in\mathcal{R}^n} \quad g_0(\mathbf{y},\mathbf{z}) \qquad \text{s.t.} \;\; g_j(\mathbf{y},\mathbf{z}) \leq 0, \quad j = 1,2,...,m \tag{6.16}$$

for arbitrary functions $g_i(\cdot,\cdot)$, $i = 0,1,...,m$. The difference between the above optimization and (6.1) is that the functions $g_i(\cdot,\cdot)$'s may not be increasing in $\mathbf{z}$. In line with the technique used in Section 6.2 for the nonconvex optimization (6.1), an SDP relaxation can be defined for the above optimization. As expected, this relaxation may not have a rank-1 solution, in which case the relaxation is not exact. Nevertheless, it is beneficial to find out how small the rank of an optimal solution of this relaxation could be. This problem will be addressed next for an acyclic graph $\mathcal{G}$.

Theorem 4. *Assume that the graph $\mathcal{G}$ is acyclic. The SDP relaxation for Optimization (6.16) always has a solution X^* whose rank is at most 2.*

Proof: The SDP relaxation for Optimization (6.16) is as follows:

$$\min_{X\in\mathcal{S}^n} \quad g_0(l_1(X),l_2(X)) \quad \text{s.t.} \;\; g_j(l_1(X),l_2(X)) \leq 0 \quad j = 1,...,m, \quad X \succeq 0. \tag{6.17}$$

This is indeed a real-valued SDP relaxation. One can consider a complex-valued SDP

relaxation as

$$\min_{\tilde{X}\in\mathcal{H}^n} \quad g_0(l_1(\tilde{X}), l_2(\tilde{X})) \quad \text{s.t.} \quad g_i(l_1(\tilde{X}), l_2(\tilde{X})) \leq 0 \quad j = 1, ..., m, \quad \tilde{X} \succeq 0 \tag{6.18}$$

where its matrix variable, denoted as $\tilde{X}$, is complex. Observe that $l_1(\tilde{X}) = l_1(\text{Re}\{\tilde{X}\})$ and $l_2(\tilde{X}) = l_2(\text{Re}\{\tilde{X}\})$ for every arbitrary Hermitian matrix $\tilde{X}$, due to the fact that the edge weights of the graph $\mathcal{G}$ are all real. This implies that the real and complex SDP relaxations have the same optimal objective value (note that $\text{Re}\{\tilde{X}\} \succeq 0$ if $\tilde{X} \succeq 0$). In particular, if $\tilde{X}^*$ denotes an optimal solution of the complex SDP relaxation, $\text{Re}\{\tilde{X}^*\}$ will be an optimal solution of the real SDP relaxation. As will be shown later in Theorem 7, Optimization (6.18) has a rank-1 solution $\tilde{X}^*$. Therefore, $\tilde{X}^*$ can be decomposed as $(\tilde{\mathbf{x}}^*)(\tilde{\mathbf{x}}^*)^H$ for some complex vector $\tilde{\mathbf{x}}^*$. Now, one can write:

$$\text{Re}\{\tilde{X}^*\} = \text{Re}\{\tilde{\mathbf{x}}\}\text{Re}\{\tilde{\mathbf{x}}\}^H + \text{Im}\{\tilde{\mathbf{x}}\}\text{Im}\{\tilde{\mathbf{x}}\}^H.$$

Hence, $\text{Re}\{\tilde{X}^*\}$ is a real-valued matrix with rank at most 2 (as it is the sum of two rank-1 matrices), which is also a solution of the real SDP relaxation. ■

Theorem 4 states that the SDP relaxation of the general optimization (6.16) always has a rank 1 or 2 solution if its sparsity can be captured by an acyclic graph. This result makes no assumptions on the monotonicity of the functions $g_j(\cdot, \cdot)$'s. The SDP relaxation for Optimization (6.16) may not have a unique solution. Hence, if a sample of this optimization is solved numerically, the obtained solution may be high rank, in which case the low-rank solution X^* is hidden and needs to be recovered (following the constructive proof of the theorem).

If the functions $g_j(\cdot, \cdot)$'s are convex, then the SDP relaxation becomes a convex program. In this case, a low-rank solution X^* can be found in polynomial time. If X^* has rank-1, then the relaxation is exact. Otherwise, X^* has rank 2 from which an approximate rank-1 solution can be found by making the smallest nonzero eigenvalue of X^* equal to 0. A more powerful strategy is to kill the undesired nonzero eigenvalue by penalizing the objective function of the SDP relaxation via a regularization term such as $\alpha \times \text{Trace}\{X\}$ for an appropriate value of α. The graph of the penalized SDP relaxation is still acyclic and therefore the penalized optimization will have a rank-1 or 2 solution. Since X^* has only

one undesired eigenvalue that needs to be killed, the wealth of results in the literature of compressed sensing justifies the idea that this might be an effective heuristic method.

Theorem 4 studies the SDP relaxation for only acyclic graphs. Partial results will be provided below for cyclic graphs.

Theorem 5. *Assume that $\mathcal{G}$ is a weakly-cyclic bipartite graph, and that*

$$\sigma_{ij} \neq 0 \qquad \forall (i,j) \in \mathcal{O}_1 \cup \mathcal{O}_2 \cup \cdots \cup \mathcal{O}_p.$$

The SDP relaxation (6.3) for Optimization (6.1) in the real-valued case $\mathcal{D} = \mathcal{R}$ has a solution X^* whose rank is at most 2.

Proof: Consider the complex-valued SDP relaxation:

$$\min_{\tilde{X} \in \mathcal{H}^n} \quad f_0(l_1(\tilde{X}), l_2(\tilde{X})), \tag{6.19a}$$

$$\text{s.t.} \quad f_j(l_1(\tilde{X}), l_2(\tilde{X})) \leq 0, \quad j = 1, ..., m, \tag{6.19b}$$

$$\tilde{X} \succeq 0. \tag{6.19c}$$

As discussed in the proof of Theorem 4, three properties hold:

- The real and complex SDP relaxations have the same optimal objective value.
- If $\tilde{X}^*$ denotes an optimal solution of the complex SDP relaxation, $\text{Re}\{\tilde{X}^*\}$ turns out to be an optimal solution of the real SDP relaxation.
- If $\tilde{X}^*$ is positive semidefinite and rank-1, its real part $\text{Re}\{\tilde{X}^*\}$ is positive semidefinite and rank 1 or 2.

Hence, to prove the theorem, it suffices to show that the complex-valued optimization (6.19) has a rank-1 solution. Since every cycle of $\mathcal{G}$ has an even number of vertices (as it is bipartite), a diagonal matrix T with entries from the set $\{0, 1, \text{i}\}$ can be designed in such a way that

$$T_{ii} \times T_{jj} = \text{i}, \qquad \forall (i,j) \in \mathcal{G}. \tag{6.20}$$

The next step is to change the variable $\tilde{X}$ in Optimization (6.19) to $T\bar{X}T^H$, where $\bar{X}$ is a

Hermitian matrix variable. Equation (6.20) yields

$$\tilde{X}_{ii} = \bar{X}_{ii}, \qquad \forall i \in \mathcal{G}, \tag{6.21a}$$

$$\tilde{X}_{ij} = \alpha_{ij}\bar{X}_{ij}, \qquad \forall (i,j) \in \mathcal{G} \tag{6.21b}$$

where $\alpha_{ij} \in \{-\mathrm{i}, \mathrm{i}\}$. Therefore, by defining $\bar{c}_{ij}^t$ as $\alpha_{ij}c_{ij}^t$, one can write:

$$\mathrm{Re}\{c_{ij}^t \tilde{X}_{ij}\} = \mathrm{Re}\{\bar{c}_{ij}^t \bar{X}_{ij}\} \tag{6.22}$$

for every $t \in \{1, 2..., k\}$. It results from (6.21a) and (6.22) that if the complex-valued SDP relaxation (6.19) is reformulated in terms of $\bar{X}$, its underlying graph looks like $\mathcal{G}$ with the only difference that the weights c_{ij}^t's are replaced by $\bar{c}_{ij}^t$'s. On the other hand, since c_{ij}^t is a real number, $\bar{c}_{ij}^t$ is purely imaginary. Hence, it follows from Theorem 11 (stated later in the chapter) that the reformulated complex SDP relaxation has a rank-1 solution $\bar{X}^*$ because its graph is weakly cyclic with purely imaginary weights. Now, $\tilde{X}^* = T\bar{X}^*T^H$ becomes rank one. In other words, the complex SDP relaxation has a rank-1 solution $\tilde{X}^*$. This completes the proof. ■

There are several applications where the goal is to find a low-rank positive semidefinite matrix X satisfying a set of constraints (such as linear matrix inequalities). Theorems 4 and 5 provide sufficient conditions under which the feasibility problem

$$\begin{aligned} & f_j(l_1(X), l_2(X)) \leq 0, \quad j = 1, ..., m, \\ & X \succeq 0, \end{aligned} \tag{6.23}$$

has a low rank solution, where the rank does not depend on the size of the problem.

6.5 Complex-Valued Optimization

In this section, Optimization (6.1) will be studied in the complex-valued case $\mathcal{D} = \mathcal{C}$. Several scenarios will be explored below.

6.5.1 Acyclic Graph with Complex Edge Weights

Consider the case where each edge weight set is complex and sign definite with respect to $\mathcal{C}$.

Theorem 6. *The relations $f^*_{SOCP} = f^*_{r\text{-}SDP} = f^*_{SDP} = f^*$ hold in the complex-valued case $\mathcal{D} = \mathcal{C}$, provided that the graph $\mathcal{G}$ is acyclic and the weight set $\{c^1_{ij}, c^2_{ij}, ..., c^k_{ij}\}$ is sign definite for all $(i,j) \in \mathcal{G}$.*

Proof: The decomposition technique developed in [123] will be deployed to prove this theorem. Similar to Theorem 2, it is enough to show that $f^* \leq f^*_{\text{SOCP}}$. To this end, consider an arbitrary feasible solution of optimization (6.5), denoted as X. Given an edge $(i,j) \in \mathcal{G}$, since the set $\{c^1_{ij}, c^2_{ij}, ..., c^k_{ij}\}$ is sign definite, it follows from the hyperplane separation theorem that there exists a nonzero real vector $(\alpha_{ij}, \beta_{ij})$ such that

$$\text{Re}\{c^t_{ij}(\alpha_{ij} + \beta_{ij}\text{i})\} = \text{Re}\{c^t_{ij}\}\alpha_{ij} - \text{Im}\{c^t_{ij}\}\beta_{ij} \leq 0 \tag{6.24}$$

for every $t \in \{1, 2, ..., k\}$. On the other hand, (6.5c) yields

$$|X_{ij}| \leq \sqrt{X_{ii}}\sqrt{X_{jj}}, \qquad \forall (i,j) \in \mathcal{G}. \tag{6.25}$$

Consider the function

$$\left|X_{ij} + \gamma_{ij}(\alpha_{ij} + \beta_{ij}\text{i})\right|^2 - X_{ii}X_{jj}$$

in which γ_{ij} is an unknown real number. This function is negative at $\gamma = 0$ (because of (6.25)) and positive at $\gamma = +\infty$. Hence, due to the continuity of this function, there exists a positive number γ_{ij} such that

$$\left|X_{ij} + \gamma_{ij}(\alpha_{ij} + \beta_{ij}\text{i})\right|^2 = X_{ii}X_{jj}. \tag{6.26}$$

Define θ_{ij} as the phase of the complex number $X_{ij} + \gamma_{ij}(\alpha_{ij} + \beta_{ij}\text{i})$. A set of angles $\{\theta_1, \theta_2, ..., \theta_n\}$ can be found iteratively by exploiting the tree topology of the graph $\mathcal{G}$ in such a way that

$$\theta_i - \theta_j = \theta_{ij}, \quad \forall (i,j) \in \mathcal{G}. \tag{6.27}$$

Define the vector $\mathbf{x}$ as

$$\left[\begin{array}{cccc} \sqrt{X_{11}}e^{-\theta_1 \mathrm{i}} & \sqrt{X_{22}}e^{-\theta_2 \mathrm{i}} & \cdots & \sqrt{X_{nn}}e^{-\theta_n \mathrm{i}} \end{array} \right]^H. \tag{6.28}$$

Using (6.24), (6.26) and (6.27), one can write:

$$\begin{aligned} \mathrm{Re}\{c_{ij}^t x_i x_j^*\} &= \mathrm{Re}\left\{c_{ij}^t \sqrt{X_{ii}}\sqrt{X_{jj}}e^{(\theta_i-\theta_j)\mathrm{i}}\right\} = \mathrm{Re}\left\{c_{ij}^t \sqrt{X_{ii}}\sqrt{X_{jj}}e^{\theta_{ij}\mathrm{i}}\right\} \\ &= \mathrm{Re}\big\{c_{ij}^t (X_{ij} + \gamma_{ij}(\alpha_{ij} + \beta_{ij}\mathrm{i}))\big\} \\ &= \mathrm{Re}\{c_{ij}^t X_{ij}\} + \gamma_{ij}\mathrm{Re}\big\{c_{ij}^t(\alpha_{ij} + \beta_{ij}\mathrm{i})\big\} \leq \mathrm{Re}\{c_{ij}^t X_{ij}\} \end{aligned}$$

for every $t \in \{1, 2, ..., k\}$. Having shown the above relation, the rest of the proof is in line with the proof of Theorem 2. More precisely, the above inequality implies that

$$l_1\left(\mathbf{x}\mathbf{x}^H\right) = l_1(X), \qquad l_2\left(\mathbf{x}\mathbf{x}^H\right) \leq l_2(X)$$

and therefore

$$f_j(\mathbf{y}, \mathbf{z}) \leq f_i(l_1(X), l_2(X)), \qquad j = 0, 1, ..., m$$

where $\mathbf{y} = l_1\left(\mathbf{x}\mathbf{x}^H\right)$ and $\mathbf{z} = l_2\left(\mathbf{x}\mathbf{x}^H\right)$. Hence, $\mathbf{x}$ is a feasible point of Optimization (6.1) whose corresponding objective value is smaller than or equal to the objective value for the feasible point X of Optimization (6.5). Consequently, $f^* \leq f^*_{\mathrm{SOCP}}$. This completes the proof. ■

The quadratically-constrained quadratic optimization (6.7) is a special case of optimization (6.1). Hence, the SDP relaxation is tight for this QCQP problem if $\mathcal{G}$ is acyclic with sign definite weight sets. This result improves upon the result developed in [125] by removing the assumption $M_0 \succeq 0$ (see Section 6.2.3).

