Optimization, Learning, and Control for Energy Networks

Manish K Singh

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Vasileios Kekatos, Chair

Chen-Ching Liu

Virgilio A Centeno

Ryan K Williams

Serkan Gugercin

June 07, 2021

Blacksburg, Virginia

Keywords: Network flow, optimal power flow, convex relaxation, mixed-integer programming, deep learning, distributed control, dynamic stability

Optimization, Learning, and Control for Energy Networks

Manish K Singh

(ABSTRACT)

Massive infrastructure networks such as electric power, natural gas, or water systems play a pivotal role in everyday human lives. Development and operation of these networks is extremely capital-intensive. Moreover, security and reliability of these networks is critical. This work identifies and addresses a diverse class of computationally challenging and time-critical problems pertaining to these networks. This dissertation extends the state of the art on three fronts. *First*, general proofs of uniqueness for network flow problems are presented, thus addressing open problems. Efficient network flow solvers based on energy function minimizations, convex relaxations, and mixed-integer programming are proposed with performance guarantees. *Second*, a novel approach is developed for sample-efficient training of deep neural networks (DNN) aimed at solving optimal network dispatch problems. The novel feature here is that the DNNs are trained to match not only the minimizers, but also their sensitivities with respect to the optimization problem parameters. *Third*, control mechanisms are designed that ensure resilient and stable network operation. These novel solutions are bolstered by mathematical guarantees and extensive simulations on benchmark power, water, and natural gas networks.

Optimization, Learning, and Control for Energy Networks

Manish K Singh

(GENERAL AUDIENCE ABSTRACT)

Massive infrastructure networks play a pivotal role in everyday human lives. A minor service disruption occurring locally in electric power, natural gas, or water networks is considered a significant loss. Uncertain demands, equipment failures, regulatory stipulations, and most importantly complicated physical laws render managing these networks an arduous task. Oftentimes, the first principle mathematical models for these networks are well known. Nevertheless, the computations needed in real-time to make spontaneous decisions frequently surpass the available resources. Explicitly identifying such problems, this dissertation extends the state of the art on three fronts: *First*, efficient models enabling the operators to tractably solve some routinely encountered problems are developed using fundamental and diverse mathematical tools; *Second*, quickly trainable machine learning based solutions are developed that enable spontaneous decision making while learning offline from sophisticated mathematical programs; and *Third*, control mechanisms are designed that ensure a safe and autonomous network operation without human intervention. These novel solutions are bolstered by mathematical guarantees and extensive simulations on benchmark power, water, and natural gas networks.

Dedication

To Guru, Ghar, Desh, and Dost!

Acknowledgments

I can not be content with any acknowledgement I put together for thanking my mentor, Dr. Vassilis Kekatos. I take refuge in the words of Kabir Das,

Sab dharti kagaj karu, lekhni sab banray,

Saat samunder ki masi karu, gurugun likha na jaaye!

These lines roughly translate to: "Even if the whole earth is transformed into paper with all the big trees made into pens, and if the entire water in the seven oceans is transformed into writing ink, even then the glories of the Guru cannot be written. So much is the greatness of the Guru." Apart from his obvious role in the technical works presented in this dissertation, I am indebted to Dr. Kekatos for imparting profound research acumen, ethics, inclusiveness, and tender yet strong mentoring skills.

I next wish to thank my PhD committee members for guiding me through the work and motivating me with several possible research directions to proceed with. Apart from this thesis, the confidence that I have gained in power systems comes from the valuable courses and extended discussions with Dr. Chen-Ching Liu, Dr. Lamine Mili and Dr. Virgilio Centeno. I am extremely thankful to them for their support and contributions. I am also thankful for the personal and academic support from the close-knit group of students and faculty at the power and energy center (PEC), and continuous financial support from the national science foundation (NSF). I am particularly thankful for to NSF grants 1751085 and 1711587 for supporting me. During the summer of 2020, I did an internship with the National Renewable Energy Loboratory, where I was fortunate to work closely with Dr. Andrey Bernstein and Dr. Guido Cavraro, while learning from Dr. Ahmed Zamzam and other group members. What started as a remote internship amidst a pandemic has continued as a jubilant collaboration which I am very thankful for.

While grad-school is tough, it is surmountable merely because of the grad-friends. The numerous (non-)meaningful discussions and laughter at *419*, the lesser known Faraday's lab and Cellar, with friends from the PEC and the "*Broce Drive People*" form adorable memories. I thank you all for being a supportive part of good and less good times of my journey so far. I'll be there for you! Finally, I cannot be more thankful for my family that taught me "all for one, and one for all". All that I could do is for you, and because of you. Thank you god, for giving me the people I've got. You are the best!

The ideation of most works presented in this dissertation happened in the company of either Dr. Kekatos, or my friend Aarushi. I treasure them, an impeccable guide and an amazing co-traveler, with gratitude.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Energy, quite literally, is driving the sustenance and advancements of human civilizations. To fulfill the energy needs of intercontinental societies in an economical, secure, and efficient manner, massive infrastructure networks have gradually emerged. The complexity of these networks is continually increasing due to growing geographical expanse, proliferation of sensors and actuators, emerging uncertainties, transforming demand-supply patterns and unforeseen inter-dependencies. Such challenges have motivated researchers from diverse disciplines to make a concerted effort to keep the *lights on*, and the *pipes clear*.

The goal of this dissertation is to expand the understanding and enrich the toolboxes for management of electric power, natural gas, and water networks. By drawing fundamental mathematical ideas from optimization, control, and machine learning literature, this dissertation provides answers to some classical questions on network flow problems; provides novel network control perspectives; and develops physics-aware deep learning based network dispatch solutions. The separate works collated in this dissertation share key underlying ideas and mathematical tools. Moreover, the schemes developed in several of these works apply to more than one infrastructure network. Nevertheless, the dissertation adopts a *manuscript-style* presentation to emphasize and benefit from their individual completeness. An overview of the dissertation is provided next.

1.1 Dissertation Outline

This dissertation can be broadly partitioned into three sections. The *first* section comprising of Chapter 2-4 studies gas and water flow problems in detail, using tools from convex optimization, combinatorics, and mixed-integer programming. The *second* section, consisting of Chapter 5-6, proposes a novel approach towards training of deep neural networks for the task of learning to optimize. Finally, the *third* section, formed by Chapter 7-8, draws ideas from control theory to develop schemes that enhance resiliency and stability of energy networks. Chapter 2-5, and 7 are reproduced from published works [113], [114], [110], [111], and [109], respectively. Chapter 6 has been submitted for publication [112], and is currently under review; while Chapter 8 is under preparation for submission. An overview of the specific problems solved in the subsequent chapters is provided next.

1.1.1 Chapter 2-4

Solving the equations relating gas(water) injections to pressures and pipeline flows lies at the heart of natural gas (water) network operation, yet existing solvers require careful initialization and uniqueness has been an open question. Chapter 2-3 consider the nonlinear steady-state version of the gas flow (GF) problem. It is first established that the solution to the GF problem is unique under arbitrary network topologies, compressor types, and sets of specifications; then a suite of GF solvers with complementary capabilities is developed. The physical laws governing natural gas and water networks portray fascinating similarities. However, the respective pressure control devices, i.e. the gas compressors and water pumps behave differently. The differences indeed render the proof of uniqueness for GF solutions, and the accompanying solvers inapplicable to water networks. Thus, while drawing insights from the GF problem, Chapter 4 employs a fresh approach and succeeds at establishing the uniqueness of a water flow solution. Chapter 4 also develops efficient water flow solvers that apply to diverse network topologies.

1.1.2 Chapter 5-6

Smart inverters have been advocated as a fast-responding voltage control solution in distribution grids with renewables. Nevertheless, finding the optimal inverter setpoints for reactive power in near real time can be computationally and communication-wise taxing. A classical variant of the aforementioned problem is the task of dispatching generators in bulk power systems for varying load conditions. These problems are formulated with varying complexities, depending on usage of nonlinear AC power flow equations, or their linearized versions. Selecting the inverter and generator dispatch problems subject to (non)linear power flow model, Chapters 5-6 put forth deep neural network (DNN)-based approaches. Specifically, a learning-to-optimize scheme is developed, where a DNN is used by the electric utility to predict the optimal power flow (OPF) minimizers once presented with the new grid loading conditions. The novel feature here is that this DNN is trained to match not only the minimizers, but also their sensitivities with respect to the OPF problem parameters. This is a unique feature for the learning-to-optimize setup yielding a dramatic improvement on data efficiency. The findings were also presented at [117].

1.1.3 Chapter 7-8

Chapter 7 and 8 draw classical concepts from control theory to develop novel approaches that increase the resiliency and stability of networked systems. Chapter 7 proposes a control scheme that brings together the benefits of distributed, local, and event-triggered control to enable networked systems in autonomously tackling unforeseen disruptions, subject to resource sufficiency. Numerical tests for the proposed scheme are provided for both water and power networks. An extension of this work, not included in this dissertation, is provided in [28]. Chapter 8 identifies that the small-signal stability of a system can be strongly influenced by its operating point. Thus, a computationally tractable formulation is developed that dispatches generators in power systems such that the resulting operating point relates to maximal dynamic stability.

Chapter 2

Natural Gas Flow Equations: Uniqueness and an MI-SOCP Solver

2.1 Publication Details

M. K. Singh and V. Kekatos, "Natural gas flow equations: Uniqueness and an MI-SOCP solver," in *Proc. IEEE American Control Conf.*, Philadelphia, PA, Jul. 2019.

2.2 Abstract

The critical role of gas fired-plants to compensate renewable generation has increased the operational variability in natural gas networks (GN). Towards developing more reliable and efficient computational tools for GN monitoring, control, and planning, this work considers the task of solving the nonlinear equations governing steady-state flows and pressures in GNs. It is first shown that if the gas flow equations are feasible, they enjoy a unique solution. To the best of our knowledge, this is the first result proving uniqueness of the steady-state gas flow solution over the entire feasible domain of gas injections. To find this solution, we put forth a mixed-integer second-order cone program (MI-SOCP)-based solver relying

on a relaxation of the gas flow equations. This relaxation is provably exact under specific network topologies. Unlike existing alternatives, the devised solver does not need proper initialization or knowing the gas flow directions beforehand, and can handle gas networks with compressors. Numerical tests on tree and meshed networks indicate that the relaxation is exact even when the derived conditions are not met.

2.3 Introduction

Natural gas has been a critical energy source for decades with uses across the residential, commercial, industrial, and electric generation sectors [100]. The importance of natural gas in the energy sector has further increased, and the same is anticipated in future. The main reasons for increasing emphasis on natural gas include the discovery of substantial new supplies of natural gas in the U.S., and its recognition as a clean low-carbon solution to meet rising energy demands [1]. The thrust for renewable energy in power sector demands for technical solutions to handle the high variability, intermittency, and uncertainty involved with wind and solar generations. Natural gas has come up as an economically viable and low-carbon solution to the said problem because of high ramping abilities of natural gas-fired generators [1].

The primary transportation mode for natural gas is through a large continent-wide network of pipelines [100]. Along a pipeline, pressure drops in the direction of flow due to friction. Gas contracts necessitate the operators to maintain a minimum pressure at consumer nodes. Therefore, compressors are placed on some pipelines to increase the pressure at the output. Given gas injection/withdraws, operators need to solve the *gas flow* (GF) equations, a set of nonlinear equations governing the distribution of gas flows and nodal pressures [92]. The increasing variability in gas withdrawals by gas-fired generators, the complex interdependence of gas-electric infrastructure, and increased focus on reliability, they all motivate well efficient GF solvers [83]. Solving the GF problem is hard for non-tree networks even under-steady state and balanced conditions [45].

The GF problem is typically solved using the Newton-Raphson (NR) scheme, though its convergence is conditioned on proper initialization [83]. A semidefinite program (SDP)-based GF solver, attaining a higher success probability than the NR scheme, is developed in [92]. Nevertheless, the SDP-based solver fails to solve the GF problem if the network state is far from the states considered in designing the solver. The necessity of good initialization for the NR scheme and suitable design points for SDP-based solver limit their suitability for reliability studies. For simpler networks without compressors, the flows and pressures are the optimal primal-dual solutions of a convex minimization [41]. Broadening the scope to tree networks, a GF solver based on the monotone operator theory has been developed in [45]. The latter applies to meshed GNs presuming that the directions of gas flows are given. Reference [45] establishes also the uniqueness of a GF solution but only within a monotonicity domain; still this domain is hard to characterize for non-tree networks or networks with compressors. Reference [131] establishes the uniqueness of a GF solution when the compressors are modeled as *additive* with a constant pressure increase.

The contribution of this work is on three fronts: First, Section 2.5 proves that the nonlinear steady-state GF equations enjoy a unique solution, if a solution exists. To the best of our knowledge, this is the first claim corroborating the numerical observations of [45]. Second, Section 2.6.1 puts forth an MI-SOCP-based GF solver. The GF task is posed as a minimization problem where flow directions are captured by binary variables; the nonlinear GF equations are relaxed to second-order cone constraints; and a judiciously designed function is appended in the objective. Thanks to the latter, the third contribution is to show that the relaxation is exact if there are no compressors in cycles and every pipe belongs to at most one cycle. Numerical tests on tree and meshed networks demonstrate that the devised solver finds the unique GF solution even when the assumed conditions are not met.

Notation: lower- (upper-) case boldface letters denote column vectors (matrices). Calligraphic symbols are reserved for sets. Symbol $^\top$ stands for transposition, and $\odot$ denotes entry-wise multiplication between vectors. Vectors $\mathbf{0}$ and $\mathbf{1}$ are the all-zero and all-one vectors. The sign function $\text{sign}(x)$ returns $+1$ if $x > 0$; -1 if $x < 0$; and 0 otherwise.

2.4 Gas Flow Problem

Consider a natural gas network (GN) modeled by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{P})$. The graph vertices $\mathcal{N} = \{1, \cdots, N\}$ model nodes where gas is injected or withdrawn from the network, or simple junctions. The graph edges $\mathcal{P} = \{1, \ldots, P\}$ correspond to gas pipelines connecting two nodes. Let $p_n > 0$ be the gas pressure at node n for all $n \in \mathcal{N}$. One of the nodes (conventionally one hosting a large gas producer) is selected as the reference node. The reference node is indexed by r, and its pressure is fixed to a known value p_r. The gas injection q_n at node $n \in \mathcal{N}$ is positive for an injection node; negative for a withdrawal node; and zero for junction nodes.

Without loss of generality, edges are assigned an arbitrary direction denoted by $\ell = (m, n) \in \mathcal{P}$ with m and $n \in \mathcal{N}$. The gas flow ϕ_ℓ on pipeline $\ell = (m, n) \in \mathcal{P}$ is positive when gas flows from node m to node n, and negative, otherwise. Conservation of mass implies that for all $n \in \mathcal{N}$

$$q_n = \sum_{\ell:(n,k)\in\mathcal{P}} \phi_\ell - \sum_{\ell:(k,n)\in\mathcal{P}} \phi_\ell. \tag{2.1}$$

To express (2.1) in matrix-vector form, collect all gas injections in $\mathbf{q} := [q_1 \ldots q_N]^\top$ and edge flows in $\boldsymbol{\phi} := [\phi_1 \ldots \phi_P]^\top$.The connectivity of the GN graph is captured by the $P \times N$

edge-node incidence matrix $\mathbf{A}$ with entries

$$A_{\ell,k} := \begin{cases} +1 & , \ k = m \\ -1 & , \ k = n \\ 0 & , \ \text{otherwise} \end{cases} \qquad \forall \ \ell = (m,n) \in \mathcal{P}. \tag{2.2}$$

The mass conservation in (2.1) may thus be expressed as

$$\mathbf{A}^\top \boldsymbol{\phi} = \mathbf{q}. \tag{2.3}$$

Observe that $\mathbf{A}\mathbf{1} = \mathbf{0}$ by definition, and so $\mathbf{1}^\top \mathbf{q} = 0$. The latter is intuitive since the network should be balanced under steady-state conditions. This also implies that (2.1) provides $N-1$ rather than N independent linear equations on $\boldsymbol{\phi}$.

For high- and medium-pressure networks, the pressure drop and energy loss across a pipeline are captured by a set of partial differential equations evolving across time and spatially along the pipeline length [93], [124]. Ignoring friction, pipeline tilt, and assuming time-invariant gas injections, this set of partial differential equations simplifies to the so termed Weymouth equation [136]

$$p_m^2 - p_n^2 = a_\ell \phi_\ell |\phi_\ell| \tag{2.4}$$

describing the pressure difference across the endpoints of pipeline $\ell = (m,n) \in \mathcal{P}$. The parameter $a_\ell > 0$ depends on the physical properties of the pipeline [93]. The Weymouth equation asserts that pressure drops within a pipeline in the direction of gas flow. To be precise, the difference of squared pressures is proportional to the squared gas flow. To simplify notation, define the squared pressure at node $n \in \mathcal{N}$ as $\psi_n := p_n^2$. Then, equation

(2.4) can be written as

$$\psi_m - \psi_n = a_\ell \operatorname{sign}(\phi_\ell)\phi_\ell^2 \quad , \ \forall \ell = (m,n) \in \mathcal{P} \tag{2.5a}$$

$$\psi_n \geq 0 \quad , \ \forall n \in \mathcal{N}. \tag{2.5b}$$

By slightly abusing the terminology, we will oftentimes refer to ψ_m as pressure rather than squared pressure, but the distinction will be clear from the context.

To avoid unacceptably low or high pressures, GN operators install compressors at selected pipelines comprising the set $\mathcal{P}_a \subseteq \mathcal{P}$ with $P_a = |\mathcal{P}_a|$. Its complement set $\bar{\mathcal{P}}_a := \mathcal{P} \setminus \mathcal{P}_a$ includes the remaining lossy pipelines satisfying (2.4)–(2.5). A compressor amplifies the squared pressure between its input and output by a ratio α_ℓ. Moreover, a compressor allows a unidirectional flow in the direction of compression. Therefore, compressor $\ell = (m,n) \in \mathcal{P}_a$ can be modeled as

$$\psi_n = \alpha_\ell \psi_m \tag{2.6a}$$

$$\phi_\ell \geq 0. \tag{2.6b}$$

Equation (2.6) assumes an ideal (lossless) compressor, that is $a_\ell = 0$. This is wlog since an actual non-ideal compressor on pipe (m,n) can be modeled by inserting an additional node n' between nodes m and n: Then, the lossless pipe (m,n') hosts an ideal compressor and pipe (n',n) is lossy. Both pipes have identical flows $\phi_{mn'} = \phi_{n'n}$.

Based on (2.5)–(2.6), it is not hard to verify that an NG network can be uniquely described by either the vector of nodal pressures $\boldsymbol{\psi}$, or the vector of edge flows $\boldsymbol{\phi}$.

Lemma 2.1. *Given a reference squared pressure ψ_r for some $r \in \mathcal{N}$, a pair $(\boldsymbol{\phi}, \boldsymbol{\psi})$ satisfying (2.3), (2.5), and (2.6), is uniquely described by either $\boldsymbol{\phi}$ or $\boldsymbol{\psi}$.*

In fact, finding one of these two vectors constitutes the *gas flow* (GF) problem formally described as follows.

Definition 2.2. Given the pressure $\psi_r = p_r^2$ at the reference node; balanced nodal injections $\mathbf{q}$; the compression ratios α_ℓ for all compressors $\ell \in \mathcal{P}_a$; and the friction parameters a_ℓ for all lossy pipes $\ell \in \bar{\mathcal{P}}_a$, the GF problem aims at finding the nodal pressures $\boldsymbol{\psi}$ and pipe flows $\boldsymbol{\phi}$ satisfying the GF equations (2.3), (2.5), and (2.6).

The task involves $N-1+P$ equations over $N-1+P$ unknowns. The GF task can be posed as the feasibility task

$$\begin{aligned} \text{find} \quad & \{\boldsymbol{\phi}, \boldsymbol{\psi}\} \\ \text{s.to} \quad & (2.3), (2.5), (2.6). \end{aligned} \tag{GF1}$$

Albeit (2.3) and (2.6) are linear, the Weymouth equation in (2.5) is piecewise quadratic and non-convex. In addition, the requirement $\{\phi_\ell \geq 0\}_{\ell \in \mathcal{P}_a}$ could further complicate solving the GF equations. The GF task is typically solved using the Newton-Raphson's scheme, which converges only if initialized sufficiently close to a solution [84].

If the GN is a tree ($P = N-1$), then $\mathbf{A}$ is full row-rank and (2.3) is invertible. Once the pipe flows $\boldsymbol{\phi}$ are known, pressures $\boldsymbol{\psi}$ can be found through (2.5)–(2.6). In practice though, gas networks can exhibit non-radial structure [100], [136]. Therefore, solving the GF problem remains non-trivial. Before devising a new GF solver in Section 2.6, the next section establishes that the GF equations enjoy a unique solution.

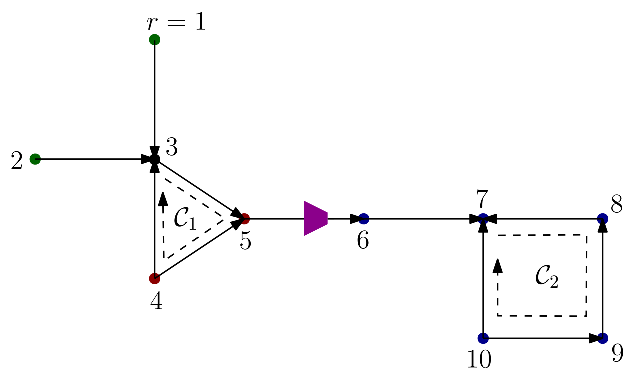

Figure 2.1: Example gas network with 11 edges, 2 cycles, and 1 compressor.

2.5 Uniqueness of the GF Solution

The analysis requires some concepts from graph theory, which are briefly reviewed next. A directed graph $\mathcal{G} = (\mathcal{M}, \mathcal{P})$ is connected if there exists a sequence of adjacent edges between any two of its nodes. All graphs in this work are assumed to be connected. A minimal set of edges $\mathcal{P}_\mathcal{T}$ preserving the connectivity of a graph constitutes a *spanning tree* of $\mathcal{G}$; is denoted by $\mathcal{T} := (\mathcal{M}, \mathcal{P}_\mathcal{T})$; and $|\mathcal{P}_\mathcal{T}| = |\mathcal{M}| - 1$. The edges not belonging to a spanning tree $\mathcal{T}$ are referred to as *links* with respect to $\mathcal{T}$.

A *path* between nodes m and n is defined as a sequence of adjacent edges between the two nodes. If a path starts and ends at the same node (without edge or node repetition), it constitutes a *cycle*. A *tree* is a connected graph with no cycles. The statements 'cycle $\mathcal{C}$ contains node i' or 'node i belongs to cycle $\mathcal{C}$' mean that there exists an edge in $\mathcal{C}$ that is incident to node i. For any cycle $\mathcal{C}$, we select an arbitrary direction and define its indicator vector $\mathbf{n}^\mathcal{C}$ with ℓ-th entry

$$n_\ell^\mathcal{C} := \begin{cases} 0 & \text{, if edge } \ell \notin \mathcal{C} \\ +1 & \text{, if direction of } \ell \text{ agrees with cycle direction} \\ -1 & \text{, otherwise.} \end{cases} \tag{2.7}$$

For example, the network of Fig. 2.1 has 11 edges and 2 cycles. Based on the assigned direction for $\mathcal{C}_2$, the entries of $\mathbf{n}^{\mathcal{C}_2}$ corresponding to edges $\{(7,8),(8,9),(9,10),(10,11)\}$ are $\{-1,-1,-1,+1\}$; the remaining entries are zero. Given a spanning tree, a *fundamental cycle* is a cycle formed by adding a link to the spanning tree.

After the graph preliminaries, we proceed with the uniqueness of the GF solution. The proof relies on the next result for a single-cycle graph, which is proved in the appendix.

Lemma 2.3. *Consider a graph comprising a single cycle $\mathcal{C}$ over $k+1$ nodes indexed by $i = 0, \ldots, k$, and define $\mathbf{n}^{\mathcal{C}} \in \mathbb{R}^k$ as the indicator vector for this cycle. For fixed squared pressure ψ_0, if two flow vectors $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ with $\tilde{\boldsymbol{\phi}} \neq \boldsymbol{\phi}$ satisfy (2.5)–(2.6), they cannot satisfy*

$$\operatorname{sign}(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}) \odot \mathbf{n}^{\mathcal{C}} < \mathbf{0}; \ \textit{or}$$
$$\operatorname{sign}(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}) \odot \mathbf{n}^{\mathcal{C}} > \mathbf{0}$$

where the inequalities are understood entrywise.

The proof of Lemma 2.3 exploits the fact that the pressure differences across a loop should sum up to zero no matter what the flows $\boldsymbol{\phi}$ or $\tilde{\boldsymbol{\phi}}$ are. The usefulness of this lemma is to eliminate scenarios of flow changes. Take for example the case where $\mathbf{n}^{\mathcal{C}} = \mathbf{1}$, and assume two flow vectors $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ satisfying (2.5)–(2.6). We hypothesize the flows in $\tilde{\boldsymbol{\phi}}$ are larger than the flows in $\boldsymbol{\phi}$, that is $\tilde{\boldsymbol{\phi}} > \boldsymbol{\phi}$. Lemma 2.3 ensures that this hypothesis is not valid since $\operatorname{sign}(\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}) \odot \mathbf{n}^{\mathcal{C}} = \mathbf{1} \odot \mathbf{1} = \mathbf{1} > \mathbf{0}$. The hypothesis $\tilde{\boldsymbol{\phi}} < \boldsymbol{\phi}$ can be crossed out likewise.

The proof for the uniqueness of the GF solution builds also on the ensuing result known from network flows. Its proof follows as a special case of [70, Th. 8.8] by setting the source-destination flow to zero.

Lemma 2.4. *Suppose $\mathbf{A}$ is the edge-node incidence matrix of a directed graph. For any* *vector* $\mathbf{n} \in \text{null}(\mathbf{A}^\top)$*, there exists a set of cycles* $\mathcal{S}_\mathcal{C}$ *such that*

$$\mathbf{n} = \sum_{\mathcal{C} \in \mathcal{S}_\mathcal{C}} \lambda_\mathcal{C} \mathbf{n}^\mathcal{C} \tag{2.8}$$

where $\lambda_\mathcal{C} \geq 0$ *for all* $\mathcal{C} \in \mathcal{S}_\mathcal{C}$*, and* $\mathbf{n}^\mathcal{C} \odot \mathbf{n}^{\mathcal{C}'} \geq \mathbf{0}$ *for all* $\mathcal{C}, \mathcal{C}' \in \mathcal{S}_\mathcal{C}$*.*

Lemma 2.4 asserts that any vector in $\text{null}(\mathbf{A}^\top)$ can be expressed as a conic combination of cycle indicator vectors. Moreover, if any pair of these cycles shares an edge, this edge participates to both cycles in the same direction. The following claim follows easily from (2.8).

Corollary 2.5. *If* $\lambda_\mathcal{C} > 0$ *in the representation of* (2.8)*, then* $n_\ell \cdot n_\ell^\mathcal{C} > 0$ *for all* $\ell \in \mathcal{C}$*.*

Using Lemmas 2.3–2.4 and Corollary 2.5, we next prove the uniqueness of the solution to the steady-state GF problem.

Theorem 2.6. *The gas flow problem* (G1) *has a unique solution, if feasible.*

Proof. Proving by contradiction, assume $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ are two distinct flow vectors solving (G1). Since both vectors satisfy (2.3), their difference $\mathbf{n} := \boldsymbol{\phi} - \tilde{\boldsymbol{\phi}}$ must lie in $\text{null}(\mathbf{A}^\top)$. Then, from Lemma 2.4, vector $\mathbf{n}$ can be decomposed as in (2.8).

Select any edge ℓ corresponding to a non-zero entry of $\mathbf{n}$. Since $n_\ell \neq 0$, there exists a cycle $\mathcal{C}$ for which $\lambda_\mathcal{C} \cdot n_\ell^\mathcal{C} \neq 0$. Two cases can be identified:

C1) Cycle $\mathcal{C}$ *contains the reference node* r*.* From Corollary 2.5, it follows $n_\ell \cdot n_\ell^\mathcal{C} = (\phi_\ell - \tilde{\phi}_\ell) \cdot n_\ell^\mathcal{C} > 0$ for all edges $\ell \in \mathcal{C}$. The latter contradicts Lemma 2.3 and proves the claim.

C2) Cycle $\mathcal{C}$ *does not contain the reference node* r*.* Calculate the *distance* of cycle $\mathcal{C}$ from r that we define as

$$d(\mathcal{C}) := \min_{i \in \mathcal{C}} d_{i-r} \tag{2.9}$$

where d_{i-r} counts the edges in the shortest path between nodes i and r. The node in $\mathcal{C}$ attaining distance $d(\mathcal{C})$ will be indexed by $i_{\mathcal{C}}$. If multiple nodes satisfy the latter property, pick one arbitrarily. Consider the shortest path $i_{\mathcal{C}} - r$ between nodes $i_{\mathcal{C}}$ and r. Two cases can be identified again.

C2a) The entries of $\mathbf{n}$ *associated with the edges in path* $i_{\mathcal{C}} - r$ *are all zero.* This implies that the related entries in $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ are equal. Therefore, the nodal pressures along the path $i_{\mathcal{C}} - r$ can be recursively computed using (2.5) and (2.6) starting from ψ_r. Then, the nodal pressures along $i_{\mathcal{C}} - r$ agree between $\boldsymbol{\psi}$ and $\tilde{\boldsymbol{\psi}}$, and so $\psi_{i_{\mathcal{C}}} = \tilde{\psi}_{i_{\mathcal{C}}}$. Since the pressure at node $i_{\mathcal{C}}$ has been fixed, this node can serve as a reference node, the argument under *C1)* applies, and proves the claim.

C2b) There exists an edge ℓ' *in* $i_{\mathcal{C}} - r$ *with a non-zero entry in* $\mathbf{n}$. This edge must belong to a cycle $\mathcal{C}'$. Check whether cycle $\mathcal{C}'$ falls under *C1)* or *C2)*, and apply the previous reasoning recursively.

The previous process considers cycles $\mathcal{C}$ with progressively smaller distances $d(\mathcal{C})$, and so case *C1)* eventually occurs. The recursion terminates upon finding a cycle contradicting Lemma 2.3 under case *C1)*, and completes the proof. $\square$

Remark 2.7. Two compressors can be sometimes connected in parallel [41]. If they are assumed ideal, it is not possible to infer the gas flow on each compressor. Under the practical assumption that the two compressors are non-ideal, this identifiability concern is waived as validated by Theorem 2.6.

Remark 2.8. While the pressure drop along a pipeline is oftentimes described by the Weymouth equation in (2.4), other alternatives include the Panhandle A and B equations [100]. The uniqueness proof of Theorem 2.6 holds in general as it depends on the monotonic dependence of pressure drop on the flows. Moreover, the proof may be generalized for dissipative network flow problems as discussed in [131].

2.6 MI-SOCP Relaxation

2.6.1 Problem Reformulation

The non-convexity in (GF1) is due to the Weymouth equation in (2.5a). Using the big-M trick and upon introducing a binary variable x_ℓ for every lossy pipe $\ell = (m,n) \in \bar{\mathcal{P}}_a$, constraint (2.5a) can be relaxed to convex inequalities as

$$-M(1-x_\ell) \leq \phi_\ell \leq Mx_\ell \tag{2.10a}$$

$$-M(1-x_\ell) \leq \psi_m - \psi_n - a_\ell\phi_\ell^2 \tag{2.10b}$$

$$\psi_m - \psi_n + a_\ell\phi_\ell^2 \leq Mx_\ell \tag{2.10c}$$

$$x_\ell \in \{0,1\} \tag{2.10d}$$

for some large $M > 0$. When $x_\ell = 1$, constraint (2.10a) yields $\phi_\ell \geq 0$, constraint (2.10b) reads $\psi_m - \psi_n - a_\ell\phi_\ell^2 \geq 0$, and (2.10c) is satisfied trivially. When $x_\ell = 0$, constraint (2.10a) yields $\phi_\ell \leq 0$, constraint (2.10b) is satisfied trivially, and (2.10c) becomes $\psi_m - \psi_n + a_\ell\phi_\ell^2 \leq 0$. We therefore get that the binary variable equals $x_\ell = \operatorname{sign}(\phi_\ell)$. It hence determines the direction of flow ϕ_ℓ, and activates (2.10b) or (2.10c).

By replacing (2.5) with (2.10) in (GF1), the GF problem can be posed as an MI-SOCP. However, the minimizer of (GF2) is useful only if it satisfies (2.10b) or (2.10c) (depending on the value of x_ℓ) with equality for all lossy pipes. Then, the relaxation is termed *exact*. Otherwise, the minimizer of (GF2) is infeasible for (GF1).

To promote exact relaxations, we will substitute the GF problem (GF1) by the minimization

$$\begin{aligned} \min \quad & r(\boldsymbol{\psi}) \\ \text{over} \quad & \boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{x} \\ \text{s.to} \quad & (2.3), (2.6), (2.10) \end{aligned} \tag{GF2}$$

where vector $\mathbf{x}$ contains all x_ℓ with $\ell \in \bar{\mathcal{P}}_a$, and

$$r(\boldsymbol{\psi}) := \sum_{(m,n)\in\bar{\mathcal{P}}_a} |\psi_m - \psi_n|.$$

The cost $r(\boldsymbol{\psi})$ sums up the absolute pressure differences across all lossy pipes. A similar MI-SOCP relaxation has been proposed for optimizing flow in water distribution networks in [115]. Although problem (GF2) remains non-convex due to the binary variables $\mathbf{x}$, with advancements in MI-SOCP solvers, this minimization can be handled for moderately sized networks [7], as corroborated by our numerical tests. The next section provides well-defined network conditions under which the relaxation (2.10) in (GF2) is exact.

2.6.2 Exactness of the Relaxation

The relaxation from (3.3) to (2.10) has been proposed earlier in [23], [135], [30]; yet without optimality guarantees. For instance, reference [23] solves gas expansion planning through this relaxation. Upon fixing the binary variables to the values obtained via the relaxation, the continuous variables are then heuristically refined by a local search. The convex relaxation has been combined with a McCormick relaxation in [135]. This section provides network conditions ensuring that the relaxation of (GF2) is provably exact.

Assumption 1. Graph $\mathcal{G}$ has no compressors in cycles.

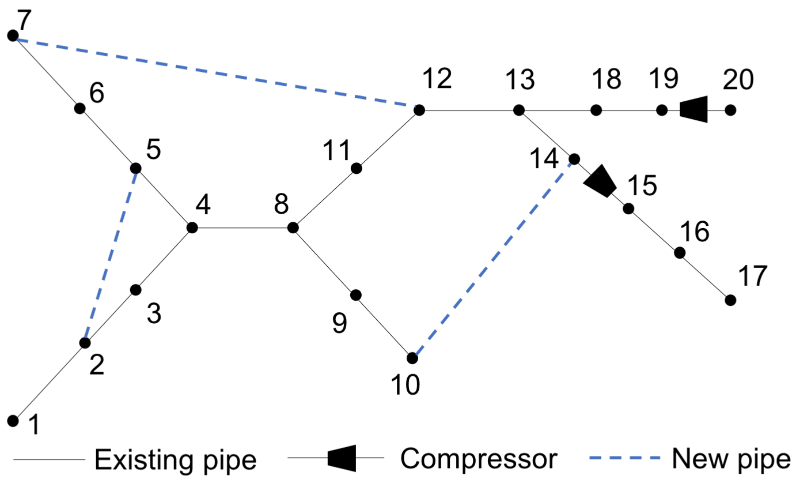

Figure 2.2: Modified Belgian natural gas network.

Assumption 2. Every edge of $\mathcal{G}$ belongs to at most one cycle.

Theorem 2.9. *Under Assumptions 1 and 2, every minimizer of* (GF2) *solves the GF problem* (GF1)*, if the latter is feasible.*

Heed that Assumptions 1 and 2 may not be always satisfied in practical GNs; see e.g., Remark 2.7. Albeit, the tests of Section 2.7 show that (GF2) solves (GF1) even when Assumption 2.9 does not hold. Either way, Theorem 2.9 guarantees that (GF2) can provably handle the GF task for a broader class of GNs than existing alternatives; recall [41] cannot handle compressors, and [45] presumes known flow directions.

2.7 Numerical Tests

Our relaxed GF solver was tested on the Belgian benchmark network shown in Fig. 2.2. Parallel pipes were replaced by their equivalents, and the compression ratios were determined based on the nodal pressures found in [41]. Problem (GF2) was solved using the MATLAB-based optimization toolbox YALMIP along with the mixed-integer solver CPLEX [78], [64],

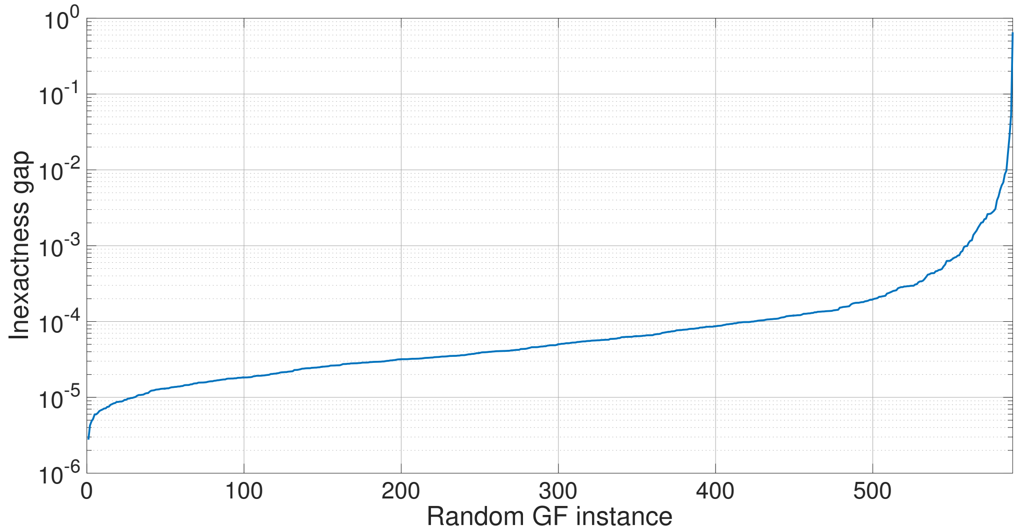

Figure 2.3: Inexactness gap attained by (GF2) over random feasible GF instances.

on a 2.7 GHz Intel Core i5 computer with 8 GB RAM. For the big–M trick, we set $M = 10^4$.

We first tested the (GF2) solver on the original tree Belgian network. The obtained pressure and flow values agreed with those of [41]. We then inserted additional pipelines on the original benchmark to get a non-radial network as shown in Fig. 2.2. Even though Theorem 2.9 requires non-overlapping cycles (Assumption 2), the modified network does not comply with this assumption. To use reasonable friction coefficients, for every new line (m, n), the coefficient a_{mn} was set equal to the sum of a_ℓ's along the $m - n$ path, yielding $a_{2,5} = 0.1936$, $a_{10,14} = 0.0439$, $a_{7,12} = 0.0419$.

The reference pressure at node 1 and the compression ratios were kept constant as in [41]. Note that the benchmark gas injections $\mathbf{q}_o$ lie in the range of $[-15.61, 22.01]$. To test the exactness of the (G2) relaxation under various conditions, we drew $1,000$ random gas injection vectors $\mathbf{q}$'s by adding a zero-mean unit-variance deviation on the entries of $\mathbf{q}_o$. To ensure balanced injections, the injection at node 20 was set to the negative sum of the remaining injections.

Due to the randomness in selecting gas injections, the resulting pipe flows are not guaranteed

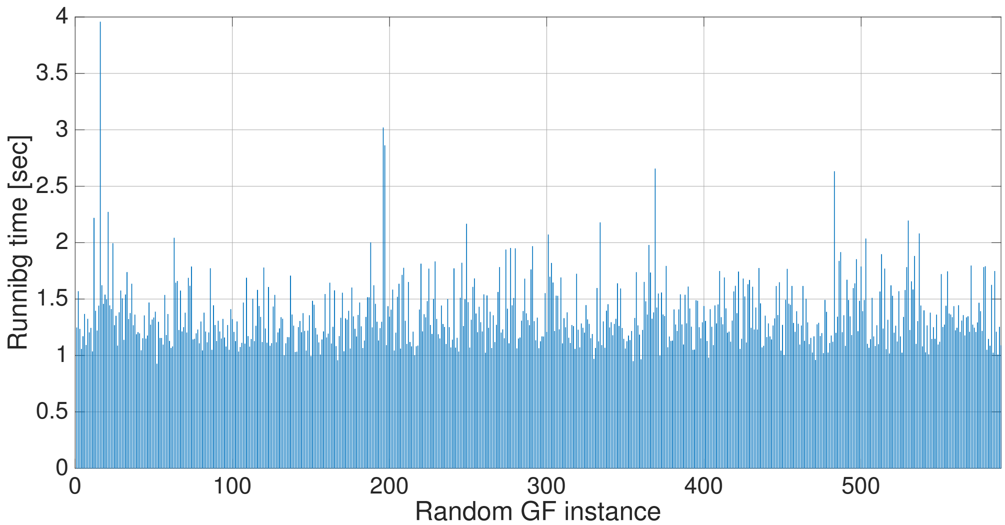

Figure 2.4: Running time for (G2) over random feasible GF instances.

to satisfy (2.6b): The problem was found to be infeasible for 411 out of the 1,000 random injection vectors. Since (GF2) is a relaxation of (GF1), these cases are apparently infeasible for (GF1) too. The performance of (GF2) was henceforth tested on the remaining 589 gas injection instances that were feasible.

To numerically quantify the success of (GF2) in solving (G1), we define the *inexactness gap* for injection vector $\mathbf{q}$ as

$$g(\mathbf{q}) := \max_{(m,n)\in\bar{\mathcal{P}}_a} \frac{|\psi_m - \psi_n| - a_{mn}\phi_{mn}^2}{a_{mn}\phi_{mn}^2} \geq 0$$

for the pressures $\{(\psi_m, \psi_n)\}$ and flows $\{\phi_{mn}\}$ obtained by solving (GF2) for the injection vector $\mathbf{q}$. Fig. 2.3 depicts the ranked inexactness gap along the feasible GF instances. Based on this curve, the gap was less than 10^{-4} for more than 72% of the instances, and less than 10^{-3} for more than 95%. This corroborates that (GF2) performs well even when Assumption 2 is not met. Fig. 2.4 shows the running time for solving (GF2) over the 589 feasible instances. The average (median) running time was 1.39 sec (1.28 sec).

2.8 Conclusions

This work has established the uniqueness of the nonlinear steady-state GF equations, put forth an MI-SOCP-based GF solver, and provided conditions under which this solver succeeds. Numerical tests have shown that the relaxation is exact. The average time of the solver for a 20-node and 22-pipe network is less than 2 seconds; yet its scalability to real-world networks is still to be explored. The success of the relaxation even when the postulated assumptions were not met motivates further research on broadening the conditions. Combining the MI-SOCP solver with existing solvers and adopting it to tackle the GF task under dynamic operation and towards state estimation constitute timely directions.

Appendix A

A.1 Proof of Lemma 2.3

Select the cycle direction $0 \to 1 \cdots \to k \to 0$. Without loss of generality (wlog), suppose the i-th entry of $\boldsymbol{\phi}$ corresponds to the flow in the pipe connecting nodes $(i-1)$ and i. We will prove the first claim by contradiction; the second claim follows similarly.

Suppose the flow vectors $\boldsymbol{\phi}$ and $\tilde{\boldsymbol{\phi}}$ satisfy (2.5)–(2.6) and $\text{sign}(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})\odot\mathbf{n} < 0$. The assumption $\text{sign}(\tilde{\boldsymbol{\phi}}-\boldsymbol{\phi})\odot\mathbf{n} < 0$ apparently implies that

$$\begin{aligned} &\phi_i > \tilde{\phi}_i, \text{ if } n_i = +1, \text{and} \\ &\phi_i < \tilde{\phi}_i, \text{ if } n_i = -1. \end{aligned} \tag{2.11}$$

We will show by induction on i that $\tilde{\psi}_i \geq \psi_i$ for all $i \neq 0$.

Starting with the base step of $i = 1$, the edge between nodes 0 and 1 can be either a lossy pipe or a compressor. If it is a lossy pipe, denote the RHS of (2.5a) by $w(\phi_\ell)$. It is not hard to see that $w(\phi_\ell)$ is monotonically increasing in ϕ_ℓ. Depending on the value of n_1, two cases can be identified:

- If $n_1 = +1$, it follows that

$$\psi_0 - \tilde{\psi}_1 = w(\tilde{\phi}_1) < w(\phi_1) = \psi_0 - \psi_1$$

  where the two equalities stem from (2.5a) and the inequality from (2.11). This implies $\tilde{\psi}_1 > \psi_1$.

- If $n_1 = -1$, we similarly get that

$$\tilde{\psi}_1 - \psi_0 = w(\tilde{\phi}_1) > w(\phi_1) = \psi_1 - \psi_0$$

  which again implies $\tilde{\psi}_1 > \psi_1$.

If the edge between nodes 0 and 1 is a compressor, the linear dependence in (2.6a) yields $\tilde{\psi}_1 = \psi_1 = \alpha_1\psi_0$. Thus, the claim $\tilde{\psi}_1 \geq \psi_1$ holds for all cases.

Proceeding with the induction step, we will assume $\tilde{\psi}_i \geq \psi_i$ and prove that $\tilde{\psi}_{i+1} \geq \psi_{i+1}$. If the edge between node i and $i+1$ is a lossy pipe, the following cases arise:

- If $n_{i+1} = +1$, it holds that $\phi_{i+1} > \tilde{\phi}_{i+1}$ from (2.11) and

$$\begin{aligned} &\tilde{\psi}_i - \tilde{\psi}_{i+1} = w(\tilde{\phi}_{i+1}) < w(\phi_{i+1}) = \psi_i - \psi_{i+1} \\ &\implies \tilde{\psi}_{i+1} > \psi_{i+1} + \tilde{\psi}_i - \psi_i \geq \psi_{i+1}. \end{aligned}$$

- If $n_{i+1} = -1$, it holds that $\phi_{i+1} < \tilde{\phi}_{i+1}$ from (2.11) and

$$\tilde{\psi}_{i+1} - \tilde{\psi}_i = w(\tilde{\phi}_{i+1}) > w(\phi_{i+1}) = \psi_{i+1} - \psi_i$$
$$\implies \tilde{\psi}_{i+1} > \psi_{i+1} + \tilde{\psi}_i - \psi_i \geq \psi_{i+1}.$$

If the edge between node i and $i+1$ is a compressor, we get $\tilde{\psi}_{i+1} = \alpha_{i+1}\tilde{\psi}_i \geq \alpha_{i+1}\psi_i = \psi_{i+1}$. Therefore, the claim $\tilde{\psi}_{i+1} \geq \psi_{i+1}$ holds for all cases.

The claim $\tilde{\psi}_i \geq \psi_i$ holds with equality only if all edges from node 0 to i are compressors. However, this is practically impossible for $i \geq 2$ since every compressor is modeled as an ideal compressor followed by a lossy pipe. Therefore $\tilde{\psi}_i > \psi_i$ for all $i \geq 2$. Applying the latter around the cycle gives $\tilde{\psi}_0 > \psi_0$, which contradicts the hypothesis of fixed ψ_0.

A.2 Proof of Theorem 2.9

Let $(\boldsymbol{\phi}, \boldsymbol{\psi})$ be the unique solution to (GF1), and $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\psi}})$ a minimizer of (GF2). Proving by contradiction, suppose there exists an edge ℓ for which $\tilde{\phi}_\ell \neq \phi_\ell$. Since both flow vectors satisfy (2.3), their difference

$$\mathbf{n} := \tilde{\boldsymbol{\phi}} - \boldsymbol{\phi} \tag{2.12}$$

must lie in null$(\mathbf{A}^\top)$. The nullspace of $\mathbf{A}^\top$ is spanned by the indicator vectors of all fundamental cycles in the network graph [57, Corollary 14.2.3]. Therefore, for all edges not belonging to a cycle, the corresponding entries of $\mathbf{n}$ are zero. Then, the edge ℓ must belong to at least one cycle. In fact, by Assumption 2, edge ℓ belongs to a single cycle, which will be henceforth termed cycle $\mathcal{C}$.

Based on cycle $\mathcal{C}$, the remainder of the proof is organized in three steps. The first step constructs a flow vector $\hat{\boldsymbol{\phi}}$ that satisfies constraints (2.3) and (2.6b). The second step constructs a pressure vector $\hat{\boldsymbol{\psi}}$ so that the pair $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (GF2). The third step shows that $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ attains a smaller objective for (GF2), thus contradicting the optimality of $(\tilde{\boldsymbol{\phi}}, \tilde{\boldsymbol{\psi}})$.