Corollary 2. *The relations $f^*_{SOCP} = f^*_{r\text{-}SDP} = f^*_{SDP} = f^*$ hold in the complex-valued case $\mathcal{D} = \mathcal{C}$ if the graph $\mathcal{G}$ is acyclic and $k \leq 2$.*

Proof: The proof is an immediate consequence of Theorem 6 and the fact that every complex set with one or two elements is sign definite. ■

Corollary 2 states that Optimization (6.1) in the complex-valued case can be solved through three relaxations if its structure can be captured by an acyclic graph with at most two weights on each of its edges.

6.5.2 Weakly Cyclic Graph with Real Edge Weights

It is shown in the preceding subsection that the SDP relaxation is exact, provided $\mathcal{G}$ is acyclic and each weight set is sign definite with respect to $\mathcal{C}$. This result requires the assumption of monotonicity of $f_j(\mathbf{y},\mathbf{z})$ in $\mathbf{z}$ for $j = 0, 1, ..., m$. The first objective of this part is to show that this assumption is not needed as long as the weight sets are real. To this end, consider the optimization

$$\min_{\mathbf{x}\in\mathcal{C}^n} \quad g_0(\mathbf{y},\mathbf{z}) \qquad \text{s.t.} \quad g_j(\mathbf{y},\mathbf{z}) \leq 0, \quad j = 1, 2, ..., m \tag{6.29}$$

for arbitrary functions $g_i(\cdot,\cdot)$, $i = 0, 1, ..., m$. The difference between the above optimization and (6.1) is that the functions $g_j(\cdot,\cdot)$'s may not be increasing in $\mathbf{z}$. One can derive the SDP, reduced SDP and SOCP relaxations for the above optimization by replacing $f_0, ..., f_m$ with $g_0, ..., g_m$ in (6.3)-(6.5). This part aims to investigate the case when the edge weights are all real numbers, while the unknown parameter $\mathbf{x}$ is complex.

Theorem 7. *Consider the complex-valued case $\mathcal{D} = \mathcal{C}$ and assume that the edge weights of $\mathcal{G}$ are all real numbers. The SDP, reduced SDP and SOCP relaxations associated with Optimization (6.29) are all exact if the graph $\mathcal{G}$ is acyclic.*

Proof: It is straightforward to show that every real set is sign definite with respect to $\mathcal{C}$. Therefore, the edge weight sets of $\mathcal{G}$ are all sign definite. Let X denote an arbitrary feasible point of the SOCP relaxation. Define $(\alpha_{ij}, \beta_{ij})$ as $(0, 1)$ for every $(i, j) \in \mathcal{G}$. Then,

$$\mathrm{Re}\{c_{ij}^t(\alpha_{ij} + \beta_{ij}\mathrm{i})\} = \mathrm{Re}\{c_{ij}^t\}\alpha_{ij} - \mathrm{Im}\{c_{ij}^t\}\beta_{ij} = 0$$

for every $t \in \{1, ..., k\}$ (note that $c_{ij}^t \in \mathcal{R}$ by assumption). Following the proof of Theorem 6, define $\mathbf{x}$ as the vector given in (6.28). Therefore,

$$\mathrm{Re}\{c_{ij}^t x_i x_j^*\} = \mathrm{Re}\{c_{ij}^t X_{ij}\} + \gamma_{ij}\mathrm{Re}\big\{c_{ij}^t(\alpha_{ij} + \beta_{ij}\mathrm{i})\big\} = \mathrm{Re}\{c_{ij}^t X_{ij}\}.$$

Now, the rest of the proof is in line with the proof of Theorem 6. More precisely,

$$l_1\left(\mathbf{x}\mathbf{x}^H\right) = l_1(X), \qquad l_2\left(\mathbf{x}\mathbf{x}^H\right) = l_2(X).$$

Given an arbitrary feasible point X for the SOCP relaxation, the above equality implies that $\mathbf{x}$ is a feasible point of the original optimization (6.29) and that X and $\mathbf{x}$ both give rise to the same objective value. This completes the proof. ■

Consider the general optimization (6.29) in the case when $\mathcal{G}$ is acyclic with real edge weights. As discussed before, the associated SDP relaxation may not be tight if its variable $\mathbf{x}$ is restricted to real numbers. However, Theorem 7 shows that the relaxation is exact if $\mathbf{x}$ is a complex-valued variable. In what follows, the results of Theorem 7 will be generalized to cyclic graphs for Optimization (6.1).

Theorem 8. *Assume that $\{c_{ij}^1, ..., c_{ij}^k\}$ is a positive or negative real set for every $(i,j) \in \mathcal{G}$. The relations $f_{r\text{-}SDP}^* = f_{SDP}^* = f^*$ hold for Optimization (6.1) in the complex-valued case $\mathcal{D} = \mathcal{C}$ if the graph $\mathcal{G}$ is bipartite and weakly cyclic.*

Proof: Following the proof of Theorem 5, consider the matrix T defined in (6.20), and change the variable X in the SDP relaxation to $\bar{X}$ through the relation $X = T\bar{X}T^H$. This implies that the real weights c_{ij}^t's will change to the imaginary weights $\bar{c}_{ij}^t$'s defined in the proof of Theorem 5. Hence, the reformulated SDP optimization is over a graph with purely imaginary weights. The existence of a rank-1 solution $\bar{X}^*$ (and hence a rank-1 matrix X^*) is guaranteed by Theorem 10. ■

Note that the SOCP relaxation may not be exact under the assumptions of Theorem 8. As a direct application of this theorem, the class of quadratic optimizations proposed later in Example 3 is polynomial-time solvable.

6.5.3 Cyclic Graph with Real and Imaginary Edge Weights

In this part, there is no specific assumption on the topology of the graph $\mathcal{G}$, but it is assumed that each edge weight is either real or purely imaginary. The definition of the edge sign σ_{ij} introduced earlier for real-valued weight sets can be extended as follows:

$$\sigma_{ij} = \begin{cases} 1 & \text{if} \quad c_{ij}^1, ..., c_{ij}^k \geq 0 \\ -1 & \text{if} \quad c_{ij}^1, ..., c_{ij}^k \leq 0 \\ \mathrm{i} & \text{if} \quad c_{ij}^1 \times \mathrm{i}, ..., c_{ij}^k \times \mathrm{i} \geq 0 \\ -\mathrm{i} & \text{if} \quad c_{ij}^1 \times \mathrm{i}, ..., c_{ij}^k \times \mathrm{i} \leq 0 \\ 0 & \text{otherwise} \end{cases} , \quad \forall (i,j) \in \mathcal{G}.$$

The parameter σ_{ij} being nonzero implies that the elements of each edge weight set $\{c_{ij}^1, ..., c_{ij}^k\}$ are homogeneous in type (real or imaginary) and in sign (positive or negative).

Theorem 9. *The relations $f^*_{SOCP} = f^*_{r\text{-}SDP} = f^*_{SDP} = f^*$ hold for Optimization (6.1) in the complex-valued case $\mathcal{D} = \mathcal{C}$ with real and purely imaginary edge weight sets if*

$$\sigma_{ij} \neq 0, \qquad \forall (i,j) \in \mathcal{G}, \tag{6.30a}$$

$$\prod_{(i,j)\in\mathcal{O}_r} \sigma_{ij} = (-1)^{|\mathcal{O}_r|}, \qquad \forall r \in \{1, ..., p\}. \tag{6.30b}$$

Proof: Consider an arbitrary feasible point X for the SOCP relaxation. Choose a spanning tree of $\mathcal{G}$ and denote it as $\mathcal{T}$. In light of (6.30a), n numbers $\sigma_1, \sigma_2, ..., \sigma_n$ can be iteratively obtained from σ_{ij}'s with the property that $\sigma_i\sigma_j = -\sigma_{ij}$ for every $(i,j) \in \mathcal{T}$. This relation together with (6.30b) yields that $\sigma_i\sigma_j = -\sigma_{ij}$ for every $(i,j) \in \mathcal{G}$. Now, define $\mathbf{x}$ as $\begin{bmatrix} \sigma_1\sqrt{X_{11}} & \sigma_2\sqrt{X_{22}} & \cdots & \sigma_n\sqrt{X_{nn}} \end{bmatrix}^H$. In line with the proofs of Theorems 2 and 6, it can be shown that $l_1\left(\mathbf{x}\mathbf{x}^H\right) = l_1(X)$ and $l_2\left(\mathbf{x}\mathbf{x}^H\right) \leq l_2(X)$; therefore $f_j(\mathbf{y}, \mathbf{z}) \leq f_j(l_1(X), l_2(X))$ for $j = 0, 1, ..., m$, where $\mathbf{y} = l_1\left(\mathbf{x}\mathbf{x}^H\right)$ and $\mathbf{z} = l_2\left(\mathbf{x}\mathbf{x}^H\right)$. This means that corresponding to every feasible point X of the SOCP relaxation, the original optimization has a feasible point $\mathbf{x}$ with a lower or equal objective value. Therefore, $f^* \leq f^*_{\text{SOCP}}$. The proof is complete by combining this inequality with $f^*_{\text{SOCP}} \leq f^*_{\text{r-SDP}} \leq f^*_{\text{SDP}} \leq f^*$. ■

6.5.4 Weakly Cyclic Graph with Imaginary Edge Weights

If $\mathcal{G}$ has at least one odd cycle whose edge weights sets are all imaginary sets, then the conditions given in Theorem 9 are violated. The reason is that the product of an odd number of imaginary numbers (edge signs) can never become a real number. The high-level goal of this part is to show that the SDP relaxation can still be tight in presence of such cycles, while the SOCP relaxation is not guaranteed to be exact. In this subsection, we assume that $\mathcal{G}$ is weakly cyclic.

To proceed with this chapter, a new SOCP relaxation needs to be introduced. This optimization assigns one real scalar variable q_i to every vertex $i \in \mathcal{G}$ and one 2×2 block matrix variable

$$\begin{bmatrix} U(\mathcal{G}_s) & V(\mathcal{G}_s) \\ V(\mathcal{G}_s)^H & W(\mathcal{G}_s) \end{bmatrix}$$

to every subgraph $\mathcal{G}_s \in \Omega$, where $U(\mathcal{G}_s), W(\mathcal{G}_s) \in \mathcal{S}^{|\mathcal{G}_s|}$ and $V(\mathcal{G}_s) \in \mathcal{R}^{|\mathcal{G}_s|\times|\mathcal{G}_s|}$. Let U, V and W denote the parameter sets $\{U(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega\}$, $\{V(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega\}$ and $\{W(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega\}$, respectively.

Notation 7. *For every $\mathcal{G}_s \in \Omega$, we arrange the elements in the vertex set of $\mathcal{G}_s$ in an increasing order. Then, we index the rows and columns of each of the matrices $U(\mathcal{G}_s), V(\mathcal{G}_s), V(\mathcal{G}_s)$ according to the ordered vertex set of $\mathcal{G}_s$. For example, if $\mathcal{G}_s$ has three vertices $5, 7, 1$, the ordered set becomes $\{1, 5, 7\}$, and therefore the three rows of $U(\mathcal{G}_s)$ are called row 1, row 5 and row 7. As an example, $U_{17}(\mathcal{G}_s)$ refers to the last entry on the first row of $U(\mathcal{G}_s)$.*

For every $r \in \{1, 2, ..., p\}$, let μ_r denote the largest index in the vertex set of $\mathcal{O}_r$. Define $\mathbf{q}$ as the vector corresponding to the set $\{q_1, ..., q_n\}$. Recall that

$$l_2(\mathbf{x}\mathbf{x}^H) = \left\{ \mathrm{Re}\left\{ c_{ij}^t x_i x_j^H \right\} \;\middle|\; \forall (i,j) \in \mathcal{G},\ t \in \{1, ..., k\} \right\}.$$

Define $\bar{l}(V)$ as a vector obtained from $l_2(\mathbf{x}\mathbf{x}^H)$ by replacing each entry $\mathrm{Re}\left\{ c_{ij}^t x_i x_j^H \right\}$ with a new term $\mathrm{Im}\{c_{ij}^t\} \times (V_{ij}(\mathcal{G}_s) - V_{ji}(\mathcal{G}_s))$, where $\mathcal{G}_s$ denotes the unique subgraph in Ω containing the edge (i, j) (the uniqueness of such subgraph is guaranteed by the weakly cyclic property of $\mathcal{G}$).

Expanded SOCP: This optimization is defined as

$$\min_{\mathbf{q}, U, V, W} \quad f_0(\mathbf{q}, \bar{l}(V)), \tag{6.31a}$$

subject to:

$$f_j(\mathbf{q}, \bar{l}(V)) \leq 0, \qquad j = 1, 2, ..., m, \tag{6.31b}$$

$$U_{ii}(\mathcal{G}_s) + W_{ii}(\mathcal{G}_s) = q_i, \quad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s, \tag{6.31c}$$

$$\begin{bmatrix} U_{ii}(\mathcal{G}_s) & V_{ij}(\mathcal{G}_s), \\ V_{ij}(\mathcal{G}_s) & W_{jj}(\mathcal{G}_s) \end{bmatrix} \succeq 0,\ \forall \mathcal{G}_s \in \Omega,\ (i,j) \in \mathcal{G}_s, \tag{6.31d}$$

$$\begin{bmatrix} U_{jj}(\mathcal{G}_s) & V_{ji}(\mathcal{G}_s) \\ V_{ji}(\mathcal{G}_s) & W_{ii}(\mathcal{G}_s) \end{bmatrix} \succeq 0,\ \forall \mathcal{G}_s \in \Omega,\ (i,j) \in \mathcal{G}_s, \tag{6.31e}$$

$$W_{\mu_r \mu_r}(\mathcal{O}_r) = 0, \qquad r = 1, 2, ..., p. \tag{6.31f}$$

Similar to the argument made for the SOCP relaxation (6.5), the above optimization

is in the form of an SOCP program because its constraints (6.31d) and (6.31e) can be replaced by linear and norm constraints. Moreover, this optimization can be regarded as an expanded version of the SOCP relaxation (6.5). Denote the optimal objective value of this optimization as $f^*_{\text{e-SOCP}}$.

Theorem 10. *Consider Optimization (6.1) in the complex-valued case $\mathcal{D} = \mathcal{C}$, and assume that the graph $\mathcal{G}$ is weakly cyclic with only purely imaginary edge weights. The following statements hold:*

i) The expanded SOCP is a relaxation for Optimization (6.1), meaning that $f^*_{e\text{-}SOCP} \leq f^*$.

ii) The expanded SOCP relaxation is exact if and only if it has a solution $(\mathbf{q}^*, U^*, V^*, W^*)$ for which all 2×2 matrices given in (6.31d) and (6.31e) have rank 1.

iii) $f^*_{SOCP} \leq f^*_{e\text{-}SOCP}$.

iv) $f^*_{e\text{-}SOCP} \leq f^*_{r\text{-}SDP}$.

v) The relations $f^*_{e\text{-}SOCP} = f^*_{r\text{-}SDP} = f^*_{SDP} = f^*$ hold if $\sigma_{ij} \neq 0$ for every $(i,j) \in \mathcal{G}$.

Proof: Since the proof is long and involved, it has been moved to the appendix of Chapter 6. ■

Assume that the graph $\mathcal{G}$ is weakly cyclic and its edge weights are all imaginary numbers. Theorem 10 shows that $f^*_{\text{SOCP}} \leq f^*_{\text{e-SOCP}} \leq f^*_{\text{r-SDP}} \leq f^*_{\text{SDP}} \leq f^*$, and that the relations $f^*_{\text{e-SOCP}} = f^*_{\text{r-SDP}} = f^*_{\text{SDP}} = f^*$ hold if each edge weight set has homogeneous elements ($\sigma_{ij} = i$ or $-i$). Note that the SOCP relaxation may not be exact, and one needs to use the expanded SOCP relaxation in this case. Interestingly, this result makes no assumption concerning the signs of the edges belonging to the same cycle in the cycle basis (unlike (6.30b)).

Although Theorem 10 deals with imaginary coefficients, some of the results derived in this chapter for complex/real optimizations with real coefficients are based on this powerful theorem. This is due to the fact that real numbers may be converted to imaginary numbers through a simple multiplication.

6.5.5 General Graph with Complex Edge Weight Sets

Given an arbitrary subgraph $\tilde{\mathcal{G}}_s$ of the graph $\mathcal{G}$, four important types will be defined for this subgraph in the following:

- **Type I:** $\tilde{\mathcal{G}}_s$ is acyclic with complex weight sets with the property that $\{c_{ij}^1, ..., c_{ij}^k\}$ is sign definite with respect to $\mathcal{C}$ for every $(i,j) \in \tilde{\mathcal{G}}_s$.
- **Type II:** $\tilde{\mathcal{G}}_s$ is weakly cyclic with imaginary weight sets and nonzero sign σ_{ij} (i.e., $\sigma_{ij} = \pm \mathrm{i}$) for every $(i,j) \in \tilde{\mathcal{G}}_s$.
- **Type III:** $\tilde{\mathcal{G}}_s$ is bipartite and weakly cyclic with the property that $\{c_{ij}^1, ..., c_{ij}^k\}$ is a real weight set with nonzero sign σ_{ij} (i.e., $\sigma_{ij} = \pm 1$) for every $(i,j) \in \tilde{\mathcal{G}}_s$.
- **Type IV:** $\tilde{\mathcal{G}}_s$ has only real and imaginary weights with the property that

$$\sigma_{ij} \neq 0, \qquad \forall (i,j) \in \tilde{\mathcal{G}}_s, \tag{6.32a}$$

$$\prod_{(i,j)\in\mathcal{O}_r} \sigma_{ij} = (-1)^{|\mathcal{O}_r|}, \quad \forall \mathcal{O}_r \in \{\mathcal{O}_1, ..., \mathcal{O}_p\} \cap \tilde{\mathcal{G}}_s. \tag{6.32b}$$

By assuming $\tilde{\mathcal{G}}_s = \mathcal{G}_s$, it follows from the theorems developed in this section that the SDP relaxation is exact for Optimization (6.1) if $\mathcal{G}$ is Type I, II, III or IV. In this part, the objective is to show that the relaxation is still tight if $\mathcal{G}$ can be decomposed into a number of Type I-IV subgraphs in an acyclic way.

Theorem 11. *Assume that $\mathcal{G}$ can be decomposed as the union of a number of edge-disjoint subgraphs $\tilde{\mathcal{G}}_1, ..., \tilde{\mathcal{G}}_\omega$ in such a way that:*

i) $\tilde{\mathcal{G}}_s$ is Type I, II, III or IV for every $s \in \{1, ..., \omega\}$.

ii) The cycle $\mathcal{O}_r$ is entirely inside one of the subgraphs $\tilde{\mathcal{G}}_1, ..., \tilde{\mathcal{G}}_\omega$ for every $r \in \{1, ..., p\}$.

Then, the relations $f_{r\text{-}SDP}^* = f_{SDP}^* = f^*$ hold for Optimization (6.1) in the complex-valued case $\mathcal{D} = \mathcal{C}$.

Proof: Given an arbitrary solution X^* of the reduced SDP relaxation, consider the optimization:

$$\min_{X} \quad f_0(l_1(X), l_2(X)), \tag{6.33a}$$

$$\text{s.t.} \quad f_j(l_1(X), l_2(X)) \leq 0, \qquad j = 1, ..., m, \tag{6.33b}$$

$$X\{\mathcal{O}_r\} \succeq 0, \qquad r = 1, ..., p, \tag{6.33c}$$

$$X\{(i,j)\} \succeq 0, \qquad \forall (i,j) \in \mathcal{G}, \tag{6.33d}$$

$$X_{ii} = X^*_{ii}, \qquad \forall i \in \mathcal{G}, \tag{6.33e}$$

$$X_{ij} = X^*_{ij}, \qquad \forall (i,j) \in \mathcal{G} \backslash \tilde{\mathcal{G}}_s, \tag{6.33f}$$

for any subgraph $\tilde{\mathcal{G}}_s \in \{\tilde{\mathcal{G}}_1, ..., \tilde{\mathcal{G}}_\omega\}$ ($\mathcal{G}\backslash\tilde{\mathcal{G}}_s$ means to exclude the edges of $\tilde{\mathcal{G}}_s$ from $\mathcal{G}$). The above optimization is obtained from the reduced SDP relaxation by setting certain entries of the variable X equal to their optimal values extracted from X^*. More precisely, this optimization aims to optimize the off-diagonal entries of X corresponding to the edges of $\tilde{\mathcal{G}}_s$. It is obvious that $X = X^*$ is a solution of the above optimization. On the other hand, since $\tilde{\mathcal{G}}_s$ is Type I, II, III or IV, it follows from Theorems 6, 8, 9 and 10 that the above optimization has an optimal solution for which the matrices given in (6.33c) and (6.33d) become rank-1 for every (i,j) and $\mathcal{O}_r$ belonging to $\tilde{\mathcal{G}}_s$. By making this argument on all subgraphs $\tilde{\mathcal{G}}_1, ..., \tilde{\mathcal{G}}_\omega$ and using Property (ii) stated in the theorem, one can design a solution for the reduced SDP relaxation for which Condition (6.8) holds. Therefore, the SDP and reduced SDP relaxations will both be exact in light of Theorem 1. ■

6.5.6 Roles of Graph Topology and Sign Definite Weight Sets

Part (iii) of Theorem 1 states that Optimization (6.1) is polynomial-time solvable if the SOCP relaxation (6.5) has a solution X^* satisfying two conditions:

1) $X^*\{(i,j)\}$ has rank 1 for every $(i,j) \in \mathcal{G}$.

2) $\sum \measuredangle X^*_{ij}$ is equal to zero for every $r \in \{1, 2, ..., p\}$, where the sum is taken over all directed edges (i,j) of the oriented cycle $\vec{\mathcal{O}}_r$.

Since $X^*\{(i,j)\}$ is a 2×2 matrix corresponding to a single edge of the graph, Condition (1) is strongly related to the properties of the edge set $\{c^1_{ij}, ..., c^k_{ij}\}$. In contrast, the graph

topology (namely its cycle basis) plays an important role in Condition (2). The goal of this part is to understand how these conditions are satisfied for various graphs studied earlier in the complex-valued case $\mathcal{D} = \mathcal{C}$.

To explore Condition (1), consider an edge $(i, j) \in \mathcal{G}$. Observe that the set $\{c_{ij}^1, ..., c_{ij}^k\}$ can be mapped into k vectors

$$\vec{c}_{ij}^{\,t} = \left[\begin{array}{cc} \mathrm{Re}\{c_{ij}^t\} & \mathrm{Im}\{c_{ij}^t\} \end{array} \right]^H, \qquad t = 1, 2, ..., k$$

in $\mathcal{R}^2$. Define the following vector corresponding to X_{ji}^*:

$$\vec{X}_{ji}^* = \left[\begin{array}{cc} \mathrm{Re}\{X_{ij}^*\} & -\mathrm{Im}\{X_{ij}^*\} \end{array} \right]^H.$$

Recall that X_{ij}^* plays the role of $(x_i^*)(x_j^*)^H$ whenever the SOCP relaxation is tight. Now, one can write

$$\mathrm{Re}\{c_{ij}^t X_{ij}^*\} = \vec{c}_{ij}^{\,t} \cdot \vec{X}_{ji}^* = |\vec{c}_{ij}^{\,t}||\vec{X}_{ji}^*| \cos(\measuredangle \vec{c}_{ij}^{\,t} - \measuredangle \vec{X}_{ji}^*) \tag{6.34}$$

where "$\cdot$" stands for *inner product*. Define $\mathcal{C}_{ij}$ as the smallest convex cone in $\mathcal{R}^2$ containing the vectors $\vec{c}_{ij}^{\,1}, ..., \vec{c}_{ij}^{\,k}$. Let $\mathrm{B}\{\mathcal{C}_{ij}\}$ denote the boundary of the cone $\mathcal{C}_{ij}$. The set $\{c_{ij}^1, ..., c_{ij}^k\}$ being sign definite is equivalent to the condition

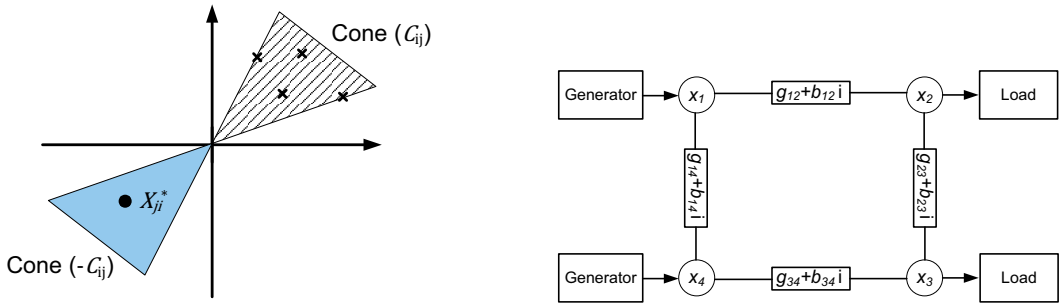
$$\{\mathcal{C}_{ij} \cap (-\mathcal{C}_{ij})\} \subseteq \mathrm{B}\{\mathcal{C}_{ij}\}, \tag{6.35}$$

meaning that $\mathcal{C}_{ij}$ and its mirror set can have common points only on their boundaries. This fact is illustrated in Figure 6.2(a). Suppose that the weight set $\{c_{ij}^1, ..., c_{ij}^k\}$ is sign definite. Since $f_0, ..., f_m$ are all increasing in $\mathbf{z}$ or equivalently in $\vec{c}_{ij}^{\,t} \cdot \vec{X}_{ji}^*$ for every $(i, j) \in \mathcal{G}$ and $t \in \{1, ..., k\}$, it is easy to verify that (see the proof of Theorem 6):

$$\vec{X}_{ji}^* \in -\mathcal{C}_{ij}. \tag{6.36}$$

This property is illustrated in Figure 6.2(a). Moreover, the monotonicity of $f_0, ..., f_m$ forces $|\vec{X}_{ij}^*|$ to have the largest possible value, i.e.,

$$|\vec{X}_{ji}^*| = |X_{ij}^*| = \sqrt{X_{ii}^* X_{jj}^*},$$



(a) (b)

Figure 6.2: (a) This figure shows the cones $\mathcal{C}_{ij}$ and $-\mathcal{C}_{ij}$, in addition to the position of the complex point X^*_{ji}. (b) An example of the power circuit studied in Section 6.6.

which makes $X^*\{(i,j\})$ rank 1. This implies that the sign definiteness of the set $\{c^1_{ij},...,c^k_{ij}\}$ guarantees the satisfaction of Condition (1) stated above.