Commencing with the first step, define the flow vector $\hat{\boldsymbol{\phi}}$

$$\hat{\phi}_\ell := \begin{cases} \phi_\ell & , \ \ell \in \mathcal{C} \\ \tilde{\phi}_\ell & , \ \text{otherwise.} \end{cases}$$

The constructed flow vector $\hat{\boldsymbol{\phi}}$ differs from $\tilde{\boldsymbol{\phi}}$ only on cycle $\mathcal{C}$. Due to (2.12), we can write

$$\hat{\boldsymbol{\phi}} = \tilde{\boldsymbol{\phi}} - \lambda \mathbf{n}^{\mathcal{C}} \tag{2.13}$$

for a $\lambda \neq 0$ and where $\mathbf{n}^{\mathcal{C}}$ is the indicator vector of cycle $\mathcal{C}$. Since $\tilde{\boldsymbol{\phi}}$ satisfies constraint (2.3) and $\mathbf{A}^\top \mathbf{n}^{\mathcal{C}} = \mathbf{0}$, then

$$\mathbf{A}^\top \hat{\boldsymbol{\phi}} = \mathbf{A}^\top \tilde{\boldsymbol{\phi}} - \lambda \mathbf{A}^\top \mathbf{n}^{\mathcal{C}} = \mathbf{A}^\top \tilde{\boldsymbol{\phi}} = \mathbf{q}.$$

This proves that $\hat{\boldsymbol{\phi}}$ satisfies constraint (2.3) as well. Granted there are no compressors in cycles (Assumption 1) and since $\tilde{\boldsymbol{\phi}}$ satisfies constraint (2.6b), then $\hat{\phi}_p = \tilde{\phi}_p \geq 0$ for all compressor edges $p \in \mathcal{P}_a$.

We proceed with the second step and construct the pressure vector $\hat{\boldsymbol{\psi}}$. To define its i-th entry, we identify three cases depending on the location of node i relative to cycle $\mathcal{C}$.

a) Node $i \notin \mathcal{C}$, and the shortest path between i and r has no edge in $\mathcal{C}$. Define the modified pressure at node i as

$$\hat{\psi}_i := \tilde{\psi}_i.$$

b) Node $i \in \mathcal{C}$. Identify the node $i_{\mathcal{C}} \in \mathcal{C}$ with the shortest path to the reference node r.

Define the modified pressure at node i as

$$\hat{\psi}_i := \psi_i + (\tilde{\psi}_{i_{\mathcal{C}}} - \psi_{i_{\mathcal{C}}}).$$

The node $i_{\mathcal{C}}$ may be i itself, implying $\hat{\psi}_{i_{\mathcal{C}}} = \tilde{\psi}_{i_{\mathcal{C}}}$.

c) Node $i \notin \mathcal{C}$, but the shortest path between i and r has an edge in $\mathcal{C}$. Identify the node $k \in \mathcal{C}$ that is closest to node i, that is $d_{i-k} = \min_{j \in \mathcal{C}} d_{i-j}$. Note that the nodal pressures from r to k have been defined under cases *a)* and *b)*. Starting from node k and its updated pressure $\hat{\psi}_k$, we next define the constructed pressures along the shortest path from k to i in a sequential fashion. Moving along the $k-i$ path, say the first node is $k+1$ and is connected to k by edge $\ell = (k, k+1)$. Then, we define the pressure at node $k+1$ as

$$\hat{\psi}_{k+1} := \begin{cases} \alpha_\ell \hat{\psi}_k & , \text{ if } \ell \in \mathcal{P}_a \\ \hat{\psi}_k + (\tilde{\psi}_{k+1} - \tilde{\psi}_k) & , \text{ if } \ell \in \bar{\mathcal{P}}_a \end{cases} .$$

The process is repeated until we reach node i.

As an example, let us construct $\hat{\boldsymbol{\psi}}$ for Fig. 2.1. Suppose cycle $\mathcal{C}_2$ is the cycle over which flows differ. Then, nodes $\{1, 2\}$ fall under case *a)*; nodes $\{3, 4, 5\}$ under case *b)*; and nodes $\{6, 7, 8, 910\}$ under *c)*, with node 5 acting as node k. Then, the constructed pressure vector is

$$\hat{\boldsymbol{\psi}} = \begin{bmatrix} \hat{\psi}_1 \\ \hat{\psi}_2 \\ \hat{\psi}_3 \\ \hat{\psi}_4 \\ \hat{\psi}_5 \\ \hat{\psi}_6 \\ \hat{\psi}_7 \\ \hat{\psi}_8 \\ \hat{\psi}_9 \\ \hat{\psi}_{10} \end{bmatrix} = \begin{bmatrix} \tilde{\psi}_1 \\ \tilde{\psi}_2 \\ \tilde{\psi}_3 \\ \psi_4 + (\tilde{\psi}_3 - \psi_3) \\ \psi_5 + (\tilde{\psi}_3 - \psi_3) \\ \alpha_{5,6}\hat{\psi}_5 \\ \hat{\psi}_6 + (\tilde{\psi}_7 - \tilde{\psi}_6) \\ \hat{\psi}_7 + (\tilde{\psi}_8 - \tilde{\psi}_7) \\ \hat{\psi}_8 + (\tilde{\psi}_9 - \tilde{\psi}_8) \\ \hat{\psi}_9 + (\tilde{\psi}_{10} - \tilde{\psi}_9) \end{bmatrix}.$$

We next show that the constructed $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (GF2). From case *b)*, it follows that for all edges $(m, n) \in \mathcal{C}$

$$\hat{\psi}_m - \hat{\psi}_n = \psi_m - \psi_n = a_{mn}\,\mathrm{sign}(\phi_{mn})\phi_{mn}^2. \tag{2.14}$$

The latter implies that constraint (2.10) is satisfied with equality for all $(m, n) \in \mathcal{C}$. Moreover, for all edges $(m, n) \notin \mathcal{C}$, cases *b)* and *c)* yield that

$$\hat{\psi}_m - \hat{\psi}_n = \tilde{\psi}_m - \tilde{\psi}_n.$$

Then, since $(\tilde{\psi}_m, \tilde{\psi}_n)$ satisfy (2.10), the same holds for $(\hat{\psi}_m, \hat{\psi}_n)$ for all $(m, n) \notin \mathcal{C}$. The previous two arguments show that $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (GF2).

Continuing with the third step of this proof, note that the objective $r(\boldsymbol{\psi})$ sums up the absolute pressure differences along lossy pipes. Since by construction these differences have changed only along $\mathcal{C}$, we get that

$$r(\tilde{\boldsymbol{\psi}}) - r(\hat{\boldsymbol{\psi}}) = \sum_{(m,n)\in\mathcal{C}} |\tilde{\psi}_m - \tilde{\psi}_n| - |\hat{\psi}_m - \hat{\psi}_n|. \tag{2.15}$$

As the pressure differences depend on flows, we next compare $\hat{\boldsymbol{\phi}}$ and $\tilde{\boldsymbol{\phi}}$ using (2.13). Since the edge directions are assigned arbitrarily, assume wlog that $\hat{\phi}_{mn} \geq 0$ for all $(m,n) \in \mathcal{C}$. Given $\mathbf{n}^{\mathcal{C}}$ and (2.13), one can find the value of λ. If $\lambda < 0$, reverse the reference direction of cycle $\mathcal{C}$ to redefine its indicator vector as $-\mathbf{n}^{\mathcal{C}}$, and get a positive λ. Then we then assume $\lambda > 0$ without loss of generality.

Recall that $\mathbf{n}^{\mathcal{C}} \in \{0, \pm 1\}^P$. Define the set of edges in $\mathcal{C}$ corresponding to positive entries of $\mathbf{n}^{\mathcal{C}}$ as $\mathcal{P}^+ \subseteq \mathcal{C}$. Likewise, define the set of edges in $\mathcal{C}$ corresponding to negative entries of $\mathbf{n}^{\mathcal{C}}$ as $\mathcal{P}^- \subseteq \mathcal{C}$. Take for example cycle $\mathcal{C}_1$ in Fig. 2.1: Its edges are grouped as $\mathcal{P}^+ = \{(3,5),(4,3)\}$ and $\mathcal{P}^- = \{(5,4)\}$. From (2.13), it follows that

$$\begin{aligned} 0 \leq \hat{\phi}_\ell < \tilde{\phi}_\ell, &\quad \forall \ell \in \mathcal{P}^+; \quad \text{and} \\ \tilde{\phi}_\ell < \hat{\phi}_\ell, &\quad \forall \ell \in \mathcal{P}^-. \end{aligned}$$

Summing up the pressure drops along cycle $\mathcal{C}$ for $\hat{\boldsymbol{\psi}}$ should be zero. In Fig. 2.1 for example we have

$$(\hat{\psi}_3 - \hat{\psi}_5) + (\hat{\psi}_5 - \hat{\psi}_4) + (\hat{\psi}_4 - \hat{\psi}_3) = 0.$$

Since the pressure drop is positive along the edges in $\mathcal{P}^+$, and negative along the edges in

$\mathcal{P}^-$, it holds that

$$\begin{aligned}\sum_{(m,n)\in\mathcal{P}^+}(\hat{\psi}_m-\hat{\psi}_n) &= \sum_{(m,n)\in\mathcal{P}^-}(\hat{\psi}_m-\hat{\psi}_n)\\ \implies \sum_{(m,n)\in\mathcal{C}}|\hat{\psi}_m-\hat{\psi}_n| &= 2\sum_{(m,n)\in\mathcal{P}^+}(\hat{\psi}_m-\hat{\psi}_n) \end{aligned}\tag{2.16}$$

where the absolute value is trivial since $\hat{\phi}_{mn}\geq 0$ for all $(m,n)\in\mathcal{C}$. Referring to Fig. 2.1, the equations in (2.16) imply $(\hat{\psi}_3-\hat{\psi}_5)+(\hat{\psi}_4-\hat{\psi}_3)=(\hat{\psi}_4-\hat{\psi}_5)$.

To draw similar relations on $\tilde{\boldsymbol{\psi}}$, define the set $\tilde{\mathcal{P}}^+\subseteq\mathcal{C}$ containing any edge $(m,n)\in\mathcal{C}$ such that the flow $\tilde{\phi}_{mn}$ is along the direction of $\mathbf{n}^{\mathcal{C}}$. Similarly, define $\tilde{\mathcal{P}}^-:=\mathcal{C}\setminus\tilde{\mathcal{P}}^+$. Using the same argument as in (2.16) for $\tilde{\boldsymbol{\psi}}$, we obtain

$$\sum_{(m,n)\in\mathcal{C}}|\tilde{\psi}_m-\tilde{\psi}_n| = 2\sum_{(m,n)\in\tilde{\mathcal{P}}^+}(\tilde{\psi}_m-\tilde{\psi}_n).\tag{2.17}$$

Since the flows in $\hat{\boldsymbol{\phi}}$ for the edges in $\mathcal{P}^+$ are aligned with $\mathbf{n}^{\mathcal{C}}$ and also $\tilde{\phi}_\ell>\hat{\phi}_\ell$ for these edges, it follows that $\mathcal{P}^+\subseteq\tilde{\mathcal{P}}^+$. Using this fact in (2.17), we get that

$$\begin{aligned}2\sum_{(m,n)\in\mathcal{P}^+}(\tilde{\psi}_m-\tilde{\psi}_n) &\leq 2\sum_{(m,n)\in\tilde{\mathcal{P}}^+}(\tilde{\psi}_m-\tilde{\psi}_n)\\ &= \sum_{(m,n)\in\mathcal{C}}|\tilde{\psi}_m-\tilde{\psi}_n|.\end{aligned}\tag{2.18}$$

For every edge $\ell=(m,n)\in\mathcal{P}^+$, it holds

$$\hat{\psi}_m-\hat{\psi}_n=a_\ell\hat{\phi}_\ell^2<a_\ell\tilde{\phi}_\ell^2\leq\tilde{\psi}_m-\tilde{\psi}_n\tag{2.19}$$

where the equality comes from the definition in (2.14); the first inequality in stems from

$\tilde{\phi}_\ell > \hat{\phi}_\ell \geq 0$; and the second inequality is from (2.10). Summing (2.19) over all $\ell \in \mathcal{P}^+$, and multiplying by 2 gives

$$\begin{aligned} 2 \sum_{(m,n)\in\mathcal{P}^+} (\hat{\psi}_m - \hat{\psi}_n) &< 2 \sum_{(m,n)\in\mathcal{P}^+} (\tilde{\psi}_m - \tilde{\psi}_n) \\ \Longrightarrow \quad \sum_{(m,n)\in\mathcal{C}} |\hat{\psi}_m - \hat{\psi}_n| &< \sum_{(m,n)\in\mathcal{C}} |\tilde{\psi}_m - \tilde{\psi}_n| \end{aligned}$$

where the inequality stems from (2.16) and (2.18). From (2.15), this implies $r(\tilde{\boldsymbol{\psi}}) > r(\hat{\boldsymbol{\psi}})$ contradicting the optimality of $\tilde{\boldsymbol{\psi}}$.

Chapter 3

Natural Gas Flow Solvers using Convex Relaxation

3.1 Publication Details

M. K. Singh and V. Kekatos, “Natural gas flow solvers using convex relaxation,” *IEEE Trans. Control of Network Systems*, vol. 7, no. 3, pp. 1283–1295, Sep. 2020.

3.2 Abstract

The vast infrastructure development, gas flow dynamics, and complex interdependence of gas with electric power networks call for advanced computational tools. Solving the equations relating gas injections to pressures and pipeline flows lies at the heart of natural gas network (NGN) operation, yet existing solvers require careful initialization and uniqueness has been an open question. In this context, this work considers the nonlinear steady-state version of the gas flow (GF) problem. It first establishes that the solution to the GF problem is unique under arbitrary NGN topologies, compressor types, and sets of specifications. For GF setups where pressure is specified on a single (reference) node and compressors do no

appear in cycles, the GF task is posed as an convex minimization. To handle more general setups, a GF solver relying on a mixed-integer quadratically-constrained quadratic program (MI-QCQP) is also devised. This solver can be used for any GF setup at any NGN. It introduces binary variables to capture flow directions; relaxes the pressure drop equations to quadratic inequality constraints; and uses a carefully selected objective to promote the exactness of this relaxation. The relaxation is provably exact in NGNs with non-overlapping cycles and a single fixed-pressure node. The solver handles efficiently the involved bilinear terms through McCormick linearization. Numerical tests validate our claims, demonstrate that the MI-QCQP solver scales well, and that the relaxation is exact even when the sufficient conditions are violated, such as in NGNs with overlapping cycles and multiple fixed-pressure nodes.

3.3 Introduction

Natural gas has served as a critical energy source for decades, mainly for heating and electric power generation [100]. Thanks to the higher ramping capabilities of gas-fired generators, electric power system operators could achieve higher penetration of uncertain and intermittent renewable generation. In addition, the discovery of substantial new supplies of natural gas in the U.S. has led to a new thrust in development of gas-centered technologies and analytical tools [1].

Natural gas produced at gas pits and refineries is primarily transported to customer locations via a continent-wide network of pipelines [100]. The safe, reliable, and economical transportation of gas across these networks is ensured by gas system operators [155]. Considering the scale of natural gas networks (NGN), and their coupling with electric power grids, a plethora of analytical and computational challenges can be envisaged. Stand-alone and gas-electric coupled versions of network expansion planning, optimal scheduling, least-cost procurement, and security analysis are examples of problems that have gained increasing research interest; see [155], [23], [15], [154], [106]. These problems aim at optimizing varying objectives, while respecting network limitations and gas flow physics.

The flow of natural gas on pipelines is governed by partial differential equations, which under steady-state assumptions, yield nonlinear equations relating pressures and gas flows [113]. These equations reveal that the pressure drops along a pipe in the direction of flow due to friction. However, a minimum pressure needs to be maintained at consumer nodes to satisfy gas contracts. Therefore, compressors are placed on selected pipelines to increase the pressure at their output based on a typically multiplicative [100], and rarely additive law [131]. Operators need to solve the set of nonlinear equations governing gas flow in an NGN [92]: For each node, the operator fixes the gas pressure or gas injection rate to specified values. Given also the compression ratios, the GF task aims at finding the injections and pressures at all nodes, as well as the gas flows on all pipes. While solving the GF task is central for numerous NGN operations, it is hard to do so even under steady-state and balanced conditions for non-tree networks [100].

The GF task is usually handled by Newton-Raphson (NR)-based solvers. However, their convergence can be sensitive to initialization [83]. A semidefinite program (SDP)-based GF solver attaining a higher success probability than the NR scheme, is developed in [92]. Nevertheless, the SDP based solver fails to solve the GF problem if the network state is far from the states considered in designing the solver. The necessity of proper initialization may be avoided for simpler networks without compressors as the flows and pressures may be found as optimal primal-dual solutions of a convex minimization [41]. Nevertheless, for practical meshed NGNs with compressors, an initialization-independent GF solver is still a research pursuit [113]. Setting scalability aside, if one uses a nonlinear solver for the GF task, the uniqueness of a solution becomes critical. References [131] and [88] prove the uniqueness of a GF solution for NGNs with *additive* compressors.

The contribution of this work is on four fronts: First, Section 3.5 establishes that the nonlinear steady-state GF equations enjoy a unique solution even with multiplicative compressors. Building on [113] where uniqueness was shown for GF setups with a single fixed-pressure node, here uniqueness is non-trivially generalized to setups with multiple fixed-pressure nodes. Second, Section 3.6 reformulates the GF task as a convex minimization. The obtained solver can handle GF setups with a single fixed-pressure node and compressors not on cycles. Third, Section 3.7 expands the analytical claims for the MI-QCQP gas flow solver of [113]. Different from the convex minimization approach, this solver applies to any GF setup and any network. The MI-QCQP solver introduces binary variables to capture flow directions; relaxes the nonlinear GF equations to quadratic inequalities; and uses a carefully selected objective to promote the exactness of the relaxation. The relaxation is provably exact in NGNs with non-overlapping cycles and a single fixed-pressure node. This significantly extends the claim of [113], where exactness was proved for non-overlapping cycles and a single fixed-pressure node, but did not allow for compressors in cycles. Having compressors in cycles is a typical arrangement, e.g., when two compressors are connected in parallel. Fourth, to accelerate the MI-QCQP solver, the bilinear terms involved are handled through McCormick linearization. Numerical tests on meshed networks with overlapping cycles and multiple fixed-pressure nodes demonstrate that the MI-QCQP solver finds the unique GF solution even when the assumed sufficient conditions are violated.

3.4 Gas Flow Problem

A natural gas network (NGN) can be represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{P})$. The nodes in the graph represent points of gas supply, demand, or network junctions. The edges are *directed*, and represent pipelines or compressors. Nodes are indexed by $n \in \mathcal{N} := \{1, \cdots, N\}$ and edges by $\ell \in \mathcal{P} := \{1, \ldots, P\}$. Each edge $\ell = (m, n)$ is assigned a direction from the origin node m to the destination node n. If $(m, n) \in \mathcal{P}$, then $(n, m) \notin \mathcal{P}$. For edges corresponding to pipes, this direction is selected arbitrarily. For edges denoting compressors, the direction coincides with the direction of gas flow, since compressors allow only unidirectional flow of gas.

For each node $n \in \mathcal{N}$, let q_n be the gas injection rate from node n to the NGN. By convention, the gas injection q_n is positive for gas source nodes; negative for demand nodes; and zero for junction nodes. Vector $\mathbf{q} \in \mathbb{R}^N$ collects the gas injections across all nodes.

For each edge $\ell = (m, n) \in \mathcal{P}$, let ϕ_ℓ denote its gas flow rate. By convention, the flow ϕ_ℓ is positive if gas flows from node m to n; and negative, otherwise. The conservation of mass at each node $n \in \mathcal{N}$ dictates that

$$q_n = \sum_{\ell:(n,k)\in\mathcal{P}} \phi_\ell - \sum_{\ell:(k,n)\in\mathcal{P}} \phi_\ell. \tag{3.1}$$

Under steady-state conditions, the input and output flows on a pipe are identical, and so gas injections are balanced at all times, that is $\sum_{n=1}^{N} q_n = 0$. Because of this, from the N linear equations in (3.1), only $(N-1)$ are linearly independent.

The topology of the NGN is captured by its edge-node incidence matrix $\mathbf{A} \in \mathbb{R}^{P\times N}$ with entries

$$A_{\ell,k} := \begin{cases} +1 & , \ k = m \\ -1 & , \ k = n \\ 0 & , \ \text{otherwise} \end{cases} \qquad \forall\ \ell = (m,n) \in \mathcal{P}.$$

Using $\mathbf{A}$, equation (3.1) can be compactly expressed as

$$\mathbf{A}^\top \boldsymbol{\phi} = \mathbf{q}. \tag{3.2}$$

where vector $\boldsymbol{\phi} \in \mathbb{R}^P$ stacks the flows ϕ_ℓ's along all edges.

For medium- and high-pressure networks, the gas flows on pipelines relate to nodal pressures through a set of nonlinear partial differential equations [93], [124]. These equations model the gas flow dynamics evolving across time and spatially along the pipeline length. However, simplifying assumptions such as ignoring friction, geographical tilt, variations in ambient temperature, and time-varying gas injections, yield the popular steady-state Weymouth equation [136]. If $\psi_n > 0$ denotes the *squared* gas pressure at node $n \in \mathcal{N}$, the pressure drop across pipeline $\ell = (m,n) \in \mathcal{P}$ is given by

$$\psi_m - \psi_n = a_\ell \operatorname{sign}(\phi_\ell)\phi_\ell^2 \tag{3.3a}$$

$$\psi_n \geq 0 \tag{3.3b}$$

where parameter $a_\ell > 0$ depends on physical properties of the pipeline [93]. The function $\operatorname{sign}(x)$ returns $+1$ if $x > 0$; -1 if $x < 0$; and 0 if $x = 0$. The absolute value in (3.3a) signifies that pressure drops along the direction of flow. In particular, the drop in squared pressures is proportional to the squared flow. We will henceforth refer to ψ_m as pressure rather than squared pressure for brevity. Let us collect all ψ_n's in $\boldsymbol{\psi} \in \mathbb{R}^N$.

To enable the desired flow of gas in an NGN while maintaining pressures within acceptable limits, system operators install compressors at selected pipelines. A pipeline hosting a compressor can be modeled by an ideal compressor which increases the gas pressure, followed by a lossy pipeline that incurs a pressure drop per (3.3). Apparently, the gas flows on the two edges are identical. Let the subset of edges hosting ideal compressors be $\mathcal{P}_a \subset \mathcal{P}$. The edges in $\mathcal{P}_a$ are also referred to as *active pipelines.* The pressures across an active pipeline or compressor $\ell = (m, n) \in \mathcal{P}_a$ are related as

$$\psi_n = \alpha_\ell \psi_m \tag{3.4a}$$

$$\phi_\ell \geq 0, \tag{3.4b}$$

where $\alpha_\ell > 0$ is the multiplicative compression factor for compressor ℓ. The unidirectional flow permitted for a compressor is enforced by (3.4b). The remaining edges, that is the edges not hosting ideal compressors, constitute the set $\bar{\mathcal{P}}_a := \mathcal{P} \setminus \mathcal{P}_a$ and abide by (3.3) instead of (3.4).

In an NGN, a node $r \in \mathcal{N}$ is selected as a reference node. Its pressure is kept fixed. Given ψ_r, if the nodal pressures $\boldsymbol{\psi}$ are known, the flows $\boldsymbol{\phi}$ can be readily computed; and vice versa. This fact follows immediately from (3.3)–(3.4), and is itemized as the next lemma to be used in subsequent arguments.

Lemma 3.1. *Given a reference pressure ψ_r for some $r \in \mathcal{N}$, a pair $(\boldsymbol{\phi}, \boldsymbol{\psi})$ satisfying* (3.3)*–*(3.4) *is uniquely characterized by either $\boldsymbol{\phi}$ or $\boldsymbol{\psi}$.*

The task of finding $\boldsymbol{\phi}$ or $\boldsymbol{\psi}$ given a combination of nodal injections and pressures constitutes the *gas flow* (GF) problem. Oftentimes, gas supply nodes are tuned to maintain a fixed pressure while injecting variable amounts of gas to meet the prescribed pressure under variable demands [155], [77]. Let set $\mathcal{N}_\psi \subset \mathcal{N}$ consist of all nodes with fixed pressures ψ_n's. The reference node r belongs to $\mathcal{N}_\psi$ by definition. Its complement set $\mathcal{N}_q := \mathcal{N} \setminus \mathcal{N}_\psi$ consists of all nodes with fixed injections q_n's. Then, the GF problem can be formally stated now.

Definition 3.2. Given pressures ψ_n for $n \in \mathcal{N}_\psi$; injections q_n for $n \in \mathcal{N}_q$; the ratios α_ℓ for all compressors $\ell \in \mathcal{P}_a$; and the friction parameters a_ℓ for all lossy pipes $\ell \in \bar{\mathcal{P}}_a$, the GF problem aims at finding the triplet $(\boldsymbol{\psi}, \boldsymbol{\phi}, \mathbf{q})$ satisfying the GF equations (3.2)–(3.4).

The GF task involves $N-1+P$ equations over $N-1+P$ unknowns. It can be posed as the feasibility problem

$$\begin{aligned} \text{find} \quad & \{\mathbf{q}, \boldsymbol{\phi}, \boldsymbol{\psi}\} \\ \text{s.to} \quad & (3.2) - (3.4) \\ & \text{given } \{q_n\}_{n\in\mathcal{N}_q} \text{ and } \{\psi_n\}_{n\in\mathcal{N}_\psi}. \end{aligned} \tag{G1}$$

Albeit (3.2) and (3.4) are linear, the piecewise quadratic Weymouth equation in (3.3) is non-convex, while the requirement $\{\phi_\ell \geq 0\}_{\ell\in\mathcal{P}_a}$ further complicates the task. The GF problem is typically solved using the Newton-Raphson's method, yet its convergence depends on the initialization [83], [77], [92]. Commercially available software require careful manual tuning by gas network operator personnel, though that could be attributed to more detailed models of NGN components.

A popular rendition of the GF problem considers the reference node as the only fixed-pressure node, and all other nodes as fixed-injection nodes [87], [92], [113]. For this rendition, solving the GF problem becomes trivial for a tree network by inverting (3.2) and using Lemma 3.1. However, for a meshed NGN, solving the GF problem remains non-trivial. Before developing new GF solvers, the next section establishes that the GF problem in (G1) enjoys a unique solution.

3.5 Uniqueness of the GF Solution

We commence with the uniqueness of the GF task under the setup of a single fixed-pressure node, proved in [113, Th. 1].

Theorem 3.3 ([113]). *If $\mathcal{N}_\psi = \{r\}$ and $\mathcal{N}_q = \mathcal{N} \setminus \{r\}$, the gas flow problem* (G1) *has a unique solution, if feasible.*

Although the single fixed-pressure setup has been studied widely, setups with multiple fixed-pressure nodes are of critical interest too. This is because gas is typically injected at supplier sites using a controller that maintains constant pressure, rather than constant rate. To address this need, this section builds on Th. 3.3 and establishes the uniqueness of the steady-state GF equations for any $(\mathcal{N}_\psi, \mathcal{N}_q)$ setup. Before doing so, let us briefly review some graph theory preliminaries.

A directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{P})$ is connected if there exists a sequence of adjacent edges between any two nodes. All graphs considered in this work are assumed to be connected. A sequence of adjacent edges between nodes m and n constitutes a *path* $\mathcal{P}_{mn} \subset \mathcal{P}$. The directionality assigned to path $\mathcal{P}_{mn}$ is from m to n. Note that nodes m and n could be connected by multiple paths. Thus, with slight abuse in notation, path $\mathcal{P}_{mn}$ shall represent any arbitrary path between m and n, unless additional conditions are provided. For path $\mathcal{P}_{mn}$, we can define an *indicator vector* $\boldsymbol{\pi}^{mn} \in \{0,1\}^P$ with $\ell-$th entry

$$\pi_\ell^{mn} := \begin{cases} 0 & , \text{ if edge } \ell \notin \mathcal{P}_{mn} \\ +1 & , \text{ if direction of } \ell \text{ agrees with path direction} \\ -1 & , \text{ otherwise.} \end{cases}$$

A *cycle* is a sequence of adjacent edges (without edge or node repetition) that starts and ends at the same node. With a slight abuse of terminology, the statement 'cycle $\mathcal{C}$ *contains* node i' will mean that there exists an edge in $\mathcal{C}$ that is incident to node i. For any cycle $\mathcal{C}$, we can select an arbitrary direction and define its indicator vector $\mathbf{n}^{\mathcal{C}}$ with ℓ-th entry

$$n_\ell^{\mathcal{C}} = \begin{cases} 0 & \text{, if edge } \ell \notin \mathcal{C} \\ +1 & \text{, if direction of } \ell \text{ agrees with cycle direction} \\ -1 & \text{, otherwise.} \end{cases}$$

A *tree* is a connected graph with no cycles.

After the graph theoretic preliminaries, we proceed with the uniqueness of the GF solution for the general GF setup. This proof builds upon the ensuing two lemmas, which are proved in the appendix.

Lemma 3.4. *Consider path $\mathcal{P}_{mn}$ along edges $\{\ell_1, \ldots, \ell_k\}$ with indicator $\boldsymbol{\pi}^{mn}$. For fixed pressures ψ_m and ψ_n, if flow vectors $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ with $\boldsymbol{\phi} \neq \boldsymbol{\phi}'$, satisfy (3.3)–(3.4), they cannot satisfy*

$$\text{sign}(\boldsymbol{\phi}' - \boldsymbol{\phi}) \odot \boldsymbol{\pi}^{mn} > \mathbf{0} \quad \textit{or} \tag{3.5a}$$

$$\text{sign}(\boldsymbol{\phi}' - \boldsymbol{\phi}) \odot \boldsymbol{\pi}^{mn} < \mathbf{0} \tag{3.5b}$$

where the strict inequalities are understood entrywise.

To get some intuition, suppose that $\boldsymbol{\pi}^{mn}$ takes the value of $+1$ for edges $\{\ell_1, \ldots, \ell_k\}$, and 0 for the remaining edges. According to Lemma 3.4, if two pairs $(\boldsymbol{\phi}, \boldsymbol{\psi})$ and $(\boldsymbol{\phi}', \boldsymbol{\psi}')$ satisfy (3.3)–(3.4) with $\psi_m = \psi'_m$ and $\psi_n = \psi'_n$, then the flows along $\mathcal{P}_{mn}$ cannot uniformly increase from $\boldsymbol{\phi}$ to $\boldsymbol{\phi}'$. In other words, $\phi'_\ell > \phi_\ell$ cannot occur simultaneously for all $\ell \in \mathcal{P}_{mn}$. Flows cannot uniformly decrease either ($\phi'_\ell < \phi_\ell$ for all $\ell \in \mathcal{P}_{mn}$). This holds merely because the pressure drop across a pipe decreases monotonically with gas flow from and compressors perform a linear scaling [cf. (3.3) and (3.4)].

The next lemma describes an interesting effect on how gas flows get redistributed when gas injections change.

Lemma 3.5. *Consider two pairs* $(\mathbf{q}, \boldsymbol{\phi})$ *and* $(\mathbf{q}', \boldsymbol{\phi}')$ *satisfying* (3.2)*. If* $\mathbf{q} \neq \mathbf{q}'$*, there exists a path* $\mathcal{P}_{mn}$ *between nodes* m *and* n *such that*

$$\operatorname{sign}(\boldsymbol{\phi}' - \boldsymbol{\phi}) \odot \boldsymbol{\pi}^{mn} > \mathbf{0} \tag{3.6a}$$

$$q'_m > q_m \quad \textit{and} \quad q'_n < q_n \tag{3.6b}$$

where $\boldsymbol{\pi}^{mn}$ *is the indicator vector for* $\mathcal{P}_{mn}$*.*

Lemma 3.5 predicates that if gas injections change, there exists a path: *i)* along which flows increase uniformly; *ii)* the source node of the path has increased injection; and *iii)* the destination node has decreased injection. Lemma 3.5 has been established in [131] via mathematical induction; see the appendix for an alternative perhaps more intuitive proof.

Using Theorem 3.3 and Lemmas 3.4–3.5, we next prove the uniqueness of the GF task under the general setup.

Theorem 3.6. *The gas flow problem* (G1) *has a unique solution, if feasible.*

Proof. Proving by contradiction, assume $(\mathbf{q}, \boldsymbol{\phi}, \boldsymbol{\psi})$ and $(\mathbf{q}', \boldsymbol{\phi}', \boldsymbol{\psi}')$ are two distinct solutions of (G1). Consider the GF setup where $|\mathcal{N}_\psi| > 1$; the special case of $|\mathcal{N}_\psi| = 1$ is covered by Theorem 3.3. If $\mathbf{q} \neq \mathbf{q}'$, then Lemma 3.5 implies that there exists a path $\mathcal{P}_{mn}$ with indicator vector $\boldsymbol{\pi}^{mn}$ satisfying (3.6a). Moreover, it holds that $q'_m > q_m$ and $q'_n < q_n$ from (3.6b). By definition, gas injections q_i are fixed for all nodes $i \in \mathcal{N}_q$. Therefore, nodes m

and n cannot be fixed-injection nodes. They have to be fixed-pressure nodes belonging to $\mathcal{N}_\psi$, implying $\psi_m = \psi'_m$ and $\psi_n = \psi_n$. However, with the pressures at nodes m and n fixed, the inequality (3.6a) contradicts Lemma 3.4. Hence, the assumption of unequal injections is refuted, implying $\mathbf{q} = \mathbf{q}'$.

Given $\mathbf{q}$ and the reference pressure ψ_r, Theorem 3.3 asserts that there is unique triplet $(\mathbf{q}, \boldsymbol{\phi}, \boldsymbol{\psi})$ satisfying the GF equations. Since $\mathbf{q} = \mathbf{q}'$, the triplets $(\mathbf{q}, \boldsymbol{\phi}, \boldsymbol{\psi})$ and $(\mathbf{q}', \boldsymbol{\phi}', \boldsymbol{\psi}')$ have to coincide, which completes the proof. □

The uniqueness claim of Theorem 3.6 is fairly general, since it applies to any NGN topology and any GF setup with a single or multiple fixed-pressure nodes. Having established uniqueness, the next two sections develop a suite of GF solvers: Section 3.6 builds upon an existing convex solver for GF setups with a single fixed-pressure node and no compressors. We develop an unconstrained convex solver as well as an extension that handles compressors on non-overlapping cycles. Section 3.7 adopts a convex relaxation and puts forth an MI-QCQP to handle more general GF setups. The relaxation is provably exact for NGNs with a single fixed-pressure node and non-overlapping cycles. Nonetheless, numerical tests demonstrate that this MI-QCQP succeeds in finding the unique GF solution in NGNs with multiple fixed-pressure nodes and overlapping cycles as long as compressors are not on overlapping cycles.

3.6 Energy Function Minimization

This section studies the GF task for the special case of $|\mathcal{N}_\psi| = 1$. In an NGN without compressors, the GF task is posed as a convex minimization. The approach can be extended to networks having compressors, but not on cycles.

3.6.1 Existing Constrained Energy Function-based GF Solver

Consider solving the GF task for a single fixed-pressure node (the reference node r) and in an NGN without compressors. This task boils down to solving equations (3.2)–(3.3). As shown in [41], the gas flows $\boldsymbol{\phi}$ for this GF setup can be found as the minimizer of the convex minimization

$$\min_{\boldsymbol{\phi}} \quad \sum_{\ell \in \mathcal{P}} \frac{a_\ell}{3} |\phi_\ell|^3 \tag{3.7a}$$

$$\text{s.to} \quad \mathbf{A}^\top \boldsymbol{\phi} = \mathbf{q}. \tag{3.7b}$$

This can be readily verified by the first-order optimality conditions of (3.7). In addition, the pressures $\boldsymbol{\psi}$ can be recovered from the optimal Lagrange multipliers $\boldsymbol{\xi} \in \mathbb{R}^N$ associated with constraint (3.7b): If $\boldsymbol{\xi}$ is shifted by a constant so that its r-th entry equals ψ_r, the remaining entries of this shifted $\boldsymbol{\xi}$ equal $\boldsymbol{\psi}$. Problem (3.7) can be reformulated as a second-order cone program or tackled via dual decomposition; see [110].

3.6.2 Novel Unconstrained Energy Function-based GF Solver

Rather than solving (3.7) over $\boldsymbol{\phi}$, here we show that one can alternatively find the GF solution via an unconstrained convex minimization over $\boldsymbol{\psi}$ as

$$\min_{\boldsymbol{\psi}} \; \frac{2}{3} \sum_{(m,n) \in \mathcal{P}} \frac{|\psi_m - \psi_n|^{\frac{3}{2}}}{\sqrt{a_{mn}}} - \mathbf{q}^\top \boldsymbol{\psi}. \tag{3.8}$$

The convexity of this objective function follows from composition rules. Since this function is convex and differentiable, its unconstrained minimization is equivalent to nulling its gradient vector. Setting the n-th entry of this gradient to zero reveals that the minimizer $\boldsymbol{\psi}^*$ of (3.8) satisfies

$$\sum_{\ell=(m,n)\in\mathcal{P}} \text{sign}(\mathbf{a}_\ell^\top \boldsymbol{\psi}^*)\sqrt{\frac{|\mathbf{a}_\ell^\top \boldsymbol{\psi}^*|}{a_\ell}} = q_n \tag{3.9}$$

where $\mathbf{a}_\ell^\top$ is the ℓ-th row of matrix $\mathbf{A}$. Equation (3.9) is equivalent to eliminating the flows $\boldsymbol{\phi}$ from (3.2) and (3.3). As with (3.7), the ambiguity in pressures could be handled by shifting $\boldsymbol{\psi}^*$ by a constant, so that ψ_r^* agrees with the given pressure at the reference node r. Once pressures $\boldsymbol{\psi}^*$ have been determined, flows can be found using Lemma 3.1.

Remark 3.7. In the absence of compressors and when $|\mathcal{N}_\psi| = 1$, the GF task becomes structurally similar to the water flow problem in water distribution networks without pumps [110]. Therefore, the (un)-constrained energy function minimization approaches of (3.7)–(3.8) apply to the gas flow and water flow problems alike. For water networks, the decomposition technique of [110] extends (3.7)–(3.8) to water network setups with $|\mathcal{N}_\psi| = 1$ and pumps, but pumps cannot lie on cycles. A similar technique can be used to solve the GF problem with compressors not on cycles and $|\mathcal{N}_\psi| = 1$. The only modification needed relates to accounting for the multiplicative pressure law in gas compressors [cf. (3.4a)] vis-à-vis the additive pressure law of water pumps. Additionally, the decomposition algorithm may be extended to accommodate compressors on non-overlapping cycles using the flow-recovery procedure provided later as Algorithm 1. Since carrying over this decomposition technique from the water flow to the gas flow context is straight-forward and due to space limitations, it is not presented here.

To handle GF setups with $|\mathcal{N}_\psi| > 1$ and/or NGNs with compressors in loops, a convex relaxation of the Weymouth equation is pursued in the next section.

3.7 MI-QCQP Relaxation

The minimization approaches of (3.7)–(3.8) provide computationally efficient methods to solve the GF problem, but exhibit three limitations: *i)* they cannot handle multiple fixed-pressure nodes ($|\mathcal{N}_\psi| > 1$); *ii)* cannot handle compressors on cycles; and *iii)* cannot be extended to optimal gas flow formulations (e.g., along the lines of [115]). To overcome these limitations, this section presents an MI-QCQP-based solver that is applicable to any GF setup.

3.7.1 Problem Reformulation

The non-convexity of (G1) is due to the Weymouth equation in (3.3a). The piecewise quadratic equalities can be relaxed to convex inequality constraints: The pressure drop along a lossy pipe $\ell = (m, n) \in \bar{\mathcal{P}}_a$ is relaxed to

- $\psi_m - \psi_n \geq a_\ell \phi_\ell^2$ for $\phi_\ell \geq 0$; or
- $\psi_n - \psi_m \geq a_\ell \phi_\ell^2$ for $\phi_\ell \leq 0$.

The two cases can be differentiated using a binary variable x_ℓ capturing the direction of flow ϕ_ℓ. The relaxed pressure drop equations can be compactly written as

$$(2x_\ell - 1)\psi_m + (1 - 2x_\ell)\psi_n \geq a_\ell \phi_\ell^2$$

where $x_\ell = 1$ corresponds to $\phi_\ell \geq 0$; and $x_\ell = 0$ to $\phi_\ell \leq 0$. Despite the relaxation, the bilinear terms $x_\ell \psi_m$ make the aforementioned constraint non-convex.

The McCormick linearization, popular for approximating multilinear terms by their linear convex envelopes, can be used to handle these bilinear terms [86]. For the special case of bilinear terms involving at least one binary term, the McCormick linearization becomes exact. In fact, it is related to the so termed big-M trick, but instead of using a single arbitrarily large value for M, it selects different values for M that are specialized per product of variables, which could potentially reduce the running time of mixed-integer programming solvers. Let us briefly review the linearization. Consider the constraint $z_{\ell n} = x_\ell \psi_n$, for which $x_\ell \in \{0, 1\}$ and $\psi_n \in [\underline{\psi}_m, \overline{\psi}_n]$. This constraint can be equivalently expressed via four linear inequalities

$$x_\ell \underline{\psi}_n \leq z_{\ell n} \leq x_\ell \overline{\psi}_n \tag{3.10a}$$

$$\psi_n + (x_\ell - 1)\overline{\psi}_n \leq z_{\ell n} \leq \psi_n + (x_\ell - 1)\underline{\psi}_n. \tag{3.10b}$$

To verify the exactness, observe that when $x_\ell = 1$, constraint (3.10b) yields $z_{\ell n} = \psi_n$ and (3.10a) holds trivially. When $x_\ell = 0$, constraint (3.10a) enforces $z_{\ell n} = 0$ and (3.10b) holds trivially. Hence, the constraints in (3.10) ensure that $z_{\ell n} = x_\ell \psi_n$.

To arrive at an MI-QCQP relaxation of (G1), for all lossy pipes $\ell \in \bar{\mathcal{P}}_a$, the pressure drop constraint of (3.3a) is replaced by (3.10) and

$$2z_{\ell m} - 2z_{\ell n} + \psi_n - \psi_m \geq a_\ell \phi_\ell^2, \tag{3.11a}$$

$$-\overline{\phi}_\ell(1 - x_\ell) \leq \phi_\ell \leq \overline{\phi}_\ell x_\ell, \tag{3.11b}$$

where $\overline{\phi}_\ell$ is an upper bound on $|\phi_\ell|$. Constraint (3.11a) represents the relaxed Weymouth equation, and constraint (3.11b) defines $x_\ell = \text{sign}(\phi_\ell)$. Similar relaxations have been previously used in [23], [104], [113]; see Section 3.8 for a detailed comparison.

When solving the GF problem with the Weymouth equations relaxed, the obtained solution is useful only if the relaxation is *exact*, that is when (3.11a) holds with equality for all ℓ.

To render the relaxation provably exact, we convert the feasibility problem (G1) to the MI-QCQP minimization

$$\begin{aligned} \min \quad & r(\boldsymbol{\psi}) \\ \text{over} \quad & \mathbf{q}, \boldsymbol{\phi}, \boldsymbol{\psi}, \mathbf{x} \\ \text{s.to} \quad & (3.2), (3.4), (3.10), (3.11). \end{aligned} \tag{G2}$$

The optimization variable $\mathbf{x}$ stacks $\{x_\ell\}_{\ell \in \bar{\mathcal{P}}_a}$, and the objective function is judiciously selected as

$$r(\boldsymbol{\psi}) := \sum_{\substack{(m,n)\in\bar{\mathcal{P}}_a \\ (m,n)\notin\mathcal{S}_{\mathcal{C}}^a}} |\psi_m - \psi_n|$$

where $\mathcal{S}_{\mathcal{C}}^a$ is the set of cycles with compressors. These cycles will be also termed as *active cycles*. The cost $r(\boldsymbol{\psi})$ sums up the absolute pressure differences across all lossy pipes not in active cycles. Despite the non-convexity of (G2) due to the binary variables, this minimization can be handled for moderately sized networks thanks to the advancements in mixed-integer second-order cone solvers. The computational performance of (G2) is further corroborated by our tests. The next section provides network conditions under which the exactness of (G2) can be guaranteed analytically. The tests in Section 3.9 demonstrate numerically that solving (G2) renders the relaxation exact for a much broader class of networks.

For solving tasks such as (G1), NR-based or fixed-point iteration solvers are often preferred as opposed to optimization-based solvers due to computational superiority. However, in addition to guaranteeing convergence irrespective of initialization, problem (G2) can also be used as follows:

- *Infeasibility:* As a relaxation of (G1), (G2) can be used to screen infeasible GF instances; see Section 3.9 for tests. Such screening is of practical use as suggested in

[43].

- *Initialization:* Problem (G2) could be terminated before reaching optimality to yield initializations for NR solvers, hence combining the benefits of both approaches.

- *Optimal gas flow:* The cost of (G2) could be useful as a penalty term that can be added to optimization problems [110], [115]. However, guaranteeing exact relaxation for such problems would need further analysis.

3.7.2 Exactness of the Relaxation

The relaxation in (G2) will be analytically shown to be exact under the following network conditions.

Condition 1. The GF setup has a single fixed-pressure node, that is $|\mathcal{N}_\psi| = 1$.

Condition 2. Each edge of the NGN belongs to at most one cycle.

Condition 3. The NGN does not exhibit circulation of gas, that is $\mathbf{n}^c \odot \boldsymbol{\phi} \not> \mathbf{0}$ and $\mathbf{n}^c \odot \boldsymbol{\phi} \not< \mathbf{0}$ for every cycle $\mathcal{C}$.

Under Condition 1, the nodal injections are fixed *a priori* and the GF task aims at finding the associated $(\boldsymbol{\psi}, \boldsymbol{\phi})$. Albeit Definition 3.2 considered the GF task with multiple fixed-pressure nodes, the setup of a single fixed-pressure node is commonly met; see [87], [92], [113], [6]. Regarding Condition 2, although it may seem restrictive at the outset, it is satisfied by several practical gas networks [14]. For Condition 3, a circulation occurs when gas flows around a cycle along the same direction. It is easy to verify that gas cannot circulate in a cycle without compressors, since the incurred pressure drops along the cycle will all be in the same direction and thus cannot sum up to zero. In cycles with compressors, gas circulation can occur though it would cause an undesirable loss of energy. However, the tests of Section 3.9 demonstrate that the relaxation in (G2) is exact even in setups where the sufficient Conditions 1–3 are all violated.

The next exactness claim applies to the GF setup with known injections. From Lemma 3.1, we know that solving the GF task is equivalent to finding the correct flows $\boldsymbol{\phi}$. The next result provides conditions under which (G2) yields flows $\boldsymbol{\phi}$ with partially correct entries. An algorithm to retrieve the entire $\boldsymbol{\phi}$ and thus eventually solve (G1) is presented afterwards.

Theorem 3.8. *Let $\boldsymbol{\phi}$ be the unique flow vector solving (G1), and $\boldsymbol{\phi}'$ the flow vector minimizing (G2). Under Cond. 1–3, it holds $\phi'_\ell = \phi_\ell$ for all edges ℓ not belonging to active cycles.*

Theorem 3.8 establishes that the only possible mismatches between $\boldsymbol{\phi}'$ and $\boldsymbol{\phi}$ occur only at the edges lying on cycles with compressors. Then, if there are no cycles with compressors, the GF problem is solved correctly; see also [113, Th. 2].

Corollary 3.9. *Under Cond. 1–2, for an NGN without compressors in cycles, the minimizer of (G2) solves (G1) as well.*

Corollary 3.9 identifies a setup where (G2) is equivalent to solving (G1). Nonetheless, if there are no compressors in cycles, one would prefer tackling (G1) using the solvers of Section 3.6. This is because running the decomposition technique discussed in Remark 3.7 and solving (3.7), are simpler than solving the MI-QCQP of (G2).

3.7.3 Recovering the GF Solution

Returning to the general setup, we next provide a procedure to retrieve the solution $\boldsymbol{\phi}$ of (G1) given a minimizer $\boldsymbol{\phi}'$ of (G2). From Theorem 3.8, vector $\boldsymbol{\phi}'$ needs to be corrected only at the entries corresponding to edges in active cycles. To this end, we first put forth an algorithm to correct the flows within a single active cycle, and then delineate the steps to systematically correct the flows for all active cycles of the network.