So far, it is shown that $\vec{X}^*_{ji}$ belongs to the cone $-\mathcal{C}_{ij}$. Now, to satisfy Condition (2) required for the exactness of the SOCP relaxation, the sum of the angles of the vectors $\vec{X}^*_{ji}$'s must be zero over each cycle in the cycle basis. This trivially happens in two cases:

- If the graph $\mathcal{G}$ is acyclic, then there is no cycle to be concerned about.
- Consider the cycle $\mathcal{O}_r$ for some $r \in \{1,2,...,k\}$. If each cone $\mathcal{C}_{ij}$ is one dimensional for every $(i,j) \in \mathcal{O}_r$, then it suffices to have $\sum \measuredangle(-\mathcal{C}_{ij}) = 0$, where the sum is taken over all directed edges (i,j) of the oriented cycle $\vec{\mathcal{O}}_r$ (note that $\measuredangle(-\mathcal{C}_{ij})$ denotes the angle of the 1-d cone $-\mathcal{C}_{ij}$).

To understand the merit of the above insights, consider Optimization (6.1) in the case when the graph $\mathcal{G}$ is bipartite and each complex weight c^t_{ij} has positive real and imaginary parts for every $(i,j) \in \mathcal{G}$ and $t \in \{1,...,k\}$. Denote the two disjoint vertex sets of the bipartite graph $\mathcal{G}$ as $\mathcal{S}_1$ and $\mathcal{S}_2$, and with no loss of generality, assume that $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$ for every $(i,j) \in \mathcal{G}$. Suppose that the constraints of Optimization (6.1) are such that the inequality

$$|\measuredangle x^*_i - \measuredangle x^*_j| \leq \frac{\pi}{2}, \quad \forall (i,j) \in \mathcal{G} \tag{6.37}$$

is satisfied for an optimal solution $\mathbf{x}^*$ of this optimization. For instance, as will be discussed later in Example 5, phasor voltages in a power network are forced to satisfy the above condition due to the operational constraints of such networks. Under this circumstance, one can modify the SOCP relaxation by including the extra constraints $\text{Re}\{X_{ij}\} \geq 0$,

$\forall(i,j) \in \mathcal{G}$, to account for (6.37). Since $\mathcal{C}_{ij}$ is a subset of a first quadrant in $\mathcal{R}^2$, $\{c_{ij}^1, ..., c_{ij}^k\}$ is a sign definite set and therefore Condition (1) holds. Let X^* denote a solution of the modified SOCP problem. Following the argument leading to (6.36), it can be shown that X_{ji}^* is a negative imaginary number for every $(i,j) \in \mathcal{G}$, meaning that $\vec{X}_{ji}^*$ has the maximum possible angle with respect to all vectors $\vec{c}_{ij}^{\,1}, ..., \vec{c}_{ij}^{\,k}$. Since $\mathcal{G}$ is assumed to be bipartite, Condition (2) holds as a result of this property. Hence, the SOCP, reduced SDP, and SDP relaxations are all exact for such graphs $\mathcal{G}$.

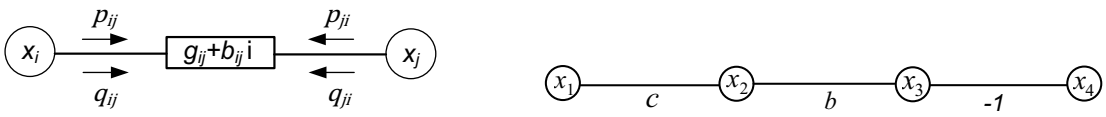
The above insight into Conditions (1) and (2) was based on the SOCP relaxation. The same argument can be made about the expanded SOCP relaxation to understand Theorem 10 for weakly cyclic graphs with imaginary weights, for which the regular SOCP relaxation may not be tight.

6.6 Application in Power Systems

A majority of real-world optimizations are naturally "optimization over graph", meaning that the optimization is defined over the graph characterizing a physical system. For example, optimizations in circuits, antenna systems and communication networks can easily be regarded as "optimization over graph". Then, the question of interest is: how does the computational complexity of an optimization relate to the structure of the system over which the optimization is performed? This question will be explored here in the context of electrical power grids. Assume that the graph $\mathcal{G}$ corresponds to an AC power network, where:

- The power network has $|\mathcal{G}|$ nodes.
- For every $(i,j) \in \mathcal{G}$, nodes i and j are connected to each other in the power network via a transmission line with the impedance $g_{ij} + b_{ij}\mathrm{i}$.
- Each node $i \in \mathcal{G}$ of the network is connected to an external device, which exchanges electrical power with the power network.

Figure 6.2(b) exemplifies a sample power network in which two external devices generate power while the remaining ones consume power. As shown in Figure 6.3(a), each line $(i,j) \in \mathcal{G}$ is associated with four power flows:



(a) (b)

Figure 6.3: (a) This figure illustrates that each transmission line has four flows. (b) Graph $\mathcal{G}$ corresponding to minimization of $f_0(x_1, x_2)$ given in (6.41).

- p_{ij}: Active power entering the line from node i.
- p_{ji}: Active power entering the line from node j.
- q_{ij}: Reactive power entering the line from node i.
- q_{ji}: Reactive power entering the line from node j.

Note that $p_{ij} + p_{ji}$ and $q_{ij} + q_{ji}$ represent the active and reactive losses incurred in the line. Let x_i denote the complex voltage (phasor) for node $i \in \mathcal{G}$. One can write:

$$p_{ij}(\mathbf{x}) = \mathrm{Re}\left\{x_i(x_i - x_j)^H \frac{1}{g_{ij} - b_{ij}\mathrm{i}}\right\}, \quad p_{ji}(\mathbf{x}) = \mathrm{Re}\left\{x_j(x_j - x_i)^H \frac{1}{g_{ij} - b_{ij}\mathrm{i}}\right\},$$
$$q_{ij}(\mathbf{x}) = \mathrm{Im}\left\{x_i(x_i - x_j)^H \frac{1}{g_{ij} - b_{ij}\mathrm{i}}\right\}, \quad q_{ji}(\mathbf{x}) = \mathrm{Im}\left\{x_j(x_j - x_i)^H \frac{1}{g_{ij} - b_{ij}\mathrm{i}}\right\}.$$

Note that since the flows all depend on $\mathbf{x}$, the argument $\mathbf{x}$ has been added to the above equations (e.g., $p_{ij}(\mathbf{x})$ instead of p_{ij}). The flows $p_{ij}(\mathbf{x})$, $p_{ji}(\mathbf{x})$, $q_{ij}(\mathbf{x})$ and $q_{ji}(\mathbf{x})$ can all be expressed in terms of $|x_i|^2$, $|x_j|^2$ and $\mathrm{Re}\left\{c_{ij}^k x_i x_j^H\right\}$ for $k = 1, 2, 3, 4$, where

$$c_{ij}^1 = \frac{-1}{g_{ij} - b_{ij}\mathrm{i}}, \quad c_{ij}^2 = \frac{-1}{g_{ij} + b_{ij}\mathrm{i}}, \quad c_{ij}^3 = \frac{\mathrm{i}}{g_{ij} - b_{ij}\mathrm{i}}, \quad c_{ij}^4 = \frac{-\mathrm{i}}{g_{ij} + b_{ij}\mathrm{i}}$$

(note that $\mathrm{Re}\{\alpha x_j x_i^H\} = \mathrm{Re}\{\alpha^H x_i x_j^H\}$ and $\mathrm{Im}\{\alpha x_j x_i^H\} = \mathrm{Re}\{(-\alpha\mathrm{i}) x_i x_j^H\}$ for every value of α). Define

$$\mathbf{p}(\mathbf{x}) = \big\{p_{ij}(\mathbf{x}), p_{ji}(\mathbf{x}) \;\big|\; \forall (i,j) \in \mathcal{G}\big\}, \quad \mathbf{q}(\mathbf{x}) = \big\{q_{ij}(\mathbf{x}), q_{ji}(\mathbf{x}) \;\big|\; \forall (i,j) \in \mathcal{G}\big\}.$$

Consider the optimization

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{C}^n} \quad & h_0(\mathbf{p}(\mathbf{x}),\mathbf{q}(\mathbf{x}),\mathbf{y}(\mathbf{x})) \\ \text{s.t.} \quad & h_j(\mathbf{p}(\mathbf{x}),\mathbf{q}(\mathbf{x}),\mathbf{y}(\mathbf{x})) \leq 0, \quad j=1,2,...,m \end{aligned} \tag{6.38}$$

for given functions $h_0,...,h_m$, where $y(\mathbf{x})$ is the vector of $|x_i|^2$'s. This optimization aims to optimize the flows in a power grid. The constraints of this optimization are meant to limit line flows, voltage magnitudes, power delivered to each load, and power supplied by each generator. Observe that $\mathbf{p}(\mathbf{x})$ and $\mathbf{q}(\mathbf{x})$ are both quadratic in $\mathbf{x}$. Assume that $h_j(\cdot,\cdot,\cdot)$ is increasing (or decreasing) in its first and second vector arguments. Since the above optimization can be cast as (6.1), the SDP, reduced SDP and SOCP relaxations introduced before can be used to eliminate the effect of quadratic terms. To study under what conditions the relaxations are exact, note that each edge (i,j) of $\mathcal{G}$ has the weight set $\{c_{ij}^1, c_{ij}^2, c_{ij}^3, c_{ij}^4\}$. Due to the physics of a transmission line, g_{ij} and b_{ij} are both nonnegative real numbers. As a result of this property, the set $\{c_{ij}^1, c_{ij}^2, c_{ij}^3, c_{ij}^4\}$ turns out to be sign definite (see Definition 2). Now, in light of Theorem 11, the relaxations are all exact as long as $\mathcal{G}$ is acyclic. This result also holds for cyclic power networks with a sufficient number of phase shifters (the graph for a mesh power network with phase shifters can be converted to an acyclic one) [123].

Optimization of power flows is a fundamental problem, which is solved every 5 to 15 minutes in practice for power grids. This problem, named Optimal Power Flow (OPF), has several variants, which are used for different purposes (real-time operation, electricity market, security assessment, etc.). Nevertheless, a more realistic form of this optimization often has two more constraints, which cannot be described in terms of $\mathbf{p}(\mathbf{x}),\mathbf{q}(\mathbf{x}),\mathbf{y}(\mathbf{x})$:

- *Line flow constraint:* For every $(i,j)\in\mathcal{G}$, the line current magnitude $\left|\frac{x_i-x_j}{g_{ij}+b_{ij}\mathrm{i}}\right|$ cannot exceed a maximum number $I_{\max}$. This constraint can be written as:

$$|x_i|^2+|x_j|^2-2\mathrm{Re}\{x_i x_j^H\} \leq |g_{ij}+b_{ij}\mathrm{i}|^2 I_{\max}^2 \tag{6.39}$$

- *Angle constraint:* For every $(i,j)\in\mathcal{G}$. the absolute angle difference $|\measuredangle x_i - \measuredangle x_j|$ should not exceed a maximum angle $\theta_{ij}^{\max}\in[0,90^\circ]$ (due to stability and thermal limits). This

constraint can be written as

$$\mathrm{Im}\{x_i x_j^H\} \leq |\tan\theta_{ij}^{\max}| \times \mathrm{Re}\{x_i x_j^H\},$$

or equivalently

$$\begin{aligned} -\tan\theta_{ij}^{\max} \times \mathrm{Re}\{x_i x_j^H\} + \mathrm{Re}\{(+\mathrm{i})\ x_i x_j^H\} \leq 0, \\ -\tan\theta_{ij}^{\max} \times \mathrm{Re}\{x_i x_j^H\} + \mathrm{Re}\{(-\mathrm{i})\ x_i x_j^H\} \leq 0. \end{aligned} \tag{6.40}$$

Since (6.39) and (6.40) are quadratic in $\mathbf{x}$, they can easily be incorporated into Optimization (6.38) and its relaxations. However, the edge set $\{c_{ij}^1, c_{ij}^2, c_{ij}^3, c_{ij}^4\}$ should be extended to $\{c_{ij}^1, c_{ij}^2, c_{ij}^3, c_{ij}^4, -1, \mathrm{i}, -\mathrm{i}\}$ for every $(i,j) \in \mathcal{G}$. It is interesting to note that this set is still sign definite and therefore the conclusion made earlier about the exactness of various relaxations is valid under this generalization.

Another interesting case is the optimization of active power flows for lossless networks. In this case, g_{ij} is equal to zero for every $(i,j) \in \mathcal{G}$. Hence, $p_{ji}(\mathbf{x})$ can be simply replaced by $-p_{ij}(\mathbf{x})$. Motivated by this observation, define the reduced vector of active powers as $\mathbf{p}_r(\mathbf{x}) = \big\{p_{ij}(\mathbf{x}) \;\big|\; \forall (i,j) \in \mathcal{G}\big\}$, and consider the optimization

$$\min_{\mathbf{x} \in \mathcal{C}^n} \quad \bar{h}_0(\mathbf{p}_r(\mathbf{x}), \mathbf{y}(\mathbf{x})) \qquad \text{s.t.} \;\; \bar{h}_j(\mathbf{p}_r(\mathbf{x}), \mathbf{y}(\mathbf{x})) \leq 0, \quad j = 1, 2, ..., m$$

for some functions $\bar{h}_0(\cdot,\cdot), ..., \bar{h}_m(\cdot,\cdot)$, which are assumed to be increasing in their first vector argument. Now, each edge (i,j) of the graph $\mathcal{G}$ is accompanied by the singleton weight set $\left\{\frac{-\mathrm{i}}{b_{ij}}\right\}$. Due to Theorem 10, the SDP and reduced SDP relaxations are exact if $\mathcal{G}$ is weakly cyclic. This is the generalization of the result obtained in [105] for optimization over lossless networks.

6.7 Examples

In this section, four examples will be provided to illustrate various contributions of this work in certain special cases.

Example 1: Consider the problem of minimizing the bivariate polynomial

$$f_0(x_1, x_2) = x_1^4 + ax_2^2 + bx_1^2 x_2 + cx_1 x_2 \tag{6.41}$$

with the real-valued variables x_1 and x_2, where the parameters $a, b, c \in \mathcal{R}$ are known. In order to find the global minimum of this optimization, the standard convex optimization technique cannot readily be used due to the non-convexity of $f(x_1, x_2)$ for generic values of a, b and c. To address this issue, the above unconstrained minimization problem will be converted to a constrained quadratic optimization. More precisely, the problem of minimizing $f_0(x_1, x_2)$ can be reformulated in terms of x_1, x_2 and two auxiliary variables x_3, x_4 as:

$$\min_{\mathbf{x} \in \mathcal{R}^4} \quad x_3^2 + ax_2^2 + bx_3 x_2 + cx_1 x_2 \tag{6.42a}$$

$$\text{subject to} \quad x_1^2 - x_3 x_4 = 0, \qquad x_4^2 - 1 = 0 \tag{6.42b}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^H$. The above optimization can be recast as follows:

$$\min_{\mathbf{x} \in \mathcal{R}^4, X \in \mathcal{R}^{4\times 4}} \quad X_{33} + aX_{22} + bX_{23} + cX_{12} \tag{6.43a}$$

$$\text{subject to} \quad X_{11} - X_{34} \leq 0, \qquad X_{44} - 1 = 0 \tag{6.43b}$$

and subject to the additional constraint $X = \mathbf{x}\mathbf{x}^H$. Note that $X_{11} - X_{34} \leq 0$ should have been $X_{11} - X_{34} = 0$, but this modification does not change the solution. To eliminate the non-convexity induced by the constraint $X = \mathbf{x}\mathbf{x}^H$, one can use an SDP relaxation obtained by replacing the constraint $X = \mathbf{x}\mathbf{x}^H$ with the convex SDP constraint $X = X^H \succeq 0$. To understand the exactness of this relaxation, the weighted graph $\mathcal{G}$ capturing the structure of Optimization (6.42) should be constructed. This graph is depicted in Figure 6.3(b). Due to Corollary 1, since $\mathcal{G}$ is acyclic, the SDP relaxation is exact for all values of a, b, c. Note that this does not imply that every solution X of the SDP relaxation has rank 1. However, there is a simple systematic procedure for recovering a rank-1 solution from an arbitrary optimal solution of this relaxation. Note also that one can use an SOCP relaxation instead.

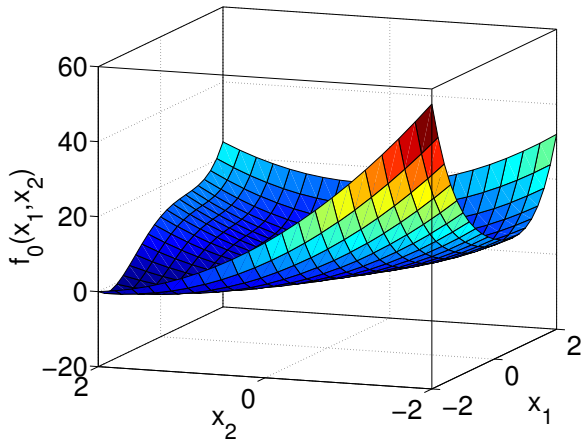


Figure 6.4: Function $f_0(x_1, x_2)$ given in (6.41) for $a = 3$, $b = -2$ and $c = 3$.

Now, assume that a set of constraints

$$f_j(x_1, x_2) = x_1^4 + a_j x_2^2 + b_j x_1^2 x_2 + c_j x_1 x_2 \leq d_j \quad j = 1, ..., m$$

has been added to Optimization (6.41) for given coefficients a_j, b_j, c_j, d_j. In this case, the graph $\mathcal{G}$ depicted in Figure 6.3(b) needs to be modified by replacing its edge sets $\{b\}$ and $\{c\}$ with $\{b, b_1, ..., b_m\}$ and $\{c, c_1, ..., c_m\}$, respectively. Due to Corollary 1, the SDP relaxation corresponding to the new optimization is exact as long as the sets $\{c, c_1, ..., c_m\}$ and $\{b, b_1, ..., b_m\}$ are both sign definite. Moreover, in light of Theorem 4, if these sets are not sign definite, then the SDP relaxation will still have a low rank (rank 1 or 2) solution.

Example 2: Consider the optimization

$$\min_{\mathbf{x} \in \mathcal{C}^7} \ \mathbf{x}^H M \mathbf{x} \qquad \text{s.t.} \ \ |x_i| = 1, \quad i = 1, 2, ..., 7 \tag{6.44}$$

where M is a given Hermitian matrix. Assume that the weighted graph $\mathcal{G}$ depicted in Figure 6.1(c) captures the structure of this optimization, meaning that (i) $M_{ij} = 0$ for every pair $(i, j) \in \{1, 2, ...7\}$ such that $(i, j) \notin \mathcal{G}$, $(j, i) \notin \mathcal{G}$, and $i \neq j$, (ii) M_{ij} is equal to the edge weight c_{ij} for every $(i, j) \in \mathcal{G}$. The SDP relaxation of this optimization is as follows:

$$\min_{X \in \mathcal{C}^{7 \times 7}} \ \text{Trace}\{MX\} \qquad \text{s.t.} \qquad X_{11} = \cdots = X_{77} = 1, \qquad X = X^H \succeq 0$$

Define $\mathcal{O}_1$ and $\mathcal{O}_2$ as the cycles induced by the vertex sets $\{1,2,3\}$ and $\{1,4,5\}$, respectively. Now, the reduced SDP and SOCP relaxations can be obtained by replacing the constraint $X = X^H \succeq 0$ in the above optimization with certain small-sized constraints based on $\mathcal{O}_1$ and $\mathcal{O}_2$, as mentioned before. In light of Theorem 11, the following statements hold:

- The SDP, reduced SDP and SOCP relaxations are all exact in the case when $c_{12}, c_{13}, c_{14}, c_{15}, c_{23}, c_{45}$ are real numbers satisfying the inequalities $c_{12}c_{13}c_{23} \leq 0$ and $c_{14}c_{15}c_{45} \leq 0$.
- The SDP and reduced SDP are exact in the case when $c_{12}, c_{13}, c_{14}, c_{15}, c_{23}, c_{45}$ are imaginary numbers (note that the SOCP relaxation may not be tight).
- The SDP, reduced SDP and SOCP relaxations are all exact in the case when each of the sets $\{c_{12}, c_{13}, c_{23}\}$ and $\{c_{14}, c_{15}, c_{45}\}$ has at least one zero element.

The above results demonstrate how the combined effect of the graph topology and the edge weights make various relaxations become exact for the quadratic optimization (6.44).

Example 3: Consider the optimization

$$\min_{\mathbf{x}\in\mathcal{C}^n} \quad \mathbf{x}^H M\mathbf{x} \qquad \text{s.t.} \quad |x_j| = 1, \quad j = 1,2,...,m \tag{6.45}$$

where M is a symmetric real-valued matrix. It has been proven in [119] that this problem is NP-hard even in the case when M is restricted to be positive semidefinite. Consider the graph $\mathcal{G}$ associated with the matrix M. As an application of Theorem 8, the SDP and reduced SDP relaxations are exact for this optimization and therefore this problem is polynomial-time solvable, provided that $\mathcal{G}$ is bipartite and weakly cyclic. To understand how well the SDP relaxation works, we pick $\mathcal{G}$ as a cycle with 4 vertices. Consider a randomly generated matrix M:

$$M = \begin{bmatrix} 0 & -0.0961 & 0 & -0.1245 \\ -0.0961 & 0 & -0.1370 & 0 \\ 0 & -0.1370 & 0 & 0.7650 \\ -0.1245 & 0 & 0.7650 & 0 \end{bmatrix}.$$

After solving the SDP relaxation numerically, an optimal solution X^* is obtained as

$$X^* = \begin{bmatrix} 1.0000 & 0.1767 & -0.5516 & 0.6505 \\ 0.1767 & 1.0000 & 0.7235 & -0.6327 \\ -0.5516 & 0.7235 & 1.0000 & -0.9923 \\ 0.6505 & -0.6327 & -0.9923 & 1.0000 \end{bmatrix}.$$

This matrix has rank-2 and thus it seems as if the SDP relaxation is not exact. However, the fact is that this relaxation has a hidden rank-1 solution. To recover that solution, one can write X^* as the sum of two rank-1 matrices, i.e., $X^* = (\mathbf{u}_1)(\mathbf{u}_1)^H + (\mathbf{u}_2)(\mathbf{u}_2)^H$ for two real vectors $\mathbf{u}_1$ and $\mathbf{u}_1$. It is straightforward to inspect that the complex-valued rank-1 matrix $(\mathbf{u}_1 + \mathbf{u}_2\mathrm{i})(\mathbf{u}_1 + \mathbf{u}_2\mathrm{i})^H$ is another solution of the SDP relaxation. Thus, $\mathbf{x}^* = \mathbf{u}_1 + \mathbf{u}_2\mathrm{i}$ is an optimal solution of Optimization (6.45).

Example 4: Consider the optimization

$$\min_{\mathbf{x}\in\mathcal{C}^n} \quad \mathbf{x}^H M_0 \mathbf{x} \qquad \text{s.t.} \quad \mathbf{x}^H M_j \mathbf{x} \leq 0, \quad j = 1, 2, ..., m$$

where $M_0, ..., M_m$ are symmetric real matrices, while $\mathbf{x}$ is an unknown complex vector. Similar to what was done in Example 1, a generalized weighted graph $\mathcal{G}$ can be constructed for this optimization. Regardless of the edge weights, as long as the graph $\mathcal{G}$ is acyclic, the SDP, reduced SDP and SOCP relaxations are all tight (see Theorem 6). As a result, this class of optimization problems is polynomial-time solvable.

6.8 Summary

This work deals with three conic relaxations for a broad class of nonlinear real/complex optimization problems, where the argument of each objective and constraint function is quadratic (as opposed to linear) in the optimization variable. Several types of optimizations, including polynomial optimization, can be cast as the problem under study. To explore the exactness of the proposed relaxations, the structure of the optimization is mapped into a generalized weighted graph with a weight set assigned to each edge. In the case of real-valued optimization, it is shown that the relaxations are exact if a set of conditions are satisfied, which depend on some weak properties of the underlying generalized weighted

graph. A similar result is derived in the complex-valued case after introducing the notion of "sign-definite complex weight sets", under the assumption that the graph is acyclic. The complex case is further studied for general graphs, and it is shown that if the graph can be decomposed as the union of edge-disjoint subgraphs, each satisfying one of the four derived structural properties, then two of the relaxations are exact. As an application, it is finally shown that the weight sets are sign definite for power networks due to the passivity of transmission lines, and this makes a broad class of energy optimizations easy to solve.

6.9 Appendix

In what follows, Theorem 10 will be proved.

Proof of Part (i): Let $\mathbf{x}^*$ denote an arbitrary solution of (6.1). For every $\mathcal{G}_s \in \Omega$, define $\alpha(\mathcal{G}_s)$ as:

- If $\mathcal{G}_s \in \Omega \backslash (\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_p)$, then we set $\alpha(\mathcal{G}_s)$ equal to any arbitrary complex number with norm 1.
- If $(\mathcal{G}_s) = \mathcal{O}_r$ for some $r \in \{1, ..., p\}$, then we set $\alpha(\mathcal{G}_s)$ equal to $e^{-(\measuredangle x^*_{\mu_r} \mathrm{i})}$.