Consider an active cycle $\mathcal{C}$ with $N_{\mathcal{C}}$ nodes. Let ψ_0 be a known pressure on node $0 \in \mathcal{C}$, and $\boldsymbol{\phi}'_{\mathcal{C}}$ be the $N_{\mathcal{C}}$-length subvector of $\boldsymbol{\phi}'$ collecting the flows on $\mathcal{C}$. Similarly, let $\mathbf{n}_{\mathcal{C}}$ be the $N_{\mathcal{C}}$-length subvector of the indicator vector for cycle $\mathcal{C}$. The next lemma explains how $\boldsymbol{\phi}_{\mathcal{C}}$ can be recovered from $\boldsymbol{\phi}'_{\mathcal{C}}$.

Lemma 3.10. *Given a known pressure ψ_0, and flows $\boldsymbol{\phi}'_{\mathcal{C}}$ on active cycle $\mathcal{C}$ obtained from* (G2)*, Algorithm 1 determines the corrected gas flows $\boldsymbol{\phi}_{\mathcal{C}}$ such that the relaxed Weymouth equations in* (3.11) *are satisfied with equality.*

Proof. Because $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ both satisfy (3.2), it follows that $(\boldsymbol{\phi} - \boldsymbol{\phi}') \in \text{null}(\mathbf{A}^\top)$. Since there are no overlapping cycles, we have that $\boldsymbol{\phi}_{\mathcal{C}} = \boldsymbol{\phi}'_{\mathcal{C}} + \lambda_{\mathcal{C}}\mathbf{n}_{\mathcal{C}}$ for some $\lambda_{\mathcal{C}} \in \mathbb{R}$. To recover $\boldsymbol{\phi}_{\mathcal{C}}$, we next provide a method for finding $\lambda_{\mathcal{C}}$.

Suppose one is given a $\lambda \in \mathbb{R}$. Given pressure ψ_0 and the candidate flow vector $\boldsymbol{\phi}'_{\mathcal{C}} + \lambda\mathbf{n}_{\mathcal{C}}$, one can calculate the pressures along $\mathcal{C}$ sequentially using (3.3a) and (3.4a). Upon completing the cycle, the pressure at node $0 \in \mathcal{C}$ will be evaluated to the value of $\hat{\psi}_0(\lambda)$. The value $\hat{\psi}_0(\lambda)$ may not be equal to ψ_0. Note that for $\lambda > \lambda_{\mathcal{C}}$, it holds that

$$\text{sign}(\boldsymbol{\phi}'_{\mathcal{C}} - \boldsymbol{\phi}_{\mathcal{C}} + \lambda\mathbf{n}_{\mathcal{C}}) \odot \mathbf{n}_{\mathcal{C}} = \text{sign}\left((\lambda - \lambda_{\mathcal{C}})\,\mathbf{n}_{\mathcal{C}}\right) \odot \mathbf{n}_{\mathcal{C}} > \mathbf{0}.$$

Using the above along with the argument used in the proof of Lemma 3.4, it can be shown that $\hat{\psi}_0(\lambda) < \psi_0$. In a similar fashion, if $\lambda < \lambda_{\mathcal{C}}$, then $\hat{\psi}_0(\lambda) > \psi_0$. Therefore, the function $\hat{\psi}_0(\lambda) - \psi_0$ is monotonic in λ, and $\hat{\psi}_0(\lambda) = \psi_0$ if and only if $\lambda = \lambda_{\mathcal{C}}$. Thanks to this monotonicity, one can find $\lambda_{\mathcal{C}}$ iteratively using bisection, tabulated as Algorithm 1. $\square$

Lemma 3.10 shows that $\boldsymbol{\phi}_\mathcal{C}$ can be recovered from ψ_0 and $\boldsymbol{\phi}'_\mathcal{C}$ using a bisection technique on λ. The limits for the search space $[\underline{\lambda}, \overline{\lambda}]$ of λ can be found using engineering constraints on gas flows. In fact, these limits can be tightened since the entries of $\boldsymbol{\phi}'_\mathcal{C}$ and $\boldsymbol{\phi}_\mathcal{C}(\lambda) = \boldsymbol{\phi}'_\mathcal{C} + \lambda \mathbf{n}_\mathcal{C}$ corresponding to any compressor in $\mathcal{C}$ must have the same sign due to (3.4b).

We next provide the steps to find the correct GF solution using the flow $\boldsymbol{\phi}'$ obtained from (G2):

T1) Select a spanning tree $\mathcal{T}$ of the NGN graph $\mathcal{G}$ rooted at the reference node r.

T2) Starting from node r, traverse $\mathcal{T}$ via a depth-first search.

T3) If a node n does not belong to an active cycle of $\mathcal{G}$, calculate its pressure as follows: If the edge connecting node n to its parent node in $\mathcal{T}$ is a lossy pipe, use (3.3a); else, if this edge is a compressor, use (3.4a).

T3) If a node n belongs to an active cycle $\mathcal{C}$, check if the flows in cycle $\mathcal{C}$ have been corrected. If the flows are already corrected or if i is the first node in $\mathcal{C}$ that is encountered, compute the nodal pressure as in step *T3)*. Else, pass the pressure at the parent node of i (which is also in $\mathcal{C}$) along with the non-corrected flow subvector $\boldsymbol{\phi}'_\mathcal{C}$ to Algorithm 1 and obtain the corrected flows on $\mathcal{C}$.

T4) Continue until all nodes in $\mathcal{T}$ have been traversed.

3.8 Comparison to Prior Work

This work puts forth three novel components: *c1)* proving the uniqueness of the GF problem solution under steady-state conditions; *c2)* proposing GF solvers based on energy function minimization; and *c3)* devising a provably exact MI-QCQP relaxation. These components

Algorithm 1 Recover flows on active cycles upon solving (G2)

Input : $\psi_0, \boldsymbol{\phi}'_{\mathcal{C}}, \mathbf{n}_{\mathcal{C}}, \underline{\lambda}, \overline{\lambda}$; tolerance ϵ; pipe and compressor parameters along $\mathcal{C}$
Output : flow vector $\boldsymbol{\phi}_{\mathcal{C}}$ and pressure vector $\boldsymbol{\psi}_{\mathcal{C}}$ along $\mathcal{C}$
Initialize: Set $\lambda \leftarrow \frac{\underline{\lambda}+\overline{\lambda}}{2}$ and $\psi_0(\lambda) \leftarrow \infty$
while $|\psi_0(\lambda) - \psi_0| \geq \epsilon$ **do**
  Try flow vector $\boldsymbol{\phi}'_{\mathcal{C}} + \lambda \mathbf{n}_{\mathcal{C}}$. Starting from node 0, compute pressures $\psi_n(\lambda)$ along all nodes in $n \in \mathcal{C}$ using (3.3a) and (3.4a) until you return to node 0.
  if $\psi_0(\lambda) > \psi_0$ **then**
    Set $\underline{\lambda} \leftarrow \lambda,\ \lambda \leftarrow \frac{\underline{\lambda}+\overline{\lambda}}{2}$
  else
    Set $\overline{\lambda} \leftarrow \lambda,\ \lambda \leftarrow \frac{\underline{\lambda}+\overline{\lambda}}{2}$
  end
end
Return : flows $\boldsymbol{\phi}_{\mathcal{C}} = \boldsymbol{\phi}'_{\mathcal{C}} + \lambda \mathbf{n}_{\mathcal{C}}$ and pressures $\boldsymbol{\psi}_{\mathcal{C}}(\lambda)$

are next contrasted to existing related works:

c1) Uniqueness: For an NGN with no compressors, the GF solution may be found as a minimizer of (3.7); see [41], [43]. A linearization technique has also been put forth to accelerate solving (3.7) [43]. References [131] and [88] broaden the uniqueness claim for NGNs with *additive* compressors of constant gain. These works formulate strictly convex problems that yield a GF solution; hence proving uniqueness by convexity. However, gas compressors are oftentimes *multiplicative*, so that the previous uniqueness claims do not carry over. In our work [113], uniqueness was proved for multiplicative compressors under any network topology, but with a single fixed-pressure node. Theorem 3.6 generalizes all past claims for NGNs with multiplicative compressors, any topology, and an arbitrary number of fixed-pressure nodes.

c2) Energy Function Minimization: Problem (3.7) dates back to [41], and has since been used for solving the GF task; verifying the feasibility and uniqueness of a GF instance [43]; and initializing optimization problems. However, its applicability was limited to NGNs without compressors. As explained in Remark 3.7, this work suggests using (3.7) to handle NGNs with compressors on non-overlapping cycles. We also present the unconstrained energy function formulation of (3.8).

c3) MI-QCQP relaxation of GF: The key difficulty in solving (optimal) GF problems stems from the non-linear Weymouth equation. A disjunctive convex relaxation of this equation was found to be efficient in [23], [104]. Numerous studies have thereon employed similar convex relaxations; see [15], [106], [30]. Unfortunately, it is hard to guarantee the exactness of these relaxations. An effective heuristic is to fix the binary variables involved to the values obtained by the convex relaxation and handle the resultant non-convex nonlinear program through a general solver [23], [15]. A gas-electric flow problem was solved in [30], wherein a cost function was proposed that was numerically found useful towards attaining exact relaxation. Unlike previous works, the MI-QCQP formulation of Section 3.7 provides theoretical guarantees for exact relaxation, while expanding the claims of [113]. The GF solver developed in [113] was applicable to NGN's with compressors not on cycles. However, in this work, the cost function of (G2) is meticulously designed to ensure that correct flows are obtained outside active cycles. Additionally, Algorithm 1 is developed to enable flow correction on active cycles efficiently. Although the GF problem is intrinsically simpler than the optimal gas (and possible electric) flow problem considered in prior works, this work lays a foundation towards analytical guarantees for exact relaxation. It has been recently shown that exact relaxation of network flow optimization problems may be guaranteed using a convex penalty [149]. It is worth mentioning that a related MI-QCQP formulation of the water flow problem in [110], can also provably yield an *exact* convex relaxation for the optimal water flow task [115].

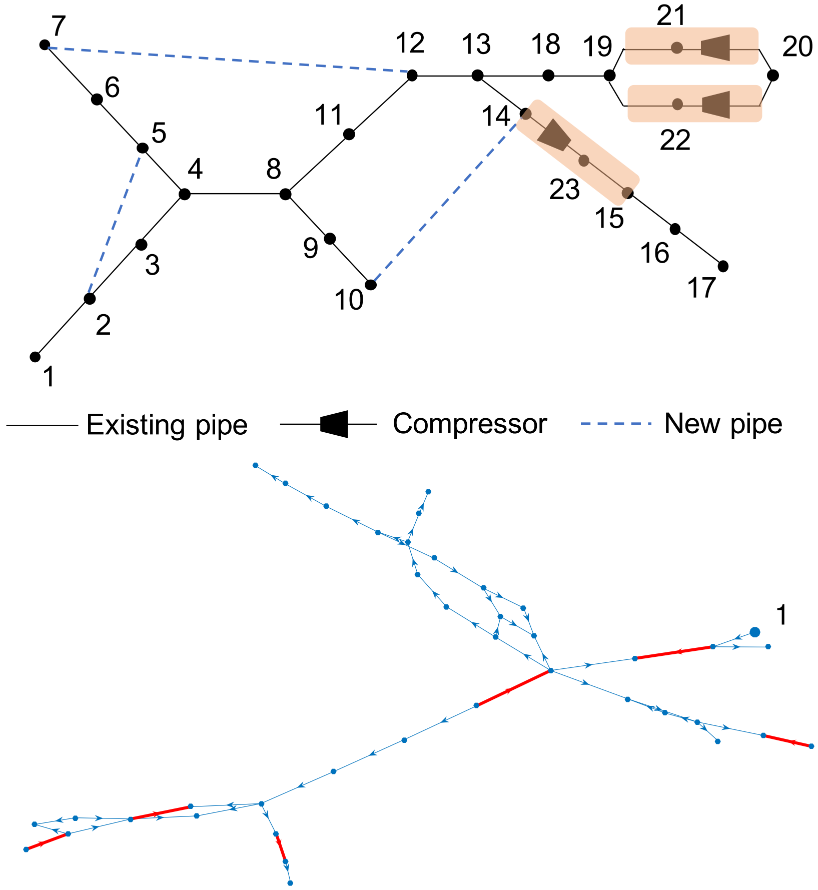

Figure 3.1: *Top:* Modified Belgian natural gas network. *Bottom*: GasLib-40 network with red edges representing compressors.

3.9 Numerical Tests

The proposed GF solver based on the relaxed MI-QCQP (G2) and Algorithm 1 was tested on the modified Belgian benchmark NGN and the GasLib-40 NGN of Fig. 3.1. Starting with the Belgian NGN, the pipe coefficients and compressor ratios were derived based on the nodal pressures and edge flows reported in [41]. The network contains three compressors, which are modeled as ideal compressors followed by lossy pipes. Problem (G2) was solved using the MATLAB-based optimization toolbox YALMIP using CPLEX as the MI-QCQP solver [78], [64]. All tests were conducted on a 2.7 GHz Intel Core i5 computer with 8 GB

RAM.

As a model validation step, we first tested the (G2) solver on the original Belgian network, which is a tree, except for one cycle formed by parallel compressors, see Fig. 3.1. The pressure at node 1 was treated as reference. The flow values obtained from (G2) agreed with those of [41] for all edges except for the edges along the active cycle. Similarly, the pressures agreed for all nodes other than node 20. Therefore, the pressure at node 19 and the flows on edges $(19, 21)$, $(19, 22)$, $(20, 21)$, $(20, 22)$ were passed to Algorithm 1 for correction. The final result was found to coincide with [41].

The Belgian network was subsequently augmented by additional pipelines; see Fig. 3.1. The resulting modified network has overlapping cycles, thus violating Condition 2 required in Theorem 3.8. To get reasonable friction coefficients, for every added line (m, n), the coefficient a_{mn} was set equal to the sum of $a_\ell' s$ along the $m - n$ path, yielding $a_{2,5} = 0.1936$, $a_{10,14} = 0.0439$, $a_{7,12} = 0.0419$. We kept the reference pressure at node 1 and the compression ratios constant as in [41], and drew $1,500$ random gas injections $\mathbf{q}$. To construct these samples, we perturbed the benchmark injections $\mathbf{q}_0$ that lie in the range $[-15.61, 22.01]$ by a standardized normal deviation. The injection at node 20 was set to the negative sum of the remaining injections to get $\mathbf{1}^\top\mathbf{q} = 0$ for all samples.

Using the modified meshed Belgian NGN of Fig. 3.1 and the random gas injections, we tested the exactness of (G2) and the performance of Algorithm 1. Not all of the random injections were feasible for the GF problem – some violated (3.4b) or (3.3b). Problem (G2) was infeasible for 876 out of the $1,500$ random instances. Since (G2) is a relaxation of (G1), these instances are apparently infeasible for (G1) too. The performance of (G2) and Algorithm 1 was tested on the remaining 624 gas injection instances. To evaluate the success

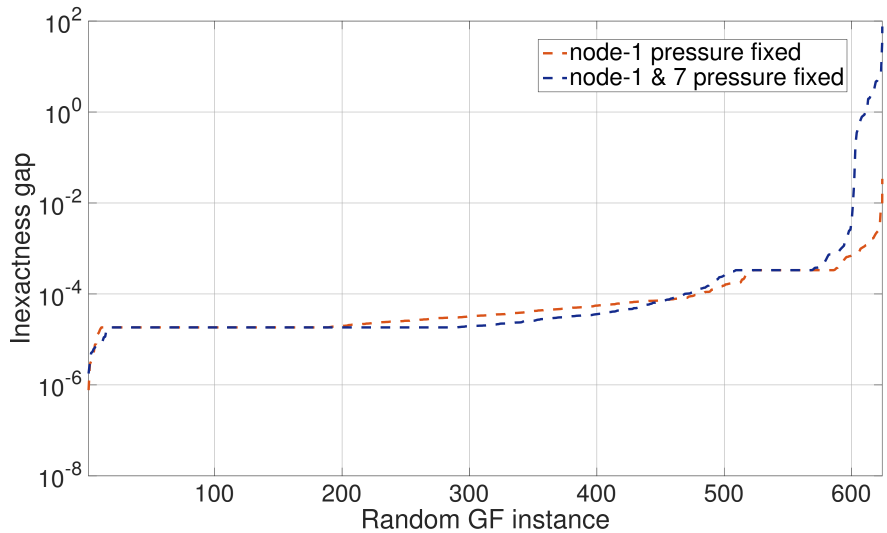

Figure 3.2: Inexactness gap attained by (G2) followed by Algorithm 1 over random feasible instances of the GF problem.

of (G2) in solving (G1), we calculated the *inexactness gap* G defined as

$$G := \max_{(m,n)\in\bar{\mathcal{P}}_a} \frac{|\psi_m - \psi_n| - a_{mn}\phi_{mn}^2}{a_{mn}\phi_{mn}^2} \geq 0$$

for the pressures and flows obtained by (G2) and Algorithm 1.

The ranked inexactness gap for the feasible GF instances is shown by the first curve in Fig. 3.2. The gap was less than 10^{-3} for more than 97% of the feasible instances, while the maximum gap over all instances was 0.009. This corroborates that the proposed solver performs well even when Condition 2 is not met. Fig. 3.3 shows the running time for solving (G2) and Algorithm 1 over the 624 feasible instances. The average (median) running time was 0.96 sec (0.89 sec).

Considering Condition 1, we used the fixed pressure at node 1 and the pressures obtained at node 7 for the feasible GF instances, we solved (G2) again. Although the hypothesis of Th. 3.8 does not hold anymore, the inexactness gap was found to be less than 10^{-3} for more than 94% of the instances; see the second curve in Fig. 3.2. Thus, the tests reveal that

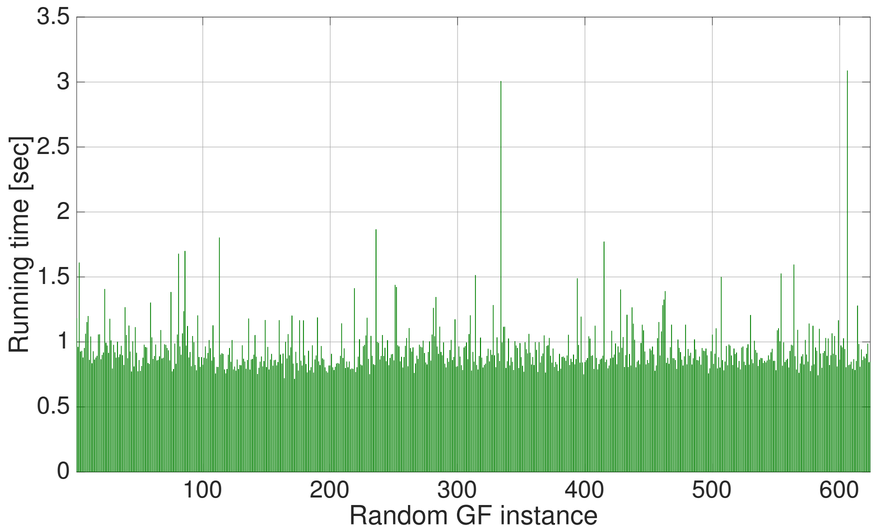

Figure 3.3: Running time for (G2) and Alg. 1 over random feasible GF instances.

the novel solver successfully finds the GF solution even when the sufficient Conditions 1–2 are violated. However, Condition 3 prohibiting gas circulations could not be violated for the Belgian NGN because the only active cycle in this NGN has parallel compressors, hence avoiding circulations from (3.4b). We next deal with GF instances on the GasLib-40 network, wherein a circulation could potentially occur.

GasLib-40 roughly represents a part of the German gas transport network [105]. The network exhibits 40 nodes, 39 pipes, and 6 compressors; see Fig. 3.1 (*Bottom*). The pipe dimensions, roughness coefficients, and a nominal demand vector $\mathbf{q}_0$ were derived from [105]. The goals for conducting additional tests on GasLib-40 include: *i)* Evaluating our solvers on a realistic setup; *ii)* Testing our MI-QCQP when Condition 3 is violated; and *iii)* Benchmarking the performance of our solvers against NR-based solver. We next briefly introduce the NR-based solver used for benchmarking. Given an injection $\mathbf{q}$, compressor ratios α_ℓ's, and reference pressure ψ_1, stack the unknowns as $\mathbf{y} = [\phi_1, \ldots, \phi_L, \psi_2, \ldots, \psi_N]^\top$. Define the equality constraints (3.2), (3.3a), and (3.4a) collectively as $\mathbf{g}(\mathbf{y}) = 0$. Given an initial estimate $\mathbf{y}_0$, the

NR-based solver would iterate as

$$\mathbf{y}_{t+1} = \mathbf{y}_t - \mu[\mathbf{J}(\mathbf{y}_t)]^{-1}\mathbf{g}(\mathbf{y}_t)$$

where t is the iteration count; matrix $\mathbf{J}(\mathbf{y}_t)$ is the Jacobian of $\mathbf{g}(\mathbf{y})$ evaluated at $\mathbf{y}_t$; and μ a step size. A solution $\mathbf{y}^\star$ obtained on convergence of NR updates would be deemed feasible if the inequalities (3.3b) and (3.4b) are satisfied. Since, the NR updates target at attaining $\mathbf{g}(\mathbf{y}) = \mathbf{0}$, the performance evaluation criteria for our results would be $\|\mathbf{g}(\mathbf{y})\|_2$ in lieu of the inexactness gap G.

In the first set of tests on GasLib-40, we generated 500 gas injection instances $\mathbf{q}$ by scaling the entries of $\mathbf{q}_0$ independently, by random factors chosen uniformly on $[0.75, 1.25]$. The pressure at node 1 was set to 50 bar and its injection was set to the negative sum of other nodes for all instances. Next, the compression ratios for the 6 compressors were drawn uniformly within $[1, 2]$. All 500 instances were solved using three approaches: *a1)* the MI-QCQP and Algorithm 1; *a2)* NR with flows initialized at $(\mathbf{A}^\top)^\dagger\mathbf{q}$, and all pressures initialized at ψ_1; and *a3)* NR with flows and pressures initialized at the solution of MI-QCQP and Algorithm 1. The stopping criteria for NR was set to $\|\mathbf{g}(\mathbf{y})\|_2 < 10^{-3}$, subject to a maximum iteration count of 50. The step size for both initialization scenarios was kept as $\mu = 1$. The MI-QCQP deemed 5 out of the 500 instances as infeasible and the performance criteria $\|\mathbf{g}(\mathbf{y})\|_2$ was found to lie in $[0.005, 0.183]$ with the median at 0.009. To compare to the index of inexactness gap, the range for G for the 495 feasible cases was $[8 \cdot 10^{-5},\ 6 \cdot 10^{-2}]$. Thus, the MI-QCQP alongside Algorithm 1 was successful in finding the GF solution for all 495 instances. Interestingly, 474 of the 495 feasible GF instances exhibit circulations, and hence violate Condition 3. Thus, the numerical results empirically demonstrate that the developed MI-QCQP alongside Algorithm 1 successfully solves the GF problem even when the conditions of Theorem 3.8 are violated. The NR solver, if initialized at the solution of

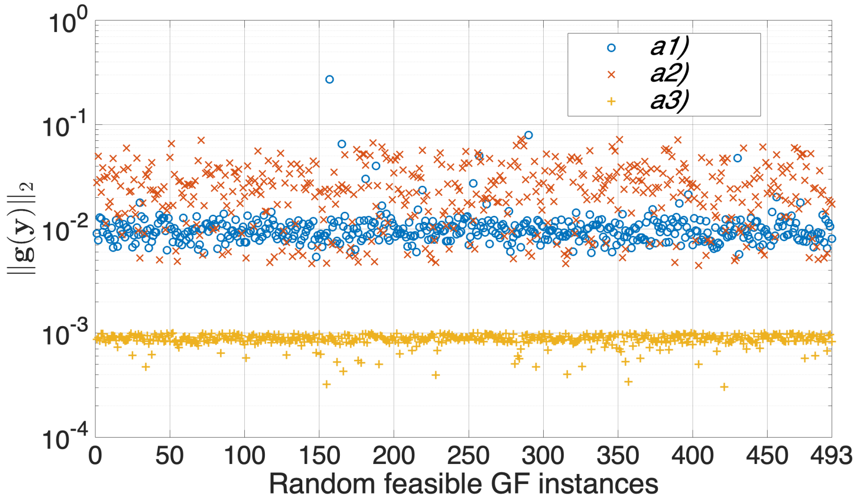

Figure 3.4: Accuracy measure $\|\mathbf{g}(\mathbf{y})\|_2$ for GF solutions obtained by MI-QCQP in (G2) followed by Algorithm 1, and GF solutions found by the Newton Raphson iterations for different initializations.

MI-QCQP improves the solution accuracy, resulting in $\|\mathbf{g}(\mathbf{y}^\star)\|_2$ within $1.2 \cdot 10^{-4} - 0.13$. For the 5 instances deemed infeasible by MI-QCQP, the NR solver was initialized at all zero flows and pressures; all 5 instances failed to converge. Surprisingly, when the NR solver was initialized with $(\mathbf{A}^\top)^\dagger\mathbf{q}$ as flows and $\psi_0's$ as pressures, all 500 instances failed to converge. The non-convergence of the NR solver is however alleviated when μ was reduced as discussed next.

A second set of tests were conducted on the GasLib-40 NGN with 500 random injections and compressor ratios generated as described earlier. The MI-QCQP solver deemed 7 of the 500 instances as infeasible. All 500 instances were then solved with the NR-based solver with flows initialized at $(\mathbf{A}^\top)^\dagger\mathbf{q}$ and pressures at ψ_1, and μ was set to $\mu = 0.9$. A steep decline in $\|\mathbf{g}(\mathbf{y}_t)\|_2$ was observed in the first few (roughly 10) iterations, while the tolerance of 10^{-3} was not attained within the 50 iterations limit. However, if the NR solver is initialized at the solution of MI-QCQP and Algorithm 1, the convergence criteria of 10^{-3} was attained at an average of 7.8 iterations. The values of $\|\mathbf{g}(\mathbf{y})\|_2$ attained by three solution techniques *a1)–a3)* are shown in Fig. 3.4. The results suggest that the accuracy of the MI-QCQP solver is better than that of *a2)*, which is a prudent initialization. However, if the NR-based solver is warm-started with the solution of MI-QCQP, an order of magnitude improvement in accuracy is observed. On the computational front, the MI-QCQP solver alongside Algorithm 1 is efficient with median solving time of 1.52 sec. However, as anticipated, the NR solvers have superior performance with median solving time of 0.17 sec. Finally, inspecting the 7 instances deemed infeasible by the MI-QCQP solver, the solution obtained by *a2)* indicates violation of (3.4b); demonstrating the merit of the proposed MI-QCQP towards certifying infeasibility of GF instances.

3.10 Conclusions

Exploiting recent results from graph theory and convex relaxations, this work provides a fresh perspective on the steady-state GF problem. The uniqueness of the GF solution has been established in a generalized setting for arbitrary NGN topologies, multiplicative compressors and multiple fixed-pressure nodes. Granted that the GF solution is unique, constrained and unconstrained versions of convex energy function minimization-based GF solvers have been proposed. These solvers can efficiently solve any GF task instance with a single fixed-pressure node and networks with compressors not on cycles. To expand the scope, an MI-QCQP GF solver had been also proposed relying on a convex relaxation of the Weymouth equation. The relaxation has been shown to be exact under specific network conditions. Numerical tests reveal that the developed MI-QCQP solver succeeds in finding the unique GF solution even when the needed conditions are violated. The success of the MI-QCQP relaxation is attributed to a judiciously designed objective. The developed approach sets forth an analytical platform for ensuring exact relaxation. Evaluating the performance of the developed approach for various optimal gas flow tasks constitutes an interesting research direction.

Appendix B

Proof of Lemma 3.4. For an edge $\ell_i \in \mathcal{P}_{mn}$, let us name the incident node closer to m as m_i, and the other node as m_{i+1}, as shown in Fig. 3.5.

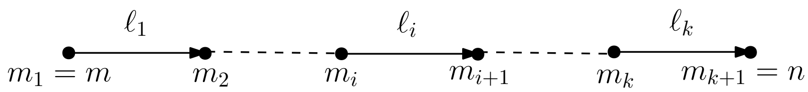

Figure 3.5: Nomenclature for nodes and edges along $\mathcal{P}_{mn}$.

Let $\boldsymbol{\psi}$ and $\boldsymbol{\psi}'$ be the pressure vectors corresponding to $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$. Since pressures ψ_m and ψ_n are fixed, it follows $\psi_m = \psi'_m$ and $\psi_n = \psi'_n$. Proving by contradiction, suppose (3.5a) holds. If that is the case, first it will be shown that $\psi'_{m_i} - \psi'_{m_{i+1}} > \psi_{m_i} - \psi_{m_{i+1}}$ for every lossy pipe $\ell_i \in \mathcal{P}_{mn}$.

Suppose that $\operatorname{sign}(\boldsymbol{\phi}' - \boldsymbol{\phi}) \cdot \boldsymbol{\pi}^{mn} > \mathbf{0}$. Let us denote the RHS of (3.3a) by $w(\phi_\ell)$. It is evident that $w(\phi_\ell)$ is monotonically increasing in ϕ_ℓ. Hence, for any lossy pipe $\ell_i \in \mathcal{P}_{mn}$, it holds

$$\begin{aligned} 0 &\overset{a}{<} \pi_{\ell_i}^{mn} \operatorname{sign}(\phi'_{\ell_i} - \phi_{\ell_i}) \\ &\overset{b}{=} \pi_{\ell_i}^{mn} \operatorname{sign}(w(\phi'_{\ell_i}) - w(\phi_{\ell_i})) \\ &\overset{c}{=} \operatorname{sign}(\pi_{\ell_i}^{mn}) \operatorname{sign}(w(\phi'_{\ell_i}) - w(\phi_{\ell_i})) \\ &\overset{d}{=} \operatorname{sign}(\pi_{\ell_i}^{mn} w(\phi'_{\ell_i}) - \pi_{\ell_i}^{mn} w(\phi_{\ell_i})) \\ &\overset{e}{=} \operatorname{sign}((\psi'_{m_i} - \psi'_{m_{i+1}}) - (\psi_{m_i} - \psi_{m_{i+1}})), \end{aligned} \tag{3.12}$$

where (a) holds by hypothesis; (b) stems from the monotonicity of $w(\phi_\ell)$; (c) holds because $\pi_{\ell_i}^{mn} \in \{0, 1, -1\}$; (d) holds from the property of sign by definition; and (e) from the definition of $\boldsymbol{\pi}^{mn}$ and (3.3a). The inequality (3.12) implies

$$\psi'_{m_i} - \psi'_{m_{i+1}} > \psi_{m_i} - \psi_{m_{i+1}}. \tag{3.13}$$

Let us now apply (3.13) and (3.4a) for the edges ℓ_1 to ℓ_k along $\mathcal{P}_{mn}$. For the fixed pressure node m, we have $\psi_m = \psi'_m$. If ℓ_1 is a lossy pipe, we get $\psi'_{m_2} < \psi_{m_2}$ from (3.13); otherwise $\psi'_{m_2} = \psi_{m_2}$ from (3.4a). Similarly, we can show that $\psi'_{m_3} \leq \psi_{m_3}$, where the equality holds only if both ℓ_1 and ℓ_2 are compressors. However, this is practically impossible as every compressor is modeled as an ideal compressor followed by a lossy pipe, necessitating $\psi'_{m_3} < \psi_{m_3}$. Continuing the process for all edges along $\mathcal{P}_{mn}$ yields $\psi'_n < \psi_n$, which contradicts with node n being a fixed-pressure node. Similarly, the assumption $\text{sign}(\boldsymbol{\phi}' - \boldsymbol{\phi}) \cdot \boldsymbol{\pi}^{mn} < 0$ leads to a contradiction by yielding $\psi'_n > \psi_n$. □

Proof of Lemma 3.5. Given the two pairs $(\mathbf{q}, \boldsymbol{\phi})$ and $(\mathbf{q}', \boldsymbol{\phi}')$ satisfying (3.2) and $\mathbf{q} \neq \mathbf{q}'$, let us define $\tilde{\boldsymbol{\phi}} := \boldsymbol{\phi}' - \boldsymbol{\phi}$ and $\tilde{\mathbf{q}} := \mathbf{q}' - \mathbf{q}$. By applying (3.2) on $(\mathbf{q}, \boldsymbol{\phi})$ and $(\mathbf{q}', \boldsymbol{\phi}')$, and taking the difference, we get

$$\mathbf{A}^\top \tilde{\boldsymbol{\phi}} = \tilde{\mathbf{q}}. \tag{3.14}$$

Since $\mathbf{1} \in \text{null}(\mathbf{A})$, premultiplying (3.14) by $\mathbf{1}^\top$ provides

$$\mathbf{1}^\top \tilde{\mathbf{q}} = \mathbf{0}. \tag{3.15}$$

From (3.14)–(3.15), the pair $(\tilde{\mathbf{q}}, \tilde{\boldsymbol{\phi}})$ qualifies as a set of balanced gas injections. By definition of $(\tilde{\mathbf{q}}, \tilde{\boldsymbol{\phi}})$, proving (3.6) is equivalent to showing there exists a path $\mathcal{P}_{mn}$ for which

$$\tilde{\boldsymbol{\phi}} \odot \boldsymbol{\pi}^{mn} > \mathbf{0} \tag{3.16a}$$

$$\tilde{q}_m > 0 \ \text{ and } \ \tilde{q}_n < 0. \tag{3.16b}$$

To prove the existence of such a path, we use the ensuing result based on [70, Th. 8.8].

Lemma 3.11 ([70]). *Given a graph with injection q at node s, demand q at node t, and zero injections at all other nodes, there exists an s-t path with flow directions along the path from s to t.*

Lemma 3.11 considers a single-source single-destination network flow setup. We transform our problem to this setup through the next steps; see also Fig. 3.6:

1) The nodes of graph $\mathcal{G}$ are partitioned into the subset with positive $\mathcal{N}_+ : \{n \in \mathcal{N} : \tilde{q}_n > 0\}$; negative $\mathcal{N}_- : \{n \in \mathcal{N} : \tilde{q}_n < 0\}$; and zero injections $\mathcal{N}_0 : \{n \in \mathcal{N} : \tilde{q}_n = 0\}$. Because $\tilde{\mathbf{q}} \neq \mathbf{0}$, the sets $\mathcal{N}_+$ and $\mathcal{N}_-$ are non-empty.

2) Augment $\mathcal{G}$ by adding nodes s and t.

3) All nodes in $\mathcal{N}_+$ are connected to node s, and all nodes in $\mathcal{N}_-$ are connected to node t.

4) The injections in $\mathcal{N}_+$ are lumped in node s by setting the flows $\tilde{\phi}_{sn} = \tilde{q}_n$ for all $n \in \mathcal{N}_+$. Similarly, the demands in $\mathcal{N}_-$ are lumped in node t by setting the flows $\tilde{\phi}_{nt} = -\tilde{q}_n$ for all $n \in \mathcal{N}_-$.

Applying Lemma 3.11 on this augmented graph, there exists a path $\mathcal{P}_{st}$ with flow directions from s to t. For any such path $\mathcal{P}_{st}$, eliminate the first and last edges to get a path $\mathcal{P}_{mn}$ with $m \in \mathcal{N}_+$ and $n \in \mathcal{N}_-$. Claim (3.16b) follows by construction. We next show (3.16a): For each edge $\ell \in \mathcal{P}_{mn}$, it was shown that the direction of $\tilde{\phi}_\ell$ is along the path $\mathcal{P}_{mn}$. If $\pi_\ell^{mn} = +1$, the direction of edge ℓ agrees with the direction of $\mathcal{P}_{mn}$. Since $\tilde{\phi}_\ell$ is along $\mathcal{P}_{mn}$, then $\tilde{\phi}_\ell > 0$. If $\pi_\ell^{mn} = -1$, the direction of edge ℓ is opposite to the direction of $\mathcal{P}_{mn}$. Since $\tilde{\phi}_\ell$ is along $\mathcal{P}_{mn}$, then $\tilde{\phi}_\ell < 0$. Either way, it holds that $\tilde{\phi}_\ell \pi_\ell^{mn} > 0$ for all $\ell \in \mathcal{P}_{mn}$, which proves (3.6a). $\square$

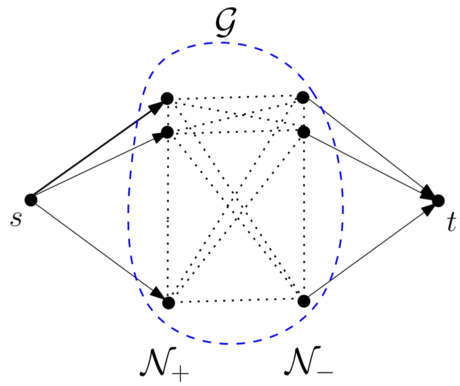

Figure 3.6: Augmented NGN graph.

Proof of Theorem 3.8. Before proving the main result, we will need two preliminary results.

Lemma 3.12. *For a lossy pipe $\ell = (m, n)$ not on an active cycle, if the triplet $(\psi_m, \psi_n, \phi_\ell)$ satisfies (3.11), then the triplet $(\psi_m + \delta, \psi_n + \delta, \phi_\ell)$ also satisfies (3.11) for any finite δ.*

Lemma 3.12 follows directly from the fact that (3.11) involves pressure differences rather than pressures.

Lemma 3.13. *Consider an active cycle $\mathcal{C}_0$ and index its nodes as $\{0, \ldots, k\}$. Given a fixed pressure ψ_0 and flows $\{\phi_\ell\}_{\ell \in \mathcal{C}_0}$ satisfying Condition 3 and (3.4b), there exists a set of pressures $\{\psi_i\}_{i=1}^k$ satisfying (3.11) and (3.4a).*

Proof. From Condition 3 and the fact that a compressor is modeled as an ideal compressor followed by a lossy pipe, it is not hard to see that there must exist a node $k \in \mathcal{C}_0$ that leads to one of the four flow scenarios shown in Fig. 3.7.

Proving by construction, we will next define pressures $\{\psi_i\}_{i=1}^k$ such that (3.11) and (3.4a) are satisfied for all edges in $\mathcal{C}_0$. Traversing the paths $0 \to k_1$ and $0 \to k_2$, one can recursively define pressures for all nodes using ψ_0 and flows $\{\phi_\ell\}_{\ell \in \mathcal{C}_0}$ based on the exact Weymouth equation (3.3) and (3.4a). The pressures on the remaining nodes of $\mathcal{C}_0$ can be defined for

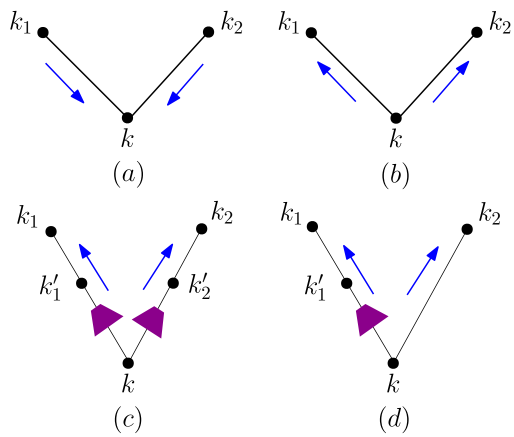

Figure 3.7: Four possible scenarios for a cycle with non-circulating gas flow. The arrows represent the actual gas flow directions.

the four scenarios of Fig. 3.7 as follows:

$$
\begin{aligned}
&(a)\ \psi_k := \min\{\psi_{k_1} - a_{k_1k}\phi^2_{k_1k}, \psi_{k_2} - a_{k_2k}\phi^2_{k_2k}\} \\
&(b)\ \psi_k := \max\{\psi_{k_1} + a_{kk_1}\phi^2_{kk_1}, \psi_{k_2} + a_{kk_2}\phi^2_{kk_2}\} \\
&(c)\ \psi_k := \max\left\{\frac{\psi_{k_1} + a_{k_1'k_1}\phi^2_{k_1'k_1}}{\alpha_{kk_1'}}, \frac{\psi_{k_2} + a_{k_2'k_2}\phi^2_{k_2'k_2}}{\alpha_{kk_2'}}\right\} \\
&\quad\ \psi_{k_1'} := \alpha_{kk_1'}\psi_k, \quad \psi_{k_2'} := \alpha_{kk_2'}\psi_k \\
&(d)\ \psi_k := \max\left\{\frac{\psi_{k_1} + a_{k_1'k_1}\phi^2_{k_1'k_1}}{\alpha_{kk_1'}}, \psi_{k_2} + a_{kk_2}\phi^2_{kk_2}\right\} \\
&\quad\ \psi_{k_1'} := \alpha_{kk_1'}\psi_k.
\end{aligned}
$$

To see that the constructed pressures satisfy (3.11), take for example scenario (a). Applying (3.11) along the edges (k_1, k) and (k_2, k) yield that ψ_k should satisfy $\psi_k \leq \psi_{k_1} - a_{k_1k}\phi^2_{k_1k}$ and $\psi_k \leq \psi_{k_2} - a_{k_2k}\phi^2_{k_2k}$. This is indeed the case by selecting ψ_k as the minimum of the two RHS. Similar reasoning applies to the other scenarios. □

Proceeding with the proof of Theorem 3.8, let $(\boldsymbol{\phi}, \boldsymbol{\psi})$ be the unique solution to (G1), and

$(\boldsymbol{\phi}', \boldsymbol{\psi}')$ a minimizer of (G2). Proving by contradiction, assume that there exists an edge ℓ not belonging to an active cycle, such that $\phi'_\ell \neq \phi_\ell$. Recall that the set of all active cycles is denoted by $\mathcal{S}^a_\mathcal{C}$. Since both flow vectors satisfy (3.2), their difference $\mathbf{n} := \boldsymbol{\phi} - \boldsymbol{\phi}'$ must lie in the nullspace of $\mathbf{A}^\top$. The nullspace of $\mathbf{A}^\top$ is spanned by the indicator vectors for all fundamental cycles in the gas network graph [57, Corollary 14.2.3]. Therefore, the entries of $\mathbf{n}$ related to edges not on a cycle must be zero. Since by hypothesis $\ell \notin \mathcal{S}^a_\mathcal{C}$, edge ℓ should belong to one of the cycles in $\mathcal{S}_\mathcal{C} \setminus \mathcal{S}^a_\mathcal{C}$. This non-active cycle will be henceforth termed $\mathcal{C}$.

The rest of the proof is organized in three parts: Part I constructs a flow vector $\hat{\boldsymbol{\phi}}$ that satisfies (3.2) and (3.4b). Part II shows there exists a $\hat{\boldsymbol{\psi}}$ so that the pair $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (G2). Part III shows that $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ attains a smaller objective for (G2), thus contradicting the optimality of $(\boldsymbol{\phi}', \boldsymbol{\psi}')$.

Part I: Define the flow vector $\hat{\boldsymbol{\phi}}$ as

$$\hat{\phi}_\ell := \begin{cases} \phi_\ell, & \ell \in \mathcal{C} \\ \phi_\ell, & \ell \text{ belongs to any active cycle} \cdot \\ \phi'_\ell, & \text{otherwise} \end{cases} \tag{3.17}$$

By construction, vector $\hat{\boldsymbol{\phi}}$ satisfies

$$\boldsymbol{\phi}' - \hat{\boldsymbol{\phi}} = \lambda \mathbf{n}^\mathcal{C} + \mathbf{n}^a \tag{3.18}$$

where $\mathbf{n}^\mathcal{C}$ is the indicator vector for cycle $\mathcal{C}$; the constant λ is nonzero; and vector $\mathbf{n}^a \in \text{null}(\mathbf{A}^\top)$ can have nonzero entries only for edges in active cycles. Since $\boldsymbol{\phi}'$ satisfies constraint (3.2) and $\mathbf{A}^\top \mathbf{n}^\mathcal{C} = \mathbf{A}^\top \mathbf{n}^a = \mathbf{0}$, then $\mathbf{A}^\top \hat{\boldsymbol{\phi}} = \mathbf{A}^\top \boldsymbol{\phi}' = \mathbf{q}$. This proves that $\hat{\boldsymbol{\phi}}$ satisfies (3.2). Note that $\hat{\boldsymbol{\phi}}$ is constructed by selecting entries from $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$. Granted both $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ satisfy

(3.4b), vector $\hat{\boldsymbol{\phi}}$ trivially satisfies (3.4b) too.

Part II: We will delineate the steps for constructing a vector of pressures $\hat{\boldsymbol{\psi}}$ such that $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (G2). Let us select a spanning tree $\mathcal{T}$ of the NGN graph $\mathcal{G}$ rooted at the reference r. We shall define the pressures $\hat{\psi}_n$'s while traversing $\mathcal{T}$ via depth-first search. In such a traversal, the following three cases may be identified on arriving at any node n:

Case 1: Node n is neither in $\mathcal{C}$ nor on an active cycle. Let $n-1$ be the parent node of n in $\mathcal{T}$ and define

$$\hat{\psi}_n := \begin{cases} \alpha_{n-1,n}\hat{\psi}_{n-1} & , \text{ if } (n-1,n) \in \mathcal{P}_a \\ \hat{\psi}_{n-1} + (\psi'_n - \psi'_{n-1}) & , \text{ if } (n-1,n) \in \bar{\mathcal{P}}_a \end{cases}.$$

Since the edge $(n-1,n)$ is not in $\mathcal{C} \cup \mathcal{S}^a_{\mathcal{C}}$, we have $\hat{\phi}_{n-1,n} = \phi'_{n-1,n}$ from (3.17). Therefore, if $(n-1,n)$ is a lossy pipe, Lemma 3.12 ensures that the defined pressure $\hat{\psi}_n$ satisfies (3.11). Moreover, if $(n-1,n)$ is a compressor, constraint (3.4a) is satisfied trivially by definition.

Case 2: Node n is in $\mathcal{C}$. If n is the first node in $\mathcal{C}$ to be visited, define $\hat{\psi}_n$ as in *Case 1*. Then, define the pressures for the remaining nodes $i \in \mathcal{C}$ as $\hat{\psi}_i := \psi_i + (\hat{\psi}_n - \psi_n)$. Note from (3.17) that the flows along $\mathcal{C}$ are assigned from $\boldsymbol{\phi}$, the pair $(\boldsymbol{\phi}, \boldsymbol{\psi})$ satisfies (3.3) and hence the relaxed Weymouth (3.11) as well. The constructed pressures $\hat{\psi}_i$'s for $i \in \mathcal{C}$ are simply a shifted version of the pressures ψ_i's. Therefore, the pressures $\hat{\psi}_i$'s satisfy (3.11) from Lemma 3.12. Mark all nodes in $\mathcal{C}$ as traversed and continue.

Case 3: Node n is in an active cycle $\mathcal{C}_a$. If n is the first node in $\mathcal{C}_a$ to be traversed, define the $\hat{\psi}_n$ as in *Case 1*. Then, define the pressure for the remaining nodes $i \in \mathcal{C}_a$ using Lemma 3.13. Mark all nodes in $\mathcal{C}_a$ as traversed and continue.

Since the constructed pressures satisfy (3.11) and (3.4a), the pair $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}})$ is feasible for (G2). Observe that the pressure drop across lossy pipes not in $\mathcal{C}$ is $\hat{\psi}_m - \hat{\psi}_n = \psi'_m - \psi'_n$ for *Case 1*; and $\hat{\psi}_m - \hat{\psi}_n = \psi_m - \psi_n$ for lossy pipes in $\mathcal{C}$ under *Case 2*. This fact is imperative for the ensuing Part III.

Part III: We will next show that $r(\boldsymbol{\psi}') > r(\hat{\boldsymbol{\psi}})$ to contradict the optimality of $\boldsymbol{\psi}'$. Note that the objective $r(\boldsymbol{\psi})$ in (G2) sums up the absolute pressure differences along lossy pipes, but not on active cycles. Since by construction these differences have changed only along $\mathcal{C}$, we get

$$r(\boldsymbol{\psi}') - r(\hat{\boldsymbol{\psi}}) = \sum_{(m,n)\in\mathcal{C}} |\psi'_m - \psi'_n| - |\hat{\psi}_m - \hat{\psi}_n|. \tag{3.19}$$

As the pressure differences depend on flows, we next compare the entries of $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}'$ along $\mathcal{C}$ using (3.18). Since the edge directions are assigned arbitrarily, assume wlog that $\hat{\phi}_{mn} \geq 0$ for all $(m,n) \in \mathcal{C}$. Given $\mathbf{n}_{\mathcal{C}}$ and (3.18), one can find the value of λ. If $\lambda < 0$, reverse the reference direction for cycle $\mathcal{C}$ to get a positive λ. Because of this, we can assume $\lambda > 0$.