For every $i \in \mathcal{G}$, define q_i^o as $|x_i^*|^2$. In addition, for every $\mathcal{G}_s \in \Omega$ and $(i,j) \in \mathcal{G}_s$, define:

$$\begin{aligned} &U_{ii}^o(\mathcal{G}_s) = \mathrm{Re}\{x_i^* \alpha(\mathcal{G}_s)\}^2, \quad U_{jj}^o(\mathcal{G}_s) = \mathrm{Re}\{x_j^* \alpha(\mathcal{G}_s)\}^2, \\ &W_{ii}^o(\mathcal{G}_s) = \mathrm{Im}\{x_i^* \alpha(\mathcal{G}_s)\}^2, \quad W_{jj}^o(\mathcal{G}_s) = \mathrm{Im}\{x_j^* \alpha(\mathcal{G}_s)\}^2, \\ &V_{ij}^o(\mathcal{G}_s) = \mathrm{Re}\{x_i^* \alpha(\mathcal{G}_s)\}\mathrm{Im}\{x_j^* \alpha(\mathcal{G}_s)\}, \quad V_{ji}^o(\mathcal{G}_s) = \mathrm{Re}\{x_j^* \alpha(\mathcal{G}_s)\}\mathrm{Im}\{x_i^* \alpha(\mathcal{G}_s)\}. \end{aligned} \tag{6.46}$$

Consider those entries of $U^o(\mathcal{G}_s), V^o(\mathcal{G}_s), W^o(\mathcal{G}_s)$ that are not specified above as arbitrary. The first goal is to show that $(\mathbf{q}, U, V, W) = (\mathbf{q}^o, U^o, V^o, W^o)$ is a feasible solution of the expanded SOCP problem. To this end, it is straightforward to verify that (6.31d) and (6.31e) are satisfied. Moreover, for every $\mathcal{G}_s \in \Omega$ and $(i,j) \in \mathcal{G}_s$, one can write:

$$q_i^o = |x_i^*|^2 = |x_i^* \alpha(\mathcal{G}_s)|^2 = U_{ii}^o(\mathcal{G}_s) + W_{ii}^o(\mathcal{G}_s). \tag{6.47}$$

Besides,

$$W_{\mu_r \mu_r}^o(\mathcal{O}_r) = \mathrm{Im}\{x_{\mu_r}^* \alpha(\mathcal{O}_r)\}^2 = \mathrm{Im}\{x_{\mu_r}^* e^{-(\measuredangle x^*_{\mu_r} \mathrm{i})}\}^2 = 0$$

for every $r \in \{1,2,...,p\}$. Hence, $(\mathbf{q}^o, U^o, V^o, W^o)$ satisfies (6.31c)-(6.31f). On the other hand, for every $(i,j) \in \mathcal{G}$, there is a unique subgraph $\mathcal{G}_s \in \Omega$ such that $(i,j) \in \mathcal{G}_s$ (because $\mathcal{G}$ is weakly cyclic by assumption). Now, since the edge weights are imaginary numbers, one can write:

$$\begin{aligned} \mathrm{Re}\{c_{ij}^t (x_i^*)(x_j^*)^H)\} &= -\mathrm{Im}\{c_{ij}^t\} \times \mathrm{Im}\left\{(x_i^* \alpha(\mathcal{G}_s))(x_j^* \alpha(\mathcal{G}_s))^H)\right\} \\ &= \mathrm{Im}\{c_{ij}^t\}(V_{ij}^o(\mathcal{G}_s) - V_{ji}^o(\mathcal{G}_s)) \end{aligned} \tag{6.48}$$

for every $t \in \{1,...,k\}$. It follows from (6.47) and (6.48) that

$$\mathbf{q}^o = l_1\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right), \quad \bar{l}(V^o) = l_2\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right). \tag{6.49}$$

Therefore,

$$0 \geq f_j\left(l_1\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right), l_2\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right)\right) = f_j(\mathbf{q}^o, \bar{l}(V^o)), \qquad j = 1,2,...,m$$

This means that $(\mathbf{q}, U, V, W) = (\mathbf{q}^o, U^o, V^o, W^o)$ is a feasible solution of the expanded SOCP problem. Similarly,

$$f* = f_0\left(l_1\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right), l_2\left((\mathbf{x}^*)(\mathbf{x}^*)^H\right)\right) = f_0(\mathbf{q}^o, \bar{l}(V^o)) \geq f^*_{\text{e-SOCP}}.$$

Proof of Part (ii): Given an arbitrary solution $\mathbf{x}^*$ of Optimization (6.1), consider $(\mathbf{q}^o, U^o, V^o, W^o)$ defined in (6.46). As shown in the proof of Part (i), this is a feasible solution of the expanded SOCP relaxation. Furthermore, observe that

$$\begin{bmatrix} U_{ii}^o(\mathcal{G}_s) & V_{ij}^o(\mathcal{G}_s) \\ V_{ij}^o(\mathcal{G}_s) & W_{jj}^o(\mathcal{G}_s) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\{x_i^*\alpha(\mathcal{G}_s)\} \\ \mathrm{Im}\{x_j^*\alpha(\mathcal{G}_s)\} \end{bmatrix} \begin{bmatrix} \mathrm{Re}\{x_i^*\alpha(\mathcal{G}_s)\} & \mathrm{Im}\{x_j^*\alpha(\mathcal{G}_s)\} \end{bmatrix},$$
$$\begin{bmatrix} U_{jj}^o(\mathcal{G}_s) & V_{ji}^o(\mathcal{G}_s) \\ V_{ji}^o(\mathcal{G}_s) & W_{ii}^o(\mathcal{G}_s) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}\{x_j^*\alpha(\mathcal{G}_s)\} \\ \mathrm{Im}\{x_i^*\alpha(\mathcal{G}_s)\} \end{bmatrix} \begin{bmatrix} \mathrm{Re}\{x_j^*\alpha(\mathcal{G}_s)\} & \mathrm{Im}\{x_i^*\alpha(\mathcal{G}_s)\} \end{bmatrix}.$$

This implies that the above matrices have rank 1, which completes the proof of the "only if" part. To prove the "if" part, let $(\mathbf{q}^*, U^*, V^*W^*)$ be a solution of the expanded SOCP relaxation satisfying the rank condition stated in Part (ii). Therefore, for every $\mathcal{G}_s \in \Omega$ and $(i,j) \in \mathcal{G}_s$, one can decompose the 2×2 matrices in (6.31d) and (6.31e) at the point

$(\mathbf{q}, U, V, W) = (\mathbf{q}^*, U^*, V^*W^*)$ as

$$\begin{bmatrix} U^*_{ii}(\mathcal{G}_s) & V^*_{ij}(\mathcal{G}_s) \\ V^*_{ij}(\mathcal{G}_s) & W^*_{jj}(\mathcal{G}_s) \end{bmatrix} = \begin{bmatrix} u^*_i(\mathcal{G}_s) \\ w^*_j(\mathcal{G}_s) \end{bmatrix} \begin{bmatrix} u^*_i(\mathcal{G}_s) \\ w^*_j(\mathcal{G}_s) \end{bmatrix}^H,$$
$$\begin{bmatrix} U^*_{jj}(\mathcal{G}_s) & V^*_{ji}(\mathcal{G}_s) \\ V^*_{ji}(\mathcal{G}_s) & W^*_{ii}(\mathcal{G}_s) \end{bmatrix} = \begin{bmatrix} u^*_j(\mathcal{G}_s) \\ w^*_i(\mathcal{G}_s) \end{bmatrix} \begin{bmatrix} u^*_j(\mathcal{G}_s) \\ w^*_i(\mathcal{G}_s) \end{bmatrix}^H$$

for some real numbers $u^*_i(\mathcal{G}_s), u^*_j(\mathcal{G}_s), w^*_i(\mathcal{G}_s), w^*_j(\mathcal{G}_s)$. Following the proof of Part (i) and by making a comparison with (6.50), it suffices to show the existence of a vector $\mathbf{x}^*$ and a complex set $\{\sigma(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega\}$ satisfying the relations:

$$u^*_i(\mathcal{G}_s) + w^*_i(\mathcal{G}_s)\mathrm{i} = x^*_i \alpha(\mathcal{G}_s), \quad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s, \tag{6.51a}$$
$$|\sigma(\mathcal{G}_s)| = 1, \qquad \forall \mathcal{G}_s \in \Omega \backslash (\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_p), \tag{6.51b}$$
$$\sigma(\mathcal{O}_r) = e^{-(\measuredangle x^*_{\mu_r}\mathrm{i})}, \qquad \forall r \in \{1, ..., p\}. \tag{6.51c}$$

It can be verified that

$$q^*_i = |u^*_i(\mathcal{G}_s) + w^*_i(\mathcal{G}_s)\mathrm{i}|^2, \quad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s.$$

Hence, the equations in (6.51) consistently find $|x^*_i|$ as $|x^*_i|^2 = \sqrt{q^*_i}$ for every $i \in \mathcal{G}$. Now, it remains to find the phase of x^*_i. To this end, (6.51) can be equivalently expressed as:

- If $\mathcal{G}_s = \mathcal{O}_r$ for some $r \in \{1, 2, ..., p\}$, then

$$\measuredangle x^*_i - \measuredangle x^*_{\mu_r} = \tan^{-1} \frac{w^*_i(\mathcal{O}_r)}{u^*_i(\mathcal{O}_r)}. \tag{6.52}$$

- If $\mathcal{G}_s \in \Omega \backslash (\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_p)$, then

$$\measuredangle x^*_i + \measuredangle \sigma(\mathcal{G}_s) = \tan^{-1} \frac{w^*_i(\mathcal{G}_s)}{u^*_i(\mathcal{G}_s)}. \tag{6.53}$$

Note that if the index i in (6.52) is chosen as μ_r, then the left side of this equation becomes zero. Equation (6.31f) guarantees that the right side of (6.52) is also zero in this case. The goal is to show that (6.52) and (6.53) have a solution $\{\measuredangle x^*_1, ..., \measuredangle x^*_n\}$. For this purpose, we order the subgraphs in the set Ω in such a way that every two consecutive subgraphs in

the ordered set share a vertex. Denote the ordered set as $\{\mathcal{G}_1, ..., \mathcal{G}_{|\Omega|}\}$. Since the graph $\mathcal{G}$ is weakly cyclic, $\mathcal{G}_1 \cup \cdots \cup \mathcal{G}_s$ and the subgraph $\mathcal{G}_{s+1}$ share exactly one vertex for every $r \in \{1, 2, ..., |\Omega| - 1\}$. Hence, the following algorithm can be used to find $\{\measuredangle x_1^*, ..., \measuredangle x_n^*\}$:

Step 1: Set $s = 1$ and $\measuredangle x_i^* = 0$ for an arbitrary vertex i of the subgraph $\mathcal{G}_1$.

Step 2: So far, the elements of $\mathbf{x}$ corresponding to all vertices of $\mathcal{G}_1 \cup \cdots \cup \mathcal{G}_{s-1}$ and only one vertex of $\mathcal{G}_s$ have been found. Let j denote the index of the only vertex of $\mathcal{G}_s$ for which x_j^* has been obtained. Now, depending on whether or not $\mathcal{G}_s$ belongs to $\Omega \backslash (\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_p)$, (6.52) or (6.53) can be uniquely solved to find all entries of $\mathbf{x}^*$ corresponding to the vertices of $\mathcal{G}_s$.

Step 3: Increment s unless $s = |\Omega|$.

Proof of Part (iii): Given an arbitrary feasible point $(\mathbf{q}, U, V, W)$ of the expanded SOCP relaxation, consider the entries of X in the SOCP relaxation (6.5) as:

- For every $i \in \mathcal{G}$, set X_{ii} equal to q_i.
- For every $(i, j) \in \mathcal{G}$, find the unique subgraph $\mathcal{G}_s \in \Omega$ such that $(i, j) \in \mathcal{G}_s$, and set $X_{ij} = X_{ji}^H = V_{ji}(\mathcal{G}_s) - V_{ij}(\mathcal{G}_s)$.
- Choose the remaining entries of X arbitrarily.

By adopting the argument leading to (6.49), it can be shown that

$$f_j(l_1(X), l_2(X)) = f_j(\mathbf{q}, \bar{l}(V)), \qquad j = 0, 1, ..., m. \tag{6.54}$$

Thus, it only remains to prove that the defined X is a feasible point of the SOCP relaxation (6.5). Given an edge $(i, j) \in \mathcal{G}$, let $\mathcal{G}_s \in \Omega$ be the subgraph containing this edge. One can write:

$$X\{(i,j)\} = \begin{bmatrix} U_{ii}(\mathcal{G}_s) & -V_{ij}(\mathcal{G}_s)\mathrm{i} \\ V_{ij}(\mathcal{G}_s)\mathrm{i} & W_{jj}(\mathcal{G}_s) \end{bmatrix} + \begin{bmatrix} W_{ii}(\mathcal{G}_s) & V_{ji}(\mathcal{G}_s)\mathrm{i} \\ -V_{ji}(\mathcal{G}_s)\mathrm{i} & U_{jj}(\mathcal{G}_s). \end{bmatrix}$$

Since $X\{(i, j)\}$ has been expressed as the sum of two positive semidefinite matrices, it must be a positive semidefinite matrix. This implies that X is a feasible point of the SOCP relaxation.

Proof of Part (iv): Let X denote an arbitrary feasible point of the reduced SDP relaxation. Given a subgraph $\mathcal{G}_s \in \Omega$, the matrix $X\{\mathcal{G}_s\}$ can be decomposed as $D\{\mathcal{G}_s\}D\{\mathcal{G}_s\}^H$, where $D\{\mathcal{G}_s\}$ is a matrix in $\mathcal{C}^{|\mathcal{G}_s|\times|\mathcal{G}_s|}$ whose last row is entirely real valued. Such a decomposition can be obtained using the eigen-decomposition method. Now, consider the matrix variable $\begin{bmatrix} U(\mathcal{G}_s) & V(\mathcal{G}_s) \\ V(\mathcal{G}_s)^H & W(\mathcal{G}_s) \end{bmatrix}$ in the expanded SOCP relaxation as

$$\begin{bmatrix} \mathrm{Re}\{D(\mathcal{G}_s)\}\mathrm{Re}\{D(\mathcal{G}_s)\}^H & \mathrm{Re}\{D(\mathcal{G}_s)\}\mathrm{Im}\{D(\mathcal{G}_s)\}^H \\ \mathrm{Im}\{D(\mathcal{G}_s)\}\mathrm{Re}\{D(\mathcal{G}_s)\}^H & \mathrm{Im}\{D(\mathcal{G}_s)\}\mathrm{Im}\{D(\mathcal{G}_s)\}^H \end{bmatrix}.$$

Moreover, consider q_i as X_{ii} for every $i \in \mathcal{G}$. It is straightforward to show that (6.54) holds for this choice of $(\mathbf{q}, U, V, W)$, and that $(\mathbf{q}, U, V, W)$ is a feasible point of the expanded SOCP relaxation. This completes the proof.