Recall that $\mathbf{n}_{\mathcal{C}} \in \{0, \pm 1\}^P$. Partition the set of edges in $\mathcal{C}$ into mutually exclusive sets $\hat{\mathcal{P}}_+$ and $\hat{\mathcal{P}}_-$ based on positive and negative entries of $\mathbf{n}_{\mathcal{C}}$, respectively. From (3.18), it follows

$$0 \leq \hat{\phi}_\ell < \phi'_\ell, \quad \forall \ell \in \hat{\mathcal{P}}_+. \tag{3.20}$$

Summing up the pressure drops along $\mathcal{C}$ for $\hat{\boldsymbol{\psi}}$ should be zero. Since the pressure drops along $\mathcal{C}$ are positive for the edges in $\hat{\mathcal{P}}_+$, and negative along the edges in $\hat{\mathcal{P}}_-$, it holds that

$$\begin{aligned} \sum_{(m,n)\in\hat{\mathcal{P}}_+} (\hat{\psi}_m - \hat{\psi}_n) &= \sum_{(m,n)\in\hat{\mathcal{P}}_-} (\hat{\psi}_m - \hat{\psi}_n) \\ \implies \quad \sum_{(m,n)\in\mathcal{C}} |\hat{\psi}_m - \hat{\psi}_n| &= 2 \sum_{(m,n)\in\hat{\mathcal{P}}_+} (\hat{\psi}_m - \hat{\psi}_n) \end{aligned} \tag{3.21}$$

where the absolute value is trivial since $\hat{\phi}_{mn} \geq 0$ for all $(m,n) \in \mathcal{C}$.

Drawing similar relations on $\boldsymbol{\psi}'$, define the set $\mathcal{P}'_+ \subset \mathcal{C}$ containing any edge $(m,n) \in \mathcal{C}$ such that the flow ϕ'_{mn} is along the direction of $\mathbf{n}_{\mathcal{C}}$. Using the same argument as in (3.21) for $\boldsymbol{\psi}'$, we obtain

$$\sum_{(m,n)\in\mathcal{C}} |\psi'_m - \psi'_n| = 2 \sum_{(m,n)\in\mathcal{P}'_+} (\psi'_m - \psi'_n). \tag{3.22}$$

Because the flows in $\hat{\boldsymbol{\phi}}$ for the edges in $\hat{\mathcal{P}}_+$ are aligned with $\mathbf{n}_\mathcal{C}$ and $\phi'_\ell > \hat{\phi}_\ell$ for these edges from (3.20), it follows that $\hat{\mathcal{P}}_+ \subseteq \mathcal{P}'_+$. Using the latter in (3.22), we get

$$\begin{aligned} 2 \sum_{(m,n)\in\hat{\mathcal{P}}_+} (\psi'_m - \psi'_n) &\leq 2 \sum_{(m,n)\in\mathcal{P}'_+} (\psi'_m - \psi'_n) \\ &= \sum_{(m,n)\in\mathcal{C}} |\psi'_m - \psi'_n|. \end{aligned} \tag{3.23}$$

For every edge $\ell = (m,n) \in \hat{\mathcal{P}}_+$, it holds that

$$\hat{\psi}_m - \hat{\psi}_n \overset{(a)}{=} a_\ell \hat{\phi}_\ell^2 \overset{(b)}{<} a_\ell \phi_\ell'^2 \overset{(c)}{\leq} \psi'_m - \psi'_n \tag{3.24}$$

where (a) comes from the definition of pressures in *Case 2* of Part II; (b) descends from $\phi'_\ell > \hat{\phi}_\ell > 0$; and (c) from (3.11). Summing (3.24) over all $\ell \in \hat{\mathcal{P}}_+$ and multiplying by 2 gives

$$\begin{aligned} 2 \sum_{(m,n)\in\hat{\mathcal{P}}_+} (\hat{\psi}_m - \hat{\psi}_n) &< 2 \sum_{(m,n)\in\hat{\mathcal{P}}_+} (\psi'_m - \psi'_n) \\ \implies \sum_{(m,n)\in\mathcal{C}} |\hat{\psi}_m - \hat{\psi}_n| &< \sum_{(m,n)\in\mathcal{C}} |\psi'_m - \psi'_n| \end{aligned}$$

where the inequality stems from (3.22) and (3.23). From (3.19), the latter implies that $r(\boldsymbol{\psi}') > r(\hat{\boldsymbol{\psi}})$, hence contradicting the optimality of $\boldsymbol{\psi}'$. $\square$

Chapter 4

On the Flow Problem in Water Distribution Networks: Uniqueness and Solvers

4.1 Publication Details

M. K. Singh and V. Kekatos, "On the flow problem in water distribution networks: Uniqueness and solvers," *IEEE Trans. Control of Network Systems*, vol. 8, no. 1, pp. 462–474, Mar. 2021.

4.2 Abstract

Increasing concerns on the security and quality of water distribution systems (WDS), call for computational tools with performance guarantees. To this end, this work revisits the physical laws governing water flow and provides a hierarchy of solvers of complementary value. Given the water injection or pressure at each WDS node, finding the water flows within pipes and pumps along with the pressures at all WDS nodes constitutes the water flow (WF)

problem. The latter entails solving a set of (non)-linear equations. We extend uniqueness claims on the solution to the WF equations in setups with multiple fixed-pressure nodes and detailed pump models. For networks without pumps, the WF solution is already known to be the minimizer of a convex function. The latter approach is extended to networks with pumps but not in cycles, through a stitching algorithm. For networks with non-overlapping cycles, a provably exact convex relaxation of the pressure drop equations yields a mixed-integer quadratically-constrained quadratic program (MI-QCQP) solver. A hybrid scheme combining the MI-QCQP with the stitching algorithm can handle WDS with overlapping cycles, but without pumps on them. Each solver is guaranteed to converge regardless of initialization, as numerically validated on a benchmark WDS.

4.3 Introduction

Water distribution systems serve as a critical infrastructure across the world. The direct dependence of human lives on the availability of water has motivated research on the security, resiliency, and quality of water supply systems [82], [103]. The high cost of installation for different WDS components renders long-term network planning an important problem [107], [40]. Furthermore, the relatively expensive operation of a WDS is primarily attributed to the electricity cost for running pumps to properly circulate water [128]. Thus, optimal pump scheduling for the daily operation of a WDS is a pertinent research problem [52], [53], [115], [132]. An inevitable component of the aforementioned computational problems is satisfying the physical laws governing water flow. Mathematically, the *water flow (WF) equations* consist of a set of linear equations ensuring mass conservation, along with a set of non-linear equations arising from energy and momentum conservation [125].

Solving the WF equations constitutes the *water flow problem*. Specifically, given water demand at all nodes, the standard WF task aims at finding the water flows within all pipes and the pressures at all nodes complying with the WF equations. Despite nonlinear, these equations enjoy a unique solution for networks with a single fixed-pressure node and when the pressure added by pumps is approximated as constant [54]. Modern renditions of the WF problem may incorporate pumps, valves, and pressure-based demands [68], [102]. Either way, handling the non-linear equations stemming from the conservation of energy and momentum remain the core challenge [125].

Existing WF solvers update iteratively a set of WF variables, which could be the pipe flows, loop flows, nodal pressures, or combinations thereof [126]. These solvers can be broadly classified into those relying on successive linear approximations, and those relying on Newton-Raphson-type of updates [125]. For example, the WF solver of [13] constitutes a fixed-point iteration and belongs to the former class, while EPANET (perhaps the most widely used WF solver) to the latter [102]. In fact, most of the schemes within each class have been shown to be equivalent to each other upon a (non)-linear transformation of variables [126], [125]. To cope with the problem of dimensionality, different preprocessing, partitioning, and reformulations have been reported in [47], [4], [48].

The aforesaid solvers exhibit two major shortcomings. First, their convergence and rate of convergence depend critically on initialization. For example, it has been numerically demonstrated that EPANET fails to find a WF solution for some practical water networks [143], [49]. Nonetheless, proper initialization may be challenging when dealing with stochastic planning or risk analysis, where the WF task has to be solved repeatedly and under varying demands [4]. As a second shortcoming, the existing solvers do not naturally extend to optimal water flow (OWF) formulations. Therefore, most OWF efforts resort to linear approximations; non-linear local optimization; or slow zero-order algorithms building on an independent WF solver such as EPANET; see [82]. Such OWF approaches lack scalability and/or optimality guarantees.

The contribution of this work is in two fronts. After a brief modeling of water networks (Section 4.4), this work first generalizes the uniqueness of the WF solution to WDS setups with multiple fixed-pressure nodes and more practical pump models with flow-dependent pressure gains. Second, it puts forth a suite of solvers that can provably recover the WF solution under different network setups; see also Fig. 4.1:

i) In networks without pumps, it is already known that the WF solution can be recovered as the minimizer of a convex problem [54]. Sections 4.6.1 devises SOCP- and dual decomposition-based solvers, while Section 4.6.3 applies them to other pressure drop laws.

ii) This energy function-based approach is extended to networks where pumps do not lie on cycles through the *stitching algorithm* of Section 4.6.2.

iii) In networks with no overlapping cycles, the mixed-integer quadratically-constrained quadratic program (MI-QCQP) of Section 4.7 can find the WF solution.

iv) A hybrid procedure combining the stitching algorithm and the MI-QCQP solver handles networks having overlapping cycles, but without pumps on them (Section 4.8).

These novel solvers not only operate under different network configurations, but constitute a hierarchy: Solver *ii)* builds on *i)*; and solver *iv)* builds on *ii)* and *iii)*. Numerical tests on benchmark networks evaluate the correctness and running times for *i)* and *iii)*; and validate them against the EPANET solver (Section 4.9).

Regarding notation, lower- (upper-) case boldface letters denote column vectors (matrices). Calligraphic symbols are reserved for sets. The vectors of all zeros, all ones, and the n-th canonical vector are denoted respectively by $\mathbf{0}$, $\mathbf{1}$, and $\mathbf{e}_n$, while their dimension will be clear from the context. The symbol $^\top$ stands for transposition.

4.4 Water Distribution System Modeling

A WDS can be represented by a directed graph $\mathcal{G} := (\mathcal{N}, \mathcal{P})$. Its nodes are indexed by $n \in \mathcal{N} := \{1, \ldots, N\}$ and correspond to water reservoirs, tanks, and points of water demand. Let d_n be the rate of water injected into the WDS from node n. For reservoirs apparently $d_n \geq 0$; for nodes with water consumers $d_n \leq 0$; tanks may be filling or emptying; and $d_n = 0$ for junction nodes.

The edges in set $\mathcal{P}$ with cardinality $P := |\mathcal{P}|$, are associated with pipes and pumps. The directed edge $p = (m, n) \in \mathcal{P}$ models the water pipe between nodes m and n. Its water flow is denoted by f_{mn} or f_p depending on the context. If water flows from node m to n, then $f_{mn} \geq 0$; otherwise $f_{mn} < 0$.

Conservation of water flow dictates that for all $n \in \mathcal{N}$

$$d_n = \sum_{k:(n,k)\in\mathcal{P}} f_{nk} - \sum_{k:(k,n)\in\mathcal{P}} f_{kn}.$$

The connectivity of the WDS is captured by the edge-node incidence matrix $\mathbf{A} \in \mathbb{R}^{P\times N}$ with entries

$$A_{p,k} = \begin{cases} +1, & k = m \\ -1, & k = n \\ 0, & \text{otherwise} \end{cases} \quad \forall\ p = (m, n) \in \mathcal{P}. \tag{4.1}$$

Given $\mathbf{A}$ and upon stacking flows and injections respectively in $\mathbf{f} \in \mathbb{R}^P$ and $\mathbf{d} \in \mathbb{R}^N$, the conservation of water flow across the WDS can be compactly expressed as

$$\mathbf{A}^\top \mathbf{f} = \mathbf{d}. \tag{4.2}$$

The operation of WDS is also governed by pressures. Water pressure is surrogated by *pressure head*, defined as the equivalent height of a water column in meters, which exerts the surrogated pressure at its bottom. The pressure heads at all WDS nodes are measured with respect to a common geographical elevation level. The pressure head (henceforth *pressure*) at node n is denoted by h_n. Moreover, the pressure h_r at a reservoir node $r \in \mathcal{N}$ typically serves as the pressure of reference.

With water flowing in a pipe, pressure drops along the direction of flow due to friction. The pressure drop across pipe $(m, n) \in \mathcal{P}$ is described by the Darcy-Weisbach law [129]

$$h_m - h_n = c_{mn} \operatorname{sign}(f_{mn}) f_{mn}^2 \tag{4.3}$$

where the constant c_{mn} depends on pipe dimensions [115], and the sign function is defined as

$$\operatorname{sign}(x) := \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}.$$

The pressures at all nodes are collected in vector $\mathbf{h} \in \mathbb{R}^N$.

To maintain pressures at desirable levels, water utilities use pumps on specific pipes. Let $\mathcal{P}_a \subset \mathcal{P}$ be the subset of pipes hosting a pump. The pipes in $\mathcal{P}_a$ can be considered lossless; this is without loss of generality since a pump can be modeled by an ideal pump followed by a short pipe. The remaining edges form the subset $\bar{\mathcal{P}}_a := \mathcal{P} \setminus \mathcal{P}_a$, and correspond to lossy pipes governed by (4.3). When pump $p = (m, n) \in \mathcal{P}_a$ is running, it adds pressure $g_{mn} \geq 0$ so that

$$h_n - h_m = g_{mn}.$$

As detailed below, the pressure added by a pump decreases with the water flow through the pump [37], [127]. Moreover, for variable speed pumps, the pressure added increases with the pump speed. The exact relation is provided by manufacturers in the form of pump operation curves, and are oftentimes approximated with quadratic curve fits [37], [127], [53]. In detail, the pressure added by pump $p = (m, n)$ is modeled as

$$g_{mn}(f_{mn}, \omega_{mn}) = \lambda_{mn} f_{mn}^2 + \mu_{mn}\omega_{mn} f_{mn} + \nu_{mn}\omega_{mn}^2 \tag{4.4}$$

where ω_{mn} is the pump speed; and $\lambda_{mn} < 0$, $\mu_{mn} \geq 0$, and $\nu_{mn} \geq 0$ are known pump parameters [53]. When pump (m, n) is running, its flow has to be maintained within the range $0 \leq \underline{f}_{mn} \leq f_{mn} \leq \bar{f}_{mn}$ due to engineering limitations [129]. Whereas the speed for fixed-speed pumps is constant; for variable-speed pumps, it is controlled by the operator. Either way, for a WF task the speed of every pump is given; otherwise the problem would become under-determined. Given its speed ω_{mn}^0, the pressure added by pump (m, n) is

$$g_{mn}(f_{mn}; \omega_{mn}^0) = h_n - h_m = \lambda_{mn} f_{mn}^2 + \bar{\mu}_{mn} f_{mn} + \bar{\nu}_{mn} \tag{4.5}$$

where $\bar{\mu}_{mn} := \mu_{mn}\omega_{mn}^0$ and $\bar{\nu}_{mn} := \nu_{mn}(\omega_{mn}^0)^2$. Because the pump parameters satisfy $2\lambda_{mn}\underline{f}_{mn} + \mu_{mn}\omega_{mn} < 0$ over the operating range of pump speeds ω_{mn}, the pressure gain due to any pump is a *strictly decreasing function* of the water flow in the range $[\underline{f}_{mn}, \bar{f}_{mn}]$. This observation is instrumental in establishing the uniqueness of the WF solution in Section 4.5.

When pump $p = (m, n) \in \mathcal{P}_a$ is not running, water can flow freely in either directions through a bypass valve [37]. Because bypass valve sections are typically short, one can ignore the pressure drop along them to get $h_m = h_n$ and $g_{mn} = 0$. In this case, the WDS graph can be reduced by removing pipe p and node n, and connecting to node m the edges previously incident to n. Alternatively, the valve can be modeled as a short lossy pipe, whose pressure drop is governed by (4.3).

Valves constitute a vital component for water flow control. They can be modeled by an on/off switch; a linear pressure-reducing model; a flow-dependent non-linear model; or a flow control model [53], [102]. Under the typical operational setup, the valve at the reference node regulates pressure, whereas the valves at the remaining reservoirs and tanks regulate flows.

Summarizing this section, a WDS operating point is described by the triplet $(\mathbf{d}, \mathbf{f}, \mathbf{h})$ satisfying the WF equations of (4.2), (4.3), and (4.5). This work deals with the uniqueness of a WF solution and efficient solvers for finding this solution.

4.5 Uniqueness of Water Flow Solution

Different from the OWF problem where tanks and pumps are scheduled over a time horizon, the WF task aims at solving the WF equations given the water injections $\mathbf{d}$. As such, it constitutes a key component of WDS operation and planning. The WF problem is formally stated next.

Definition 4.1. Given: i) water injections $\mathbf{d}$; ii) the statuses and speeds $\{\omega_{mn}\}$ for all pumps $(m, n) \in \mathcal{P}_a$; and iii) the pressure h_r at the reference node $r \in \mathcal{N}$; the WF task aims at finding the flows $\mathbf{f}$ and pressures $\mathbf{h}$ satisfying (4.2), (4.3), (4.5).

The WF task involves $N + P - 1$ equations over $N + P - 1$ unknowns. The water balance in (4.2) yields $N - 1$ linearly independent equations. In addition, the pressure drops across lossy pipes [cf. (4.3)], and the pressure gains due to pumps [cf. (4.5)] provide P non-linear equations.

According to Definition 4.1, both $\mathbf{f}$ and $\mathbf{h}$ are unknown. Nevertheless, to find a triplet $(\mathbf{d}, \mathbf{f}, \mathbf{h})$ that satisfies (4.2), (4.3), and (4.5), it suffices to find either $\mathbf{f}$ or $\mathbf{h}$. To see this, note that if $\mathbf{f}$ is known, then $\mathbf{h}$ can be calculated from (4.3)–(4.5) and h_r. On the other hand, if $\mathbf{h}$ is known, the flows $\mathbf{f}$ can be found thanks to the monotonicity of (4.3) and the monotonicity of (4.5) in $[\underline{f}_{mn}, \bar{f}_{mn}]$. In a nutshell, solving the WF task amounts to finding either $\mathbf{f}$ or $\mathbf{h}$. Because this simple observation is used throughout our analysis, it is summarized as a lemma.

Lemma 4.2. *Given* $\mathbf{d}$*, a triplet* $(\mathbf{d}, \mathbf{f}, \mathbf{h})$ *satisfying* (4.2)*,* (4.3)*, and* (4.5) *is uniquely characterized by* $\mathbf{f}$ *or* $\mathbf{h}$*.*

The WF task can be posed as the feasibility problem

$$\begin{aligned} \text{find} \quad & \{\mathbf{f}, \mathbf{h}\} \\ \text{s.to} \quad & (4.2), (4.3), (4.5). \end{aligned} \tag{W1}$$

Since (4.3)–(4.5) are quadratic equalities, problem (W1) is non-convex. For a tree WDS graph $\mathcal{G}$ (for which $P = N - 1$), the edge-node incidence matrix $\mathbf{A}$ has $N - 1$ linearly independent rows [57]. Then the flows $\mathbf{f}$ can be found uniquely from (4.2), and the WF task is readily solved according to Lemma 4.2. However, handling the WF task in a loopy $\mathcal{G}$ containing pumps remains non-trivial. Naturally, analyzing the uniqueness of a WF solution is a critical task. Reference [54] establishes the uniqueness of a WF solution under the assumption that the pressure gain added by a pump is constant. The next claim extends this uniqueness result regardless of the structure of $\mathcal{G}$ and to the more detailed pump model of (4.5).

Theorem 4.3. *If the WF equations are feasible for some* $\mathbf{d}$*, they feature a unique solution.*

Proof. Proving by contradiction, assume $(\mathbf{h}, \mathbf{f})$ and $(\tilde{\mathbf{h}}, \tilde{\mathbf{f}})$ are two distinct solutions of (W1). Since the two flow vectors $\mathbf{f}$ and $\tilde{\mathbf{f}}$ satisfy (4.2), the vector $\mathbf{n} := \tilde{\mathbf{f}} - \mathbf{f}$ lies in the nullspace of $\mathbf{A}^\top$, that is $\mathbf{A}^\top \mathbf{n} = \mathbf{0}$. Then it follows that

$$\mathbf{n}^\top \mathbf{A}(\tilde{\mathbf{h}} - \mathbf{h}) = 0. \tag{4.6}$$

Let us decompose $\mathbf{n}$ into its positive and negative entries as $\mathbf{n} = \mathbf{n}_+ - \mathbf{n}_-$, where $\mathbf{n}_+ \geq \mathbf{0}$ and $\mathbf{n}_- \geq \mathbf{0}$. Plugging this decomposition into (4.6) yields

$$\mathbf{n}_+^\top \left(\mathbf{A}\tilde{\mathbf{h}} - \mathbf{A}\mathbf{h}\right) = \mathbf{n}_-^\top \left(\mathbf{A}\tilde{\mathbf{h}} - \mathbf{A}\mathbf{h}\right). \tag{4.7}$$

Recall from (4.3) that the pressure drop across a lossy pipe is monotonically increasing in flow. Similarly, the pressure added by a pump described by (4.5) is monotonically decreasing in flow. In other words, the pressure drop along a pump is monotonically increasing in flow. Hence, for each edge $p \in \mathcal{P}$:

$$\mathbf{a}_p^\top \tilde{\mathbf{h}} > \mathbf{a}_p^\top \mathbf{h} \quad \text{if and only if} \quad \tilde{f}_p > f_p \tag{4.8}$$

where $\mathbf{a}_p^\top$ is the p-th row of matrix $\mathbf{A}$.

Consider the p-th entry of vector $\mathbf{n}$. If $\tilde{f}_p > f_p$, then $n_{+,p} > 0$ and $n_{-,p} = 0$ by definition of $\mathbf{n}_+$ and $\mathbf{n}_-$. Then edge p contributes to the left-hand side (LHS) of (4.7). The monotonicity in (4.8) entails that $\mathbf{a}_p^\top(\tilde{\mathbf{h}} - \mathbf{h}) > 0$ and so $n_{+,p} \cdot \mathbf{a}_p^\top(\tilde{\mathbf{h}} - \mathbf{h}) > 0$. The latter holds for all edges contributing to the LHS of (4.8), so that

$$\mathbf{n}_+^\top \left(\mathbf{A}\tilde{\mathbf{h}} - \mathbf{A}\mathbf{h}\right) > 0.$$

On the other hand, if $\tilde{f}_p < f_p$, then $n_{+,p} = 0$ and $n_{-,p} > 0$. Then, edge p contributes to the right-hand side (RHS) of (4.8). The monotonicity in (4.8) entails $n_{-,p} \cdot \mathbf{a}_p^\top(\tilde{\mathbf{h}} - \mathbf{h}) < 0$. Applying the claim over all edges participating in the RHS of (4.8) provides

$$\mathbf{n}_-^\top \left(\mathbf{A}\tilde{\mathbf{h}} - \mathbf{A}\mathbf{h}\right) < 0.$$

The signs of the LHS and RHS contradict the equality in (4.7), thus proving the claim. $\square$

Regarding the pressure drop law of (4.3), the Hazen-Williams equation is sometimes used wherein the flow f_{mn} is raised to the exponent of 1.852. This exponent is different from the exponent of 2 in the Darcy-Weisbach equation; see for example [13]. While the Darcy-Weisbach equation is a theoretical formula, the Hazen-Williams equation is based on curve fitting of experimental data [40]. Either way, the uniqueness argument of Theorem 4.3 holds for any positive exponent on the water flow f_{mn} involved in the pressure drop equation of (4.3).

The WF problem posed in Definition 4.1 follows the most traditional setup pursued in WF literature [54], [38], [85]. However, with multiple sources feeding a WDS an alternate form of WF becomes pertinent. Specifically, multiple supply nodes may operate as fixed-pressure nodes while the remaining nodes act as fixed-injection nodes [133]. The ensuing result extends the uniqueness of a WF solution to this more general setting.

Proposition 4.4. *Given: i) pressures h_i for i in a subset of nodes $\mathcal{N}_r \subset \mathcal{N}$; ii) injections d_i for all $i \in \mathcal{N} \setminus \mathcal{N}_r$; and iii) the statuses and speeds for all pumps; the set of equations* (4.2), (4.3), *and* (4.5) *has at most one solution.*

Proof outline. Consider two pairs $(\mathbf{d}, \mathbf{f})$ and $(\mathbf{d}', \mathbf{f}')$ satisfying (4.2). If $\mathbf{d} = \mathbf{d}'$, the uniqueness of the WF solution follows from Theorem 4.3. Otherwise, [114, Lemma 3] dictates that there

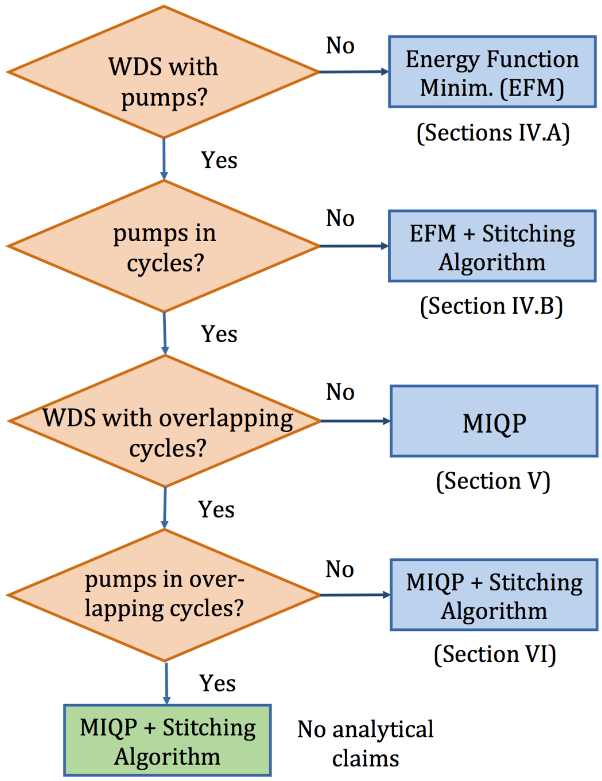

Figure 4.1: Given a WDS, this flowchart suggests the most suitable WF solver based on its analytical guarantees and computational complexity.

exists a path $\mathcal{P}_{mn}$ between fixed-pressure nodes m and n, along which the flows in $\mathbf{f}$ differ from those in $\mathbf{f}'$ in a consistent direction, i.e., either $f_p > f'_p$ or $f_p < f'_p$ for all edges $p \in \mathcal{P}_{mn}$. However, in that case, the monotonicity argument used in (4.8) would entail for the respective pressures $h_m - h_n \neq h'_m - h'_n$, which contradicts the hypothesis of m and n being fixed-pressure nodes. $\square$

Having established the uniqueness of the WF solution, the ensuing sections develop a suite of WF solvers of complementary value: Each solver finds provably the WF solution under different network setups. The devised solvers exhibit different complexity, yet none of them requires a proper initialization. Figure 4.1 summarizes the particular WDS setups each solver can handle, and serves as a roadmap for the following sections. The solvers developed in this work are for the WF task as posed in Definition 4.1, whereas WF solvers for the setting considered in Proposition 4.4 are beyond the scope of this work.

4.6 Energy Function-Based Water Flow Solvers

For a WDS without pumps, the WF task simplifies to solving (4.2)–(4.3). These equations are structurally similar to the equations governing the flows and pressures in a natural gas network under steady-state conditions [114], [113]. Reference [38] finds the solution to the nonlinear WF equations in networks without pumps as the minimizers of a convex *energy function*; see also [85] and [41] for counterparts in gas networks. Upon briefly reviewing these reformulations, this section expands their scope to WDS with no pumps in cycles and other pressure drop laws; see roadmap of Figure 4.1.

4.6.1 Finding the WF Solution in WDS without Pumps

In the absence of pumps, the WF problem (W1) has been posed as a constrained minimization over $\mathbf{f}$ [38].

Lemma 4.5 ([38])**.** *In a WDS without pumps ($\mathcal{P} = \bar{\mathcal{P}}_a$), the vector of water flows* $\mathbf{f}$ *satisfying* (4.2)–(4.3) *can be found as the unique minimizer of the convex minimization problem*

$$\min_{\mathbf{f}} \quad \frac{1}{3}\sum_{p\in\mathcal{P}} c_p|f_p|^3 \tag{4.9a}$$

$$\text{s.to} \quad \mathbf{A}^\top\mathbf{f} = \mathbf{d}. \tag{4.9b}$$

Moreover, if $\boldsymbol{\xi}^*$ is the vector of optimal Lagrange multipliers corresponding to (4.9b)*, then the nodal pressures are provided by* $\mathbf{h} = (\mathbf{I} - \mathbf{1}\mathbf{e}_r^\top)\boldsymbol{\xi}^* + h_r\mathbf{1}$.

The claim follows from the optimality conditions of (4.9). The strict convexity of (4.9a) shows that the WF solution is unique in networks *without pumps*. This claim has also been established using a contraction argument in [13]. Theorem 4.3 and Proposition 4.4 generalize this uniqueness claim to networks having pumps and multiple fixed-pressure nodes.

We next present two ways for handling (4.9): an second-order cone program (SOCP) reformulation, and a dual decomposition approach. It is known that ℓ_p-norms can be handled using a hierarchy of second-order cone (SOC) constraints; see [3, Eq. (11)]. In the context of WDS, a more compact hierarchy of SOCs was originally put forth in [52] to handle the cubic powers of (4.9a) involved in an OWF problem. Adopting the latter reformulation, the minimizer of (4.9) can be found by solving the SOCP

$$\min_{\{f_p, w_p, y_p, t_p\}} \quad \frac{1}{3}\sum_{p\in\mathcal{P}} c_p t_p \tag{4.10a}$$

$$\text{s.to} \quad \mathbf{A}^\top \mathbf{f} = \mathbf{d}. \tag{4.10b}$$

$$-w_p \leq f_p \leq w_p, \qquad \forall p \tag{4.10c}$$

$$w_p^2 \leq y_p,\ y_p^2 \leq w_p t_p, \qquad \forall p. \tag{4.10d}$$

The fact that the f_p's minimizing (4.9) and (4.10) coincide can be established by showing that the solution of (4.10) satisfies $w_p = |f_p|$; $y_p = w_p^2 = f_p^2$; and $t_p = w_p^3 = |f_p|^3$ for all $p \in \mathcal{P}$; see also [52]–[53] for details. Each constraint in (4.10d) is a rotated second-order cone [3].

Alternatively, problem (4.9) can be tackled via dual decomposition: During its k-th iteration, the primal variable $\mathbf{f}$ is updated by minimizing the Lagrangian function $L(\mathbf{f}; \boldsymbol{\xi}^k)$ evaluated at the latest estimate of dual variables $\boldsymbol{\xi}^k$. The latter problem decouples across pipes and enjoys a closed-form solution as

$$f_p^{k+1} := \arg\min_{f_p} \frac{c_p|f_p|^3}{3} - f_p \mathbf{a}_p^\top \boldsymbol{\xi}^k = \operatorname{sign}(\mathbf{a}_p^\top \boldsymbol{\xi}^k)\sqrt{\frac{|\mathbf{a}_p^\top \boldsymbol{\xi}^k|}{c_p}}.$$

For a step-size $\mu > 0$, the dual variables are updated as

$$\boldsymbol{\xi}^{k+1} := \boldsymbol{\xi}^k + \mu\left(\mathbf{d} - \mathbf{A}^\top \mathbf{f}^{k+1}\right).$$

Again for WDS without pumps, the WF task can alternatively be posed as an *unconstrained* minimization over $\mathbf{h}$ [38]

$$\min_{\mathbf{h}} \ \frac{2}{3} \sum_{(m,n)\in\mathcal{P}} \frac{|h_m - h_n|^{\frac{3}{2}}}{\sqrt{c_{mn}}} - \mathbf{d}^\top \mathbf{h}. \tag{4.11}$$

The objective of (4.11) is convex (by composition rules) and differentiable. The n-th entry of its gradient $\mathbf{g}_u(\mathbf{h})$ is

$$[\mathbf{g}_u(\mathbf{h})]_n = \sum_{p=(m,n)\in\mathcal{P}} \operatorname{sign}(\mathbf{a}_p^\top \mathbf{h})\sqrt{\frac{|\mathbf{a}_p^\top \mathbf{h}|}{c_p}} - d_n.$$

Setting this gradient equal to zero yields the WF equations after eliminating $\mathbf{f}$ from (4.2)–(4.3). The ambiguity in pressures can be waived by shifting the minimizer of (4.11) to match the reference pressure h_r. Once the pressures $\mathbf{h}$ are found, the flow vector $\mathbf{f}$ can be retrieved by Lemma 4.2. Problem (4.11) is amenable to any first-order method for unconstrained optimization, such as the gradient descent iterations $\mathbf{h}^{k+1} := \mathbf{h}^k - \mu \mathbf{g}_u(\mathbf{h}^k)$ for a step-size $\mu > 0$, or accelerated variants.

4.6.2 Extension to WDS with no Pumps in Cycles

Since problems (4.9) and (4.11) cannot handle water networks with pumps, this section extends their applicability to networks with pumps, but not on cycles. This is accomplished by adopting the reduction technique of [101] to build what we term *stitching algorithm*:

S1) Remove the edges of $\mathcal{G}$ corresponding to pumps $\mathcal{P}_a$. The obtained graph contains $|\mathcal{P}_a|+1$ disconnected components $\mathcal{G}_c$ for $c=1,\dots,|\mathcal{P}_a|+1$.

S2) Replace each component $\mathcal{G}_c$ by a supernode, and connect the supernodes using the edges in $\mathcal{P}_a$ to create the supergraph $\mathcal{G}'$. Graph $\mathcal{G}'$ features a *tree structure.*

S3) Find the total water injection per supernode $\mathcal{G}_c$. Since $\mathcal{G}'$ is a tree, the water flows on $\mathcal{P}_a$ can be found readily.

S4) If the flow along pump $(m,n)\in\mathcal{P}_a$ is f_{mn}, modify the injections at nodes m and n as $\hat{d}_m = d_m - f_{mn}$ and $\hat{d}_n = d_n + f_{mn}$.

S5) Solve (4.9) per connected component $\mathcal{G}_c$ to find the water flows at all lossy pipes.

S6) Having acquired vector $\mathbf{f}$, the pressure vector $\mathbf{h}$ can be found from Lemma 4.2.

In *S5)*, rather than solving the constrained minimization of (4.9), one could use its unconstrained counterpart of (4.11). Steps *S1)–S4)* are still needed to find a meaningful water injection vector per component $\mathcal{G}_c$. Once a pressure vector has been found per component, the pressures must be revised as follows: The pressures within the component containing the reference node, say component $\mathcal{G}_1$, are kept unaltered. Consider a pump running from node $m\in\mathcal{G}_1$ to node $n\in\mathcal{G}_2$. Using the flow f_{mn} computed in step *S3)*, the pressure gain $g_{mn}=g_{mn}(f_{mn};\omega^0_{mn})$ can be found from (4.5). Having found h_m by solving (4.11) in $\mathcal{G}_1$, the pressure at node n can be computed as $h_n = h_m + g_{mn}$. If the pressure at node n recovered by solving (4.11) in $\mathcal{G}_2$ is $\tilde{h}_n$, then all pressures within $\mathcal{G}_2$ should be shifted by $h_n - \tilde{h}_n$. The process is repeated for all components to recover the entire pressure vector $\mathbf{h}$.

4.6.3 Handling the Hazen-Williams Pressure Drop Law

The aforementioned WF solvers can be modified to handle the Hazen-Williams in lieu of the Darcy-Weisbach pressure drop equations. Recall that according to the Hazen-Williams pressure drop law, the exponent of $|f_{mn}|$ in (4.3) changes from 2 to $\rho = 1.852$. To handle this, problem (4.9) is generalized to

$$\min_{\mathbf{f}} \quad \frac{1}{\rho+1}\sum_{p\in\mathcal{P}} c_p|f_p|^{\rho+1} \tag{4.12a}$$

$$\text{s.to} \quad \mathbf{A}^\top \mathbf{f} = \mathbf{d} \tag{4.12b}$$

where the exponent ρ depends on the pressure drop law and could vary per pipe. Problem (4.12) remains convex for $\rho > 0$.

Even though the SOCP model of (4.10) cannot be used to solve (4.12), a different SOCP reformulation can be derived as long as ρ is rational; see [3, pp. 12–13] for details. Moreover, the dual decomposition updates of Section 4.6.1 can be adapted to solve (4.12). Similarly, the unconstrained minimization of (4.11) can be modified to

$$\min_{\mathbf{h}} \frac{\rho}{\rho+1}\sum_{(m,n)\in\mathcal{P}} \frac{|h_m - h_n|^{\frac{\rho+1}{\rho}}}{c_{mn}^{1/\rho}} - \mathbf{d}^\top\mathbf{h}. \tag{4.13}$$

Heed that the energy function-based solvers for (W1) can not handle the WF setups with multiple fixed-pressure nodes described under Proposition 4.4. This is because energy function-based solvers do not take reference pressure as an input to the problem. Rather, the obtained pressures are shifted *a posteriori* to agree with a given fixed pressure. For a single fixed-pressure node, the applicability of our WF solvers can be broadened to WDS with pumps in cycles. To do so, we next pursue an MI-QCQP solver. Unlike the energy function-based WF solvers of this section, the MI-QCQP solver of Section 4.7 applies only to the Darcy-Weisbach equation; its possible extension to the Hazen-Williams pressure drop law goes beyond the scope of this work.

4.7 MI-QCQP Water Flow Solver

This section presents an MI-QCQP relaxation of (W1) along with sufficient conditions for its exactness.

4.7.1 Problem Reformulation

The non-convexity of the WF problem (W1) is due to the quadratic equality constraints of (4.3) and (4.5). Adopting the convex relaxation approach of [53]–[52], the pressure drop along pipe $(m, n) \in \bar{\mathcal{P}}_a$ is relaxed from (4.3) to

- $h_m - h_n \geq c_{mn} f_{mn}^2$ for $f_{mn} \geq 0$; or
- $h_n - h_m \geq c_{mn} f_{mn}^2$ for $f_{mn} \leq 0$.

To handle the two cases, let us introduce a binary variable x_{mn} capturing the flow direction on pipe (m, n). By using the so-termed big-M trick, the two cases can be modeled as

$$-M(1 - x_{mn}) \leq f_{mn} \leq M x_{mn} \tag{4.14a}$$

$$-M(1 - x_{mn}) \leq h_m - h_n - c_{mn} f_{mn}^2 \tag{4.14b}$$

$$h_m - h_n + c_{mn} f_{mn}^2 \leq M x_{mn} \tag{4.14c}$$

$$x_{mn} \in \{0, 1\}. \tag{4.14d}$$

The same MI-QCQP model has been also advocated for handling the pressure drop equations along the pipelines in natural gas networks; see e.g., [23], [104], [114]. Constraint (4.14a) implies that $x_{mn} = \text{sign}(f_{mn})$. If $x_{mn} = 1$, the convex quadratic constraint in (4.14b) is activated, whereas constraint (4.14c) becomes trivial. The claim reverses for $x_{mn} = 0$. Depending on the value x_{mn}, if either (4.14b) or (4.14c) is satisfied with equality, the relaxation is deemed as *exact*. If that happens for all lossy pipes, the feasible set of (4.14) has captured the original non-convex constraints in (4.3).

Similarly, the pressure added by pump $(m, n) \in \mathcal{P}_a$ is relaxed from (4.5) to

$$h_n - h_m \leq \lambda_{mn} f_{mn}^2 + \hat{\mu}_{mn} f_{mn} + \hat{\nu} \tag{4.15}$$

which is a convex constraint since $\lambda_{mn} < 0$. Again, if the inequality in (4.15) is satisfied with equality for all pumps, the relaxation of (4.5) to (4.15) is deemed as exact.

One may now try solving the feasibility problem in (W1) by replacing (4.3)–(4.5) by (4.14)–(4.15). If all the relaxed constraints are satisfied with equality, the relaxation is *exact*. Unfortunately, toy numerical tests can verify that such relaxation is inexact even for simple water networks. To favor exact relaxations, we convert the feasibility to a minimization problem with a judiciously selected cost. We pose the MI-QCQP

$$\begin{aligned} \min_{\mathbf{f},\mathbf{h},\mathbf{x}} \quad & s(\mathbf{h}) \\ \text{s.to} \quad & (4.2), (4.14), (4.15) \end{aligned} \tag{W2}$$

where the binary vector $\mathbf{x}$ contains all $\{x_{mn}\}_{(m,n)\in\bar{\mathcal{P}}_a}$, and the cost is defined as

$$s(\mathbf{h}) := \sum_{(m,n)\in\bar{\mathcal{P}}_a} |h_m - h_n| - \sum_{(m,n)\in\mathcal{P}_a} (h_n - h_m).$$

The cost sums up the absolute pressure drops across lossy pipes minus the pressure gains added by pumps. This is in contrast to the MI-QCQP of [113] that recovers a gas flow solution, whose penalty did not include compressors. It will be shown in Section 4.7.2 that the MI-QCQP formulation (W2) guarantees exactness under realistic network conditions.

Problem (W2) is non-convex due to the binary variables $\mathbf{x}$. Given the advancements in MI-QCQP solvers, this minimization can be handled for moderately sized networks. Recall that every lossy pipe is associated with one binary and one continuous optimization variable. To accelerate the computations, we next provide a simple pre-processing step to determine the flows in all edges not belonging to a cycle.

Lemma 4.6. *Let $\mathbf{f}$ be the unique flow solution of (W1), and consider any vector $\tilde{\mathbf{f}}$ satisfying $\mathbf{A}^\top\tilde{\mathbf{f}} = \mathbf{d}$. For any edge (m,n) not belonging to a cycle, it holds that $\tilde{f}_{mn} = f_{mn}$.*

Proof. Consider the minimum-norm solution $\mathbf{f}_0 := \left(\mathbf{A}^\top\right)^\dagger \mathbf{d}$ to the linear system of (4.2), where $\left(\mathbf{A}^\top\right)^\dagger$ is the pseudo-inverse of $\mathbf{A}^\top$. Any other solution of (4.2) can be expressed as $\mathbf{f} = \mathbf{f}_0 + \mathbf{n}$ for some vector $\mathbf{n} \in \text{null}(\mathbf{A}^\top)$.

The space null$(\mathbf{A}^\top)$ can be represented using the cycles of graph $\mathcal{G}$ as explained next. For cycle $\mathcal{C}$ in $\mathcal{G}$, select an arbitrary direction and define its indicator vector $\mathbf{n}_{\mathcal{C}} \in \mathbb{R}^P$ as

$$n_{\mathcal{C},p} := \begin{cases} 0 & \text{, if edge } p \notin \mathcal{C} \\ +1 & \text{, if the directions of } p \text{ and } \mathcal{C} \text{ coincide} \\ -1 & \text{, otherwise.} \end{cases}$$

We are particularly interested in a set of *fundamental cycles* defined as follows [57]: In graph $\mathcal{G} = (\mathcal{N}, \mathcal{P})$, select a spanning tree $\mathcal{T}$. Every edge $p \in \mathcal{P} \setminus \mathcal{T}$ along with some edges of $\mathcal{T}$ form a cycle. The cycles formed using all edges in $\mathcal{P} \setminus \mathcal{T}$ comprise the set $\mathcal{L}_{\mathcal{T}}$ of fundamental cycles. In general, a graph may have multiple spanning trees, making the set of fundamental cycles non-unique.

A key property of fundamental cycles is that the space $\text{null}(\mathbf{A}^\top)$ is spanned by the indicator vectors for any set of fundamental cycles [57, Corollary 14.2.3]. Therefore, the vector $\mathbf{n}$ in $\mathbf{f} = \mathbf{f}_0 + \mathbf{n}$ can be decomposed as

$$\mathbf{n} = \sum_{\ell \in \mathcal{L}_{\mathcal{T}}} \alpha_\ell \mathbf{n}_\ell \tag{4.16}$$

where $\mathbf{n}_\ell$ is the indicator vector for cycle ℓ and $\alpha_\ell \in \mathbb{R}\ \forall \ell$.

Consider the p-th entry of $\mathbf{n}$. If edge $p \in \mathcal{P}$ does not belong to any cycle, then it does not belong to any fundamental cycle. Then $n_{\ell,p} = 0$ for all $\ell \in \mathcal{L}_{\mathcal{T}}$, and $n_p = 0$ follows from (4.16). To conclude, if edge p does not belong to a cycle, then all solutions $\mathbf{f}$ of (4.2) agree in their p-th entry. $\square$

Lemma 4.6 ensures that for weakly meshed WDS, the number of variables in (W2) can be reduced markedly. Moreover, the flow on any edge not belonging to a cycle coincides with the related entry of the minimum-norm solution to (4.2).

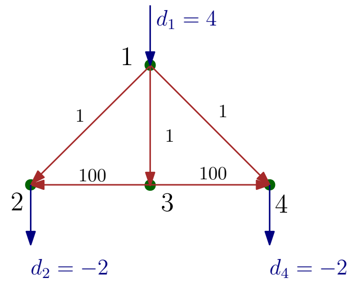

Figure 4.2: A pathological WDS for which the relaxation of (W2) is inexact.

4.7.2 Exactness of the Relaxation

The next result provides conditions under which a minimizer of (W2) satisfies (4.14)–(4.15) with equality.

Theorem 4.7. *In a WDS where no edge* $(m, n) \in \mathcal{P}$ *belongs to more than one cycle, a minimizer of* (W2) *minimizes* (W1) *as well, if* (W1) *is feasible.*

Theorem 4.7 (shown in the appendix) asserts that the MI-QCQP relaxation of (W1) to (W2) is exact in water networks with non-overlapping cycles, that is cycles sharing no edges. The claim holds regardless of water demand or the presence of pumps. This is in contrast to the exactness claims of [113], where compressors were not allowed on cycles. Although the assumptions of Theorem 4.7 may not always hold, the numerical tests of Section 4.9 indicate the relaxation is exact and (W2) solves (W1) for practical WDS with overlapping cycles.

The condition of Theorem 4.7 cannot be relaxed analytically: One could construct pathological WDS with overlapping cycles that render the relaxation of (W2) inexact. To present such a counterexample, consider the 4-node and 5-pipe WDS of Fig. 4.2. The coefficients c_{mn}'s are shown on the respective pipes. Nodes 2 and 4 host water demands, and the reference node 1 supplies water at the reference pressure of $h_1 = 10$. The edge $(1, 3)$ belongs to two cycles and hence the condition of Theorem 4.7 is violated. The minimizer of (W2) satisfies the relaxed constraint (4.14) with strict inequality. The pressures at nodes 1 and 3 were found to differ by $h_1 - h_3 = 3.435$, while the frictional drop was $c_{13}f_{13}^2 = 0.014$. This WDS though is characterized by a strong disparity of pipe coefficients. For a WDS with limited variations in pipe dimensions and material, such disparity is not anticipated in practice. Similar to the example of Fig. 4.2, one can construct setups with multiple fixed-pressure nodes for which the MI-QCQP relaxation is inexact.

Since the WDS of Fig. 4.2 does not host pumps, the solution to the previous WF problem was eventually found via (4.9). This motivates us to exploit the ability of the energy function-based approach to handle overlapping cycles alongside the merit of the MI-QCQP relaxation to handle pumps in cycles. To solve the WF problem for a broader class of WDS network topologies, a hybrid solver is deferred to Section 4.8. Before that, the formulation of (W2) is contrasted to prior work.

4.7.3 Comparison to Prior Work and Extensions

The proposed WF solver involves three ingredients:

I1) replacing (4.3) by the MIQP model of (4.14);

I2) replacing (4.4) by the convex constraint of (4.15); and

I3) solving (W2) in lieu of (W1).

The key contribution of this section is *I3)*, since *I1)* and *I2)* have appeared before: References [52]–[53] and [140] consider the OWF problem assuming known flow directions (say $f_{mn} \geq 0$ for all pipes), and relax (4.3) to [cf. (4.14)]

$$h_m - h_n \geq c_{mn}f_{mn}^2, \;\; f_{mn} \geq 0. \tag{4.17}$$

Regarding pumps, references [52]–[53] and [140] consider only variable-speed pumps. Moreover, each pump speed ω_{mn} is limited within $0 \leq \omega_{mn} \leq \bar{\omega}_{mn}$, which may be unrealistic since pumps come with positive lower speed bounds. Substituting this speed range into the monotonic formula of (4.5) yields the convex quadratic constraint

$$h_n - h_m \leq \lambda_{mn} f_{mn}^2 + \mu_{mn} \bar{\omega}_{mn} f_{mn} + \nu_{mn} \bar{\omega}_{mn}^2. \tag{4.18}$$

Heed that the actual speed ω_{mn} has been eliminated from (4.18). Nevertheless, once an OWF minimizer is found [and so a triplet (h_m, h_n, f_{mn}) satisfying (4.18)], the speed ω_{mn} making (4.18) an equality can be readily recovered [52].