Proof of Part (v): Consider the optimization

$$\min_{\mathbf{u},\mathbf{w}} \quad f_0(\mathbf{q}, \bar{l}(V)) \tag{6.55a}$$

subject to:

$$f_j(\mathbf{q}, \bar{l}(V)) \leq 0, \qquad j = 1, ..., m, \tag{6.55b}$$

$$U_{ii}(\mathcal{G}_s) + W_{ii}(\mathcal{G}_s) = q_i, \quad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s, \tag{6.55c}$$

$$U_{ii} = u_i(\mathcal{G}_s)^2, \qquad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s, \tag{6.55d}$$

$$W_{ii} = w_i(\mathcal{G}_s)^2, \qquad \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s, \tag{6.55e}$$

$$V_{ij} = u_i(\mathcal{G}_s)w_j(\mathcal{G}_s), \qquad \forall \mathcal{G}_s \in \Omega,\ (i,j) \in \mathcal{G}_s, \tag{6.55f}$$

$$V_{ji} = u_j(\mathcal{G}_s)w_i(\mathcal{G}_s), \qquad \forall \mathcal{G}_s \in \Omega,\ (i,j) \in \mathcal{G}_s, \tag{6.55g}$$

$$W_{\mu_r\mu_r}(\mathcal{O}_r) = 0, \qquad r = 1, 2, ..., p, \tag{6.55h}$$

where $\mathbf{u} = \{u_i(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s\}$ and $\mathbf{w} = \{w_i(\mathcal{G}_s) \mid \forall \mathcal{G}_s \in \Omega,\ i \in \mathcal{G}_s\}$. Note that U, V, W are considered as implicit (dependent) variables in Optimization (6.55), because they can be readily expressed in terms of $\mathbf{u}$ and $\mathbf{w}$. Optimization (6.55) is real-valued and can be cast in the form of (6.1). Therefore, one can find its SDP, reduced SDP and SOCP relaxations. It is easy to verify that the SOCP relaxation for this optimization is indeed the expanded SOCP relaxation (6.31). Assume that this relaxation is tight for (6.31).

Then, it follows from Theorem 1 that the expanded SOCP relaxation has a solution for which the matrices in (6.31d) and (6.31e) have rank 1. In this case, the proof of Part (v) is an immediate consequence of Parts (ii)-(iv). Therefore, it suffices to show that the relaxation (6.31) is tight for Optimization (6.55). To this end, according to Corollary 1, it is enough to show that the graph capturing the structure of Optimization (6.55) is acyclic. To construct this graph, notice that not every quadratic term in the matrix $\begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix}^H$ appears in the constraints of Optimization (6.55). The ones creating an edge in the graph of this optimization are given by the set $\big\{u_i(\mathcal{G}_s)w_j(\mathcal{G}_s), u_j(\mathcal{G}_s)w_i(\mathcal{G}_s) \;\big|\; \forall \mathcal{G}_s \in \Omega,\ (i,j) \in \mathcal{G}_s\big\}$. This graph is cyclic. However, since $w_{\mu_r}(\mathcal{O}_r)$ is equal to zero for $r = 1, ..., p$, all vertices associated with $w_{\mu_r}(\mathcal{O}_r)$'s can be removed from the graph. Now, the remaining graph becomes acyclic (given that $\mathcal{G}$ is weakly cyclic). This completes the proof. ■

Chapter 7

Conclusions and Future Work

This dissertation is concerned with the analysis and synthesis of complex networks using tools and techniques from the broad area of "control and optimization". The main results of this dissertation are presented in five chapters. Three chapters are dedicated to the following real-world networks: (i) the human brain networks, (ii) communication networks, (iii) electric power networks. The last two chapters aim to further study power networks, but in a general framework that can be applied to a broad set of complex networks. The problems investigated in these two chapters are: (i) flow optimization over a flow network, (ii) nonlinear optimization over a generalized weighted graph. In what follows, the main results obtained for each of the aforementioned problems will be summarized and possible future research directions will be discussed accordingly.

Brain Networks: Two popular methods for assessing the brain functional/effective connectivity are: (1) mapping the thresholded correlation matrix into a graph, which shows the marginal independence/dependence relationships among random variables, (2) mapping the inverse covariance matrix into a graph, which shows the conditional dependencies of Gaussian random variables. The latter method is based on Bayesian networks and sparse regression. An important question arises as to which of these methods provides better information about the structure (topology) of the brain networks. Due to the electrical properties of the brain, this problem is investigated in the context of electrical circuits. An electric circuit model is considered, for which it is shown that the inverse covariance matrix of the node voltages reveals the topology of the circuit. Having made this observation, another question arises as to how to find the topology of the circuit based on noisy measurements taken from node voltages. In this problem, the aim is to find the topology of the circuit when a limited number of samples are available. For this purpose, the graphical lasso technique

is used to estimate a sparse inverse covariance matrix. It is shown that the graphical lasso may find most of the circuit topology if the exact covariance matrix is well-conditioned, and may fail to work well when this matrix is ill-conditioned. To deal with ill-conditioned matrices, a small modification to the graphical lasso algorithm is proposed, which improves the recovery of the circuit topology significantly. Finally, the technique developed in this work is applied to the resting-state fMRI data of a number of healthy subjects, and useful observations are made accordingly. Some problems left for future research are as follows:

- This work assumes that the brain connectivity network is static. As a more realistic problem, it is important to study the dynamic models of the brain networks.
- Hidden (unobserved) variables play an important role in modeling the brain networks. Understanding the role of such variables in mathematical modeling of the brain networks is an important problem, which needs to be incorporated in the framework proposed in this work.

Communication Networks: Congestion control algorithms aim to allocate resources to demands in a communication network in such a way that the total utilization of the network is optimized. Despite the progress in the analysis and synthesis of the Internet congestion control, almost all existing fluid models of congestion control assume that every link in the path of a flow observes the original source rate. To address this issue, a more accurate model is derived for the behavior of the network under an arbitrary congestion controller, which takes the effect of buffering on data flows into account. By investigating this model, it is proved that the well-known Internet congestion control algorithms may no longer be stable for the common pricing schemes, unless a sufficient condition is satisfied. It is also shown that these algorithms are guaranteed to be stable if a new pricing mechanism is used. The theories developed in this work are for a fluid model of the network with no stochastic sources. The removal of these two assumptions in the proposed modeling technique is considered as future research.

Electrical Power Networks: Optimal power flow (OPF) is concerned with finding an optimal operating point of a power network minimizing the total power generation cost subject to network and physical constraints. It has been recently shown that several practical instances of OPF problem can be solved in polynomial time as long as the objective function is quadratic. The current work first generalizes this result to arbitrary convex

functions and then studies how the deployment of power electronic devices guarantees solvability of OPF in polynomial time. To this end, a convex relaxation is proposed, which solves the OPF problem exactly for every radial network and every meshed network with a sufficient number of phase shifters, provided power over-delivery is allowed. The concept of "power over-delivery" is equivalent to relaxing the power balance equations to inequality constraints. If a power network has a very limited number of phase shifters, then the present work suggests adding a sufficient number of virtual phase shifters to the network in order to find an approximate solution to the OPF problem. Studying the optimality degree of the obtained solution is left as future work. Another future research direction is to investigate how the effect of the virtual phase shifters killing the non-convexity of OPF can be penalized in the objective function.

Flow Networks: The generalized network flow (GNF) problem aims to optimize the flows over an arbitrary lossy flow network. The GNF problem is hard to solve due to the presence of nonlinear equality flow constraints. Under the assumption of monotonicity and convexity of the flow and cost functions, a convex relaxation is proposed, which always finds the optimal nodal injections. As an application in OPF, it can be concluded from this work that the relaxation of power balance equations (i.e., load over-delivery) is not needed in practice under a very mild angle assumption.

Although the convex relaxation proposed in this work is guaranteed to find the optimal nodal injections (and hence the optimal objective value), it may produce wrong flows. The reason why the convex relaxation does not necessarily find the correct flows is that the mapping from flows to injections is not invertible. Obtaining a set of flows associated with the optimal injections is considered as future work.

Generalized Weighted Graphs: Motivated by power optimizations, this work aims to find a global optimization technique for a nonlinear optimization defined over a generalized weighted graph. Every edge of this type of graph is associated with a weight set corresponding to the known parameters of the optimization (e.g., the coefficients). The motivation behind this problem is to investigate how the (hidden) structure of a given real/complex-valued optimization makes the problem easy to solve, and indeed the generalized weighted graph is introduced to capture the structure of an optimization. Various sufficient conditions are derived, which relate the polynomial-time solvability of different classes of optimization problems to weak properties of the generalized weighted graph such

as its topology and the sign definiteness of its weight sets. As an application, it is proved that a broad class of real and complex optimizations over power networks are polynomial-time solvable due to the passivity of transmission lines and transformers. Although a broad class of network topologies has been explored here, this work does not provide any concrete result for mesh networks with general complex variables. The study of this type of graph is left as future work.

Bibliography

[1] S. Ryalia, T. Chena, K. Supekara, and V. Menon, "Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty," *NeuroImage*, vol. 59, pp. 3852–3861, 2012.

[2] K. Friston, A. Holmes, K. Worsley, and J. Poline, "Statistical parametric maps in functional imaging: A general linear approach," *Human Brain Mapping*, vol. 2, pp. 189–210, 1995.

[3] J. Cao and K. Worsley, "The geometry of correlation fields, with an application to functional connectivity of the brain," *Ann. Appl. Probab.*, vol. 9, pp. 1021–1057, 1999.

[4] C. Goutte, P. Toft, E. Rostrup, F. Nielsen, and L. K. Hansen, "On clustering fMRI time series," *NeuroImage*, vol. 9, pp. 298–310, 1999.

[5] P. Filzmoser, R. Baumgartner, and E. A. Moser, "Hierarchical clustering method for analyzing functional MR images," *Magn Reson Imaging*, vol. 17, p. 81726, 1999.

[6] A. Baune, F. T. Sommer, M. Erb, D. Wildgruber, B. Kardatzki, G. Palm, and W. Grodd, "Dynamical cluster analysis of cortical fMRI activation," *NeuroImage*, vol. 9, pp. 477–89, 1999.

[7] L. Harrison, W. Penny, and K. Friston, "Multivariate autoregressive modeling of fMRI time series," *NeuroImage*, vol. 19, pp. 1477–1491, 2003.

[8] P. Valdes-Sosa, J. Sanchez-Bornot, A. Lage-Castellanos, M. Vega-Hernandez, J. Bosch-Bayard, L. Melie-Garcia, and E. Canales-Rodriguez, "Estimating brain functional connectivity with sparse multivariate autoregression," *Phil. Trans. Roy. Soc. B*, vol. 360, pp. 969–981, 2003.

[9] K. J. Friston, L. Harrison, and W. Penny, "Dynamic causal modelling," *NeuroImage*, vol. 19, pp. 1273–1302, 2003.

[10] A. Marreiros, S. Kiebel, and K. Friston, "Dynamic causal modelling for fMRI: A two-state model," *NeuroImage*, vol. 39, pp. 269–278, 2008.

[11] U. Noppeney, C. J. Price, W. Penny, and K. J. Friston, "Two distinct neural mechanisms for category-selective responses," *Cereb. Cortex*, vol. 16, pp. 437–445, 2006.

[12] K. Stephan, W. Penny, J. C. Marshall, G. R. Fink, and K. J. Friston, "Investigating the functional role of callosal connections with dynamic causal models," *Ann. N. Y. Acad. Sci.*, vol. 1064, pp. 16–36, 2005.

[13] X. Zheng and J. C. Rajapakse, "Learning functional structure from fMR images," *NeuroImage*, vol. 31, pp. 1601–1613, 2006.

[14] J. C. Rajapakse and J. Zhou, "Learning effective brain connectivity with dynamic Bayesian networks," *NeuroImage*, vol. 37, pp. 749–760, 2007.

[15] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[16] F. P. Kelly, A. Maullo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[17] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 274–285, 2004.

[18] J. Lavaei, J. C. Doyle, and S. H. Low, "Utility functionals associated with available congestion control algorithms," *IEEE International Conference on Computer Communications*, 2010.

[19] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," *Internetworking Research and Experience*, vol. 1, pp. 3–26, 1990.

[20] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, 1993.

[21] J. Carpentier, "Contribution to the economic dispatch problem," *Bulletin Societe Francaise Electriciens*, 1962.

[22] R. A. Jabr, "Radial distribution load flow using conic programming," *IEEE Transactions on Power Systems*, vol. 21, pp. 1458–1459, 2006.

[23] R. A. Jabr, "Optimal power flow using an extended conic quadratic formulation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1000–1008, 2008.

[24] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *International Journal of Electric Power & Energy Systems*, vol. 30, no. 6-7, pp. 383–392, 2008.

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

[26] W. S. Jewell, "Optimal flow through networks with gains," *Operations Research*, vol. 10, pp. 476–499, 1962.

[27] H. Brannlund, J. A. Bubenko, D. Sjelvgren, and N. Andersson, "Optimal short term operation planning of a large hydrothermal power system based on a nonlinear network flow concept," *IEEE Transactions on Power Systems*, vol. 1, pp. 75–81, 1986.

[28] J. L. Goffin, J. Gondzio, R. Sarkissian, and J. P. Vial, "Solving nonlinear multicommodity flow problems by the analytic center cutting plane method," *Mathematical Programming*, vol. 76, pp. 131–154, 1996.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, 2004.

[30] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," To appear in *Foundations and Trends in Optimization*, 2013, http://www.stanford.edu/~boyd/papers/pdf/msg_pass_dyn.pdf.

[31] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory," *Studies in Applied Mathematics, SIAM*, 1994.

[32] M. Fazel, H. Hindi, , and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," *American Control Conference*, vol. 3, pp. 2156–2162, 2003.

[33] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, pp. 471–501, 2010.

[34] A. Barvinok, "Problems of distance geometry and convex properties of quadartic maps," *Discrete and Computational Geometry*, vol. 12, pp. 189–202, 1995.

[35] G. Pataki, "On the rank of extreme matrices in semidenite programs and the multiplicity of optimal eigenvalues," *Mathematics of Operations Research*, vol. 23, pp. 339–358, 1998.