References [52]–[53] aim at minimizing the energy losses across pipes. Ignoring the details of geographical elevation and electricity prices, their cost simplifies to $\sum_{(m,n)\in\bar{\mathcal{P}}_a} c_{mn} f_{mn}^3$. Thanks to its form, minimizing this cost subject to (4.17)–(4.18) renders the relaxation in (4.17) numerically exact. This was also analytically shown granted the simplifying assumptions of: *a1)* no pumps on cycles; *a2)* known flow directions; *a3)* if a node is fed by more than one pipes, all of them have to be equipped with valves; and *a4)* no fixed-speed pumps. Reference [140] considers the joint scheduling of WDS and electric power distribution networks. Since the exactness of (4.17) is not promoted by the cost anymore, a feasible point pursuit-based approach is applied to find a stationary point satisfying (4.17) with equality.

From the preceding review, it is evident that albeit a relaxation of (4.3) is convenient, its exactness is not always guaranteed. While some OWF instances feature exactness [52]–[53]; others call for additional measures [140]. We shall next address the natural question: *Q1)* Can one use one of the aforesaid OWF solvers to solve the WF task? Numerical tests on WF instances even in simple WDS demonstrate that the approaches of [52]–[53] and [140] fail to solve (W1), that is the convex relaxation is not exact. This is because even if an instance of (W1) satisfies all other conditions needed for each one of these solvers, the common assumption *a4)* is violated since pump speeds are fixed for (W1).

One may reverse question *Q1)* and pose *Q2)*: Can the developed WF solvers be used towards solving OWF tasks? One could identify two possible uses. First, the proposed tools can be used instead of EPANET in existing zero-order OWF algorithms that rely on a WF solver. Second, the success of the proposed penalized relaxation could be adopted in OWF tasks. For example, our previous work [115] deals with optimal pump scheduling under dynamic electricity pricing. This OWF task handles unknown flow directions and is relaxed to an MI-SOCP after $g_{mn}(f_{mn};\omega_{mn}^0)$ in (4.5) is approximated as constant. The relaxation is provably exact under certain conditions and after adding a penalty to the cost of the MI-SOCP. The relaxation is numerically exact under a broader range of WDS conditions. Interestingly, the penalty that worked for [115] does not work for the WF task studied here. For this reason, this work put forth the penalty $s(\mathbf{h})$ in (W2) and took a totally different route for establishing exactness. The sufficient conditions for exactness of Theorem 4.7 are significantly simpler compared to those in prior works on OWF.

4.8 A Hybrid Water Flow Solver

The hybrid WF solver relaxes the assumption of non-overlapping cycles of Theorem 4.7 to the following condition.

Assumption 3. The water network has no pumps in overlapping cycles.

The assumption permits overlapping cycles, but these particular cycles should carry no pumps. For a network satisfying Assumption 3, the WF task can be solved through the following steps illustrated also in Fig. 4.3:

T1) Any cycle $\mathcal{C}$ with pumps is non-overlapping. Thus, for any node n belonging to $\mathcal{C}$, two cases arise:

i) Node n belongs only to cycle $\mathcal{C}$ (node 3 of $\mathcal{C}_1$); or

ii) Node n belongs to other cycle(s) $\mathcal{C}'$ as well; yet $\mathcal{C}$ and $\mathcal{C}'$ share no edge (e.g., node 4 of $\mathcal{C}_2$).

Then, reduce graph $\mathcal{G}$ to $\mathcal{G}'$ through the steps:

- If for a cycle $\mathcal{C}$ having pumps, all nodes do not belong to any other cycle, replace $\mathcal{C}$ by a supernode $n_{\mathcal{C}}$.
- If for a cycle $\mathcal{C}$ having pumps, node n belongs to other cycle(s), identify the edges (n_1, n) and (n, n_2), which belong to $\mathcal{C}$. These two edges belong to no other cycles as $\mathcal{C}$ is non-overlapping. Split the node n into n and n' connected by a lossless edge (n, n'), such that all edges in $\mathcal{G}$ other than (n_1, n) and (n, n_2) that were incident on n are now incident on n'. After repeating this step for all nodes in $\mathcal{C}$ that belong to multiple cycles, replace $\mathcal{C}$ by a supernode $n_{\mathcal{C}}$.

This process ensures that in $\mathcal{G}'$ all supernodes and the pumps left out from $\mathcal{G}$ do not appear on cycles.

T2) Use Lemma 4.6 to compute the water flows on the edges $(m, n_{\mathcal{C}})$ incident to supernodes $n_{\mathcal{C}}$'s in $\mathcal{G}'$.

T3) For each edge $(m, n_{\mathcal{C}})$ in $\mathcal{G}'$, modify the injection at node m as $\hat{d}_m = d_m - f_{mn_{\mathcal{C}}}$.

T4) Partition $\mathcal{G}'$ into a set of connected components $\mathcal{G}_c$ by removing the supernodes and their incident edges. Each $\mathcal{G}_c$ has known injections and bears no pumps in cycles.

T5) Use the stitching algorithm of Section 4.6.2 per component $\mathcal{G}_c$ to find the flows within $\mathcal{G}_c$.

T6) Each supernode $n_{\mathcal{C}}$ is split back to the cycle $\mathcal{C}$ it replaced. If the edge $(m, n_{\mathcal{C}})$ of $\mathcal{G}'$ corresponded to edge (m, n) of $\mathcal{G}$, modify the injection at node $n \in \mathcal{C}$ as $\hat{d}_n = d_n + f_{mn_{\mathcal{C}}}$.

T7) Solve (W2) per cycle $\mathcal{C}$ to find the water flows on $\mathcal{C}$.

T8) Given vector $\mathbf{f}$, find the pressure vector $\mathbf{h}$ using Lemma 4.2.

Let us apply the previous steps on the 23-node and 28-pipe water network of Fig. 4.3. There are two cycles carrying pumps, marked as $\mathcal{C}_1$ and $\mathcal{C}_2$. Node 4 of cycle $\mathcal{C}_1$ belongs to multiple cycles and hence it is split in 4 and 4′. Next, removing $\mathcal{C}_1$ and $\mathcal{C}_2$ along with the edges that connect these cycles with the rest of the graph results in the connected subgraphs $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ shown on the bottom of Fig. 4.3. The demands on boundary nodes are modified as per steps *T3)* and *T6)*. The edge flows within each connected component are subsequently found by solving (4.9). The flows on $\mathcal{C}_1$ and $\mathcal{C}_2$ are finally found by (W2).

This WDS setup could not be handled by the energy function-based approach of Section 4.6 alone due to the presence of pumps on cycles $\mathcal{C}_1$ and $\mathcal{C}_2$. The convex relaxation of Section 4.7 alone is not guaranteed to succeed either, due to the presence of overlapping cycles. Combining the merits of each method and leveraging Lemma 4.6, this hybrid method can handle successfully this WDS setup.

4.9 Numerical Tests

The new WF solvers were evaluated under the different network configurations of Fig. 4.1. First, the new WF solvers were evaluated on the EPANET *Example Network-2* representing a water network from Cherry Hills, Connecticut shown in Fig. 4.4 [103]. This network consists of $P = 40$ pipes, $N = 34$ demand nodes, one tank, and one pump station. We modified the network by representing the pump station as a reservoir with pressure 100 ft connected to a fixed-speed pump. All nodes were assumed at the same elevation. The pipe friction coefficients c_{mn}'s, and base demand vector $\mathbf{d}$, were derived from the EPANET benchmark. Note that the WDS in Fig. 4.4 has overlapping cycles but the pump is not in a loop. Thus, it qualifies for being solved using the energy function method alongside the stitching algorithm of Section 4.6.2.

We tested whether the proposed solvers yield the same solution as EPANET. For this purpose, we obtained WF solutions using the constrained energy function minimization of (4.9) and the MI-QCQP of (W2). Problem (4.9) was solved using the closed-form dual decomposition steps of Section 4.6.1 for $\mu = 10^{-4}$ within $20,000$ iterations. The MI-QCQP in (W2) was solved using the MATLAB-based toolbox YALMIP along with the mixed-integer solver CPLEX [78], [64]. All tests were run on a 2.7 GHz, Intel Core i5 computer with 8 GB RAM. The flows obtained by the two solvers were very close to the EPANET solution, as illustrated in Fig. 4.5. EPANET uses more detailed flow models, e.g., the coefficient c_{mn} in (4.3) depends weekly on flow f_{mn}. The $20,000$ iterations for the primal-dual updates of (4.9) were completed within 4.5 sec, and the running time of the MI-QCQP (W2) was 1.6 sec.

We next evaluated the performance of the MI-QCQP-based solver of (W2) in finding the correct WF solution, when the conditions of Theorem 4.7 are not satisfied. Specifically, the network of Fig. 4.4 has overlapping cycles. To represent various demand levels, we generated 100 random WF instances by scaling the benchmark demand $\mathbf{d}$ by a scalar uniformly drawn from $[0, 1.5]$. Given a minimizer of (W2), we defined the inexactness over lossy pipe (m, n) as $|h_m - h_n| - c_{mn}f_{mn}^2$. For each run of (W2) with a random input, the ranked maximum inexactness gap over all lossy pipes is displayed on Fig. 4.6 (top). Despite violating the conditions of Th. 4.7, the inexactness gap was small for all random tests. Computationally, the running time of (W2) over the 100 instances with a median value of only 1.68 sec; see Fig. 4.6 (bottom).

Our WF solvers were subsequently tested on the 20-node synthetic network of Fig. 4.7 and under different demands. The WDS of Fig. 4.7 was created by combining two copies of the popular 10-node WDS representing the modified Arava Valley network from Israel; see [115], [37] for network data. The additional pipes $(1,2)$, $(7,14)$, and $(8,20)$ have the same dimensions as pipes $(4,5)$, $(5,6)$, and $(6,8)$, respectively. These were added to generate different network conditions. The parameters for all the pumps were derived by scaling the pump parameters of [53] by 0.25, and were set to $(\lambda_{mn}, \bar{\mu}_{mn}, \bar{\nu}_{mn}) = (-2.735 \cdot 10^{-5}, 0.0129, 55.83)$.

The performance of (W2) was tested for two network configurations derived from the WDS of Fig. 4.7: *C-i)* pipe $(4,10)$ removed; *C-ii)* pipe $(8,20)$ removed. Note that configuration *C-i)* has pumps in non-overlapping cycle along side other overlapping cycles not hosting pumps. Thus, the WF problem for *C-i)* can provably be solved using the hybrid WF solver of Section 4.8. Moreover, configuration *C-ii)* has pumps in overlapping cycles and thus cannot be provably solved even by the hybrid WF solver. However, we numerically tried (W2) in solving 100 random WF instances for both *C-i)* and *C-ii)*.

The 100 random WF instances were generated by fixing pressure $h_1 = 10$ m and drawing the remaining pressures independently as Gaussian random variables of mean 10 m and variance 2 m^2, or $\mathcal{N}(10,2)$. The on/off statuses of pumps were drawn as independent Bernoulli random variables with mean of 0.5. The flow for on pumps was drawn with uniform probability on the allowable pump flow $[250, 1500]$ m^3/hr. For off pumps, flows were drawn from $\mathcal{N}(0,200)$. The pressure added by pumps was then calculated using (4.5). For pump (m,n), the receiving node pressure was updated as $h_n := h_m + g_{mn}$. The water flow in all lossy pipes was calculated from the so obtained pressures and (4.3). Once the complete flow vector was obtained, the injections were computed from (4.2). The obtained vector of injections, pump on/off status, and pressure h_1 served as a feasible input for the WF task. Similarly 100 feasible random WF instances were generated for configuration *C-ii)*, and solved using (W2). The maximum time for solving (W2) was set to 1 min in CPLEX.

The WF task for both configurations was solved within a 1-minute deadline for 98 out of the 100 random instances with the average time being 1.01 and 1.06 sec for *C-i)* and *C-ii)*, respectively. The ranked maximum inexactness gap over all lossy pipes for *C-i)* and *C-ii)* is shown in Fig. 4.8. Although the conditions of Th. 4.7 were not satisfied by either configuration, the maximum inexactness obtained was less than 10^{-3} for 74 and 92 out of the 98 instances of *C-i)* and *C-ii* that were solved. For the two cases where the MI-QCQP failed to converge within the 1-min deadline, we modified the big-M parameter value from $M = 300$ to $M = 80$. The result was that the two instances were solved in 0.13 and 0.82 seconds with the inexactness gap being less than 10^{-5}.

To evaluate whether the status of the pumps has any significant effect on runtime, we conducted the previous tests on the WDS of Fig. 4.7 with all the pumps running. The inexactness gap and running times were found similar to the tests with pump statuses randomly chosen. Interestingly, the inexactness gap and running times for the new tests did not show significant differences from the tests with random pump statuses; see Fig. 4.9. 99 out of the 100 random WF instances were solved within the 1-min deadline with a median running time of 0.75 sec. Thus, the MI-QCQP (W2) is in general a powerful WF solver for various network configurations.

4.10 Conclusions

Using recent tools from graph theory, convex relaxations, energy function-based approaches, and mixed-integer programing, this work has provided a fresh perspective on the physical laws governing water distribution networks. It has been established that the WF problem admits a unique solution even in networks with multiple fixed-pressure nodes and flow-dependent pump models. This WF solution can be provably recovered via a hierarchical stack of WF solvers suitable for different network configurations. Radial networks can be handled by simple convex minimization tasks, whereas networks with cycles call for more elaborate MI-QCQP-based solvers. Nevertheless, numerical tests demonstrate that even the MI-QCQP approach scales well for moderately-sized networks. The MI-QCQP solvers have been derived upon a convex relaxation of the pressure drop equation followed by an objective penalization, an approach that sparked a parallel line of research on the OWF problem [115]. Network configurations hosting pumps on overlapping cycles remain to be a challenging case.

Appendix C

Proof of Theorem 4.7. The cost of (W2) can be written as

$$s(\mathbf{h};\mathbf{f}) = \sum_{(m,n)\in\mathcal{P}} (h_m - h_n)\,\mathrm{sign}(f_{mn}).$$

To express pressure differences along the flow direction in a compact manner, define the $P \times N$ edge-node incidence matrix $\mathbf{A}(\mathbf{f})$: Its dependence on $\mathbf{f}$ signifies that the directionality of each edge coincides with the flow directions in $\mathbf{f}$. Therefore, if the p-th row of $\mathbf{A}(\mathbf{f})$ is associated with pipe $p = (m,n)$, then its (p,k) entry is

$$A_{p,k}(\mathbf{f}) := \begin{cases} -\,\mathrm{sign}^2(f_{mn}) + \mathrm{sign}(f_{mn}) + 1 & ,\ k = m \\ \mathrm{sign}^2(f_{mn}) - \mathrm{sign}(f_{mn}) - 1 & ,\ k = n \\ 0 & ,\ \text{otherwise.} \end{cases}$$

For zero flows ($\mathrm{sign}(f_{mn}) = 0$), the default pipe direction (m,n) is selected without loss of generality.

Based on $\mathbf{A}(\mathbf{f})$, the pressure differences along the direction of flows can be written as $\mathbf{A}(\mathbf{f})\mathbf{h}$, and so

$$s(\mathbf{h};\mathbf{f}) = \mathbf{1}^\top \mathbf{A}(\mathbf{f})\mathbf{h}. \tag{4.19}$$

If (W1) is feasible, denote its unique solution by $(\bar{\mathbf{f}}, \bar{\mathbf{h}})$. Since the base directionality of the WDS graph $\mathcal{G}$ is arbitrary, let it coincide with the water flow directions of $\bar{\mathbf{f}}$. Using this convention, it follows that $\bar{\mathbf{f}} \geq \mathbf{0}$ and $\mathbf{A}(\bar{\mathbf{f}})$ is identical to the base incidence matrix $\mathbf{A}$. Next, proving by contradiction, suppose $(\tilde{\mathbf{f}}, \tilde{\mathbf{h}})$ is a minimizer of (W2), which is not feasible for (W1). Since both $\bar{\mathbf{f}}$ and $\tilde{\mathbf{f}}$ satisfy (4.2) for $\mathbf{A} = \mathbf{A}(\bar{\mathbf{f}})$, there exists a nonzero vector $\mathbf{n} \in \text{null}(\mathbf{A}^\top)$ such that

$$\tilde{\mathbf{f}} = \bar{\mathbf{f}} + \mathbf{n}. \tag{4.20}$$

As shown in (4.16), vector $\mathbf{n}$ can be expressed in terms of a set $\mathcal{L}_\mathcal{T}$ of fundamental cycles using the flow directions of $\mathbf{A}$. To simplify notation, let us define $\tilde{\mathbf{A}} := \mathbf{A}(\tilde{\mathbf{f}})$. Since every edge is assumed to belong to at most one cycle, the set of fundamental cycles is unique irrespective of the spanning tree $\mathcal{T}$ selected. Therefore, the set of fundamental cycles in fact contains all cycles in the graph, implying $\mathcal{L}_\mathcal{T} = \mathcal{L}$.

Building on (4.16), consider a decomposition of $\mathbf{n}$ as a weighted sum of indicator vectors $\mathbf{n}_\ell$'s. If there exists a cycle ℓ, for which $\alpha_\ell < 0$, then one can reverse the direction of cycle ℓ and substitute $(\alpha_\ell, \mathbf{n}_\ell)$ in (4.16) with $(-\alpha_\ell, -\mathbf{n}_\ell)$. Hence, it can be assumed that $\alpha_\ell \geq 0$ for all $\ell \in \mathcal{L}$ without loss of generality.

Each vector $\mathbf{n}_\ell$ can be decomposed as $\mathbf{n}_\ell = \mathbf{n}_\ell^+ - \mathbf{n}_\ell^-$, where $\mathbf{n}_\ell^+ := \max\{\mathbf{n}_\ell, \mathbf{0}\}$ and $\mathbf{n}_\ell^- := \max\{-\mathbf{n}_\ell, \mathbf{0}\}$. Because every edge belongs to at most one cycle, it holds that

$$\mathbf{1} = \sum_{\ell\in\mathcal{L}} \mathbf{n}_\ell^+ + \sum_{\ell\in\mathcal{L}} \mathbf{n}_\ell^- + \mathbf{n}^0 \tag{4.21}$$

where the p-th entry of vector $\mathbf{n}^0$ is 1 if edge p does not belong to any cycle; and 0, otherwise. Using (4.21) in (4.19), the objective function of (W2) becomes

$$s(\mathbf{h};\mathbf{f}) = \sum_{\ell\in\mathcal{L}}(\mathbf{n}_\ell^+)^\top\mathbf{A}\mathbf{h} + \sum_{\ell\in\mathcal{L}}(\mathbf{n}_\ell^-)^\top\mathbf{A}\mathbf{h} + (\mathbf{n}^0)^\top\mathbf{A}\mathbf{h}. \tag{4.22}$$

Albeit $s(\mathbf{h};\mathbf{f})$ will be evaluated for different pairs $(\mathbf{h};\mathbf{f})$, the vectors $\mathbf{n}_\ell$'s remain unchanged and depend on $\mathbf{A}$. Based on (4.16) and (4.22), we will next show that $s(\bar{\mathbf{h}};\bar{\mathbf{f}}) < s(\tilde{\mathbf{h}};\tilde{\mathbf{f}})$. To do so, we consider the three terms of (4.22) separately.

First summand of (4.22). Recall $\mathbf{n}_\ell^+$ is a binary vector; and the base graph directionality is such that $\bar{\mathbf{f}} \geq \mathbf{0}$. Consider the entries of $\bar{\mathbf{f}}$ and $\tilde{\mathbf{f}}$ for the edges p related to $n_{\ell,p}^+ = 1$. If $n_{\ell,p}^+ = 1$, then $\tilde{f}_p = \bar{f}_p + \alpha_\ell \geq \bar{f}_p \geq 0$. In that case, if edge $p = (m,n)$ and relates to a lossy pipe, we get

$$(\tilde{h}_m - \tilde{h}_n)\operatorname{sign}(\tilde{f}_p) \geq c_p\tilde{f}_p^2 \geq c_p\bar{f}_p^2 = (\bar{h}_m - \bar{h}_n)\operatorname{sign}(\bar{f}_p) \tag{4.23}$$

where the first inequality stems from constraint (4.14) of (W2); the second one from $\tilde{f}_p \geq \bar{f}_p \geq 0$; and the equality from constraint (4.3) of (W1).

If edge $p = (m,n)$ relates to a pump, we can also show

$$\begin{aligned}(\tilde{h}_m - \tilde{h}_n)\operatorname{sign}(\tilde{f}_p) &\geq -g_p(\tilde{f}_p) \geq -g_p(\bar{f}_p)\\ &= (\bar{h}_m - \bar{h}_n)\operatorname{sign}(\bar{f}_p).\end{aligned} \tag{4.24}$$

Consider cycle ℓ and sum up the LHS and RHS of (4.23) or (4.24) for all p with $n_{\ell,p}^+ = 1$ to get

$$(\mathbf{n}_\ell^+)^\top\tilde{\mathbf{A}}\tilde{\mathbf{h}} \geq (\mathbf{n}_\ell^+)^\top\mathbf{A}\bar{\mathbf{h}}, \tag{4.25}$$

where the inequality is strict if $\alpha_\ell \neq 0$. Summing (4.25) over all cycles provides

$$\sum_{\ell\in\mathcal{L}}(\mathbf{n}_\ell^+)^\top\tilde{\mathbf{A}}\tilde{\mathbf{h}} > \sum_{\ell\in\mathcal{L}}(\mathbf{n}_\ell^+)^\top\mathbf{A}\bar{\mathbf{h}}, \tag{4.26}$$

with strict inequality arising from the fact that not all α_ℓ's can be zero for a nonzero $\mathbf{n}$.

Second summand of (4.22). As explained earlier, if $n_{\ell,p}^+ = 1$, then $\tilde{f}_p = \bar{f}_p + \alpha_\ell \geq \bar{f}_p \geq 0$ so $\tilde{f}_p$ remains positive. On the other hand, if $n_{\ell,p}^- = 1$, then $\tilde{f}_p = \bar{f}_p - \alpha_\ell < \bar{f}_p$. In the latter case, the flow $\tilde{f}_p$ has decreased, and its sign may have been reversed to negative. Since all $\bar{f}_p$'s are positive, the flow reversals in $\tilde{f}_p$'s can be modeled as $\mathbf{A} = \mathbf{S}\tilde{\mathbf{A}}$, where matrix $\mathbf{S} := \mathrm{dg}(\mathrm{sign}(\tilde{\mathbf{f}}))$ is diagonal with the signs of $\tilde{\mathbf{f}}$ on its main diagonal. Since the vectors $\mathbf{n}_\ell$'s form a basis for $\mathrm{null}(\mathbf{A}^\top)$, it holds that $\mathbf{A}^\top\mathbf{n}_\ell = \mathbf{0}$ and so

$$(\mathbf{n}_\ell^+)^\top\mathbf{A}\bar{\mathbf{h}} = (\mathbf{n}_\ell^-)^\top\mathbf{A}\bar{\mathbf{h}}. \tag{4.27}$$

Similar properties hold for $\tilde{\mathbf{n}}_\ell := \mathbf{S}\mathbf{n}_\ell$. To see this, the vector $\tilde{\mathbf{n}}_\ell$ belongs to $\mathrm{null}(\tilde{\mathbf{A}}^\top)$ since

$$\tilde{\mathbf{A}}^\top\tilde{\mathbf{n}}_\ell = \tilde{\mathbf{A}}^\top\mathbf{S}\mathbf{n}_\ell = \mathbf{A}^\top\mathbf{n}_\ell = \mathbf{0}.$$

Therefore, we also get that

$$(\tilde{\mathbf{n}}_\ell^+)^\top\tilde{\mathbf{A}}\tilde{\mathbf{h}} = (\tilde{\mathbf{n}}_\ell^-)^\top\tilde{\mathbf{A}}\tilde{\mathbf{h}} \tag{4.28}$$

where $\tilde{\mathbf{n}}_\ell^+ := \max\{\tilde{\mathbf{n}}_\ell, \mathbf{0}\}$ and $\tilde{\mathbf{n}}_\ell^- := \max\{-\tilde{\mathbf{n}}_\ell, \mathbf{0}\}$.

By definition of $\tilde{\mathbf{n}}_\ell$, if $n_{\ell,p} = 1$, then $S_{p,p} = 1$ and $\tilde{n}_{\ell,p} = 1$. However, if $n_{\ell,p} = -1$, then $S_{p,p} = +1$ or $S_{p,p} = -1$ depending on the sign of $\tilde{f}_p$, and so $\tilde{n}_{\ell,p} = 1$ or $\tilde{n}_{\ell,p} = -1$. It therefore follows that

$$\mathbf{n}_\ell^+ \leq \tilde{\mathbf{n}}_\ell^+ \tag{4.29a}$$

$$\mathbf{n}_\ell^- \geq \tilde{\mathbf{n}}_\ell^-. \tag{4.29b}$$

Back to the ℓ-th term of the second summand in (4.22):

$$\begin{aligned}(\mathbf{n}_\ell^-)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} &\overset{a}{\geq} (\tilde{\mathbf{n}}_\ell^-)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} \overset{b}{=} (\tilde{\mathbf{n}}_\ell^+)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} \\ &\overset{c}{\geq} (\mathbf{n}_\ell^+)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} \overset{d}{\geq} (\mathbf{n}_\ell^+)^\top \mathbf{A}\bar{\mathbf{h}} \overset{e}{=} (\mathbf{n}_\ell^-)^\top \mathbf{A}\bar{\mathbf{h}}. \end{aligned} \tag{4.30}$$

where (a) stems from (4.29b); (b) from (4.28); (c) from (4.29a); (d) from (4.25); and (e) from (4.27). Summing (4.30) over $\ell \in \mathcal{L}$

$$\sum_{\ell \in \mathcal{L}} (\mathbf{n}_\ell^-)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} > \sum_{\ell \in \mathcal{L}} (\mathbf{n}_\ell^-)^\top \mathbf{A}\bar{\mathbf{h}}, \tag{4.31}$$

where the strict inequality stems from that argument (d) in (4.30) is strict for $\alpha_\ell \neq 0$, and not all α_ℓ's can be zero.

Third summand of (4.22)*.* The third summand sums up the pressure differences along the direction of flow for all edges not lying in any cycle. If edge $p = (m, n)$ belongs to this case (i.e., $n_p^0 = 1$), then $\bar{f}_p = \tilde{f}_p$ and so as in (4.23) we get

$$(\tilde{h}_m - \tilde{h}_n)\operatorname{sign}(\tilde{f}_p) \geq c_p \tilde{f}_p^2 = c_p \bar{f}_p^2 = (\bar{h}_m - \bar{h}_n)\operatorname{sign}(\bar{f}_p).$$

Summing up over all edges with $n_p^0 = 1$ yields

$$(\mathbf{n}^0)^\top \tilde{\mathbf{A}}\tilde{\mathbf{h}} \geq (\mathbf{n}^0)^\top \mathbf{A}\bar{\mathbf{h}}. \tag{4.32}$$

Adding (4.25), (4.31), and (4.32) by parts gives $s(\tilde{\mathbf{h}};\tilde{\mathbf{f}}) > s(\bar{\mathbf{h}};\bar{\mathbf{f}})$. This contradicts that $(\tilde{\mathbf{f}},\tilde{\mathbf{h}})$ is a minimizer of (W2) since $(\bar{\mathbf{f}},\bar{\mathbf{h}})$ is feasible for (W2), and concludes the proof. $\square$

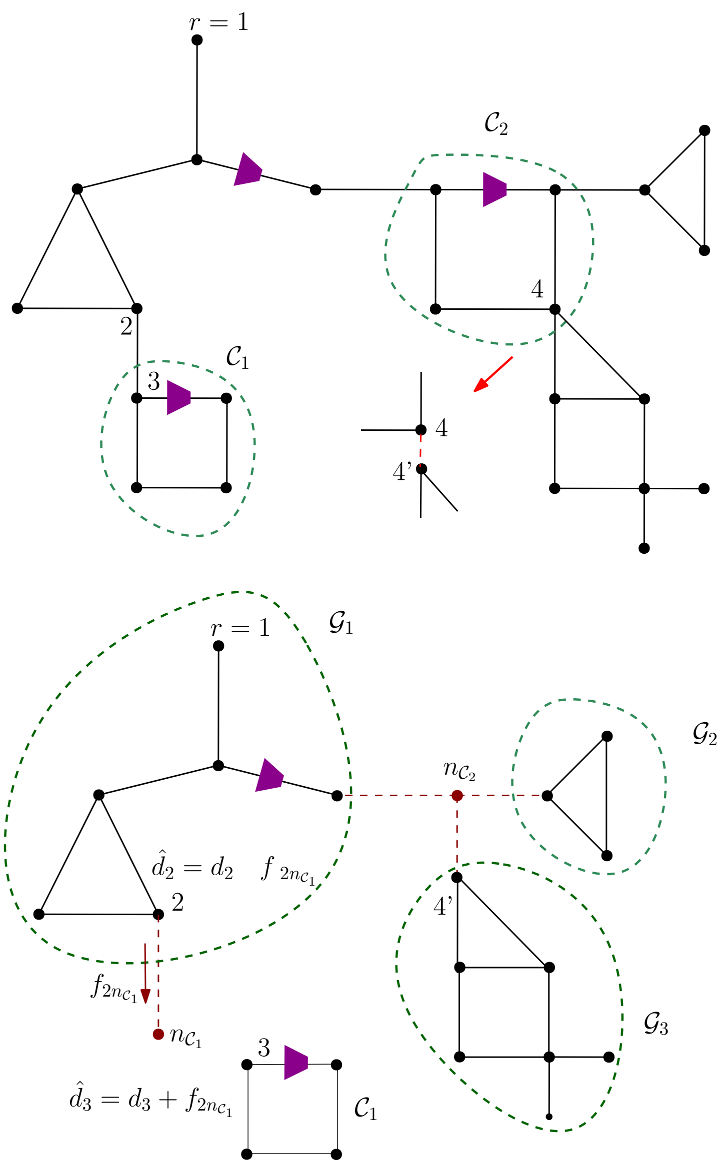

Figure 4.3: *Top:* A WDS with pumps in non-overlapping cycles; *Bottom*: Connected components after removing cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ that carry pumps.

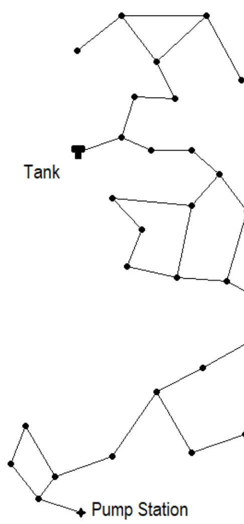

Figure 4.4: EPANET Example Network-2 of a WDS from Cherry Hills, CT [103].

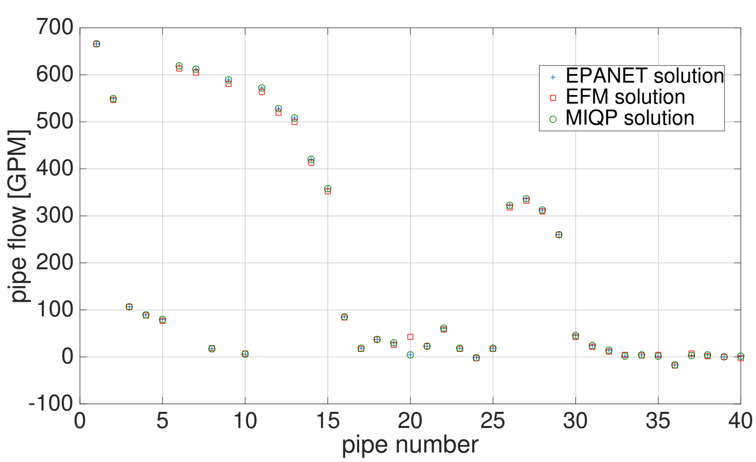

Figure 4.5: Water flows obtained by the EPANET solver, the energy function minimization of (4.9), and the MI-QCQP of (W2) for the WDS of Fig. 4.4.

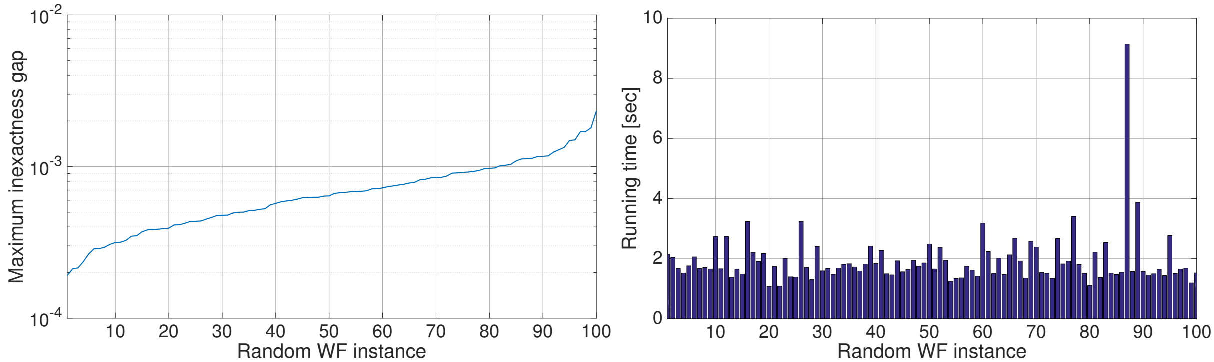

Figure 4.6: *Left:* Maximum inexactness gap attained by (W2) over 100 random WF instances. *Right*: Running time for (W2) over random WF instances.

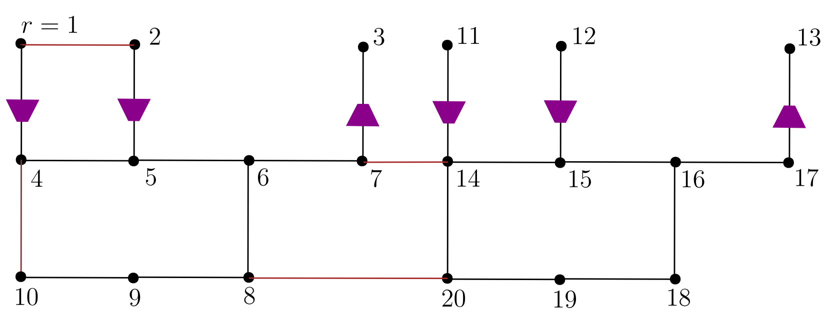

Figure 4.7: Benchmark 20-node WDS derived from the WDS of [115].

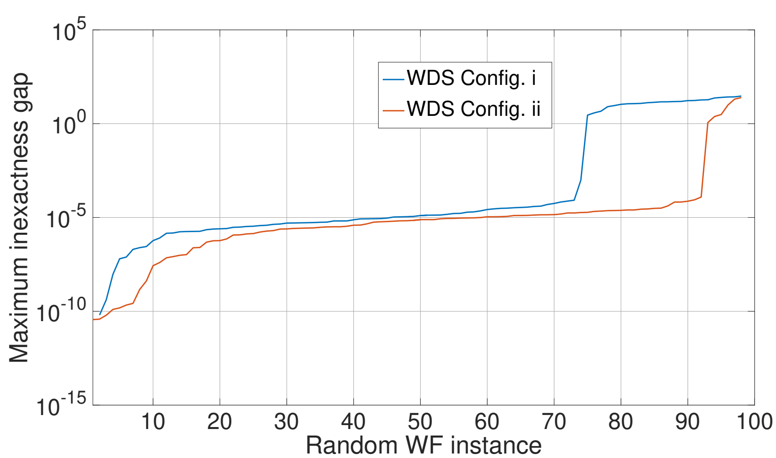

Figure 4.8: Maximum inexactness gap attained by (W2) over 100 random WF instances for configurations *C-i)* and *C-ii)* of the WDS of Fig. 4.7.

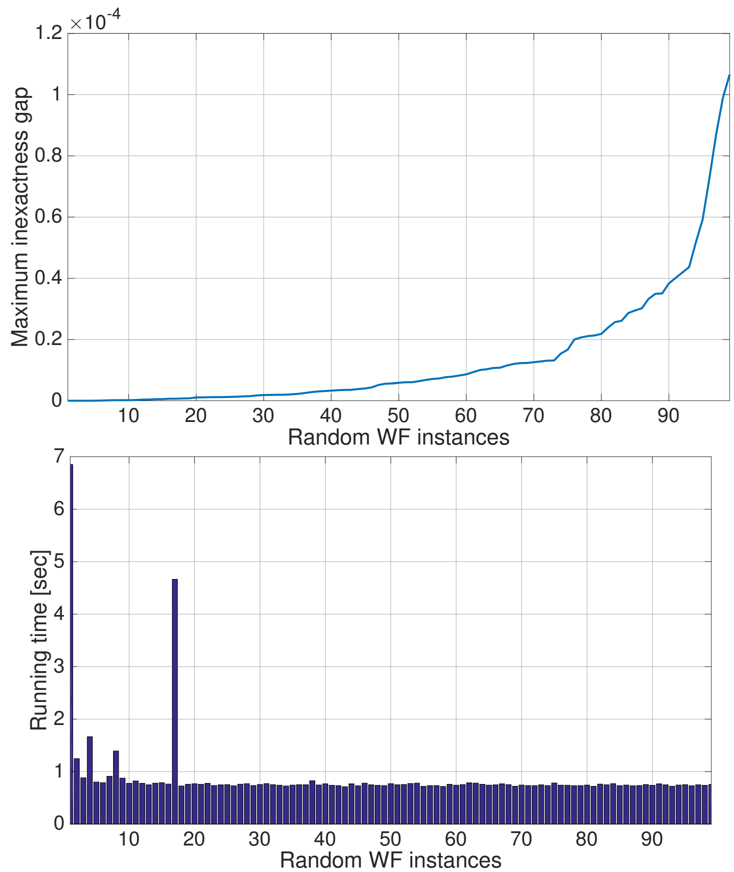

Figure 4.9: *Top:* Maximum inexactness gap for random WF instances with all pumps of Fig. 4.7 running; *Bottom*: Running times for random WF instances.

Chapter 5

Learning to Optimize Power Distribution Grids using Sensitivity-Informed Deep Neural Networks

5.1 Publication Details

M. K. Singh, S. Gupta, V. Kekatos, G. Cavraro, and A. Bernstein, "Learning to Optimize Power Distribution Grids using Sensitivity-Informed Deep Neural Networks," in *Proc. IEEE Smart Grid Communications*, Tempe, AZ, Nov. 2020.

5.2 Abstract

Deep learning for distribution grid optimization can be advocated as a promising solution for near-optimal yet timely inverter dispatch. The principle is to train a deep neural network (DNN) to predict the solutions of an optimal power flow (OPF), thus shifting the

computational effort from real-time to offline. Nonetheless, before training this DNN, one has to solve a large number of OPFs to create a labeled dataset. Granted the latter step can still be prohibitive in time-critical applications, this work puts forth an original technique for improving the prediction accuracy of DNNs by taking into account the sensitivities of the OPF minimizers with respect to the OPF parameters. By expanding on multiparametric programming, it is shown that although inverter control problems may exhibit dual degeneracy, the required sensitivities do exist in general and can be computed readily using the output of any standard quadratic program (QP) solver. Numerical tests showcase that sensitivity-informed deep learning can enhance prediction accuracy in terms of mean square error (MSE) by 2-3 orders of magnitude at minimal computational overhead. Improvements are more significant in the small-data regime, where a DNN has to learn to optimize using a few examples. Beyond multiparametric QPs, the approach is currently being generalized to parametric (non)-convex optimization problems.

5.3 Introduction

To cope with resource allocation problems with stringent latency requirements, recent research advocates using learning models that are trained to solve optimization tasks; see e.g., [46], [33]. The general workflow involves solving a large number of such problems offline to create a dataset of problem instances paired with their minimizers; train a learning model to predict those minimizers; and use the model for promptly finding near-optimal solutions in real time. References [46] and [51] impose feasibility using Lagrangian decomposition, while [119] and [144] build DNNs with structures mimicking algorithmic iterates solving the original optimization.

In the context of power systems, there has recently been an interesting interplay between deep learning and the OPF. DNNs have been employed to predict OPF solutions under a linearized DC [96], [141]; or the exact AC model [51], [59]. The standard regression loss function is sometimes augmented by a penalty on OPF constraint violations to promote feasibility [96], [59], [51]. Alternatively, predicted solutions may be converted to a physically implementable schedule upon projection using a power flow (PF) solver [96], [141]. A graph neural network leveraging the connectivity of the power system is trained to infer AC-OPF solutions in [94]. Focusing on OPF for inverter control, reference [34] constructs a DNN to be used as a digital twin of the electric feeder. To learn optimal inverter control policies, DNN-based reinforcement learning schemes have been suggested in [146], [134].

Instead of learning minimizers, DNNs have also been used to predict the active constraints of an OPF [59], [42], [33]. References [59] and [42] train DNN-based classifiers to predict individual or sets of active constraints. In [33], the active constraints of a linear programming (LP) DC-OPF are unveiled in three steps: A DNN is first trained to predict the optimal generation cost given demands; locational prices are then computed as the DNN sensitivities with respect to demands; and congested lines are identified via sparse-aware regression.

When the OPF is posed using a linearized grid model, it gives rise to parametric LPs or QPs, and hence, powerful tools from multiparametric programming (MPP) become relevant [24]. Such tools have been utilized before to study congestion patterns in electricity markets [150]; for the probabilistic characterization of primal/dual OPF solutions [67]; to accelerate hosting capacity analyses [121]; or strategic investment [122]. MPP-aided methods however are meaningful only when the related OPF enjoys few congestion patterns and/or the application of interest is of a batch nature.

This work contributes to the literature of training DNNs to solve the OPF in the following creative way. A DNN is trained to fit not only the OPF minimizer, but also its sensitivities with respect to the problem parameters. Inspired by [99], we cross-pollinate the idea of training sensitivity-aware DNNs from solving differential equations to learning for optimization problems in Section 5.4. Advances in automatic differentiation facilitate the efficient computation of DNN sensitivities as well as their integration into training. In Sections 5.5 and 5.6, we particularize properties from MPP theory to the inverter dispatch task at hand, so that the sensitivities of the OPF solution can be computed readily. The numerical tests of Section 5.7 showcase that the proposed sensitivity-informed DNN outperforms a plain DNN in terms of prediction accuracy by 2-3 orders of magnitude. The paper is concluded with generalizations of the method to more challenging OPF tasks.

Regarding *notation*, lower- (upper-) case boldface letters denote column vectors (matrices). Sets are represented by calligraphic symbols. Symbol $^\top$ stands for transposition, and inequalities are understood element-wise. All-zero and all-one vectors and matrices are represented by $\mathbf{0}$ and $\mathbf{1}$; the respective dimensions are deducible from context. A symmetric positive definite matrix is denoted as $\mathbf{X} \succ \mathbf{0}$.

5.4 Sensitivity-Informed Deep Learning

Suppose a system operator has to routinely solve a convex quadratic program (QP) over the resource vector $\mathbf{x}$

$$\mathbf{x}_{\boldsymbol{\theta}} := \arg\min_{\mathbf{x}} \quad \tfrac{1}{2}\mathbf{x}^\top \mathbf{A}\mathbf{x} - \mathbf{x}^\top \mathbf{B}\boldsymbol{\theta} \tag{5.1a}$$

$$\text{s.to} \quad \mathbf{C}\mathbf{x} \leq \mathbf{D}\boldsymbol{\theta} + \mathbf{e} : \quad \boldsymbol{\lambda}_{\boldsymbol{\theta}}. \tag{5.1b}$$

While $(\mathbf{A} \succ \mathbf{0}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{e})$ are kept fixed for some time, the parameter vector $\boldsymbol{\theta}$ may be varying frequently. Let $(\mathbf{x}_{\boldsymbol{\theta}}, \boldsymbol{\lambda}_{\boldsymbol{\theta}})$ denote a pair of primal/dual solutions for a particular $\boldsymbol{\theta}$. This multiparametric QP (MPQP) may be encountered in several resource allocation tasks in cyber-physical systems in general, and power systems in particular. Renditions of (5.1) may appear in real-time electricity markets and contingency analysis in transmission systems; or in load disaggregation, demand response, and DER management in distribution systems.

The application of interest here is dispatching inverters through an approximate OPF that minimizes power losses subject to voltage constraints. Vector $\mathbf{x}$ comprises the inverter reactive power setpoints, while $\boldsymbol{\theta}$ carries the loading conditions (demand and solar generation) across feeder buses. Since $\boldsymbol{\theta}$ may change rapidly (rooftop solar generation can fluctuate by up to 15% of its rating within one minute [10]), an operator may not be able to solve (5.1) exactly for thousands of feeders hosting thousands of inverters each.

To expedite the process of computing inverter setpoints, the operator may want to train a DNN that *learns to solve* (5.1). This DNN would be fed with $\boldsymbol{\theta}$ to compute a predictor $\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w})$ of $\mathbf{x}_{\boldsymbol{\theta}}$. The straightforward approach for training this DNN is finding its weights $\mathbf{w}$ as the minimizers of

$$\min_{\mathbf{w}} \sum_{s=1}^{S} \|\hat{\mathbf{x}}(\boldsymbol{\theta}_s;\mathbf{w}) - \mathbf{x}_{\boldsymbol{\theta}_s}\|_2^2. \tag{5.2}$$

Problem (5.2) is solved offline and aims at fitting the DNN output to the OPF minimizer over S loading scenarios $\{\boldsymbol{\theta}_s\}_{s=1}^S$, which are deemed representative of the upcoming control period. Before solving (5.2) however, the operator has to solve S instances of (5.1) to build the labeled dataset $\{(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s})\}_{s=1}^S$. Accommodating a large S could be impractical in time-critical setups, where the DNN has to be re-trained every 30–60 min or so because parameters $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{e})$ change too. This is relevant for inverter control due to grid topology reconfigurations, changes in regulator settings and capacitor statuses, and varying loading conditions that may alter the linearization point of the nonlinear power flow equations.

We improve on this process in two aspects: *i)* we reduce S so the operator has to solve fewer OPFs; and *ii)* for each OPF instance solved during training, we further exploit the sensitivity of $\mathbf{x}_s$ with respect to $\boldsymbol{\theta}_s$. Both aspects can expedite training or increase the training dataset over the same training time, while *ii)* is expected to yield better generalization to the sought DNN and stabilize its solutions against variations in $\boldsymbol{\theta}$. Inspired by [99] where a DNN was trained to solve differential equations by incorporating information on its derivatives, here we integrate sensitivity information of the optimizer $\mathbf{x}_{\boldsymbol{\theta}}$ with respect to the DNN input $\boldsymbol{\theta}$. To accomplish that, we propose augmenting the training process of (5.2) as

$$\min_{\mathbf{w}} \sum_{s=1}^{S} \|\hat{\mathbf{x}}(\boldsymbol{\theta}_s;\mathbf{w}) - \mathbf{x}_{\boldsymbol{\theta}_s}\|_2^2 + \|\hat{\mathbf{J}}(\boldsymbol{\theta}_s;\mathbf{w}) - \mathbf{J}_{\boldsymbol{\theta}_s}\|_F^2 \tag{5.3}$$

where $\mathbf{J}_{\boldsymbol{\theta}} := \nabla_{\boldsymbol{\theta}}\mathbf{x}_{\boldsymbol{\theta}}$ and $\hat{\mathbf{J}}(\boldsymbol{\theta};\mathbf{w}) := \nabla_{\boldsymbol{\theta}}\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w})$ are the Jacobian matrices of the minimizer and the DNN output accordingly with respect to the parameter vector $\boldsymbol{\theta}$. Symbol $\|\cdot\|_F$ denotes the Frobenius norm. Matrix $\hat{\mathbf{J}}(\boldsymbol{\theta}_s;\mathbf{w})$ and its derivatives with respect to $\mathbf{w}$ can be readily computed via automatic differentiation tools such as GradientTape in TensorFlow. The optimization of (5.3) aims at fitting the values of the minimizers as well as their sensitivities (partial derivatives) in terms of the DNN input $\boldsymbol{\theta}$. By incorporating such physics-aware information, the DNN is expected to predict well not only on the training scenarios $\boldsymbol{\theta}_s$'s, but in a neighborhood around them too. We term the DNN trained by (5.3) as *sensitivity-informed DNN* (SI-DNN) versus the *plain DNN* (P-DNN) obtained from (5.2).

Having posed the sensitivity-informed deep learning task of (5.3), the pertinent questions are if the OPF operator OPF : $\boldsymbol{\theta} \rightarrow \mathbf{x}_{\boldsymbol{\theta}}$ is differentiable; and if so, whether its Jacobian matrix $\mathbf{J}_{\boldsymbol{\theta}}$ can be found in a computationally efficient way. The goal is to be able to build the augmented training set $\{(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s}, \mathbf{J}_{\boldsymbol{\theta}_s})\}$ at a minimal computational effort in excess to the effort for building the training set $\{(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s})\}$ for (5.2). To this end, Section 5.5 first poses optimal inverter dispatching in the form of the MPQP in (5.1), and then Section 5.6 adapts results from multiparametric programming to compute $\mathbf{J}_{\boldsymbol{\theta}}$ efficiently.