[36] J. Sturm and S. Zhang, "On cones of nonnegative quadratic functions," *Mathematics of Operations Research*, vol. 28, pp. 246–267, 2003.

[37] Y. Huang and S. Zhang, "Complex matrix decomposition and quadratic programming," *Mathematics of Operations Research*, vol. 32, pp. 758–768, 2007.

[38] W. Ai, Y. Huang, and S. Zhang, "On the low rank solutions for linear matrix inequalities," *Mathematics of Operations Research*, vol. 33, pp. 965–975, 2008.

[39] J. Poline and B. Brett, "The general linear model and fMRI: Does love last forever?" *NeuroImage*, vol. 62, pp. 871–880, 2012.

[40] R. B. Buxton, E. C. Wong, and L. R. Frank, "Dynamics of blood flow and oxygenation changes during brain activation: the Balloon model," *MRM*, vol. 39, pp. 855–864, 1998.

[41] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, pp. 267–288, 1996.

[42] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistic*, vol. 9, pp. 432–441, 2008.

[43] N. Meinshausen, "A note on the lasso for Gaussian graphical model selection," *Stat. Probab. Lett.*, vol. 78, pp. 880–884, 2008.

[44] P. Vertes, A. F. Alexander-Bloch, N. Gogtay, J. N. Giedd, J. L. Rapoport, and E. T. Bullmore, "Simple models of human brain functional networks," *Proceedings of the National Academy of Sciences*, vol. 109, pp. 5868–5873, 2012, (the fMRI data is available online at http://intramural.nimh.nih.gov/chp/articles/matlab.htm).

[45] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, 1993.

[46] V. Jacobson and M. J. Karels, "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, pp. 314–329, 1988.

[47] V. Jacobson, "Berkeley TCP evolution from 4.3-tahoe to 4.3-reno," *In Proceedings of the Eighteenth Internet Engineering Task Force*, 1990.

[48] S. Floyd and T. Henderson, "The new reno modification to TCPs fast recovery algorithm," *RFC 2582*, 1999.

[49] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," *RFC 2018, available at http://www.icir.org/floyd/sacks.html*, 1996.

[50] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," *in Proceedings of IEEE*, vol. 95, pp. 255–312, 2007.

[51] R. Srikant, "The mathematics of internet congestion control," *Birkhauser*, 2004.

[52] S. Shakkottai and R. Srikant, "Network optimization and control," *Foundations and Trends in Networking*, vol. 2, pp. 271–379, 2008.

[53] L. S. Brakmo and L. Peterson, "TCP vegas: End to end congestion avoidance on a global internet," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1465–1480, 1995.

[54] S.Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review*, vol. 24, pp. 10–23, 1994.

[55] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, "Fluid models and solutions for large-scale ip networks," *ACM/SIGMETRICS*, 2003.

[56] V. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *ACM/SIGCOMM*, 2000.

[57] H. Han, C. Hollot, Y.Chait, and V. Misra, "Stability of buffer-based aqms: The effect of routing and departure-rate models," *Sixteenth International Symposium on Mathematical Theory of Networks and Systems*, 2004.

[58] H. Han, "Modelling and analysis of TCP network dynamics," *Electronic Doctoral Dissertations for UMass Amherst. Paper AAI3254913. http://scholarworks.umass.edu/dissertations/AAI3254913.*

[59] I. Lestas and G. Vinnicombe, "How good are deterministic models for analyzing congestion control in delayed stochastic networks?" *IEEE Conference on Decision and Control*, 2004.

[60] S. Sojoudi, S. H. Low, and J. C. Doyle, "Effect of buffers on stability of internet congestion controllers," *IEEE International Conference on Computer Communications*, pp. 471–475, 2011.

[61] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 556–567, 2000.

[62] S. H. Low, L. Peterson, and L. Wange, "Understanding vegas: a duality model," *J. of ACM*, vol. 49, pp. 207–235, 2002.

[63] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," *IEEE Transactions on Automatic Control*, vol. 47, pp. 945–959, 2002.

[64] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle, "Linear stability of TCP/RED and a scalable control," *Computer Networks Journal*, vol. 43, pp. 633–647, 2003.

[65] J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. Part I: Nonlinear and quadratic programming approaches," *IEEE Transactions on Power Systems*, 1999.

[66] J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. Part II: Newton, linear programming and interior point methods," *IEEE Transactions on Power Systems*, 1999.

[67] I. A. Hiskens and R. J. Davy, "Exploring the power flow solution space boundary," *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 389–395, 2001.

[68] M. Huneault and F. Galiana, "A survey of the optimal power flow literature," *IEEE Transactions on Power Systems*, vol. 6, pp. 762–770, 1991.

[69] H. Wang, C. E. Murillo-Sanchez, R. D. Zimmerman, and R. J. Thomas, "On computational issues of market-based optimal power flow," *IEEE Transactions on Power Systems*, vol. 22, pp. 1185–1193, 2007.

[70] K. S. Pandya and S. K. Joshi, "A survey of optimal power flow methods," *Journal of Theoretical and Applied Information Technology*, 2008.

[71] R. A. Jabr, A. H. Coonick, and B. J. Cory, "A primal-dual interior point method for optimal power flow dispatching," *IEEE Transactions on Power Systems*, vol. 17, pp. 654–662, 2002.

[72] H. Wei, H. Sasaki, J. Kubokawa, and R. Yokoyama, "An interior point nonlinear programming for optimal power flow problems with a novel data structure," *IEEE Transactions on Power Systems*, vol. 13, pp. 870–877, 1998.

[73] J. Lavaei, "Zero duality gap for classical OPF problem convexifies fundamental nonlinear power problems," *American Control Conference*, 2011.

[74] J. Lavaei and S. H. Low, "Relationship between power loss and network topology in power systems," *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

[75] R. Baldick, *Applied Optimization: Formulation and Algorithms for Engineering Systems*. Cambridge, 2006.

[76] S. Ghosh, D. A. Iancu, D. Katz-Rogozhnikov, D. T. Phan, and M. S. Squillante, "Power generation management under time-varying power and demand conditions," *IEEE Power & Energy Society General Meeting*, 2011.

[77] H. Zhu and G. B. Giannakis, "Estimating the state of ac power systems using semidefinite programming," *North American Power Symposium*, 2011.

[78] D. Gayme and U. Topcu, "Optimal power flow with distributed energy storage dynamics," *Proceedings of the 2011 American Control Conference*, 2011.

[79] A. Taylor and F. S. Hover, "Conic relaxations for transmission system planning," *North American Power Symposium*, 2011.

[80] S. Sojoudi and S. H. Low, "Optimal charging of plug-in hybrid electric vehicles in smart grids," *IEEE Power & Energy Society General Meeting*, 2011.

[81] R. Anders, "Distributed control using positive quadratic programming," *30th Chinese Control Conference*, 2011.

[82] D. Phan and S. Ghos, "A two-stage nonlinear program for optimal electrical grid power balance under uncertainty," *Proceedings of the 2011 Winter Simulation Conference*, 2011.

[83] B. Z. A. Lam and D. Tse, "Distributed algorithms for optimal power flow problem," *http://arxiv.org/abs/1109.5229*, 2011.

[84] B. Zhang and D. Tse, "Geometry of feasible injection region of power networks," *http://arxiv.org/abs/1107.1467*, 2011.

[85] S. Bose, D. F. Gayme, S. Low, and M. K. Chandy, "Optimal power flow over tree networks," *Proceedings of the Forth-Ninth Annual Allerton Conference*, 2011.

[86] M. Farivar, C. R. Clarke, S. H. Low, and K. M. Chandy, "Inverter VAR control for distribution systems with renewables," *International Conference on Smart Grid Communications*, 2011.

[87] B. Lesieutre, D. Molzahn, A. Borden, and C. L. DeMarco, "Examining the limits of the application of semidefinite programming to power flow problems," *49th Annual Allerton Conference*, 2011.

[88] University of Washington, "Power systems test case archive," http://www.ee.washington.edu/research/pstca.

[89] A. V. Goldberg, E. Tardos, and R. E. Tarjan, "Network flow algorithms," *Flows, Paths and VLSI (Springer, Berlin)*, pp. 101–164, 1990.

[90] L. R. Ford and D. R. Fulkerson, "Flows in networks," *Princeton University Press*, 1962.

[91] M. Klein, "A primal method for minimal cost flows with applications to the assignment and transportation problems," *Management Science*, vol. 14, pp. 205–220, 1967.

[92] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network flows: theory, algorithms, and applications," *Prentice-Hall*, 1993.

[93] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Mathematical Programming*, vol. 98, pp. 49–71, 2003.

[94] D. Bertsimas and S. Stock-Paterson, "The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach," *Transportation Science*, vol. 34, pp. 239–255, 2000.

[95] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, "Linear programming and network flows," *John Wiley & Sons*, 1990.

[96] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM*, vol. 19, pp. 248–264, 1972.

[97] K. E. Nygard, P. R. Chandler, and M. Pachter, "Dynamic network flow optimization models for air vehicle resource allocation," *American Control Conference*, 2001.

[98] D. Goldfarb and J. Hao, "Polynomial-time primal simplex algorithms for the minimum cost network flow problem," *Algorithmica*, vol. 8, pp. 145–160, 1992.

[99] D. Bienstock, S. Chopra, O. Gunluk, and C. Y. Tsai, "Minimum cost capacity installation for multicommodity network flows," *Mathematical Programming*, vol. 81, pp. 177–199, 1998.

[100] A. Araposthatis, S. Sastry, and P. Varaiya, "Analysis of power-flow equation," *International Journal of Electrical Power & Energy Systems*, vol. 3, pp. 115–126, 1981.

[101] S. Sojoudi and J. Lavaei, "Physics of power networks makes hard optimization problems easy to solve," *IEEE Power & Energy Society General Meeting*, 2012.

[102] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Transactions on Power Apparatus and Systems*, 1968.

[103] T. J. Overbye, X. Cheng, and Y. Sun, "A comparison of the AC and DC power flow models for LMP calculations," in *Proceedings of the 37th Hawaii International Conference on System Sciences*, 2004.

[104] Y. V. Makarov, Z. Y. Dong, and D. J. Hill, "On convexity of power flow feasibility boundary," *IEEE Transactions on Power Systems*, 2008.

[105] B. Zhang and D. Tse, "Geometry of injection regions of power networks," To appear in *IEEE Transactions on Power Systems*, 2012.

[106] J. Lavaei, B. Zhang, and D. Tse, "Geometry of power flows in tree networks," *IEEE Power & Energy Society General Meeting*, 2012.

[107] J. Lavaei and S. H. Low, "Convexification of optimal power flow problem," *48th Annual Allerton Conference*, 2010.

[108] A. Y. S. Lam, B. Zhang, A. Dominguez-Garcia, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," 2012, Submitted for publication.

[109] Y. Weng, Q. Li, R. Negi, and M. Ilic, "Semidefinite programming for power system state estimation," *IEEE Power & Energy Society General Meeting*, 2012.

[110] D. K. Molzahn, B. C. Lesieutre, and C. L. DeMarco, "A sufcient condition for power flow insolvability with applications to voltage stability margins," *http://arxiv.org/pdf/1204.6285.pdf*, 2012.

[111] J. Lavaei and S. Sojoudi, "Competitive equilibria in electricity markets with nonlinearities," *American Control Conference*, 2012.

[112] E. D. Klerk, "The complexity of optimizing over a simplex, hypercube or sphere: a short survey," *Central Eur J. Oper. Res.*, vol. 16, pp. 111–125, 2008.

[113] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisability problems using semidefinite programming," *J. ACM*, vol. 42, pp. 1115–1145, 1995.

[114] M. Goemans and D. Williamson, "Approximation algorithms for max-3-cut and other problems via complex semidefinite programming," *Journal of Computer and System Sciences*, vol. 68, pp. 422–470, 2004.

[115] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optim. Methods Softw.*, vol. 9, pp. 141–160, 1998.

[116] Y. Ye, "Approximating quadratic programming with bound and quadratic constraints," *Math. Prog.*, vol. 84, pp. 219–226, 1999.

[117] Y. Ye, "Approximating global quadratic optimization with convex quadratic constraints," *J. Glob. Optim.*, vol. 15, pp. 1–17, 1999.

[118] S. Zhang, "Quadratic maximization and semidefinite relaxation," *Math. Prog. A*, vol. 87, pp. 453–465, 2000.

[119] S. Zhang and Y. Huang, "Complex quadratic optimization and semidefinite programming," *SIAM J. Optim.*, vol. 87, pp. 871–890, 2006.

[120] Z. Luo, N. Sidiropoulos, P. Tseng, and S. Zhang, "Approximation bounds for quadratic optimization with homogeneous quadratic constraints," *SIAM J. Optim.*, vol. 18, pp. 1–28, 2007.

[121] S. He, Z. Luo, J. Nie, and S. Zhang, "Semidefinite relaxation bounds for indefinite homogeneous quadratic optimization," *SIAM J. Optim.*, vol. 19, pp. 503–523, 2008.

[122] S. He, Z. Li, and S. Zhang, "Approximation algorithms for homogeneous polynomial optimization with quadratic constraints," *Math. Program.*, vol. 125, pp. 353–383, 2010.

[123] S. Sojoudi and J. Lavaei, "Physics of power networks makes hard optimization problems easy to solve," *IEEE Power & Energy Society General Meeting*, 2012.

[124] S. Kim and M. Kojima, "Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations," *Computational Optimization and Applications*, vol. 26, pp. 143–154, 2003.

[125] S. Bose, D. F. Gayme, S. H. Low, and K. M. Chandy, "Quadratically constrained quadratic programs on acyclic graphs with application to power flow," *arXiv:1203.5599v1*, 2012.

[126] R. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," *Linear Algebra and Its Applications*, vol. 58, pp. 109–124, 1984.