5.5 Optimal Inverter Dispatch

Consider the task of optimal reactive power compensation by smart inverters. Given (re)active loads along with the active solar generation on every bus, the goal is to find the inverter setpoints for reactive power injections. Such setpoints can be found through an OPF that minimizes ohmic power losses on distribution lines, while satisfying voltage regulation limitations across buses. Before formally posing this OPF, we briefly review the postulated feeder model.

In a single-phase radial feeder with $N+1$ buses, the substation is indexed by $n=0$. Let v_n be the voltage magnitude, and $p_n + jq_n$ the complex power injection at bus n. Collect all but the substation injections and voltages in the N-length vectors $(\mathbf{p},\mathbf{q},\mathbf{v})$. Vectors $(\mathbf{p},\mathbf{q})$ can be decomposed into non-negative injections $(\mathbf{p}_i,\mathbf{q}_i)$ from inverters, and withdrawals $(\mathbf{p}_\ell,\mathbf{q}_\ell)$ from loads as

$$\mathbf{p} = \mathbf{F}_i\mathbf{p}_i - \mathbf{F}_\ell\mathbf{p}_\ell \tag{5.4a}$$

$$\mathbf{q} = \mathbf{F}_i\mathbf{q}_i - \mathbf{F}_\ell\mathbf{q}_\ell. \tag{5.4b}$$

Matrices $\mathbf{F}_i$ and $\mathbf{F}_\ell$ map buses with solar and loads to the N feeder buses, respectively. If there are I buses hosting solar and L buses hosting loads, then $\mathbf{F}_i \in \{0,1\}^{N\times I}$ and $\mathbf{F}_\ell \in \{0,1\}^{N\times L}$. We assume that each bus may host at most one (aggregate) load and at most one solar generator.

To simplify the task of dispatching inverters, we resort to an approximate feeder model obtained upon linearizing the power flow equations at the flat voltage profile of unit magnitudes and zero angles at all buses; see e.g., [121]. According to this model, voltages depend linearly on power injections as

$$\mathbf{v}(\mathbf{p},\mathbf{q}) \simeq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} + v_0\mathbf{1}_N \tag{5.5}$$

where the $N \times N$ matrices $(\mathbf{R}, \mathbf{X})$ are symmetric positive definite, and depend on the feeder topology and line impedances. We are also interested in expressing ohmic losses with respect to nodal injections. Adopting a second-order Taylor's series expansion, active losses on lines can be approximated as a convex quadratic function of power injections as

$$L(\mathbf{p},\mathbf{q}) \simeq 2\mathbf{p}^\top\mathbf{R}\mathbf{p} + 2\mathbf{q}^\top\mathbf{R}\mathbf{q}. \tag{5.6}$$

Based on the previous modeling, the reactive setpoints for inverters can be found as the minimizers of

$$\min_{\mathbf{q}_i} \quad \mathbf{q}^\top\mathbf{R}\mathbf{q} \tag{5.7a}$$

$$\text{s.to} \quad -0.03\cdot\mathbf{1}_N \leq \mathbf{R}\mathbf{p} + \mathbf{X}\mathbf{q} \leq 0.03\cdot\mathbf{1}_N \tag{5.7b}$$

$$-\bar{\mathbf{q}}_i \leq \mathbf{q}_i \leq \bar{\mathbf{q}}_i. \tag{5.7c}$$

The approximate OPF of (5.7) minimizes ohmic losses over the controllable inverter setpoints $\mathbf{q}_i$; note the first summand of (5.6) and the scaling by 2 have been ignored. Constraint (5.7b) maintains voltages within ±3% per unit (pu). Although the substation voltage v_0 has been set at 1 pu, it can be appended as an optimization variable of (5.7) without loss of generality. Our methodology can incorporate voltage regulators as in [121]; they are ignored here to keep the exposition succinct. Inverter setpoints are constrained in (5.7c) by the vector $\bar{\mathbf{q}}_i$ of reactive ratings, which is assumed known and fixed.

Under varying grid conditions, an operator may want to solve (5.7) routinely for different

OPF instances. If the topology remains fixed, OPF instances differ in the values of loads and solar $(\mathbf{p}_i, \mathbf{p}_\ell, \mathbf{q}_\ell)$. Define the vector of OPF inputs $\boldsymbol{\theta} := [\mathbf{p}^\top \ \mathbf{q}_\ell^\top]^\top$. Due to the problem structure, active loads and solar can appear collectively as $\mathbf{p}$ per (5.4a). To better distinguish the optimization variable $\mathbf{x} = \mathbf{q}_i$ from the OPF parameters $\boldsymbol{\theta}$, let us substitute (5.4b) in (5.7), and express the inverter dispatch of (5.7) as the MPQP of (5.1) with the substitutions

$$\mathbf{A} := 2\mathbf{F}_i^\top \mathbf{R}\mathbf{F}_i \succ \mathbf{0}, \qquad \mathbf{B} := 2\mathbf{F}_i^\top \mathbf{R}\mathbf{F}_\ell \left[\mathbf{0}_{L\times N} \ \ \mathbf{I}_L\right] \tag{5.8a}$$

$$\mathbf{C} := \begin{bmatrix} -\mathbf{X}\mathbf{F}_i \\ \mathbf{X}\mathbf{F}_i \\ -\mathbf{I}_I \\ \mathbf{I}_I \end{bmatrix}, \qquad \mathbf{D} := \begin{bmatrix} [\mathbf{R} \ \ -\mathbf{X}\mathbf{F}_\ell] \\ [-\mathbf{R} \ \ +\mathbf{X}\mathbf{F}_\ell] \\ \mathbf{0}_I \\ \mathbf{0}_I \end{bmatrix} \tag{5.8b}$$

$$\mathbf{e}^\top := \begin{bmatrix} 0.03 \cdot \mathbf{1}_N^\top & 0.03 \cdot \mathbf{1}_N^\top & \bar{\mathbf{q}}_i^\top & \bar{\mathbf{q}}_i^\top \end{bmatrix}. \tag{5.8c}$$

5.6 Computing the Jacobian Matrix via MPP

Consider solving (5.1) for a particular $\boldsymbol{\theta}$. The corresponding optimal primal-dual pair $(\mathbf{x}_{\boldsymbol{\theta}}, \boldsymbol{\lambda}_{\boldsymbol{\theta}})$ should satisfy the Karush-Kuhn-Tucker (KKT) optimality conditions [17]. Among the inequality constraints of (5.1), those satisfied with equality at the optimum are termed *active or binding constraints*. Let $(\tilde{\mathbf{C}}, \tilde{\mathbf{D}}, \tilde{\mathbf{e}}, \tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}})$ be the row partitions of $(\mathbf{C}, \mathbf{D}, \mathbf{e}, \boldsymbol{\lambda}_{\boldsymbol{\theta}})$ associated with active constraints. Likewise, let $(\bar{\mathbf{C}}, \bar{\mathbf{D}}, \bar{\mathbf{e}}, \bar{\boldsymbol{\lambda}}_{\boldsymbol{\theta}})$ denote the row partition related to *inactive or non-binding constraints*. Leveraging complementarity slackness, the KKT conditions can be expressed as [17]

$$\mathbf{A}\mathbf{x}_{\boldsymbol{\theta}} + \tilde{\mathbf{C}}^\top \tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} = \mathbf{B}\boldsymbol{\theta} \tag{5.9a}$$

$$\tilde{\mathbf{C}}\mathbf{x}_{\boldsymbol{\theta}} = \tilde{\mathbf{D}}\boldsymbol{\theta} + \tilde{\mathbf{e}} \tag{5.9b}$$

$$\bar{\mathbf{C}}\mathbf{x}_{\boldsymbol{\theta}} < \bar{\mathbf{D}}\boldsymbol{\theta} + \bar{\mathbf{e}} \tag{5.9c}$$

$$\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} > \mathbf{0} \tag{5.9d}$$

$$\bar{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} = \mathbf{0}. \tag{5.9e}$$

For (5.9d) in particular, it is assumed that binding constraints relate to strictly positive dual variables $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} > \mathbf{0}$. The reverse is unlikely to occur when $\boldsymbol{\theta}_s$'s are drawn at random. In the unlikely case some of the entries of $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}$ do equal zero, the OPF sensitivity for this specific $\boldsymbol{\theta}$ can be ignored and the DNN is trained only to match the OPF minimizer $\mathbf{x}_{\boldsymbol{\theta}}$.

If there are A active constraints, the equalities of (5.9a)–(5.9b) constitute a system of $(I+A)$ linear equations over the $(I+A)$ unknowns $(\mathbf{x}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}})$. Since $\mathbf{A} \succ \mathbf{0}$, the Lagrangian optimality condition of (5.9a) yields

$$\mathbf{x}_{\boldsymbol{\theta}} = \mathbf{A}^{-1}(\mathbf{B}\boldsymbol{\theta} - \tilde{\mathbf{C}}^\top \tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}). \tag{5.10}$$

Substituting (5.10) into (5.9b) provides

$$\mathbf{G}\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} = (\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}})\boldsymbol{\theta} - \tilde{\mathbf{e}} \tag{5.11}$$

where $\mathbf{G} := \tilde{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^\top$. If matrix $\tilde{\mathbf{C}}$ has linearly independent rows, then $\mathbf{G} \succ \mathbf{0}$, and (5.11) has a unique solution. Otherwise, there are infinitely many $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}$'s satisfying (5.11) within a shift invariance on the nullspace $\text{null}(\tilde{\mathbf{C}}^\top) = \text{null}(\mathbf{G})$.

For a general QP, the rows of $\tilde{\mathbf{C}}$ are typically linearly independent, thus satisfying the so termed linear independence constraint qualification (LICQ) [17]. Nonetheless, that is not

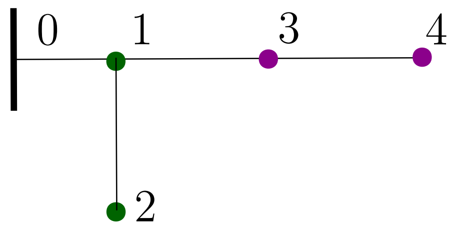

Figure 5.1: A 5-bus feeder demonstrating the violation of LICQ for (5.7).

the case for the inverter dispatch problem of (5.7) where instances of linearly dependent active constraints occur frequently. To see this, consider the feeder of Fig. 5.1, where buses $\{1, 2\}$ host inverters, bus 3 is a load bus, and 4 is a zero-injection bus. Let x_n denote the reactance for the line feeding bus n. Then, matrix $\mathbf{X}$ can be found as [121]

$$\mathbf{X} = \begin{bmatrix} x_1 & x_1 & x_1 & x_1 \\ x_1 & x_1 + x_2 & x_1 & x_1 \\ x_1 & x_1 & x_1 + x_3 & x_1 + x_3 \\ x_1 & x_1 & x_1 + x_3 & x_1 + x_3 + x_4 \end{bmatrix}.$$

Moreover, the product $\mathbf{XF}_i$ selects the first two columns of $\mathbf{X}$ corresponding to inverter locations. Now consider a loading scenario that causes the voltage at bus 3 to be at the lower limit. Being a zero-injection bus, the downstream bus 4 shares the same voltage. Thus, the minimum voltage constraints become active for buses 3 and 4. Assuming no other constraints being active, matrix $\tilde{\mathbf{C}}$ of (5.8b) reads as

$$\tilde{\mathbf{C}} = - \begin{bmatrix} x_1 & x_1 \\ x_1 & x_1 \end{bmatrix}.$$

which is clearly row rank-deficient. This counterexample emphasizes the need to explicitly address the case where the solution of (5.1) or (5.7) fails to meet the LICQ.

Regardless of whether $\tilde{\mathbf{C}}$ is full row-rank or not, the solution to (5.11) lies in the polyhedron

$$\tilde{\ }_{\boldsymbol{\theta}} := \{\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}} = \mathbf{G}^{+}(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}})\boldsymbol{\theta} - \mathbf{G}^{+}\tilde{\mathbf{e}} + \mathbf{u}, \tilde{\mathbf{C}}^{\top}\mathbf{u} = \mathbf{0}\}$$

where $\mathbf{G}^{+}$ is the pseudo-inverse of $\mathbf{G} = \tilde{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^{\top}$. Substituting $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}$ back in (5.10) and exploiting $\tilde{\mathbf{C}}^{\top}\mathbf{u} = \mathbf{0}$, we get

$$\mathbf{x}_{\boldsymbol{\theta}} = \mathbf{J}_{\boldsymbol{\theta}}\boldsymbol{\theta} + \mathbf{A}^{-1}\tilde{\mathbf{C}}^{\top}\mathbf{G}^{+}\tilde{\mathbf{e}}. \tag{5.12}$$

where the sensitivity matrix is defined as

$$\mathbf{J}_{\boldsymbol{\theta}} := \mathbf{A}^{-1}\mathbf{B} - \mathbf{A}^{-1}\tilde{\mathbf{C}}^{\top}\mathbf{G}^{+}(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \tilde{\mathbf{D}}). \tag{5.13}$$

Although there may be infinitely many $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}}$'s satisfying the KKT conditions, the optimal primal solution of (5.1) is unique if it exists. This is not surprising since (5.1) has a strictly convex objective. Moreover, the solution can be expressed as an affine function of $\boldsymbol{\theta}$. Note that the parameters of this affine function depend on the set of active constraints, and this is indicated by the subscript of $\boldsymbol{\theta}$ on $\mathbf{J}_{\boldsymbol{\theta}}$.

Because the triplet $(\boldsymbol{\theta}, \mathbf{x}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}})$ should also satisfy (5.9d) and (5.9c), there exists a $\mathbf{u} \in \text{null}(\tilde{\mathbf{C}}^{\top})$ so that $\boldsymbol{\theta}$ satisfies

$$\mathbf{G}^{+}(\tilde{\mathbf{C}}\mathbf{A}^{-1}\mathbf{B} - \mathbf{D})\boldsymbol{\theta} > \mathbf{G}^{+}\tilde{\mathbf{e}} - \mathbf{u} \tag{5.14a}$$

$$(\bar{\mathbf{C}}\mathbf{J}_{\boldsymbol{\theta}} - \bar{\mathbf{D}})\boldsymbol{\theta} < \bar{\mathbf{e}} - \bar{\mathbf{C}}\mathbf{A}^{-1}\tilde{\mathbf{C}}^{\top}\mathbf{G}^{+}\tilde{\mathbf{e}} \tag{5.14b}$$

So far, we have characterized the solution to (5.1) for a particular $\boldsymbol{\theta}$. Since we are interested in the sensitivity of the OPF minimizer with respect to $\boldsymbol{\theta}$, we ask the natural question whether (5.12) holds for all $\boldsymbol{\theta}'$'s within a vicinity of this specific $\boldsymbol{\theta}$. The answer to this question is on the affirmative. To see this, fix $\mathbf{u}$ and perturb $\boldsymbol{\theta}$ to get $\boldsymbol{\theta}'$ so it still satisfies (5.14). Using $\boldsymbol{\theta}'$ in lieu of $\boldsymbol{\theta}$, construct $\mathbf{x}_{\boldsymbol{\theta}'}$ from (5.12), and $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{\theta}'}$ from (5.11). In doing so, we row-partition matrices assuming the same constraints are active as for $\boldsymbol{\theta}$. This means we still use matrix $\mathbf{J}_{\boldsymbol{\theta}}$. We also set $\bar{\boldsymbol{\lambda}}_{\boldsymbol{\theta}'} = \mathbf{0}$. The constructed primal-dual pair $(\mathbf{x}_{\boldsymbol{\theta}'}, \boldsymbol{\lambda}_{\boldsymbol{\theta}'})$ satisfies the KKT conditions of (5.1) for $\boldsymbol{\theta}'$, and hence constitutes an optimal solution for this $\boldsymbol{\theta}'$. The aforesaid process can be repeated for any $\boldsymbol{\theta}'$ in the vicinity of the original $\boldsymbol{\theta}$ because (5.14) are strict inequalities. In other words, formula (5.12) is valid for all $\boldsymbol{\theta}'$'s around $\boldsymbol{\theta}$ and with matrix $\mathbf{J}_{\boldsymbol{\theta}}$ remaining unaltered.

The latter reveals that the OPF operator from $\boldsymbol{\theta}$ to $\mathbf{x}_{\boldsymbol{\theta}}$ is differentiable around $\boldsymbol{\theta}$. In addition, its Jacobian matrix is provided in closed form using (5.13). Calculating $\mathbf{J}_{\boldsymbol{\theta}}$ entails: *i)* Knowing the set of active constraints; *ii)* inverting matrix $\mathbf{A}$ once; and *iii)* inverting matrix $\mathbf{G}$. Although *iii)* is executed once per $\boldsymbol{\theta}$, the computation is lightweight since the number of active constraints A should be smaller than N. In a nutshell, computing $\mathbf{J}_{\boldsymbol{\theta}}$ once (5.1) has been solved for a particular $\boldsymbol{\theta}$, is much simpler than solving (5.1) per se. Hence, computing sensitivities adds insignificant complexity in the process of constructing the labeled dataset $(\boldsymbol{\theta}, \mathbf{x}_{\boldsymbol{\theta}}, \mathbf{J}_{\boldsymbol{\theta}})$.

5.7 Numerical Tests

The performance of the developed approach was numerically tested by solving the OPF in (5.7) for the IEEE 37-bus feeder upon removing regulators, incorporating 5 solar generators, and converting it to its single-phase equivalent; see Fig. 5.2. We extracted minute-based load and solar generation data for June 1, 2018, from the Pecan Street dataset [65]. The feeder has 25 buses with non-zero load. The first 75 non-zero load buses from the dataset were aggregated every 3 and normalized to obtain 25 load profiles. Similarly, 5 solar generation

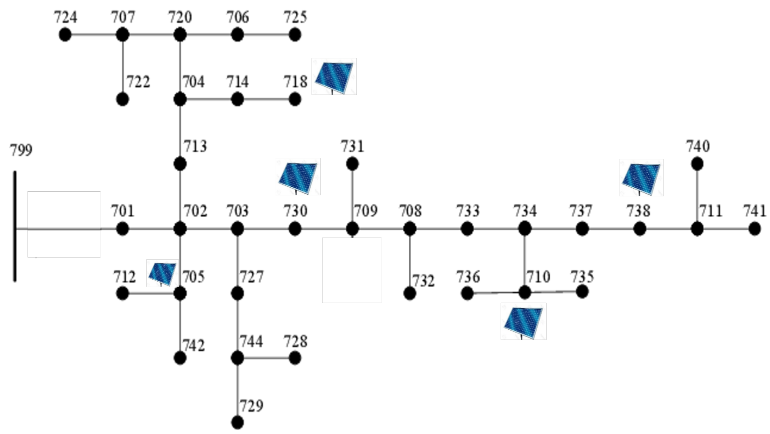

Figure 5.2: A modified IEEE 37-bus feeder showing additional solar generators.

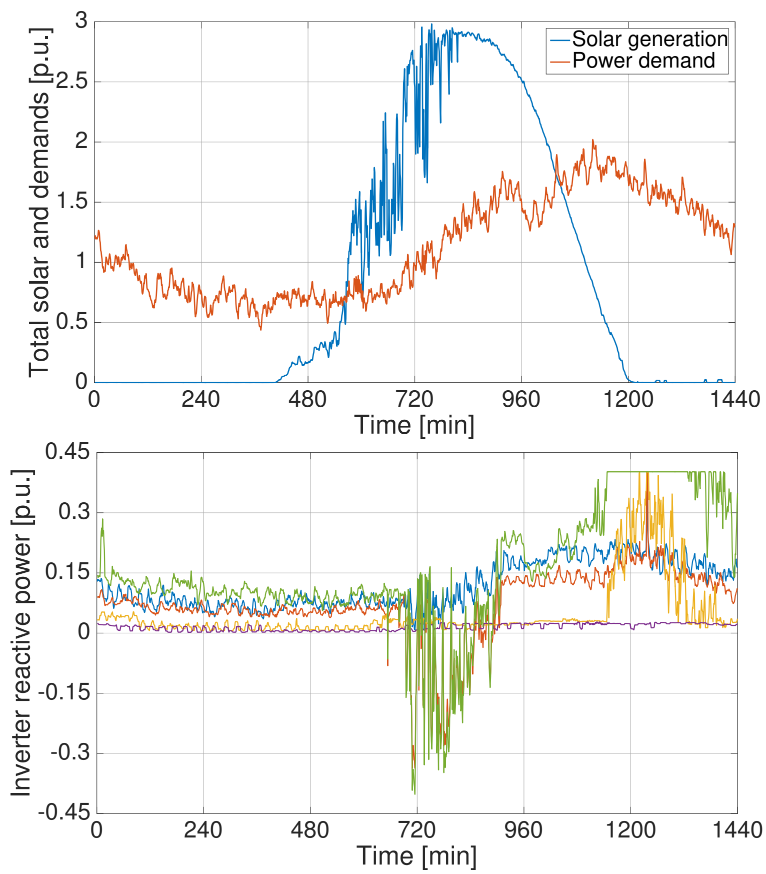

Figure 5.3: Minute-based data for the total (feeder-wise) solar power generation and active power demand (top panel); and the optimal reactive power injection setpoints for five inverters (bottom panel).

profiles were obtained. The normalized load profiles for the 24-hr period were scaled so the 97-th percentile of the total load duration curve coincided with the total nominal load. This scaling results in a peak aggregate load being 1.1 times the total nominal load. We synthesized reactive loads by scaling active demand to match the power factors of the IEEE 37-bus feeder. The 5 solar generators were of equal ratings and scaled to meet 75% of the total energy requirement over the entire day. Figure 5.3 shows the total demand and solar generation across the feeder (top), along with the optimal reactive power injection for each of the five inverters (bottom). The optimal setpoints were obtained by solving (5.7) using YALMIP and Sedumi. In solving the 1,440 OPF instances, no constraints were active for 914 instances. Out of the remaining 526 instances, the LICQ was not satisfied for 175 instances; thus necessitating the approach pursued here.

Our tests compared the P-DNN and SI-DNN, both trained to predict the minimizer of (5.7). For the first set of tests, the DNNs were assumed to be trained on an hourly basis. To evaluate the potential benefit of integrating sensitivity information, the architecture, optimizer, and training epochs were kept identical for the two DNNs. In fact, the architecture was chosen in favor of the P-DNN. It was determined so that for a sufficiently large training set, the P-DNN would be able to predict a near-optimal inverter dispatch. This ensured a sufficiently rich, yet not too complex architecture: Three hidden layers with 210, 210, and 350 neurons respectively, all using the rectified linear unit (ReLU) activation. All results were generated using the Adam optimizer with a learning rate of 0.01. The DNNs were implemented using the TensorFlow library on Google colab.

For a given hour, the two DNNs were trained using 4 OPF instances, obtained by sampling the upcoming hourly period every 15 min. The remaining 56 grid instances were used to test the two DNNs based on the mean square error (MSE) $\|\mathbf{x}_s - \hat{\mathbf{x}}_s\|_2^2$ on the predicted setpoints. To compare P-DNN and SI-DNN under varying conditions, their training and testing errors

Table 5.1: Average Test MSE [in 10^{-6} pu] after 1,000 Epochs for Different Hours of the Day

Hour	12 OPF training scenarios		4 OPF training scenarios	
	P-DNN	**SI-DNN**	**P-DNN**	**SI-DNN**
5	27.9	0.04	75.3	0.33
12	1184.4	50.4	7229.8	4755.3
16	145.1	49.6	540.3	342.5
20	359.3	45.0	1500.6	145.2

Table 5.2: Average Test MSE [in 10^{-6} pu] and training Time [in sec] after 1,000 Epochs for 10–12 am (840 Min-based Scenarios)

Training Scenarios	**P-DNN**		**SI-DNN**	
	MSE	**Time**	**MSE**	**Time**
5%	1407.8	108.9	119.9	118.7
10%	520.2	77.1	72.9	79.7
20%	219.2	100.8	55.1	113.4

were evaluated over hours 5, 12, and 20, and shown in Fig. 5.4. The tests show that even when P-DNN yields smaller training errors, SI-DNN offers an improvement in testing error by one or two orders of magnitude.

We next conducted the previous test but now over 10 Monte Carlo runs, each time selecting the training OPF scenarios at random. To evaluate the effect of the dataset size, we trained both DNNs using 4 and 12 scenarios for each of the hours 5, 12, 16, and 20. Table 5.1 reports the test MSEs attained after 1,000 epochs and averaged over the 10 Monte Carlo runs. The results corroborates that the SI-DNN features a gain in prediction accuracy over P-DNN by 1-3 orders of magnitude. Its advantage is generally more pronounced when dealing with smaller datasets, as expected.

For the last set of tests, the two DNNs were trained to learn the OPF for an entire day. The training dataset was constructed by sampling 5%, 10%, and 20% of the one-minute data. To explicitly focus on periods of high variability, we excluded the hours from midnight to 10 am from sampling. Table 5.2 summarizes the MSEs obtained after 1,000 epochs and averaged over 100 Monte Carlo draws of the training dataset. The table also reports the average training time for 1,000 epochs for the two methodologies. The runtimes on Google colab often varies with session. Thus, these times can only be compared separately for each training scenario; and not across scenarios. The results establish that the SI-DNN consistently outperforms the P-DNN without incurring any significant increase in training time. For example, the SI-DNN achieves an MSE of $119.9 \cdot 10^{-6}$ pu using 5% of the data, whereas the P-DNN achieves an MSE of $219.2 \cdot 10^{-6}$ pu although it has been trained by using 4 times more data (20%).

5.8 Conclusions and Ongoing Work

A novel procedure for training DNNs that learn to optimize has been put forth. Leaping beyond the general practice of training DNNs via labeled parameter-minimizer pairs, this work ensures that the sensitivities of DNN predictions to inputs are close to the respective sensitivities of the original optimization task. Addressing QPs in particular, the sensitivities required for training the DNN can be computed readily in virtue of results from MPP theory. The application pursued has been inverter reactive power control in distribution systems for minimizing losses subject to voltage constraints. It has been shown here that although for general QPs dual degeneracies are rare, for the inverter dispatch task such degeneracies can be encountered frequently. Fortunately, even for such instances, the required Jacobian matrices do exist in general, and the proposed approach can successfully compute them in closed form. The efficacy of the novel training method is demonstrated via numerical tests that corroborate an improvement in prediction accuracy by 2-3 orders of magnitude, as compared to the traditional regression approach. The improvements are more pronounced in the small-data regime, where a DNN has to learn to optimize using few examples.

Prompted by these promising results, we are currently generalizing this creative idea of including gradient information into DNNs towards several exciting directions. Beyond MPQPs that feature rich structure in their solutions, we are interested in improving learning for resource allocation tasks of the generic parametric form

$$\begin{aligned} \mathbf{x}_{\boldsymbol{\theta}} := \arg\min_{\mathbf{x}} \quad & f(\mathbf{x};\boldsymbol{\theta}) \\ \text{s.to} \quad & \mathbf{g}(\mathbf{x};\boldsymbol{\theta}) \leq \mathbf{0} : \ \boldsymbol{\lambda}_{\boldsymbol{\theta}}. \end{aligned} \tag{$P_{\boldsymbol{\theta}}$}$$

Thanks to results from optimization, given $(f, \mathbf{g}, \mathbf{x}_{\boldsymbol{\theta}}, \boldsymbol{\lambda}_{\boldsymbol{\theta}})$ one can compute the gradient $\nabla_{\boldsymbol{\theta}} f(\mathbf{x}_{\boldsymbol{\theta}})$, and the Jacobian matrices $\nabla_{\boldsymbol{\theta}} \mathbf{x}_{\boldsymbol{\theta}}$ and $\nabla_{\boldsymbol{\theta}} \boldsymbol{\lambda}_{\boldsymbol{\theta}}$, regardless if $(P_{\boldsymbol{\theta}})$ is convex. Leveraging these results, we are currently working towards: *d1)* incorporating sensitivities for learning to optimize different power system optimization and monitoring tasks, including the AC OPF and its various convex relaxations; *d2)* integrating dual sensitivities and weight their deviations; and *d3)* predicting binding constraints so that exact minimizers can be found by solving an OPF over a condensed feasible set.

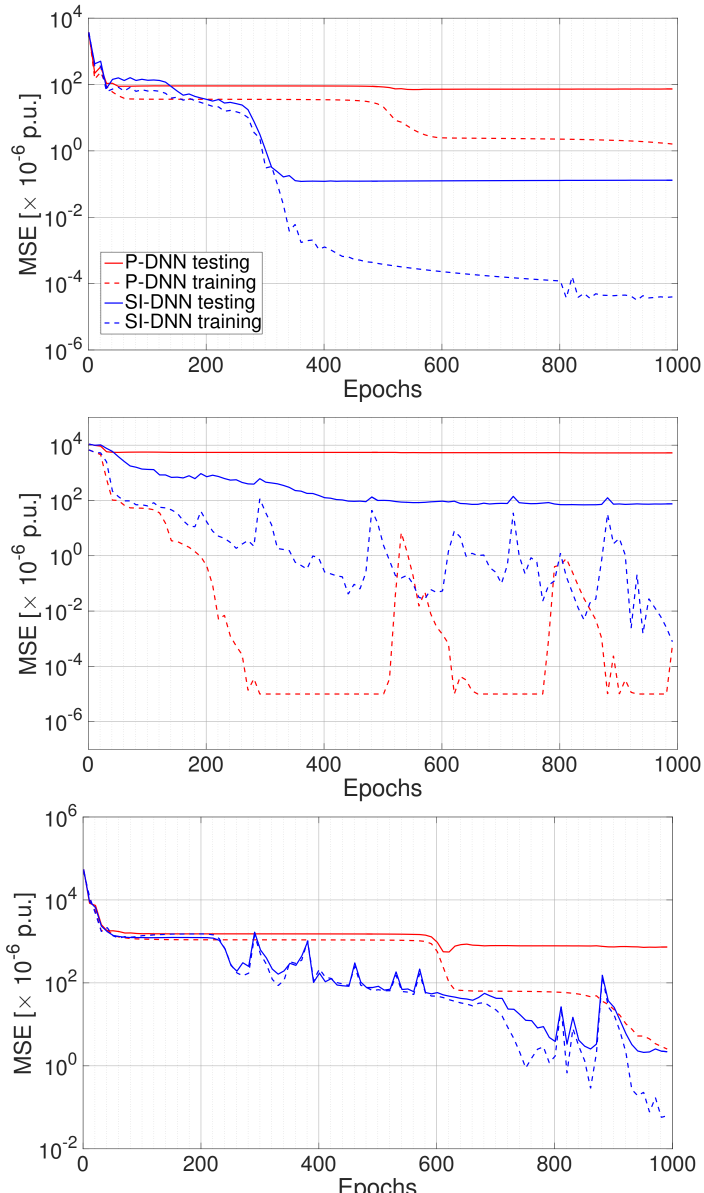

Figure 5.4: Training and testing errors across epochs in terms of MSE for hours 5 (top), 12 (middle), and 20 (bottom). Training was performed using 15-min smart meter data (4 OPF instances) to operate during the hour of interest.

Chapter 6

Learning to Solve the AC-OPF using Sensitivity-Informed Deep Neural Networks

6.1 Publication Details

M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to solve the ac-opf using sensitivity-informed deep neural networks," 2021, (submitted). [Online]. Available: https://arxiv.org/abs/2103.14779

6.2 Abstract

To shift the computational burden from real-time to offline in delay-critical power systems applications, recent works entertain the idea of using a deep neural network (DNN) to predict the solutions of the AC optimal power flow (AC-OPF) once presented load demands. As network topologies may change, training this DNN in a sample-efficient manner becomes a necessity. To improve data efficiency, this work utilizes the fact OPF data are not simple training labels, but constitute the solutions of a parametric optimization problem. We thus

advocate training a sensitivity-informed DNN (SI-DNN) to match not only the OPF optimizers, but also their partial derivatives with respect to the OPF parameters (loads). It is shown that the required Jacobian matrices do exist under mild conditions, and can be readily computed from the related primal/dual solutions. The proposed SI-DNN is compatible with a broad range of OPF solvers, including a non-convex quadratically constrained quadratic program (QCQP), its semidefinite program (SDP) relaxation, and MATPOWER; while SI-DNN can be seamlessly integrated in other learning-to-OPF schemes. Numerical tests on three benchmark power systems corroborate the advanced generalization and constraint satisfaction capabilities for the OPF solutions predicted by an SI-DNN over a conventionally trained DNN, especially in low-data setups.

6.3 Introduction

Power system operation involves routinely solving various renditions of the optimal power flow (OPF) task. Physical expansion of power networks, increasing number of dispatchable resources, and renewable generation-induced volatility call for solving large-scale OPF problems frequently. These problems naturally involve nonlinear alternating-current (AC) power flow equations as constraints. Such formulations, referred to as AC-OPF, are non-convex and are often computationally too expensive for real-time applications. Approximate formulations such as the linearized or so termed DC-OPF serve as the pragmatic resort for such setups.

A concerted effort towards handling AC-OPF efficiently has yielded popular nonlinear solvers such as MATPOWER [153], and efficient conic relaxations with optimality guarantees for frequently encountered problem instances [8], [25], [80], [79]. Despite significant advancements in numerical solvers, scalability of the AC-OPF may still be a challenge, particularly in online, combinatorial, and stochastic settings [137], [51]. To alleviate these issues, there has been growing interest in developing machine learning-based approaches (particularly deep learning) for OPF [137], [51], [96], [59], [141]; see [148] for recent applications.

To entirely bypass numerical solvers for the OPF, one can opt for unsupervised learning approaches, such as the ones suggested in [94], [60], and [73]. However, the performance of OPF solvers for offline data generation and the availability of historical data with power utilities adequately motivate supervised learning. Nevertheless, there are two main challenges central to learning for OPF. First, traditional deep learning approaches are not amenable to enforcing constraints even for the training set. Predictions for OPF minimizers may have limited standalone value if the related constraints are violated. Second, power systems undergo frequent topological and operational changes that require retraining deep neural networks (DNN), potentially at short intervals. Deep learning-based approaches are traditionally data-intense and frequent retraining for large systems may be prohibitive.

To cope with the first challenge, a DNN may be engaged to better initialize (warm-start) existing numerical solvers [141], or to predict active constraints and thus result in an OPF model with much reduced number of constraints [59], [33], [42]. Another group of approaches targets constraint satisfaction by penalizing constraint violations and the related Karush–Kuhn–Tucker (KKT) conditions [96], or explicitly resorting to a Lagrangian dual scheme for DNN training [51], [60], [139]. The third alternative involves post-processing DNN predictions by projecting them using a power flow solver [96], [141]. Although the projected point satisfies the power flow equations, it may still violate engineering limits.

To cater to the second challenge of frequent model changes, sample-efficient learning models that generalize well are well motivated. One way to achieve that is by prudently designing the DNN architecture using prior information. For instance, by seeking an input-convex DNN when the underlying input-output mapping is convex [145], or using DNNs that *unroll* an iterative optimization algorithm [144], or adopting graph-based priors [94], [138]. The previously mentioned approaches that design objective functions based on OPF constraints and KKT conditions may also be seen as including prior knowledge in training, thus enhancing learnability.

Recent works in the general area of *physics-informed learning* aim at incorporating prior information on the underlying data, not occurring in conventional training datasets [99]. Specifically, for learning solutions of differential equations, the derivatives of DNN output with respect to input carry obvious virtue. Suitably utilizing these derivatives yields significant advantages in such applications [99], [89]. While for differential equations derivatives with respect to time and/or space dimensions emerge naturally, sensitivities in an optimization problem have been underappreciated. To this end, our recent work proposed a novel approach of training sensitivity informed DNNs (SI-DNN), intended to learn OPF solutions [111]. Training SI-DNNs requires computing the sensitivities of OPF minimizers with respect to the input parameters. For DC-OPF posed as linear or quadratic programs, computing these sensitivities is simpler, and using these to train DNNs yielded remarkable improvements; see [111].

The contributions of this work are on four fronts: *c1)* We put forth a novel approach for training DNNs to predict AC-OPF solutions by matching not only the OPF minimizers, but also their sensitivities (partial derivatives collected in a Jacobian matrix) with respect to the OPF problem parameters, such as load demands. *c2)* We compute the desired sensitivities for general nonlinear OPF formulations building upon classical works on perturbation analysis of continuous optimization problems [22], [50], [39]. Existing sensitivity results assume certain constraint qualifications that may not be satisfied by OPF instances. *c3)* We relax such assumptions and establish that the sensitivities of primal OPF solutions do exist under significantly milder conditions. *c4)* In pursuit of globally optimal OPF solutions for training a DNN, we also study the SDP formulation of the AC-OPF and compute its sensitivities upon leveraging the sensitivities of the related QCQP. Such shortcut obviates the difficulty of differentiating a conic program.

The rest of the paper is organized as follows: Section 6.4 presents the key methodology for SI-DNNs. Section 6.5 poses the AC-OPF as a parametric optimization taking the form of a non-convex QCQP. Section 6.6 establishes that the sought sensivities of the AC-OPF exist under mild conditions and explains how they can be readily computed. Section 6.7 computes the sensitivities of globally optimal AC-OPF solutions obtained via the SDP relaxation of the OPF. The numerical tests of Section 6.8 contrast SI-DNN to conventionally trained DNNs using datasets generated by MATPOWER and the SDP-based OPF solver on three benchmark power systems. Conclusions are drawn in Section 6.9.

Notation: lower- (upper-) case boldface letters denote column vectors (matrices). Calligraphic symbols are reserved for sets. Symbol $^\top$ stands for transposition, vectors $\mathbf{0}$ and $\mathbf{1}$ are the all-zeros and all-ones vectors or matrices, and $\mathbf{e}_n$ is the n-th canonical vector of appropriate dimensions implied by the context. Operator $\mathcal{D}(\mathbf{x})$ returns a diagonal matrix with the entries of vector $\mathbf{x}$ on its main diagonal.

6.4 Sensitivity-Informed Training for DNNs

Consider the task of optimally dispatching generators and flexible loads to meet power balance constraints $\mathbf{h}(\cdot)$ while enforcing engineering limits captured by $\mathbf{g}(\cdot)$. OPF boils down to a *parametric optimization problem* that has to be solved routinely for different values of the problem parameters being the time-varying renewable generation and loads. The problem can be abstracted as: Given a vector $\boldsymbol{\theta} \in \mathbb{R}^P$ of problem parameters or inputs, find an optimal dispatch $\mathbf{x}_{\boldsymbol{\theta}} \in \mathbb{R}^N$ consisting of voltages and generator setpoints as the minimizer

$$\begin{aligned}\mathbf{x}_{\boldsymbol{\theta}} \in \arg\min_{\mathbf{x}} \quad & f(\mathbf{x};\boldsymbol{\theta}) \\ \text{s.to} \quad & \mathbf{h}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{0} : \ \boldsymbol{\lambda}_{\boldsymbol{\theta}} \\ & \mathbf{g}(\mathbf{x};\boldsymbol{\theta}) \leq \mathbf{0} : \ \boldsymbol{\mu}_{\boldsymbol{\theta}}\end{aligned} \tag{$P_{\boldsymbol{\theta}}$}$$

where functions $f(\mathbf{x};\boldsymbol{\theta})$, $\mathbf{h}(\mathbf{x};\boldsymbol{\theta})$, and $\mathbf{g}(\mathbf{x};\boldsymbol{\theta})$ are continuously differentiable with respect to $\mathbf{x}$ and $\boldsymbol{\theta}$; and vectors $(\boldsymbol{\lambda}_{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\boldsymbol{\theta}})$ collect the optimal dual variables corresponding to the equality and inequality constraints, respectively.

To save on running time and computational resources, rather than solving ($P_{\boldsymbol{\theta}}$), one can adopt a *learning-to-optimize approach* and train a learning model such as a DNN to predict approximate solutions of ($P_{\boldsymbol{\theta}}$); see e.g., [119]. Once presented with an instance of $\boldsymbol{\theta}$, this DNN can be trained to output a predictor $\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w})$ of $\mathbf{x}_{\boldsymbol{\theta}}$. The DNN is parameterized by weights $\mathbf{w}$, which can be selected upon minimizing a suitable distance metric or loss function between $\mathbf{x}_{\boldsymbol{\theta}}$ and $\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w})$ over a training set.

Given a choice for a DNN architecture, the straightforward approach for learning-to-optimize entails two steps:

S1) Building a labeled training dataset $\{\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s}\}_{s=1}^S$ by solving S instances of ($P_{\boldsymbol{\theta}}$); and

S2) Learning $\mathbf{w}$ by minimizing a data fitting loss over the training dataset as

$$\min_{\mathbf{w}} \sum_{s=1}^{S} \ell\left(\hat{\mathbf{x}}(\boldsymbol{\theta}_s;\mathbf{w}), \mathbf{x}_{\boldsymbol{\theta}_s}\right). \tag{6.1}$$

For a regression task such as the one considered here, commonly used loss functions ℓ include the mean squared error (MSE) $\|\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w}) - \mathbf{x}_{\boldsymbol{\theta}}\|_2^2$ or the mean absolute error (MAE) $\|\hat{\mathbf{x}}(\boldsymbol{\theta};\mathbf{w}) -$

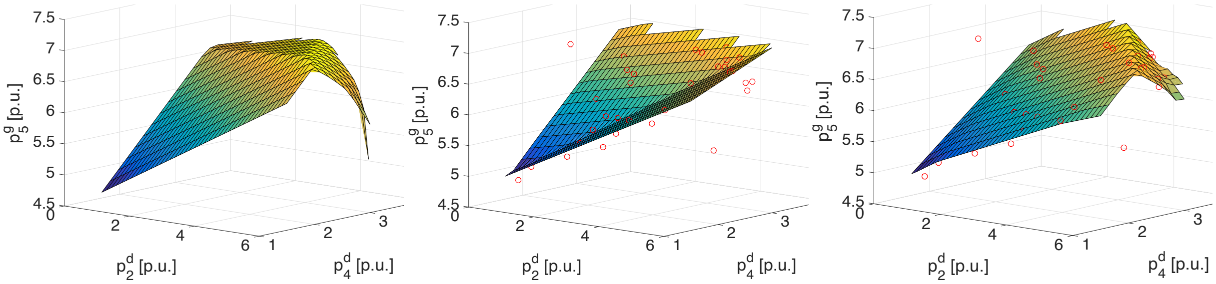

Figure 6.1: The left panel depicts the optimal generation dispatch $p_5^g(\boldsymbol{\theta})$ for bus 5 as a function of load demands at buses 2 and 4, that is $\boldsymbol{\theta} := [p_2^d\ p_4^d]^\top$. Sampling the parameter space of $\boldsymbol{\theta}$'s provided 523 feasible OPF instances, of which 37 instances constituted the training set. The center and right panel show the dispatches learned by a P-DNN and an SI-DNN. For the two DNNs, we used the same training points (red circles), architecture (two hidden layers with 16 neurons each), optimizer, and learning rates. P-DNN fails to learn the drop in p_5^g for larger load demands.

$\mathbf{x}_{\boldsymbol{\theta}}\|_1$. We refer to a DNN trained by solving (6.1) as a *plain DNN or P-DNN* for short. The conventional P-DNN approach focuses merely on the dataset $\{\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s}\}_{s=1}^S$, and is oblivious of any additional properties the mapping $\boldsymbol{\theta} \rightarrow \mathbf{x}_{\boldsymbol{\theta}}$ induced by ($P_{\boldsymbol{\theta}}$) bears.

The key idea here is to extend each training data pair $(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s})$ as $(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s}, \mathbf{J}_{\boldsymbol{\theta}_s})$, where $\mathbf{J}_{\boldsymbol{\theta}_s} := [\nabla_{\boldsymbol{\theta}} \mathbf{x}_{\boldsymbol{\theta}}]_{\boldsymbol{\theta}=\boldsymbol{\theta}_s}$ is the Jacobian matrix carrying the partial derivatives of the minimizer $\mathbf{x}_{\boldsymbol{\theta}_s}$ with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_s$ assuming such sensitivities exist. To incorporate the sensitivity information into DNN training, we propose augmenting the loss function with an additional fitting term as

$$\min_{\mathbf{w}} \sum_{s=1}^{S} \|\hat{\mathbf{x}}(\boldsymbol{\theta}_s; \mathbf{w}) - \mathbf{x}_{\boldsymbol{\theta}_s}\|_2^2 + \rho \|\hat{\mathbf{J}}(\boldsymbol{\theta}_s; \mathbf{w}) - \mathbf{J}_{\boldsymbol{\theta}_s}\|_F^2 \tag{6.2}$$

where $\rho > 0$ is a scalar weight and $\|\cdot\|_F$ denotes the matrix Frobenius norm. The DNN trained by solving (6.2) aims to match not only the target output $\mathbf{x}_{\boldsymbol{\theta}}$, but also the sensitivities of $\mathbf{x}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$. We term this neural network a *sensitivity-informed DNN or SI-DNN*.

To motivate the inclusion of sensitivities, let us provide some geometric intuition. An SI-

DNN describes a mapping whose output is intended to agree with label $\mathbf{x}_{\boldsymbol{\theta}_s}$ for an input $\boldsymbol{\theta}_s$ of the training set. Additionally, the learned mapping serves as a local first-order affine approximation to the $\boldsymbol{\theta} \rightarrow \mathbf{x}_{\boldsymbol{\theta}}$ mapping. SI-DNNs for learning optimizers were first introduced for solving multiparametric QPs (MPQP) in the conference precursor of this work [111]. For MPQPs, the minimizer $\mathbf{x}_{\boldsymbol{\theta}}$ is known to be a piecewise affine function of $\boldsymbol{\theta}$ [24], [121]. Hence, a DNN with rectified linear unit (ReLU) activations is well-motivated as it can describe such mapping. If hypothetically trained to zero training error, this SI-DNN would yield perfectly accurate predictions in a neighborhood of each training datum $\boldsymbol{\theta}_s$. Numerical tests showed improvements of 2-3 orders of magnitude for SI-DNN over P-DNN in inferring MPQP solutions [111].

This work advocates that the sample efficiency benefit of SI-DNN over P-DNN goes well beyond MPQPs. Before going into the details, we provide an illustrative example on the PJM 5-bus power system, which we dispatched via AC-OPF by varying the active load demands $\boldsymbol{\theta} := [p_2^d \ p_4^d]^\top$ on buses 2 and 4 within [1.5, 3.75] per unit (pu) and [0.4, 6] pu, respectively. Fixing the other demands, we dispatched the generators at buses 1 and 5, and removed other generators. Fig. 6.1 depicts the enhanced performance of SI-DNN over P-DNN in predicting p_5^g, the optimal dispatch at bus 5.

This new learning-to-optimize approach alters the two steps of the P-DNN workflow as follows. For step *S1)*, in addition to the minimizer $\mathbf{x}_{\boldsymbol{\theta}_s}$, we now have to compute the Jacobian matrix $\mathbf{J}_{\boldsymbol{\theta}_s}$ for all instances s, if such sensitivities exist. Sections 6.6–6.7 explain how and when such Jacobian matrices can be computed for a non-convex and a convexified rendition of the AC-OPF. The punchline is that obtaining $\mathbf{J}_{\boldsymbol{\theta}}$ requires minimal additional computational effort and no intervention to the OPF solver. Once the primal/dual solutions have been found by the OPF, finding $\mathbf{J}_{\boldsymbol{\theta}}$ is as simple as solving a linear system of equations. For step *S2)*, we migrate from solving (6.1) to (6.2). Matrix $\mathbf{J}_{\boldsymbol{\theta}_s}$ is a constant that has been evaluated for all s during step *S1)*. Matrix $\hat{\mathbf{J}}(\boldsymbol{\theta}_s;\mathbf{w})$ on the other hand is a function of $\mathbf{w}$ and is not straightforward to compute. Fortunately, computing $\hat{\mathbf{J}}(\boldsymbol{\theta}_s;\mathbf{w})$ can be performed efficiently thanks to advances in automatic differentiation [12]. Modeling (6.2) in existing DNN software platforms (such as TensorFlow) is almost as easy as modeling (6.1) modulo the coding modifications deferred to Appendix 6.9. Our numerical tests of Section 6.8 further show that the extra computational time for solving (6.2) is modest. Before computing $\mathbf{J}_{\boldsymbol{\theta}}$, we first pose AC-OPF as a parametric QCQP.

6.5 AC-OPF as a Parametric QCQP

A power network with N_b buses can be represented by an undirected connected graph $\mathcal{G} = (\mathcal{N},\mathcal{E})$, whose nodes $n \in \mathcal{N} := \{1,\ldots,N_b\}$ correspond to buses, and edges $e = (n,k) \in \mathcal{E}$ to transmission lines, with cardinality $|\mathcal{E}| := E$. Given line impedances, one can derive the $N_b \times N_b$ bus admittance matrix $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$. Let $v_n = v_n^r + jv_n^i$ and $p_n + jq_n$ denote respectively the complex voltage and power injection at bus $n \in \mathcal{N}$. Power injections are quadratically related to bus voltages through the power flow equations

$$p_n = \sum_{k=1}^{N_b} v_n^r(v_k^r G_{nk} - v_k^i B_{nk}) + v_n^i(v_k^i G_{nk} + v_k^r B_{nk})$$
$$q_n = \sum_{k=1}^{N_b} v_n^i(v_k^r G_{nk} - v_k^i B_{nk}) - v_n^r(v_k^i G_{nk} + v_k^r B_{nk}).$$

If $\mathbf{v} \in \mathbb{R}^{2N_b}$ collects the real and imaginary parts of nodal voltages as $\mathbf{v} := [(\mathbf{v}^r)^\top \ (\mathbf{v}^i)^\top]^\top$ with $\mathbf{v}^r := \{v_n^r\}_{n=1}^{N_b}$ and $\mathbf{v}^i := \{v_n^i\}_{n=1}^{N_b}$, the power flow equations can be written as

$$p_n = \mathbf{v}^\top \mathbf{M}_{p_n} \mathbf{v} \tag{6.3a}$$

$$q_n = \mathbf{v}^\top \mathbf{M}_{q_n} \mathbf{v} \tag{6.3b}$$

where $\mathbf{M}_{p_n}$ and $\mathbf{M}_{q_n}$ are $2N_b \times 2N_b$ symmetric real-valued matrices [69]. Squared voltage magnitudes can also be expressed as quadratic functions of $\mathbf{v}$ as

$$|v_n^r + jv_n^i|^2 = \mathbf{v}^\top \mathbf{M}_{v_n} \mathbf{v} \tag{6.4}$$

where $\mathbf{M}_{v_n} := \mathbf{e}_n \mathbf{e}_n^\top + \mathbf{e}_{N_b+n} \mathbf{e}_{N_b+n}^\top$. The same holds true for the squared magnitude of line currents. If y_{mn} is the series impedance of line $(m, n) \in \mathcal{E}$, the current flowing on this line is $\tilde{i}_{mn} = (\tilde{v}_m - \tilde{v}_n) y_{mn}$, and thus,

$$|\tilde{i}_{mn}|^2 = \mathbf{v}^\top \mathbf{M}_{i_{mn}} \mathbf{v} \tag{6.5}$$

where $\mathbf{M}_{i_{mn}} := |y_{mn}|^2 (\mathbf{e}_m - \mathbf{e}_n)(\mathbf{e}_m - \mathbf{e}_n)^\top + |y_{mn}|^2 (\mathbf{e}_{N_b+m} - \mathbf{e}_{N_b+n})(\mathbf{e}_{N_b+m} - \mathbf{e}_{N_b+n})^\top$.

The active power injected into bus n can be decomposed into a dispatchable component p_n^g and an inflexible component p_n^d as $p_n = p_n^g - p_n^d$. The former captures the active power dispatch of a generator or a flexible load located at bus n. The latter captures the inelastic load to be served at bus n. To simplify the exposition, each bus is assumed to be hosting at most one dispatchable resource (generator or flexible load). The reactive power injected into bus n is decomposed similarly as $q_n = q_n^g - q_n^d$. Let $\mathcal{N}_g \subseteq \mathcal{N}$ be the subset of buses hosting dispatchable power injections with cardinality N_g. Bus $n = 1$ belongs to $\mathcal{N}_g$ and serves as the angle reference, so that $v_1^i = \mathbf{v}^\top \mathbf{e}_{N_b+1} \mathbf{e}_{N_b+1}^\top \mathbf{v} = 0$. The remaining buses host non-flexible loads and constitute the subset $\mathcal{N}_\ell = \mathcal{N} \setminus \mathcal{N}_g$ with cardinality $N_l = N_b - N_g$. For simplicity, we will henceforth term the buses in $\mathcal{N}_g$ as *generator buses*, and the ones in $\mathcal{N}_\ell$ as *load buses*. Zero-injection (junction) buses belong to $\mathcal{N}_\ell$ and satisfy $p_n^g = p_n^d = q_n^g = q_n^d = 0$.

Given the inflexible loads at all buses $\{p_n^d, q_n^d\}_{n\in\mathcal{N}}$, the OPF problem aims at optimally dispatching generators and flexible loads $\{p_n^g, q_n^g\}_{n\in\mathcal{N}_g}$ while meeting resource and network limits. The OPF can be formulated as the QCQP [25], [80]

$$\min \sum_{n\in\mathcal{N}_g} c_n^p p_n^g + c_n^q q_n^g \tag{P1}$$

$$\text{over } \mathbf{v}\in\mathbb{R}^{2N_b}, \{p_n^g, q_n^g\}_{n\in\mathcal{N}_g}$$

$$\text{s.to } \mathbf{v}^\top \mathbf{M}_{p_n}\mathbf{v} = p_n^g - p_n^d, \quad \forall\, n\in\mathcal{N}_g \tag{6.6a}$$

$$\mathbf{v}^\top \mathbf{M}_{q_n}\mathbf{v} = q_n^g - q_n^d, \quad \forall\, n\in\mathcal{N}_g \tag{6.6b}$$

$$\mathbf{v}^\top \mathbf{M}_{p_n}\mathbf{v} = -p_n^d, \quad \forall\, n\in\mathcal{N}_\ell \tag{6.6c}$$

$$\mathbf{v}^\top \mathbf{M}_{q_n}\mathbf{v} = -q_n^d, \quad \forall\, n\in\mathcal{N}_\ell \tag{6.6d}$$

$$\underline{p}_n^g \le \mathbf{v}^\top \mathbf{M}_{p_n}\mathbf{v} + p_n^d \le \bar{p}_n^g, \quad \forall\, n\in\mathcal{N}_g \tag{6.6e}$$

$$\underline{q}_n^g \le \mathbf{v}^\top \mathbf{M}_{q_n}\mathbf{v} + q_n^d \le \bar{q}_n^g, \quad \forall\, n\in\mathcal{N}_g \tag{6.6f}$$

$$\underline{v}_n \le \mathbf{v}^\top \mathbf{M}_{v_n}\mathbf{v} \le \bar{v}_n, \quad \forall\, n\in\mathcal{N} \tag{6.6g}$$

$$\mathbf{v}^\top \mathbf{e}_{N+1}\mathbf{e}_{N+1}^\top \mathbf{v} = 0 \tag{6.6h}$$

$$\mathbf{v}^\top \mathbf{M}_{i_{mn}}\mathbf{v} \le \bar{i}_{mn}, \quad \forall\, (m,n)\in\mathcal{E} \tag{6.6i}$$

where (c_n^p, c_n^q) are the coefficients for generation cost or the utility function for flexible load at bus n. Constraints (6.6a)–(6.6d) enforce the power flow equations at load and generator buses. Constraints (6.6e)–(6.6f) impose limits for generators and flexible loads. Constraints (6.6g) confine squared voltage magnitudes within given ranges and (6.6h) identifies the reference bus. Finally, constraint (6.6i) limits squared current magnitudes according to line ratings.

Problem (6.6) is a parametric QCQP as it needs to be solved for different values of demands $\{p_n^d, q_n^d\}_{n\in\mathcal{N}}$; costs $\{c_n\}_{n\in\mathcal{N}_g}$; and generation capacities $\{\underline{p}_n^g, \bar{p}_n^g, \underline{q}_n^g, \bar{q}_n^g\}$. Voltage limits $\{\underline{v}_n, \bar{v}_n\}$ for $n \in \mathcal{N}$ and current limits $\{\bar{i}_{(m,n)}\}$ for $(m,n) \in \mathcal{E}$ may also be changing due to normal and emergency ratings. To keep the exposition uncluttered, we henceforth fix all but the inelastic demands to known values. In other words, we are interested in solving (6.6) over different values of the parameter vector $\boldsymbol{\theta} := \{p_n^d, q_n^d\}_{n\in\mathcal{N}} \in \mathbb{R}^{2L}$.

The optimization variables of (6.6) consist of all nodal voltages $\mathbf{v}$ and the (re)active power schedules for generators. If vector $\mathbf{x}_g$ collects generator schedules $\{p_n^g, q_n^g\}_{n\in\mathcal{N}_g}$, then vector $\mathbf{x}_{\boldsymbol{\theta}}^\top := [\mathbf{v}^\top \ \mathbf{x}_g^\top]$ denotes the minimizer of (6.6) for the specific parameter vector $\boldsymbol{\theta}$. Aiming for the complete $\mathbf{x}_{\boldsymbol{\theta}}$ is apparently an over-parameterization of the problem, adopted only to ease the formulation in (6.6). What the system operator actually needs to know in practice is only the voltage magnitude and active power schedule for each generator (modulo the reference generator for which we set the voltage magnitude and angle). In light of this and to reduce the DNN output dimension, the DNN is trained to predict the PV setpoints for generators. Given the predicted quantities and knowing the values for inflexible loads from $\boldsymbol{\theta}$, the remaining quantities can be readily computed using a power flow solver anyway.

Given a set of (locally) optimal primal/dual solutions for (6.6), we next analyze the sensitivity of the parametric AC-OPF. Sensitivities are computed for the complete $\mathbf{x}_{\boldsymbol{\theta}}$, from which the Jacobian $\mathbf{J}_{\boldsymbol{\theta}}$ needed in (6.2) can be readily obtained.

6.6 Sensitivity Analysis for QCQP-based OPF

To analyze the sensitivity of (6.6) with respect to $\boldsymbol{\theta}$, let us first express the vectors of complex power injections across all buses as

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = \mathbf{A}\mathbf{x}_g + \mathbf{B}\boldsymbol{\theta} \tag{6.7}$$

where $\mathbf{x}_g$ stacks the generator (re)active power injections $\{p_n^g, q_n^g\}_{n\in\mathcal{N}_g}$ and $(\mathbf{A}, \mathbf{B})$ are matrices assigning generators and loads to buses. We then reformulate (6.6) as

$$\min_{\mathbf{v},\mathbf{x}_g} \quad \mathbf{a}_0^\top \mathbf{x}_g \tag{6.8a}$$

$$\text{s.to} \quad \mathbf{v}^\top \mathbf{L}_\ell \mathbf{v} = \mathbf{a}_\ell^\top \mathbf{x}_g + \mathbf{b}_\ell^\top \boldsymbol{\theta}, \qquad \ell = 1 : L : \quad \lambda_\ell \tag{6.8b}$$

$$\mathbf{v}^\top \mathbf{M}_m \mathbf{v} \leq \mathbf{d}_m^\top \boldsymbol{\theta} + f_m, \qquad m = 1 : M : \quad \mu_m \tag{6.8c}$$

where $\mathbf{a}_0$ collects the generation cost coefficients (cf. (6.6)); the first $L = 2N_b + 1$ constraints in (6.8b) correspond to the power flow equations (6.6a)–(6.6d) and the angle reference constraint (6.6h), while constraints (6.8c) correspond to the $M = 4N_g + 2N_b + E$ inequality constraints of (6.6). Matrices $(\mathbf{L}_\ell, \mathbf{M}_m)$ are drawn from the $\mathbf{M}$ matrices appearing in the quadratic forms of (6.6). Vectors $(\mathbf{a}_\ell, \mathbf{b}_\ell)$ correspond to rows of matrices $(\mathbf{A}, \mathbf{B})$ in (6.7) for $\ell \leq 2N_b$; and $\mathbf{0}$ for $\ell = 2N_b + 1$. Vectors $\mathbf{d}_m$ are indicator (canonical) vectors and constants f_m relate to generation, voltage, and line limits.

Aiming at computing the sensitivity of a minimizer $\mathbf{x}_{\boldsymbol{\theta}}^\top = [\mathbf{v}^\top \ \mathbf{x}_g^\top]$ of (6.8) with respect to $\boldsymbol{\theta}$, we explored the related literature. Fortunately, there exists a rich corpus of work on perturbation analysis of continuous optimization problems with applications in operation research, economics, mechanics, and optimal control [22]. The first approaches applied the implicit function theorem to the related first-order optimality conditions [50]. Thereon, many developments have been made towards relaxing the assumptions of initial works, and expanding the scope to conic programs [39], [26], [22], [2]. For several recent applications however, the early approaches of [50] are well suited due to their simplicity; see for example [5]. Building upon [50], we next compute the sensitivities required for SI-DNN in Section 6.6.1; and relax some of the needed assumptions in Section 6.6.2.

6.6.1 Perturbing Optimal Primal/Dual Solutions

Towards instantiating ($P_{\boldsymbol{\theta}}$) with (6.8) and to reduce notational clutter, let us use symbols $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ to denote the *optimal* primal/dual variables $(\mathbf{x}_{\boldsymbol{\theta}}, \boldsymbol{\lambda}_{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\boldsymbol{\theta}})$ of (6.8) for a particular $\boldsymbol{\theta}$. Under mild technical assumptions, a local primal/dual point for (6.8) satisfies the first-order optimality conditions [39]. The goal of sensitivity analysis is to find infinitesimal changes $(\mathrm{d}\mathbf{x}, \mathrm{d}\boldsymbol{\lambda}, \mathrm{d}\boldsymbol{\mu})$, so that the perturbed point $(\mathbf{x}+\mathrm{d}\mathbf{x}, \boldsymbol{\lambda}+\mathrm{d}\boldsymbol{\lambda}, \boldsymbol{\mu}+\mathrm{d}\boldsymbol{\mu})$ still satisfies the first-order optimality conditions when the input parameters change from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}+\mathrm{d}\boldsymbol{\theta}$ [50]. To this end, we next review the optimality conditions and then differentiate them to compute the sought sensitivities.

The Lagrangian function of (6.8) is defined as

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}; \boldsymbol{\theta}) := \mathbf{a}_0^\top \mathbf{x}_g &+ \sum_{\ell=1}^{L} \lambda_\ell \left(\mathbf{v}^\top \mathbf{L}_\ell \mathbf{v} - \mathbf{a}_\ell^\top \mathbf{x}_g - \mathbf{b}_\ell^\top \boldsymbol{\theta} \right) \\ &+ \sum_{m=1}^{M} \mu_m \left(\mathbf{v}^\top \mathbf{M}_m \mathbf{v} - \mathbf{d}_m^\top \boldsymbol{\theta} - f_m \right).\end{aligned}$$

With $\mathbf{x} := \{\mathbf{v}, \mathbf{x}_g\}$, Lagrangian optimality $\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{0}$ gives

$$\underbrace{\left(\sum_{\ell=1}^{L} \lambda_\ell \mathbf{L}_\ell + \sum_{m=1}^{M} \mu_m \mathbf{M}_m\right)}_{:=\mathbf{Z}} \mathbf{v} = \mathbf{0}. \tag{6.9a}$$

$$\mathbf{a}_0 = \sum_{\ell=1}^{L} \lambda_\ell \mathbf{a}_\ell \tag{6.9b}$$

In addition to Lagrangian optimality, first-order optimality conditions include primal feasibility [cf. (6.8b)–(6.8c)], as well as complementary slackness and dual feasibility for all m:

$$\mu_m \underbrace{\left(\mathbf{v}^\top \mathbf{M}_m \mathbf{v} - \mathbf{d}_m^\top \boldsymbol{\theta} - f_m\right)}_{:=g_m} = 0 \tag{6.10a}$$

$$\mu_m \geq 0. \tag{6.10b}$$

From the aforementioned optimality conditions, let us focus on those that take the form of equalities, namely (6.9a)–(6.9b), (6.8b), and (6.10a). For these conditions, we will compute their total differentials. From the first three, we obtain

$$\mathbf{Z}\,\mathrm{d}\mathbf{v} + \mathbf{L}_\lambda\,\mathrm{d}\boldsymbol{\lambda} + \mathbf{M}_\mu\,\mathrm{d}\boldsymbol{\mu} = \mathbf{0} \tag{6.11a}$$

$$\mathbf{A}^\top\,\mathrm{d}\boldsymbol{\lambda} = \mathbf{0} \tag{6.11b}$$

$$2\mathbf{L}_\lambda^\top\,\mathrm{d}\mathbf{v} - \mathbf{A}\,\mathrm{d}\mathbf{x}_g - \mathbf{B}\,\mathrm{d}\boldsymbol{\theta} = \mathbf{0} \tag{6.11c}$$

where $\mathbf{L}_\lambda := \sum_{\ell=1}^{L} \mathbf{L}_\ell \mathbf{v}\mathbf{e}_\ell^\top$; and $\mathbf{M}_\mu := \sum_{m=1}^{M} \mathbf{M}_m \mathbf{v}\mathbf{e}_m^\top$.

For (6.10a), the total differential is

$$g_m\,\mathrm{d}\mu_m + \mu_m\,\mathrm{d}g_m = 0 \tag{6.12}$$

where $\mathrm{d}g_m := (\nabla_{\mathbf{v}} g_m)^\top \mathrm{d}\mathbf{v} + (\nabla_{\boldsymbol{\theta}} g_m)^\top \mathrm{d}\boldsymbol{\theta}$ for all m. We identify three cases:

c1) If $\mu_m = 0$ and $g_m < 0$, then (6.12) implies $\mathrm{d}\mu_m = 0$. It follows that: *i)* $\mu_m + \mathrm{d}\mu_m = 0$; *ii)* $(\mu_m + \mathrm{d}\mu_m)(g_m + \mathrm{d}g_m) = 0$; and *iii)* $g_m + \mathrm{d}g_m < 0$ for any small $\mathrm{d}g_m$. In conclusion, condition (6.12) ensures that the perturbed point satisfies conditions for optimality, including the inequalities from primal/dual feasibility.

c2) If $\mu_m > 0$ and $g_m = 0$, then (6.12) gives $\mathrm{d}g_m = 0$. It also follows that: *i)* $g_m + \mathrm{d}g_m = 0$; *ii)* $(\mu_m + \mathrm{d}\mu_m)(g_m + \mathrm{d}g_m) = 0$; and *iii)* $\mu_m + \mathrm{d}\mu_m > 0$ for any small $\mathrm{d}\mu_m$. As in case *c1)*, condition (6.12) ensures that the perturbed point satisfies all conditions for optimality.

c3) If $\mu_m = g_m = 0$, then (6.12) is inconclusive on $\mathrm{d}g_m$ and $\mathrm{d}\mu_m$. In this *degenerate* case, for the perturbed point to remain optimal, we need to explicitly impose: *i)* $\mathrm{d}g_m \leq 0$; *ii)* $\mathrm{d}\mu_m \geq 0$; and *iii)* $\mathrm{d}g_m\, \mathrm{d}\mu_m = 0$. Even though the three latter constraints can be handled by the sensitivity analysis of [39], [26], they considerably complicate the treatment. Moreover, such degeneracy is seldom encountered numerically. We henceforth rely on the so called *strict complementarity* assumption, which ignores case *c3)* [50].

Assumption 4. Given a tuple of optimal primal/dual variables $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, constraint $g_m(\mathbf{x}; \boldsymbol{\theta}) = 0$ if and only if $\mu_m > 0$.

Two observations are in order. First, the analysis under *c1)-c2)* reveals that although we perturbed only the equality conditions for optimality, the obtained perturbed point satisfies the inequality conditions for optimality as well. Therefore, under Assumption 4, the point $(\mathbf{x} + \mathrm{d}\mathbf{x}, \boldsymbol{\lambda} + \mathrm{d}\boldsymbol{\lambda}, \boldsymbol{\mu} + \mathrm{d}\boldsymbol{\mu})$ satisfying the perturbed optimality conditions is (locally) optimal for (6.8), when solved for $\boldsymbol{\theta} + \mathrm{d}\boldsymbol{\theta}$. Second, despite Assumption 4, if a degenerate instance of (6.8) does occur for some $\boldsymbol{\theta}_s$ in the training dataset, the particular pair $(\boldsymbol{\theta}_s, \mathbf{x}_{\boldsymbol{\theta}_s})$ can still be used to train the SI-DNN, yet without the additional sensitivity information. In other words, degenerate instances can contribute only to the first fitting term of (6.2).

Applying (6.12) for all m, the total derivatives for (6.10a) can be compactly expressed as

$$\mathcal{D}(\mathbf{g})\,\mathrm{d}\boldsymbol{\mu} + 2\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_{\mu}^{\top}\,\mathrm{d}\mathbf{v} - \mathbf{D}^{\top}\,\mathrm{d}\boldsymbol{\theta} = \mathbf{0}, \tag{6.13}$$

where $\mathbf{g} := \{g_m\}_{m=1}^{M}$ stacks the inequality constraint values, and matrix $\mathbf{D} := \sum_{m=1}^{M}\mu_m\mathbf{d}_m\mathbf{e}_m^{\top}$. Operator $\mathcal{D}(\mathbf{x})$ returns a diagonal matrix with vector $\mathbf{x}$ on its main diagonal. Conditions (6.11) and (6.13) can be collected in matrix-vector form as

$$\underbrace{\begin{bmatrix} \mathbf{Z} & \mathbf{0} & \mathbf{L}_{\lambda} & \mathbf{M}_{\mu} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^{\top} & \mathbf{0} \\ 2\mathbf{L}_{\lambda}^{\top} & -\mathbf{A} & \mathbf{0} & \mathbf{0} \\ 2\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_{\mu}^{\top} & \mathbf{0} & \mathbf{0} & \mathcal{D}(\mathbf{g}) \end{bmatrix}}_{:=\mathbf{S}} \begin{bmatrix} \mathrm{d}\mathbf{v} \\ \mathrm{d}\mathbf{x}_g \\ \mathrm{d}\boldsymbol{\lambda} \\ \mathrm{d}\boldsymbol{\mu} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{B} \\ \mathbf{D}^{\top} \end{bmatrix}}_{:=\mathbf{U}} \mathrm{d}\boldsymbol{\theta} \tag{6.14}$$

To compute the sensitivities of primal and dual variables with respect to the p-th entry θ_p of $\boldsymbol{\theta}$, we need to solve the previous system of $2(N_b + N_g) + L + M$ linear equations for $\mathrm{d}\boldsymbol{\theta} = \mathbf{e}_p$. The size of the system can be reduced by dropping the numerous inactive inequality constraints of (6.8) for which $\mu_m = 0$ and $g_m < 0$, and thus, $\mathrm{d}\mu_m = 0$ as discussed earlier under case *c1)*. Notably, if matrix $\mathbf{S}$ is invertible, the aforementioned sensitivities can all be found at once using the respective blocks of $\mathbf{S}^{-1}\mathbf{U}\mathbf{e}_p$. We next address two relevant questions: *q1) When is* $\mathbf{S}$ *invertible?*; and *q2) What are the implications of a singular* $\mathbf{S}$*?*

6.6.2 Existence of Primal Sensitivities

To address *q1)* for an arbitrary ($P_{\boldsymbol{\theta}}$), the existing literature identifies some assumptions on $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}; \boldsymbol{\theta})$. We first review these assumptions, and then assess if they are reasonable for the OPF task at hand. Given an optimal primal $\mathbf{x}$ for some $\boldsymbol{\theta}$, let $\mathcal{A}(\mathbf{x})$ denote the subset of inequality constraints of $\mathbf{g}(\mathbf{x}; \boldsymbol{\theta}) \leq \mathbf{0}$ that are *active or binding*, that is $\mathcal{A}(\mathbf{x}) := \{m : g_m(\mathbf{x}; \boldsymbol{\theta}) = 0\}$. A primal solution $\mathbf{x}$ is termed *regular* if the next assumption holds.

Assumption 5. The vectors $\{\nabla_{\mathbf{x}} h_\ell\}_{\forall \ell}$ and $\{\nabla_{\mathbf{x}} g_m\}_{m \in \mathcal{A}(\mathbf{x})}$ are linearly independent.

For the OPF in (6.8), the functions h_ℓ and g_m correspond to the (in)equality constraints (6.8b)–(6.8c) written in the standard form as in ($P_{\boldsymbol{\theta}}$). Assumption 5 is often referred to as linearly independent constraint qualification (LICQ). If a (locally) optimal $\mathbf{x}$ satisfies the LICQ, the corresponding optimal dual variables $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ are known to be unique [17]. In addition to satisfying first-order optimality conditions, a sufficient condition for $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}; \boldsymbol{\theta})$ to be (locally) optimal is often provided by the following second-order optimality condition.

Assumption 6. For a subspace orthogonal to the subspace spanned by the gradients of active constraints

$$\mathcal{Z} := \left\{\mathbf{z} : \mathbf{z}^\top \nabla_{\mathbf{x}} h_\ell = 0 \ \forall \ \ell, \mathbf{z}^\top \nabla_{\mathbf{x}} g_m = 0 \ \forall \ m \in \mathcal{A}(\mathbf{x})\right\}$$

it holds that $\mathbf{z}^\top \nabla^2_{\mathbf{xx}} \mathcal{L} \mathbf{z} >$ for all $z \in \mathcal{Z} \setminus \{\mathbf{0}\}$.

Under the strict complementarity, regularity, and second-order optimality conditions, matrix $\mathbf{S}$ is guaranteed to be invertible; see Theorem 2.1 and Corollary 2.1 of [50].

Lemma 6.1 ([50])**.** *If Assumptions 4–6 hold, matrix* $\mathbf{S}^{-1}$ *exists.*

Lemma 6.1 implies that under the stated assumptions, the optimal primal and dual variables of ($P_{\boldsymbol{\theta}}$) vary smoothly with changes in parameter $\boldsymbol{\theta}$, and the associated sensitivities can be found via (6.14). While Assumptions 4–6 seem to be standard in the optimization literature, our recent work on the optimal dispatch of inverters in distribution grids demonstrated analytically and numerically that LICQ (Assumption 5) is violated frequently [111]. Instances violating LICQ can be conceived for the AC-OPF in (6.8) too. To bring up one such example, consider a power system where a load bus m is connected to the rest of the system through another bus n via a single transmission line (m, n). As bus m is a load bus, it contributes two equality constraints (6.6c) and (6.6d). It can be shown that if any of the three following scenarios occurs, LICQ fails: *i)* the voltage limits in (6.6g) become binding (above or below) for both buses m and n; *ii)* line (m, n) becomes congested [cf. (6.6i)] and a voltage limit at bus m becomes binding; or *iii)* line (m, n) becomes congested and a voltage limit at bus n becomes binding. Attempting to circumvent LICQ violation via problem reformulations may be futile as their occurrences depend on $\boldsymbol{\theta}$, and are thus hard to analyze. This critical observation leads us to question *q2)*.

The implications of a singular $\mathbf{S}$ have previously been investigated in [39] and [26]: When LICQ is violated despite strict complementarity, the sensitivities of some primal/dual variables may still exist with respect to a θ_p. In detail, consider the set $\Gamma := \{\boldsymbol{\gamma} \in \mathbb{R}^{2(N_b+N_g)+L+M} : \mathbf{S}\boldsymbol{\gamma} = \mathbf{U}\mathbf{e}_p\}$, which is the solution set of (6.14). If the n-th entry of $\boldsymbol{\gamma}$ remains constant for all $\boldsymbol{\gamma} \in \Gamma$, the sensitivity of the n-th entry of $[\mathbf{x}^\top\ \boldsymbol{\lambda}^\top\ \boldsymbol{\mu}^\top]^\top$ with respect to θ_p does exist. Since for training an SI-DNN, we are interested only in the sensitivities $\nabla_{\boldsymbol{\theta}}\mathbf{x}$, we need to ensure that all solutions $\boldsymbol{\gamma} \in \Gamma$ share the same first N entries. This is equivalent to saying that the first entries of $\mathbf{n}$ are zero for all $\mathbf{n} \in \text{null}(\mathbf{S})$. The equivalence stems from the fact that if $\mathbf{S}\bar{\boldsymbol{\gamma}} = \mathbf{u}$ for a $\bar{\boldsymbol{\gamma}}$, any other solution to $\mathbf{S}\boldsymbol{\gamma} = \mathbf{u}$ takes the form $\boldsymbol{\gamma} = \bar{\boldsymbol{\gamma}} + \mathbf{n}$ for some $\mathbf{n} \in \text{null}(\mathbf{S})$. The next claim provides sufficient conditions for the first N entries of $\mathbf{n}$ to be zero.

Theorem 6.2. *If Assumptions 4 and 6 hold, then $n_i = 0$ for $i = 1, \ldots, 2(N_b + N_g)$ for all $\mathbf{n} \in \mathrm{null}(\mathbf{S})$.*

Proof. The claim holds trivially for $\mathbf{n} = \mathbf{0}$. The proof for non-zero $\mathbf{n}$ builds on Assumption 6, and thus the terms involved in these assumptions are computed for (6.8) first.

$$\nabla^2_{\mathbf{xx}}\mathcal{L} := \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \nabla_{\mathbf{x}} h_\ell := \begin{bmatrix} 2\mathbf{L}_\ell \mathbf{v} \\ -\mathbf{a}_\ell \end{bmatrix}, \nabla_{\mathbf{x}} g_m := \begin{bmatrix} 2\mathbf{M}_m \mathbf{v} \\ \mathbf{0} \end{bmatrix}.$$

Given vector $\mathbf{n} \neq \mathbf{0}$ such that $\mathbf{Sn} = \mathbf{0}$, partition $\mathbf{n}$ conformably to $[\mathbf{v}^\top\ \mathbf{x}_g^\top\ \boldsymbol{\lambda}^\top\ \boldsymbol{\mu}^\top]^\top$ as $\mathbf{n} := [\mathbf{n}_v^\top\ \mathbf{n}_{x_g}^\top\ \mathbf{n}_\lambda^\top\ \mathbf{n}_\mu^\top]^\top$. By the definition of $\mathbf{S}$ in (6.14), expanding $\mathbf{Sn} = \mathbf{0}$ yields

$$\mathbf{Z}\mathbf{n}_v + \mathbf{L}_\lambda \mathbf{n}_\lambda + \mathbf{M}_\mu \mathbf{n}_\mu = \mathbf{0} \tag{6.16a}$$

$$\mathbf{A}^\top \mathbf{n}_\lambda = \mathbf{0} \tag{6.16b}$$

$$2\mathbf{L}_\lambda^\top \mathbf{n}_v - \mathbf{A}\mathbf{n}_{x_g} = \mathbf{0} \tag{6.16c}$$

$$2\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_\mu^\top \mathbf{n}_v + \mathcal{D}(\mathbf{g})\mathbf{n}_\mu = \mathbf{0}. \tag{6.16d}$$

Assumption 4 dictates that the first and second term in (6.16d) have complementary sparsity, so that $\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_\mu^\top \mathbf{n}_v = \mathbf{0}$ and $\mathcal{D}(\mathbf{g})\mathbf{n}_\mu = \mathbf{0}$. Recalling the definition $\mathbf{M}_\mu := \sum_{m=1}^M \mathbf{M}_m \mathbf{v} \mathbf{e}_m^\top$, equation $\mathcal{D}(\boldsymbol{\mu})\mathbf{M}_\mu^\top \mathbf{n}_v = \mathbf{0}$ indeed implies $[\mathbf{n}_v^\top\ \mathbf{n}_{x_g}^\top]\nabla_{\mathbf{x}} g_m = 0$ for all $m \in \mathcal{A}(\mathbf{x})$, and together with (6.16c) ensures

$$\begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_{x_g} \end{bmatrix} \in \mathcal{Z}. \tag{6.17}$$

Pre-multiplying (6.16a)–(6.16b) by $2\mathbf{n}_v^\top$ and $\mathbf{n}_{x_g}^\top$ and summing the two resulting equations yields

$$
\begin{aligned}
&2\mathbf{n}_v^\top \mathbf{Z}\mathbf{n}_v + 2\mathbf{n}_v^\top \mathbf{L}_\lambda \mathbf{n}_\lambda + \mathbf{n}_{x_g}^\top \mathbf{A}^\top \mathbf{n}_\lambda \\
&+[\mathbf{n}_v^\top \ \mathbf{n}_x^\top] \begin{bmatrix} 2\mathbf{M}_\mu \\ \mathbf{0} \end{bmatrix} \mathbf{n}_\mu = 0
\end{aligned} \tag{6.18}
$$

where the second and third term on the left-hand side (LHS) sum up to zero per (6.16c). Since $\mathcal{D}(\mathbf{g})\mathbf{n}_\mu = \mathbf{0}$, if $g_m \neq 0$, then the m-th entry $n_{\mu,m}$ of $\mathbf{n}_\mu$ should be zero. In other words, we get that $n_{\mu,m} = 0$ for all $m \notin \mathcal{A}(\mathbf{x})$, and thus, $2[\mathbf{M}_\mu^\top \ \mathbf{0}]^\top \mathbf{n}_\mu = \sum_{m\in\mathcal{A}(\mathbf{x})} \nabla_{\mathbf{x}} g_m n_{\mu,m}$. Substituting the latter into (6.18) gives

$$
2\mathbf{n}_v^\top \mathbf{Z}\mathbf{n}_v + \sum_{m\in\mathcal{A}(\mathbf{x})} [\mathbf{n}_v^\top \ \mathbf{n}_{x_g}^\top]^\top \nabla_{\mathbf{x}} g_m n_{\mu,m} = 0.
$$

The second term on the LHS equals zero due to (6.17). Therefore $\mathbf{n}_v^\top \mathbf{Z}\mathbf{n}_v = 0$, thus implying

$$
\begin{bmatrix} \mathbf{n}_v^\top & \mathbf{n}_{x_g}^\top \end{bmatrix} \nabla^2_{xx}\mathcal{L} \begin{bmatrix} \mathbf{n}_v \\ \mathbf{n}_{x_g} \end{bmatrix} = 0
$$

which contradicts Assumption 6, unless $\mathbf{n}_v = \mathbf{0}$ and $\mathbf{n}_{x_g} = \mathbf{0}$. □

Thanks to Theorem 6.2, we can proceed with computing $\nabla_{\boldsymbol{\theta}}\mathbf{x}$ by solving (6.14) even if $\mathbf{S}$ is singular. In other words, Theorem 6.2 allows us to compute $\nabla_{\boldsymbol{\theta}}\mathbf{x}$ even if the LICQ (Assumption 5) fails. If $\mathbf{S}^\dagger$ is the pseudo-inverse of $\mathbf{S}$, the Jacobian matrix $\mathbf{J}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}}\mathbf{x}$ can be computed as the top $2(N_b + N_g)$ rows of $-\mathbf{S}^\dagger\mathbf{U}$.

The previous analysis has tacitly presumed the system $\mathbf{S}\boldsymbol{\gamma} = \mathbf{u}$ has at least one solution for all $\mathbf{u} \in \text{range}(\mathbf{U})$. The numerical tests of Section 6.8 demonstrate that for the AC-OPF in (6.8), the system $\mathbf{S}\boldsymbol{\gamma} = \mathbf{u}$ features a solution indeed.

As discussed earlier, we focus on training an SI-DNN for predicting generator voltage magnitudes and active power setpoints. Having solved (6.14) and found the sensitivity of $\mathbf{x}$ with respect to $\boldsymbol{\theta}$, the sensitivity of active power generation can be obtained readily using the corresponding entries of $\mathbf{x}_g$. The sensitivity of voltage magnitudes can be derived from the sensitivities of the real and imaginary components of voltages with respect to $\boldsymbol{\theta}$. Precisely, the voltage magnitude at bus n is given by $v_n = \sqrt{(v_n^r)^2 + (v_n^i)^2}$ and its sensitivity with respect to θ_ℓ can be found through the chain rule

$$\frac{\partial v_n}{\partial \theta_\ell} = \frac{1}{v_n}\left(v_n^r \frac{\partial v_n^r}{\partial \theta_\ell} + v_n^i \frac{\partial v_n^i}{\partial \theta_\ell}\right).$$

Evaluating the above completes the requirements of sensitivities for augmenting the SI-DNN training set.

From (6.14), the required sensitivities obviously depend on values of optimal primal/dual variables of (6.8). Problem (6.8) is a non-convex quadratic program and existing solvers may converge to a local rather than a global solution. Albeit the previous sensitivity analysis is valid even for local solutions, the performance of the trained DNN will be apparently suboptimal. To train an SI-DNN to predict global OPF solutions, we next extend the analysis to the SDP relaxation of (6.8).

6.7 SDP Relaxation of the AC-OPF

In pursuit of globally optimal AC-OPF schedules, the non-convex QCQP of (6.8) can be relaxed to the SDP [8]

$$\min_{\mathbf{x}_g, \mathbf{V}\succeq 0} \quad \mathbf{a}_0^\top \mathbf{x}_g \tag{6.19a}$$

$$\text{s.to} \quad \mathrm{Tr}(\mathbf{L}_\ell \mathbf{V}) = \mathbf{a}_\ell^\top \mathbf{x}_g + \mathbf{b}_\ell^\top \boldsymbol{\theta}, \qquad \forall \ell: \ \lambda_\ell \tag{6.19b}$$

$$\mathrm{Tr}(\mathbf{M}_m \mathbf{V}) \leq \mathbf{d}_m^\top \boldsymbol{\theta} + f_m, \qquad \forall m: \quad \mu_m. \tag{6.19c}$$

Problem (6.19) is equivalent to (6.8) if matrix $\mathbf{V}$ is rank-1 at optimality, in which case the SDP relaxation is deemed as *exact*. The relaxation turns out to be exact for several power networks and practical loading conditions; see [79] for a review of related analyses. When the relaxation is exact, the $\mathbf{V}$ minimizer of (6.19) can be expressed as $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$.

We briefly review how the solution $(\mathbf{x}_g, \mathbf{v}; \boldsymbol{\lambda}, \boldsymbol{\mu})$ obtained from the SDP formulation of (6.19) satisfies the first-order optimality conditions for the non-convex QCQP in (6.8) as well. To this end, it is not hard to derive the dual program of (6.19):

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \ -\sum_{\ell=1}^{L} \lambda_\ell \mathbf{b}_\ell^\top \boldsymbol{\theta} - \sum_{m=1}^{M} \mu_m (\mathbf{d}_m^\top \boldsymbol{\theta} + f_m) \tag{6.20a}$$

$$\text{s.to} \quad \mathbf{a}_0 = \sum_{\ell=1}^{L} \lambda_\ell \mathbf{a}_\ell \tag{6.20b}$$

$$\mathbf{Z} := \sum_{\ell=1}^{L} \lambda_\ell \mathbf{L}_\ell + \sum_{m=1}^{M} \mu_m \mathbf{M}_m \succeq 0 \tag{6.20c}$$

$$\boldsymbol{\mu} \geq \mathbf{0}. \tag{6.20d}$$

The optimality conditions for the SDP primal-dual pair (6.19)–(6.20) then include:

i) Primal feasibility (6.19b)–(6.19c) implies (6.8b)–(6.8c).

ii) Dual feasibility (6.20d) applies to (6.8) as well.

iii) Complementary slackness for (6.19c) applies to (6.8c).

iv) Complementary slackness for (6.20c) gives $\mathrm{Tr}(\mathbf{VZ}) = 0$ or $\mathbf{Zv} = \mathbf{0}$, which along with (6.20b) yield the Lagrangian optimality conditions for the QCQP shown in (6.9).

Therefore $(\mathbf{x}_g, \mathbf{v}; \boldsymbol{\lambda}, \boldsymbol{\mu})$ is a stationary point for the QCQP in (6.8). Because it further attains the optimal cost for the relaxed problem in (6.19), it is in fact the globally optimal for (6.8).

To recapitulate, we have used the non-convex QCQP formulation of the AC-OPF to derive the sensitivity formulae of (6.14). This is advantageous as the QCQP features differentiable objective and constraint functions. The obtained sensitivity formulae can be evaluated at the AC-OPF solution provided by any nonlinear programming solver, although such solution may be only locally optimal. To compute a globally optimum AC-OPF solution, we propose using (6.19) instead. If the SDP relaxation is exact, the obtained solution is globally optimal, while the sensitivity formulae derived from QCQP can still be used. Our suggested workflow avoids computing the sensitivities of the SDP formulation for the AC-OPF: Even though differentiating through convex cone constraints is possible [2], it can be perplexing.

6.8 Numerical Tests

The novel SI-DNN approach was evaluated using the IEEE 39-bus, the IEEE 118-bus, and the Illinois 200-bus system. Datasets were generated using either the nonlinear OPF MATPOWER or the globally optimal SDP-based solver.

6.8.1 DNN Architecture and Training

To ease the implementation and without loss of generality, we assumed that buses hosting generators do not host loads, i.e., $p_n^d = q_n^d = 0$ for all $n \in \mathcal{N}_g$. As discussed at the end of Section 6.5, the DNN input $\boldsymbol{\theta}$ consists of the $2(N_b - N_g)$ (re)active power demands at load buses. The DNN output is the setpoints for active power and voltage magnitude (p_n^g, v_n) at all generators $n \in \mathcal{N}_g$ excluding p_1^g for the slack bus. We collect these output quantities in $\check{\mathbf{x}}_{\boldsymbol{\theta}}$, a subvector of $\mathbf{x}_{\boldsymbol{\theta}}$.

Both for P-DNN and SI-DNN, we choose a feed-forward fully-connected architecture. For the number of hidden layers being K, denote the number of neurons in layer k by u_k, with input dimension $u_0 = 2(N_b - N_g)$ and output dimension $u_{K+1} = 2N_g - 1$. To explicitly constrain DNN outputs as per (6.6e) and (6.6g), the output layer uses tanh as its activation function, while all other layers use ReLU. For DNN training and evaluation, labels $\check{\mathbf{x}}_{\boldsymbol{\theta}}$ were suitably scaled within $[-1, 1]$.

We built all DNNs using the `TensorFlow 2.0` python platform alongside `Keras` libraries. Training an SI-DNN deviates from the default routine as gradient updates are implemented separately; see Appendix 6.9 for key differences. For DNN training, at every weight-update step, the gradients computed via the procedure in the appendix are passed to the Adam optimizer. For all tests, we set factor $\rho = 20$ in (6.2), and the initial learning rate for Adam to 5×10^{-4} with an exponential decay reducing the rate to 85% every 250 epochs. DNNs were compiled using Jupyter notebook on a 2.7 GHz Intel Core i5 computer with 8 GB RAM.

6.8.2 Learning Locally Optimal OPF Solutions

We first trained DNNs towards predicting MATPOWER AC-OPF minimizers. We contrasted SI-DNN with P-DNN in terms of the MSE and the related training times.

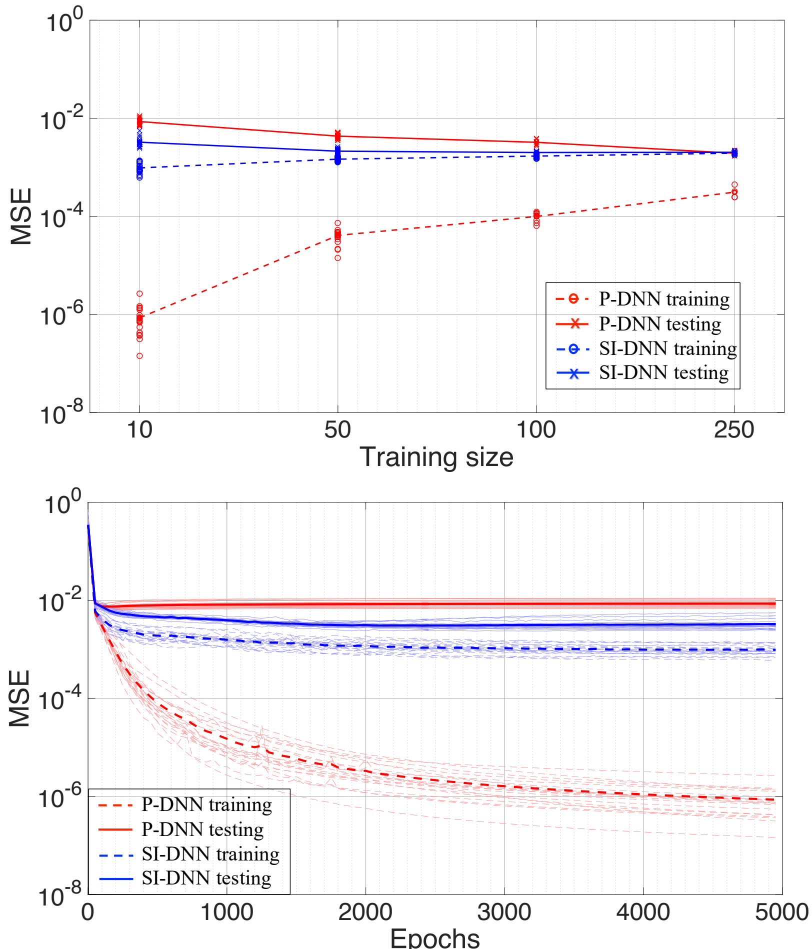

Figure 6.2: Average training and testing errors for different training sizes (*top*); and errors across epochs for different runs with training size 10 (*bottom*).

Tests on IEEE 39-bus system

The network parameters and nominal loads for the IEEE 39-bus system were fetched from the MATPOWER `casefile` [153]. The 39-bus system hosts $N_g = 10$ generators. The benchmark system has loads on two of the generator buses. Removing these, there are 29 load buses. To build a dataset for DNN testing and training, a set of 1000 random $\boldsymbol{\theta} \in \mathbb{R}^{2\cdot 29}$ was sampled, and the corresponding $\check{\mathbf{x}}_\theta$'s were obtained via MATPOWER. To represent various demand levels, we sampled the 1000 random $\boldsymbol{\theta}'s$ by scaling the benchmark demands entry-wise by a scalar drawn independently and uniformly from $[0.8, 1.2]$. A uniform active

Table 6.1: Average Test MSE [$\times 10^{-3}$] and training Time [in sec] for predicting MATPOWER solution on IEEE 39-bus system

Training Size	**P-DNN**		**SI-DNN**	
	MSE	Time	MSE	Time
10	8.6	738	3.3	746
50	4.3	739	2.1	756
100	3.2	747	2.0	776
250	1.9	302	2.0	332

power cost was used for all generators. The default OPF formulation of MATPOWER deviates from the QCQP in (P1). These differences introduce some nuances in building the linear system of (6.14) for computing sensitivities; see Appendix 6.9 for details. Having built the aforementioned dataset $\{(\boldsymbol{\theta}_s, \mathbf{J}_{\boldsymbol{\theta}_s}, \check{\mathbf{x}}_{\boldsymbol{\theta}_s})\}_{s=1}^{1000}$, the architectures for SI-DNN and P-DNN were determined next. Based on preliminary tests, identical architectures were chosen for P-DNN and SI-DNN with $K = 4$ hidden layers with $u_k = 256$ neurons for $k = 1, \ldots, 4$. Preliminary tests showed negligible effect on P-DNN performance if the number of layers is reduced to three. Nevertheless, the architecture for the two DNNs was kept identical to ensure equal expressibility.

The evaluation for the two DNNs for different training sizes was performed as follows. First, for a *training size* of 10, we created 20 different training sets by sampling 10 OPF instances from the dataset without replacement. For each of these 20 times or *runs*, the OPF instances not sampled for training consisted the training sets. We then separately trained P-DNN and SI-DNN on these 20 sets. For the training sizes of $(50, 100, 250)$, we had $(20, 10, 4)$ runs, respectively. For training sizes $(10, 50, 100)$, the entire training set was used for gradient computation at each step, with the total epochs being 5000. When the training size was 250, the batch-size was fixed to 100, and total epochs to 2000. The training and testing MSE loss for all training sizes, and runs are shown in Fig. 6.2 *(top)*. For the tests with training size 10, the evolution of DNN errors are shown in Fig. 6.2 *(bottom)*. The average test MSE

Table 6.2: Average Test MSE [$\times 10^{-3}$] and training Time [in sec] for predicting MATPOWER solution

Training Size	IEEE 118-bus				Illinois 200-bus			
	P-DNN		**SI-DNN**		**P-DNN**		**SI-DNN**	
	MSE	Time	MSE	Time	MSE	Time	MSE	Time
25	1.8	447	1.1	483	0.1894	452	0.0424	491
50	1.7	458	1.1	527	0.1469	456	0.0376	524
100	1.6	463	0.9	610	0.0913	471	0.0620	608

and training times for the two DNNs are shown in Table 6.1. From Fig. 6.2 *(top)*, we observe as anticipated, that for both DNNs, the gap between training and testing loss decreases for larger training size. Further, the errors for different *runs* are well clustered, indicating a numerically stable DNN implementation. From Table 6.1, it is fascinating to note that the test loss attained by SI-DNN is much lower than P-DNN, especially at smaller training sets. For instance, the P-DNN requires 100 samples to roughly attain the average test MSE which the SI-DNN attains with 10 samples. It is worth stressing that the improvement in sample efficiency comes at modest increase in training time.

Tests on other benchmarks

The DNN architecture chosen for the other two power systems was similar to the IEEE 39-bus case with the differences being in the number of neurons per layer. Specifically, the DNNs used for the 118$-$, and 200-bus systems had 512 neurons in hidden layers, with the input (output) layers having 302 (97), and 128 (107) neurons, respectively. For each of these systems, we created a dataset with 500 random demands. The linear cost coefficients from the respective benchmark systems were retained as c_p's, while the reactive power cost coefficients were set to zero. All DNNs were evaluated for five runs, with training sizes of 25, 50, and 100. Table 6.2 summarizes the obtained results.

Table 6.3: Average Test MSE [$\times 10^{-3}$] for predicting SDP solutions, and constraint violation statistics on the IEEE 39-bus system

Train. Size	P-DNN MSE	P-DNN (a)	P-DNN (b)	P-DNN (c)	SI-DNN MSE	SI-DNN (a)	SI-DNN (b)	SI-DNN (c)
10	6.3	2.61	0.50	9.78	0.91	2.52	0.37	3.35
50	3.6	2.45	0.55	7.38	0.62	2.58	0.27	2.06
100	2.5	2.59	0.53	6.87	0.67	2.52	0.27	1.96

(a) #violations /instance; **(b)** max. violation; **(c)** mean violation [$\times 10^{-2}$]

6.8.3 Learning Globally Optimal OPF Solutions

The SI-DNN was evaluated towards predicting the minimizer of an SDP relaxation-based OPF solver for the IEEE 39-bus system. A uniform active power cost was used for all generators while the reactive power cost coefficients were set as $c_n^q = 0.1c_n^p$. To build a dataset, a set of 1000 random $\boldsymbol{\theta}'s$ was sampled as explained earlier. The corresponding $\check{\mathbf{x}}_\theta$'s were obtained by solving (6.19) using the MATLAB-based optimization toolbox YALMIP with SDP solver Mosek [78]. For all SDP instances, the ratio of the second largest eigenvalue of matrix $\mathbf{V}$ to the largest eigenvalue was found to lie in $[3 \cdot 10^{-7}, 1 \cdot 10^{-4}]$; numerically indicating an exact relaxation. Thus, the eigenvector corresponding to the largest eigenvalue was deemed as the optimal voltage $\mathbf{v}$. As with learning MATPOWER solutions, the sample efficiency of SI-DNN was found significantly superior to P-DNN in learning globally optimal OPF solutions; see Table 6.3 for the average MSE attained during testing. The presented results with local and global OPF solvers demonstrate that SI-DNN yields a dramatic improvement in generalizability.

While emphasis has been on MSE, the importance of satisfying constraints cannot be undermined. To this end, we tested the feasibility of SI-DNN OPF predictions using the following metrics. For each of the DNNs, given a test input and DNN prediction, an AC power flow solution was obtained using MATPOWER. For each instance, the inequalities in (6.8c) not directly enforced by the tanh activation were evaluated. These included voltage limits on load buses, line flow limits, generator reactive power limits, and the slack bus active power limits, totalling to 126 constraints for the 39-bus system. For suitable scaling, the violations in flows and generation were normalized by the maximum limit, while voltage violations were maintained in pu. We evaluated the constraint violations caused by implementing the predictions of SI-DNN and P-DNN on the test instances that remained after sampling training sets of different sizes from the 1000 random instances. For different training sizes, Table 6.3 lists: *a)* the average number of violations, exceeding a normalized magnitude of 10^{-4}, per test instance; *b)* the maximum constraint violation observed; and *c)* the violations averaged over all constraints and test instances. Interestingly, while both DNNs incur similar count of violations, SI-DNN reduces the maximum violation by half and mean violation to less than a third.

6.9 Conclusions

This work has built on the fresh idea of sensitivity-informed training for learning the solutions of arbitrary AC-OPF formulations. It comprehensively delineated the steps for computing the involved sensitivities using the optimal primal/dual solutions, which are readily available by AC-OPF solvers. Such sensitivities of the primal AC-OPF solutions have been shown to exist under mild assumptions, while their computation is as simple as solving a system of linear equations with multiple right-hand sides. The approach is quite general since the OPF solutions comprising the training dataset can be obtained by off-the-shelf nonlinear OPF solvers or modern conic relaxation-based schemes. It is also worth stressing that sensitivity-informed training can readily complement other existing learn-to-OPF methodologies. Extensive numerical tests on three benchmark power systems have demonstrated that with a modest increase in training time, SI-DNNs attain the same prediction performance as conventionally trained DNNs by using roughly only 1/10 to 1/4 of the training data. Such improvement on sample efficiency reduces the time needed for generating training datasets, and is thus, relevant to delay-critical power systems applications. Furthermore, SI-DNN predictions turn out to feature better constraint satisfaction capabilities too. Sensitivity-informed learning forms the solid foundations for several exciting and practically relevant research directions, such as warm-starting key optimal primal/dual variables to accelerate decentralized OPF solvers and predicting active constraints.

Appendix D

D.1 Python Implementation for SI-DNN

A typical implementation example for computing the gradient of the MSE loss in P-DNN with respect to the DNN weights (which are the *trainable variables*) involves

```
with tensorflow.GradientTape() as tape:
    pred_x = model(theta)
    loss = keras.losses.MSE(xlabel,pred_x)
model_gradients=tape.gradient(loss, model.trainable_variables)
```

where `model` represents the DNN and `GradientTape` computes the desired gradient. In transitioning to SI-DNN, we first need to compute the gradient of the DNN output `pred_x` with respect to its input `theta` to define the loss. We then compute the gradients of the two loss terms with respect to the DNN weights. This can be implemented using nested `GradientTape` as

```
with tensorflow.GradientTape() as tape:
```

```
  with tensorflow.GradientTape() as tape2:
    tape2.watch(theta)
    pred_x=model(theta)
    Ploss=keras.losses.MSE(xlabel,pred_x)
  J_model=tape2.batch_jacobian(pred_x, theta)
  J_flat=tf.keras.backend.reshape(fgrad, shape=(1,))
  SI_loss=keras.losses.MSE(J_flat,J_label)
  total_loss=P_loss+rho*SI_loss
model_gradients=tape.gradient(loss,model.trainable_variables)
```

where the inner `tape` computes the sensitivity of DNN to compute the overall SI-DNN loss, while the outer `tape` computes the gradients for weight updates.

D.2 Sensitivity computation with MATPOWER

While solving the AC-OPF instances with MATPOWER, we used the Cartesian coordinate system, and flow limits were imposed on squared currents. For computing the desired sensitivities, we first need to build the linear system (6.14), which requires the optimal dual variables, constraint function values, and the derivatives of the constraint functions with respect to the optimization variables. Although MATPOWER can deal with the AC-OPF posed with voltages in Cartesian coordinates, it slightly differs from the QCQP in (6.6) as follows:

a1) MATPOWER enforces power flow equations as in (6.6a)–(6.6d). Different from (6.6e)–(6.6f) however, MATPOWER poses generator (re)active power limits as $\underline{p}_n^g \leq p_n^g \leq \bar{p}_n^g$. Thus, the related $\nabla_{\mathbf{v}} g_m$ becomes zero and $\nabla_{\mathbf{x}_g} g_m$ becomes a signed canonical vector corresponding to bus n.

a2) MATPOWER constraints voltages, rather than squared voltages as in (6.6g). While the two versions are equivalent, the related derivatives $\nabla_{\mathbf{v}} g$ apparently differ. The derivatives of non-squared magnitudes can be found using the chain rule as $\nabla_{\mathbf{v}} v_n = \frac{\partial v_n}{\partial v_n^2} \nabla_{\mathbf{v}} v_n^2 = \frac{1}{v_n} \nabla_{\mathbf{v}} v_n^2$ and $\nabla_{\mathbf{v}} v_n^2 = 2\mathbf{M}_{v_n}\mathbf{v}$ from (6.4).

a3) Different from (6.6h), MATPOWER sets the voltage angle reference to zero by enforcing $\arctan(v_{N+1}^i / v_{N+1}^r) = 0$. Fortunately, simply setting the imaginary part v_{N+1}^i to zero is equivalent, and the gradients of these two formulations agree. Thus, despite the difference in formulation, we use (6.6h) wherever needed in building (6.14).

a4) Finally, MATPOWER poses flow limits on both the sending and receiving ends of each line, thus doubling the number of constraints in (6.6i). The matrices $\mathbf{M}_{\mathbf{i}_{mn}}$ can be built using the *from bus* and *to bus* admittances obtained via the MATPOWER command `makeYbus()`.

Chapter 7

Ripple-Type Control for Enhancing Resilience of Networked Physical Systems

7.1 Publication Details

M. K. Singh, G. Cavraro, A. Bernstein, and V. Kekatos, "Ripple-Type Control for Enhancing Resilience of Networked Physical Systems," in *Proc. IEEE American Control Conf.*, New Orleans, LA, May 2021.

7.2 Abstract

Distributed control agents have been advocated as an effective means for improving the resiliency of our physical infrastructures under unexpected events. Purely local control has been shown to be insufficient, centralized optimal resource allocation approaches can be slow. In this context, we put forth a hybrid low-communication saturation-driven protocol for the coordination of control agents that are distributed over a physical system and are

allowed to communicate with peers over a "hotline" communication network. According to this protocol, agents act on local readings unless their control resources have been depleted, in which case they send a beacon for assistance to peer agents. Our ripple-type scheme triggers communication locally only for the agents with saturated resources and it is proved to converge. Moreover, under a monotonicity assumption on the underlying physical law coupling control outputs to inputs, the devised control is proved to converge to a configuration satisfying safe operational constraints. The assumption is shown to hold for voltage control in electric power systems and pressure control in water distribution networks. Numerical tests corroborate the efficacy of the novel scheme.

7.3 Introduction

Utility systems, such as power, water, and gas networks, are significant examples of networked systems and are undergoing rapid changes. Power systems experience significant penetration of distributed energy resources and flexible load thus increasing the system volatility. Natural gas-fired generators serve as a fast-acting balancing mechanism for power systems, inadvertently increasing the volatility in gas networks. Rising threats of clean water scarcity have motivated tremendous efforts toward judicious planning for bulk water systems and enhanced monitoring and control for water distribution networks. The increasing occurrence of natural disasters and other (cyber and physical) disruptions undermines the operation of the aforesaid systems.

Classic operation of utility systems is performed via management systems that aim at satisfying consumer demands in a cost-effective manner while meeting the related operational and physical constraints. Such tasks constitute the family of optimal dispatch problems (ODP). ODPs are typically solved at regular intervals based on anticipated demands and network conditions. Stochastic optimization formulations are often leveraged to account for the uncertainty and to ensure reliable operation within the ODP interval.

Although ODP solutions can ensure reliable system operation during normal conditions, they cannot account for the occurrence of low-probability, high-impact disruptions that might undermine system operation during the time interval between two ODP actions. To improve system resilience to these events, this paper devises a control mechanisms to ensure the satisfaction of system operational requirements. It draws features from local, distributed, and event-triggered control and is referred to as *ripple-type control*. In local control rules, agents make decisions based on locally available readings. For example, in [152], [151], power generators control their reactive power output given their local power injection and voltage; however, local schemes have limited efficacy [21]. Distributed control strategies, in which agents compute their control action after sharing information with neighbors in a communication network, have a wide spectrum of applications, e.g., energy systems [27], [16], [29] or camera networks [20]. To avoid wasting resources and to communicate only when it is really needed, event-triggered control techniques were advocated in [63], [81]. Essentially, every agent evaluates locally a *triggering function*, e.g., in a consensus setup, the mismatch between the current state and the state that was last sent to neighbors [91]. When the triggering function takes some specified values, agents communicate and update their control rule.

In the proposed ripple-type control, agents first try to satisfy their local constraints via purely local control. Only when such control efforts reach their maximum limit, an assistance is sought from neighboring agents on a communication graph. The process is continued until the control objectives of every agent are met. The proposed algorithm is *model free* in the sense that it does not require knowledge of the system model parameters. This is an essential property of real-time emergency control due to lack of accurate model information during contingency scenarios [32]. The proposed control scheme is tested on electric and water networks.

Notation: Lower- (upper-) case boldface letters denote column vectors (matrices). Sets are represented by calligraphic symbols. Symbol $^\top$ stands for transposition. Inequalities, max, and min operators are understood element-wise. All-zero and all-one vectors and matrices are represented by $\mathbf{0}$ and $\mathbf{1}$; the respective dimensions are deducible from context. Symbol $\succ$ indicates all positive real eigenvalues. The $\mathrm{dg}(\cdot)$ operator on vectors places the vector on the principal diagonal of a matrix. Symbol $\|\cdot\|$ represents the L_2 norm.

7.4 System Modeling

Consider a networked system modeled by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The set $\mathcal{N}$ is a collection of N nodes hosting controllable agents, and vector $\mathbf{u} \in \mathbb{R}^N$ represents their *control inputs*A subset of agents comprising set $\mathcal{Y} \subset \mathcal{N}$ of cardinality M is assumed to be collecting noiseless scalar observations of the local *states or outputs* stacked in $\mathbf{y} \in \mathbb{R}^M$. The entries of $\mathbf{y}$ are to be regulated within a desired range. Given control $\mathbf{u}$, the system has a locally unique output $\mathbf{y}$ determined by a mapping $\mathrm{F} : \mathbb{R}^N \to \mathbb{R}^M$. Heed that the mapping F depends implicitly on other uncontrollable system inputs that are not part of $\mathbf{u}$. Moreover, this mapping might not have an explicit form. We consider cyber-physical systems in which the related mapping F adheres to the following property.

Assumption 7. Mapping $\mathbf{y} = \mathrm{F}(\mathbf{u})$ satisfies $\nabla_{\mathbf{u}}\mathrm{F}(\mathbf{u}) \geq \mathbf{0}\ \forall\ \mathbf{u}$.

Albeit seemingly restrictive, the postulated monotonicity assumption holds for several physical systems abiding by a dissipative flow law, such as natural gas and water networks [131]. We will now establish the validity of Assumption 7 for power systems and water networks.

Then, Section 7.5 formally introduces the problem setup, and Section 7.6 presents the proposed control scheme and the associated claims.

Power Systems. An electric power network can be modeled by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with $\mathcal{E}$ capturing transmission lines. The node or bus set $\mathcal{N}$ can be partitioned into generator (PV) buses $\mathcal{N}_G$ and load (PQ) buses $\mathcal{N}_L$. Generators can control their active power injection and voltage magnitude. For loads, complex power injections can be adjustable or fixed, but they are largely independent of voltages. Let (v_n, q_n) denote the voltage magnitude and reactive power injection at bus $n \in \mathcal{N}$. Vectors $(\mathbf{v}, \mathbf{q})$ collect $\{(v_n, q_n)\}_{n\in\mathcal{N}}$.

Consider the task of maintaining the load voltages above given limits by controlling the reactive power injections at the load buses and the voltages at the generator buses. Loads can partially control their reactive injections due to inverters, capacitor banks, and flexible AC transmission systems (FACTS). To study the task, we build on the widely adopted approximate model [108]:

$$\mathbf{q} = \mathrm{dg}(\mathbf{v})\mathbf{B}\mathbf{v}. \tag{7.1}$$

where $\mathbf{B} \in \mathbb{R}^{N\times N}$ is a weighted Laplacian matrix of $\mathcal{G}$: Its (m, n)-th entry $B_{mn} < 0$ equals the negative susceptance of line $(m, n) \in \mathcal{E}$; $B_{mn} = 0$ if $(m, n) \notin \mathcal{E}$; and $B_{mm} = -\sum_{n\neq m} B_{mn} > 0$ for its diagonal entries. Model (7.1) has been derived from the AC power flow equations after ignoring transmission line resistances and assuming small voltage angle differences across neighboring buses. Partitioning $\mathbf{v}$ and $\mathbf{q}$ into generator and load buses, rewrite (7.1) as:

$$\begin{bmatrix} \mathbf{q}_G \\ \mathbf{q}_L \end{bmatrix} = \begin{bmatrix} \mathrm{dg}(\mathbf{v}_G) & \mathbf{0} \\ \mathbf{0} & \mathrm{dg}(\mathbf{v}_L) \end{bmatrix} \begin{bmatrix} \mathbf{B}_{GG} & \mathbf{B}_{LG}^\top \\ \mathbf{B}_{LG} & \mathbf{B}_{LL} \end{bmatrix} \begin{bmatrix} \mathbf{v}_G \\ \mathbf{v}_L \end{bmatrix} \tag{7.2}$$

where $\mathbf{B}$ has been partitioned accordingly.

For the control task at hand, the control variable per bus n depends on the type of the bus as:

$$u_n := \begin{cases} v_n & , \ n \in \mathcal{N}_G \\ q_n & , \ n \in \mathcal{N}_L \end{cases}.$$

We would like to adjust input $\mathbf{u} = [\mathbf{q}_L^\top \quad \mathbf{v}_G^\top]^\top$ to control output $\mathbf{y} = \mathbf{v}_L$. To validate Assumption 7, the ensuing result studies the mapping $\mathbf{y} = \mathrm{F}(\mathbf{u})$ for the case of voltage control.

Proposition 7.1. *For control input $\mathbf{u} = [\mathbf{q}_L^\top \quad \mathbf{v}_G^\top]^\top$ and output $\mathbf{y} = \mathbf{v}_L$, the mapping $\mathbf{y} = F(\mathbf{u})$ satisfies Assumption 7 if $\mathrm{dg}(\mathbf{g}_L) + \mathbf{B}_{LL} \succ 0$ where $\mathbf{g}_L := [\mathrm{dg}(\mathbf{v}_L)]^{-2}\,\mathbf{q}_L$.*

Proof. Adopting the implicit differentiation approach, apply the differential operator on the second block of (7.2):

$$\partial \mathbf{q}_L = (\mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LL})\,\partial \mathbf{v}_L + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LG}\,\partial \mathbf{v}_G \tag{7.3}$$

where $\mathbf{i}_L := \mathbf{B}_{LG}\mathbf{v}_G + \mathbf{B}_{LL}\mathbf{v}_L$. Because $(\mathbf{q}_L, \mathbf{v}_G)$ are independent, it holds $\nabla_{\mathbf{q}_L}\mathbf{v}_G = \mathbf{0}$ so that (7.3) yields:

$$\nabla_{\mathbf{q}_L}\mathbf{v}_L = (\mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LL})^{-1}. \tag{7.4}$$

And because $\nabla_{\mathbf{v}_G}\mathbf{q}_L = \mathbf{0}$, equation (7.3) also provides:

$$\nabla_{\mathbf{v}_G}\mathbf{v}_L = -\,(\mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LL})^{-1}\,\mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LG}. \tag{7.5}$$

For the Jacobian matrices $\nabla_{\mathbf{q}_L}\mathbf{v}_L$ and $\nabla_{\mathbf{v}_G}\mathbf{v}_L$ to have nonnegative entries, it suffices to show that the inverse of $\mathbf{G} := \mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LL}$ has nonnegative entries. This is because $\mathbf{v}_L > \mathbf{0}$ and $\mathbf{B}_{LG} \leq \mathbf{0}$.

To establish $\mathbf{G}^{-1} \geq \mathbf{0}$, note that the off-diagonal entries of $\mathbf{G}$ are nonpositive; hence, proving

$\mathbf{G} \succ 0$ would make $\mathbf{G}$ an M-matrix so that $\mathbf{G}^{-1} \leq \mathbf{0}$. The assumption $\mathrm{dg}(\mathbf{g}_L) + \mathbf{B}_{LL} \succ 0$ stated in this proposition ensures $\mathbf{G} \succ 0$, as we show next. From (7.3), it follows that $\mathbf{q}_L = \mathrm{dg}(\mathbf{v}_L)\mathbf{i}_L$. Substituting $\mathbf{i}_L = \mathrm{dg}(\mathbf{v}_L)^{-1}\mathbf{q}_L$ in the stated condition yields:

$$\begin{aligned}
&\mathrm{dg}(\mathbf{i}_L)\,\mathrm{dg}(\mathbf{v}_L)^{-1} + \mathbf{B}_{LL} \succ 0 \implies \\
&\mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)^{1/2}\mathbf{B}_{LL}\,\mathrm{dg}(\mathbf{v}_L)^{1/2} \succ 0 \implies \\
&\mathrm{dg}(\mathbf{i}_L) + \mathrm{dg}(\mathbf{v}_L)\mathbf{B}_{LL} = \mathbf{G} \succ 0
\end{aligned}$$

where the first transition follows from Sylvester's law of inertia for congruent matrices, and the second from the similarity transformation involved. $\square$

Proposition 7.1 identifies the conditions that ensure Assumption 7 holds for the grid model in (7.1). Numerically verifying $\mathrm{dg}(\mathbf{g}_L) + \mathbf{B}_{LL} \succ 0$ for benchmark systems revealed an interesting observation, discussed next. For the IEEE 5-39-118-bus systems, we scaled up the nominal $\mathbf{q}_L$ by a scalar until the power flow solver MATPOWER [153] failed to converge. At each step, we also noted the eigenvalues of $\mathrm{dg}(\mathbf{g}_L) + \mathbf{B}_{LL}$. Interestingly, the minimum eigenvalue kept decreasing for increasing $\mathbf{q}_L$, but it remained positive until the last successful power flow instance for all networks. This indicates a relation between $\mathrm{dg}(\mathbf{g}_L)+\mathbf{B}_{LL} \succ 0$ and the solvability of the AC power flow equations, but its analytical investigation goes beyond the scope of this work.

Water Distribution Systems (WDS). A WDS can be modeled by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the nodes in $\mathcal{N}$ correspond to water reservoirs, tanks, junctions and consumers. If d_n denotes the rate of water injection at node n, then $d_n \geq 0$ for reservoirs; $d_n \leq 0$ for water consumers; tanks might be filling or emptying; and $d_n = 0$ for junctions. Nodes are connected by edges in $\mathcal{E}$, which represent pipelines, pumps, and valves. Let σ_{mn} denote the rate of water flowing from node m to n over edge (m, n). It also holds $\sigma_{mn} = -\sigma_{nm}$. Flow conservation at node n dictates:

$$d_n = \sum_{m:(n,m)\in\mathcal{E}} \sigma_{nm}. \tag{7.6}$$

The relation between water flow σ_{mn} across edge (m,n) and the pressures π_m and π_n at nodes m and n takes the form

$$\pi_m - \pi_n = \rho_{mn}(\sigma_{mn}). \tag{7.7}$$

The operation of a WDS involves serving water demands while maintaining nodal pressures within desirable levels. The task of pressure control can include continuous-valued variables, such as reservoir and tank output pressures, as well as water injections at different nodes. Pressure control can also involve binary variables capturing the on/off status of fixed-speed pumps and valves [115]. These continuous and binary control variables are often determined by periodically solving ODPs; see e.g., [115]. Assuming the binary variables to be fixed between two ODP instances, we focus on the continuous-valued variables. Partition $\mathcal{N}$ into the subset $\mathcal{N}_S$ of reservoirs and tanks, and the subset $\mathcal{N}_L$ of loads. For nodes in $\mathcal{N}_S$, pressures are controllable. For nodes in $\mathcal{N}_L$, demands are controllable; inelastic demands can be modeled with lower and upper limits coinciding; thus, the control variables now are:

$$u_n := \begin{cases} \pi_n & , \ n \in \mathcal{N}_S \\ d_n & , \ n \in \mathcal{N}_L \end{cases}.$$

Given $\mathbf{u}$, the water injections for the nodes in $\mathcal{N}_S$ and the pressures at nodes in $\mathcal{N}_L$ are determined by the water flow equations (7.6)–(7.7). The nodal pressures over $\mathcal{N}_L$ need to be maintained at stipulated levels; thus they constitute vector $\mathbf{y}$. For the aforementioned assignments of $\mathbf{u}$ and $\mathbf{y}$, the mapping F is implicitly defined by (7.6)–(7.7). Reference [110] guarantees that F maps a given $\mathbf{u}$ to a unique $\mathbf{y}$.

To verify the validity of Assumption 7, we use a monotonicity result for dissipative flow networks from [131]. To qualify as a dissipative flow network per [131], the functions $\rho_{mn}(\cdot)$ in (7.7) should be nondecreasing and continuous. Both these requirements hold true for edges in a WDS. Specifically, for a pipe $(m, n) \in \mathcal{E}$, the function ρ_{mn} models the pressure drop due to friction described by the Darcy-Weisbach or the Hazen-Williams laws [110]. Both are non-decreasing and continuous. For pumps and valves, pressures π_m and π_n relate to flow σ_{mn} via nondecreasing empirical laws as well [37]; therefore, WDS fall within the purview of dissipative networks, and the ensuing result applies [131].

Lemma 7.2. *[131, Corollary 4] Let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be N-length pressure vectors satisfying the water flow equations (7.6)-(7.7) for demand vectors $\mathbf{d}$ and $\mathbf{d}'$. If $\pi_n \geq \pi_n'$ for all $n \in \mathcal{N}_S$, and $d_n \geq d_n'$ for all $n \in \mathcal{N}_L$, then $\pi_n \geq \pi_n'$ for all $n \in \mathcal{N}_L$.*

In terms of the assigned vectors $\mathbf{u}$ and $\mathbf{y}$, Lemma 7.2 states that controls $\mathbf{u} \geq \mathbf{u}'$ result in $\mathbf{y} \geq \mathbf{y}'$; thus, considering infinitesimal changes at node pairs m and n, Lemma 7.2 translates to $\frac{\partial y_m}{\partial u_n} \geq 0$, hence satisfying Assumption 7.

7.5 Problem Formulation

Operators manage the networked system by computing periodically the control set points for agents to implement. Such set points are typically the solution of an ODP

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}} \quad c(\mathbf{u}, \mathbf{y}) \tag{P1}$$

$$\text{s.to} \quad \mathbf{y} = \mathrm{F}(\mathbf{u}) \tag{7.8a}$$

$$\mathbf{h}(\mathbf{u}, \mathbf{y}) \leq 0, \tag{7.8b}$$

$$\underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}} \tag{7.8c}$$

$$\mathbf{y} \geq \underline{\mathbf{y}} \tag{7.8d}$$

where $c(\mathbf{u}, \mathbf{y})$ is a cost function depending on the system inputs and outputs; the mapping in (7.8a) corresponds to the physical laws governing the networked system; and the inequality constraints are grouped into two categories:

- Constraint (7.8b) captures requirements that are important to efficiently operate the system, but can in principle be safely violated (especially for a short time). These will be referred to as *soft constraints*.

- Constraints (7.8c) and (7.8d) impose limitations on input and output variables. If violated, they can lead to system failure. These constraints will be referred to as *hard constraints*.

Typically, given the set of parameters defining F, a central dispatcher solves (P1) and communicates optimal control set points $\mathbf{u}^*$ to agents. Ideally, this process shall be repeated every time there is a change in the system model that modifies the underlying definition of F, such as an abrupt load change or line tripping for the case of power systems. Constrained by communication and computational resources, however, problem (P1) is solved only at finite time intervals. As a consequence, the set points $\mathbf{u}^*$ can become obsolete or even result in network constraint violations.

Such limitations motivate the design of mechanisms to at least ensure that some important operational requirements are met between two consecutive centralized dispatch actions, i.e., to make the control vector $\mathbf{u}$ belong to the feasible set:

$$\mathcal{F} = \{\mathbf{u} : \ \underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}}, \ \mathbf{y} = \mathrm{F}'(\mathbf{u}), \mathbf{y} \geq \underline{\mathbf{y}}\}$$

in which F' is the input-output mapping defined by the system model after a disruptive event. Set $\mathcal{F}$ considers only the hard constraints of (P1), namely, (7.8c) and (7.8d). We next put forth an algorithm for steering $\mathbf{u}$ to $\mathcal{F}$. In the following, we will assume that the set $\mathcal{F}$ is non-empty, or $\mathcal{F} \neq \emptyset$.

Remark 7.3. For power systems, constraint (7.8b) models line flow limits, which can be considered soft constraints because their violation could be briefly tolerated. Limits on the power or voltage output of generators are captured by (7.8c). Inequality (7.8d) models load undervoltage constraints. Avoiding dangerously low voltages is important to prevent a voltage collapse. For WDS, the inequalities (7.8b) can represent the water flow limits on pipes and pumps. Although flow limits shall normally be explicitly enforced, in practice low-pressure limits are triggered before the flow limits [40]; thus, edge flow can be characterized as soft over short time intervals. Constraints (7.8c) represent the limits on pressure at water sources and demands by consumers. These limits cannot be violated physically because they are posed by the actual capacity of the WDS components and available demands. The inequalities in (7.8d) model the minimum pressure requirements at consumer nodes. The latter requirements must be adhered to at all times as low pressures can cause service failures and equipment malfunction at the consumer end.

7.6 Ripple-Type Network Control

Our algorithm has the ensuing key attributes:

A1) Agent $n \in \mathcal{Y}$ measures y_n and controls u_n locally. Upon a violation of (7.8d), local control resources are used first.

A2) Agent n transmits communication signals to a few peer agents over a communication network only if $u_n = \bar{u}_n$, i.e., only when local control resources reach their maximum limit and assistance is sought from neighboring nodes on a communication graph.

A3) The control scheme is agnostic to the physical system parameters, i.e., it is a model-free approach that does not require explicit knowledge of operator F'.

The communication network is modeled as an undirected graph $\mathcal{G}_c = (\mathcal{N}, \mathcal{E}_c)$, in which the communication links $\mathcal{E}_c$ do not necessarily coincide with the physical connections among agents. Graph $\mathcal{G}_c$ is henceforth assumed to be connected, and we denote its adjacency matrix as $\mathbf{A}$, where $A_{mn} = 1$ if there is a direct communication link between nodes m and n, i.e., $(m, n) \in \mathcal{E}_c$; and $A_{mn} = 0$ otherwise. Let us also introduce function $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N_+$ defined entry-wise as:

$$f_n(\mathbf{u}) := \begin{cases} \underline{y}_n - y_n(\mathbf{u}), & \underline{y}_n \geq y_n(\mathbf{u}), n \in \mathcal{Y} \\ 0, & \text{otherwise.} \end{cases} \tag{7.9}$$

Assume that, at time $t = 0$, a violation of (7.8d) occurs as a consequence of a model change. Let $\mathbf{u}(0)$ be the initial control variables, and introduce an auxiliary vector $\boldsymbol{\lambda} \in \mathbb{R}^N$, which is initialized as $\boldsymbol{\lambda}(0) = \mathbf{0}$. At subsequent times $t \geq 1$, the control scheme proceeds in four steps, as delineated next.

Step 1: Agents compute $\mathbf{f}(\mathbf{u}(t))$ according to (7.9). The entries of $\mathbf{f}(\mathbf{u}(t))$ are strictly positive if the associated nodes in $\mathcal{Y}$ experience a violation of (7.8d); and zero, otherwise.

Step 2: A target set point is computed as:

$$\hat{\mathbf{u}}(t+1) = \mathbf{u}(t) + \mathrm{dg}(\boldsymbol{\eta}_1)\mathbf{f}(\mathbf{u}(t)) + \mathrm{dg}(\boldsymbol{\eta}_2)\mathbf{A}\boldsymbol{\lambda}(t) \tag{7.10}$$

for positive $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$. Note that for node n, the target $\hat{u}_n(t)$ is computed using the local reading $y_n(t)$ and the entries of $\boldsymbol{\lambda}$ sent from its peers (neighbor nodes of node n on $\mathcal{G}_c$).

Step 3: Agents compute the auxiliary vector $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda}(t+1) = \max\{\mathbf{0}, \mathrm{dg}(\boldsymbol{\eta}_3)(\hat{\mathbf{u}}(t+1) - \bar{\mathbf{u}})\}. \tag{7.11}$$

for a positive $\boldsymbol{\eta}_3$. Vector $\boldsymbol{\lambda}$ serves as a beacon for assistance that is communicated across peer nodes.

Step 4: The target set point is projected to the feasible range:

$$\mathbf{u}(t+1) = \min\{\hat{\mathbf{u}}(t+1), \bar{\mathbf{u}}\}, \tag{7.12}$$

and is physically implemented.

To establish the effectiveness, Proposition 7.4 proves that the proposed scheme reaches an equilibrium point, and Proposition 7.5 proves that this equilibrium belongs to $\mathcal{F}$. The proofs for these results are provided in the Appendix.

Proposition 7.4. *Given any $\mathbf{u}(0)$, the sequence $\{\mathbf{u}(t)\}$ converges asymptotically.*

Proposition 7.5. *Let Assumptions 7 hold, let $\mathcal{F} \neq \emptyset$, and:*

$$\| \operatorname{dg}(\boldsymbol{\eta}_2) \operatorname{dg}(\boldsymbol{\eta}_3)\mathbf{A}\| < 1. \tag{7.13}$$

A pair $(\mathbf{u},\ \boldsymbol{\lambda})$ is an equilibrium for the proposed scheme if and only if $\mathbf{u}$ belongs to $\mathcal{F}$ and $\boldsymbol{\lambda} = \mathbf{0}$.

The novel control scheme satisfies attribute *A1)* by design. Moreover, for a node n with target set points $\hat{u}_n(t+1)$ within the local control limit $\bar{u}_n$, the corresponding entry of $\boldsymbol{\lambda}(t+1)$ is zero; thus, the computation of $\hat{u}_m$ from (7.10) for nodes m that are neighbors of n requires no communication from node n, hence fulfilling *A2)*. The scheme also meets *A3)* because it is agnostic to the physical network topology and/or demands.

Remark 7.6. Apparently, parameter $\boldsymbol{\eta}_1$ does not influence the control scheme convergence. Indeed, (7.13) provides a condition only on $\boldsymbol{\eta}_2$ and $\boldsymbol{\eta}_3$; however, $\boldsymbol{\eta}_1$ affects the equilibrium point and the convergence rate, whose analytical quantification is beyond the scope of this work. Optimizing upon the parameters $\boldsymbol{\eta}_1 - \boldsymbol{\eta}_3$, and communication graph $\mathcal{G}_c$ constitutes pertinent future research directions.

7.7 Numerical Tests

The control algorithm was tested for two utility network applications: voltage control in power systems and pressure control in water systems. Parameters $\boldsymbol{\eta}_2$ and $\boldsymbol{\eta}_3$ were chosen so that (7.13) holds true. The communication graphs $\mathcal{G}_c$ were arbitrarily chosen while ensuring connectivity.

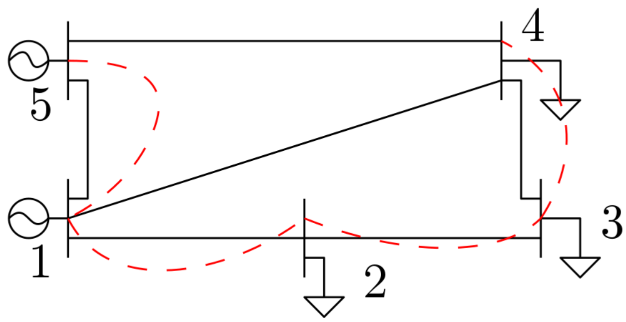

Figure 7.1: Modified PJM system. Communication links as red dashed lines.

A Power System Test Case The benchmark network used for the tests on power systems is a modified version[1] of the PJM 5-bus system [74] shown in Fig. 7.1. Buses 1 and 2 serve as generators, whereas buses 3, 4, and 5 are loads. Generators are initially at $u_n(0) = 1$ p.u., allowed to increase their voltage output to 1.02 pu; loads can reduce their power demand by 10 MW, and their voltages are required to be greater than $V_{thr} = 0.94$ p.u., which represents a safe threshold. The top panel of Figure 7.2 shows the bus voltage trajectories, whereas the bottom panel plots the ratio of control resources used $(u_n(t) - u_n(0)/(\bar{u}_n(t) - u_n(0))$. After an abnormal event, buses 2 and 3 have voltage less than V_{thr} and start performing the proposed control strategy. Because their local efforts do not manage to regulate the voltages, they seek assistance from their neighbors in the communication network, precisely, Bus 4 (around the 200-th iteration) and Bus 1 (around the 300-th iteration). Finally, when Bus 1 hits its control limits, Bus 5 kicks in (after the 400-th iteration) and is finally able to bring the voltage within the safe interval.

A Water Network Test Case The proposed algorithm was applied the pressure task on the 10-node WDS shown on Fig. 7.3. Pipe dimensions and friction coefficients were taken from [115]. The minimum pressure requirement $\underline{\pi}_n$ for nodes 3 to 10 is 10,7,10,10,5,10,10, and 10 m. The pumps $(1, 2)$, $(2, 5)$, and $(7, 3)$ are considered to be operating at fixed speeds

[1] In our numerical setup, the generators at Bus 3 and Bus 4 were neglected.

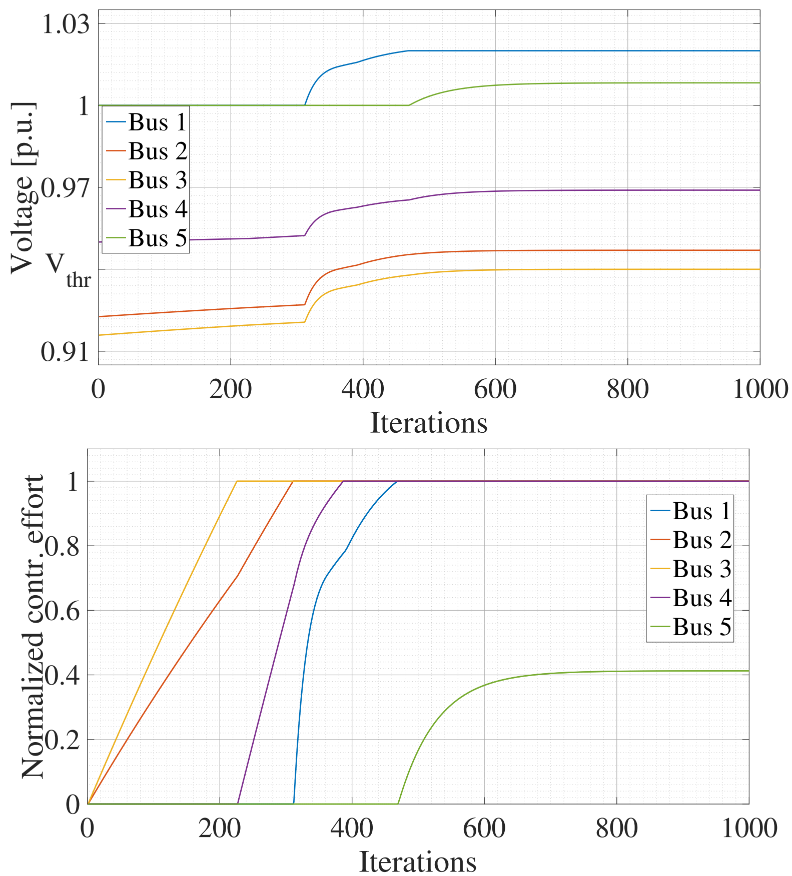

Figure 7.2: *Top:* Bus voltages. *Bottom:* Normalized control effort $(u_n(t) - u_n(0))/(\bar{u}_n(t) - u_n(0))$ used.

with constant pressure gains of 10, 10, and 5 m, respectively. All water flow instances were solved using the optimization-based solver of [110]. The base operating condition involves injection

$$\mathbf{d}_0 = [380, 300, -170, 0, 0, -220, -200, -150, -140, 200]^\top m^3/hr$$

and

$$\boldsymbol{\pi} = [3.0, 1.8, 11.6, 13, 11.9, 10.2, 6.6, 10.3, 10.9, 12.9]^\top m.$$

A disruption was modeled by considering a failure of pump $(2, 5)$, resulting in the unavailability of the Node 2 reservoir and the reservoir at Node 1 supplying $d_1 = 680\text{m}^3/\text{hr}$. This contingency would result in $\boldsymbol{\pi} = [3.0, 0, 9.1, 13, 8.8, 7.7, 4.1, 8.2, 9.6, 12.7]^\top$ m, which violates

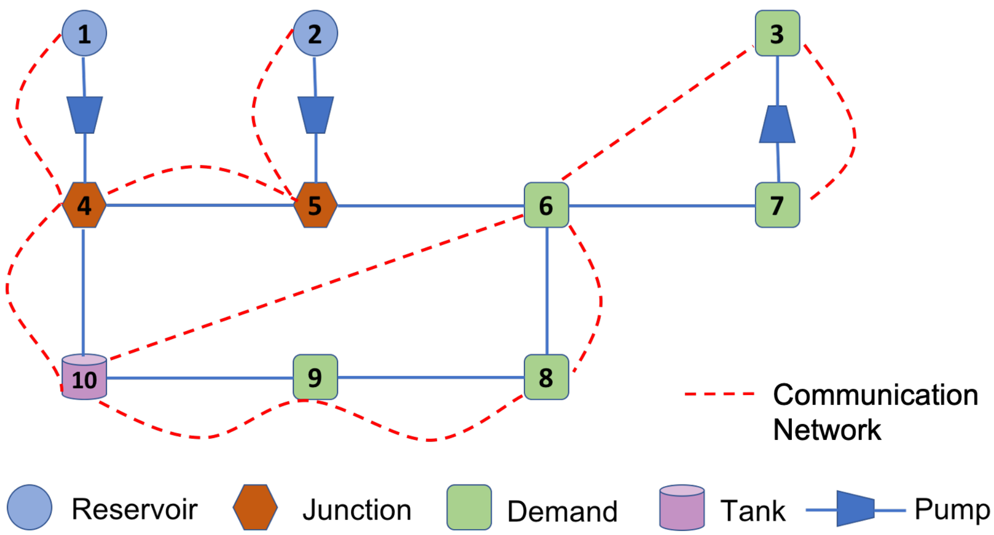

Figure 7.3: A 10-node benchmark water distribution system.

the minimum pressure needed at nodes 3 and $5-9$.

To study a scenario where not all agents have control capability, only two demand nodes –namely 3 and 8– are allowed to reduce demand by 50 m^3/h. Further, the tank node 10 has an injection flexibility of ± 200 m^3/h and reservoir 1 has a controllable pressure range of $[0, 5]$ m. The performance of the proposed algorithm in restoring the pressures is demonstrated in Fig. 7.4. The top panel shows the difference $\pi_n - \underline{\pi}_n$ for nodes $n = \{3 \ldots 10\}$. As anticipated, the pressures are nondecreasing and the algorithm succeeds in restoring them above the respective lower limits. The bottom panel plots the normalized nodal control effort. As desired, once all pressures are restored to the desired levels, the algorithm attains an equilibrium, and $\mathbf{u}$ saturates.

7.8 Conclusions

A ripple-type coordination scheme for the emergency control of networked systems has been put forth. Agents act based on local control rules as long as local resources have not been

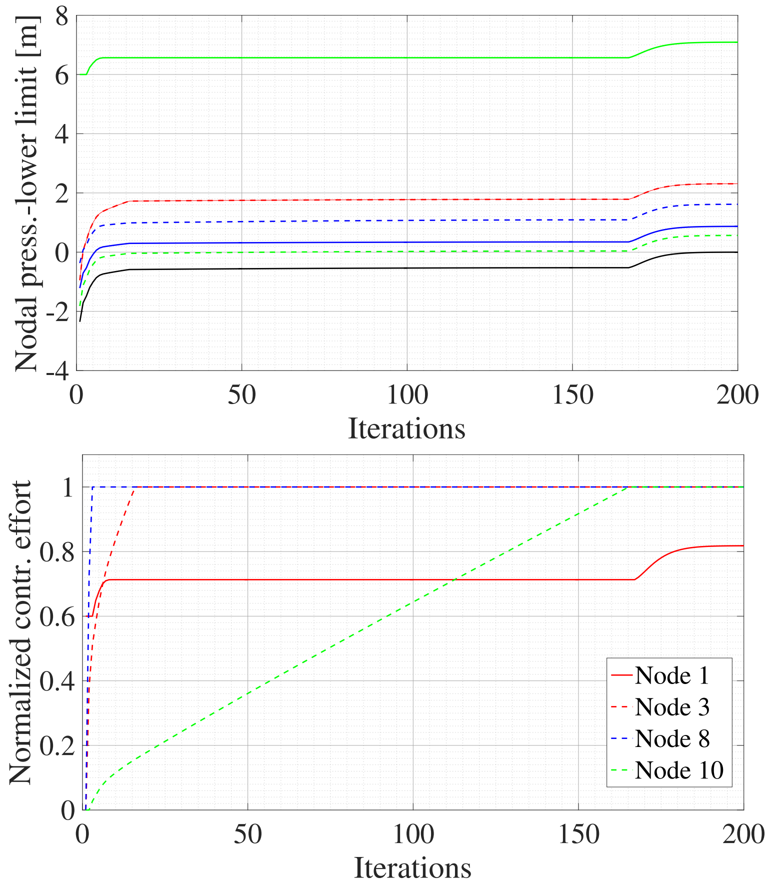

Figure 7.4: *Top:* Difference of pressures at nodes 3-10 from their lower limits. *Bottom:* Normalized control effort at nodes 1, 3, 8, and 10

saturated. Otherwise, they solicit help from peer agents through a "hotline" communication network not necessarily coinciding with the physical system graph. The algorithm provably converges to safe operating conditions under an appropriate choice of parameters. Its validity has been illustrated on power and water network examples. Future research directions include enforcing upper limits on the output variables and optimally designing control parameters.

Appendix E

Proof of Proposition 7.4 Owing to the projection in (7.12), it is evident that $\mathbf{u}(t) \leq \bar{\mathbf{u}}$ for all t; thus, proving a non-decreasing property $\mathbf{u}(t) \leq \mathbf{u}(t+1)$ for all t is sufficient for establishing convergence of the sequence $\{\mathbf{u}(t)\}$. Consider an arbitrary node n and time t. When $u_n(t+1) = \bar{u}_n$ we trivially have $u_n(t) \leq u_n(t+1)$. When $u_n(t+1) < \bar{u}_n$, *Step 4* yields $\hat{u}_n(t+1) = u_n(t+1)$ and (7.10) implies:

$$u_n(t+1) = u_n(t) + \eta_{1,n} f_n(\mathbf{u}(t)) + \eta_{2,n} \sum_{m=1}^{N} A_{nm} \lambda_m(t) \geq u_n(t)$$

because $f_n(\mathbf{u}(t))$ and $\boldsymbol{\lambda}_m(t)$ are nonnegative; $\eta_{1,n}$ and $\eta_{2,n}$ are positive; and matrix $\mathbf{A}$ has nonnegative entries.

Proof of Proposition 7.5 Assume $\mathbf{u} \in \mathcal{F}$ and $\boldsymbol{\lambda} = \mathbf{0}$. It follows that $\mathbf{y}(\mathbf{u}) \geq \underline{\mathbf{y}}$ and so $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ from (7.9). Upon initializing the proposed control scheme at $\mathbf{u}$ and $\boldsymbol{\lambda}$, *Step 1* provides $\mathbf{f}(\mathbf{u}) = \mathbf{0}$; *Step 2* yields $\hat{\mathbf{u}}(1) = \mathbf{u}$; *Step 3* provides $\boldsymbol{\lambda}(1) = \mathbf{0}$; and *Step 4* that $\mathbf{u}(1) = \mathbf{u}$. Therefore, $(\mathbf{u},\ \boldsymbol{\lambda})$ is an equilibrium for the proposed control steps.

To establish the reverse direction, we will prove the contrapositive statement, i.e., if $\mathbf{u}$ does not belong to $\mathcal{F}$ or $\boldsymbol{\lambda} \neq \mathbf{0}$, then $(\mathbf{u}, \boldsymbol{\lambda})$ is not an equilibrium. We show the two cases separately using proof by contradiction.

Case 1) Suppose $\mathbf{u} \notin \mathcal{F}$, yet there exists a $\boldsymbol{\lambda}$ such that $(\mathbf{u}, \boldsymbol{\lambda})$ is an equilibrium for the algorithm. Because $\mathbf{u}$ is projected in its permissible range by (7.12), its infeasibility means there exists a node $n \in \mathcal{Y}$ such that:

$$\underline{y}_n > y_n(\mathbf{u}). \tag{7.14}$$

Step 2 dictates $\hat{u}_n = u_n + \eta_{1,n} f_n(\mathbf{u}) + \eta_{2,n} \sum_{\ell=1}^{N} A_{n,\ell}\lambda_\ell$.

Because $f_n(\mathbf{u}) > 0$, it follows that $\hat{u}_n > u_n$; however, since *Step 4* sets $u_n = \min\{\hat{u}_n, \bar{u}_n\}$, it follows that $u_n = \bar{u}_n$ and $\hat{u}_n > \bar{u}_n$, so $\lambda_n > 0$ from (7.11).

Consider a node m neighbor of n in $\mathcal{G}_c$ and compute $\hat{u}_m = u_m + \eta_{1,m} f_m(\mathbf{u}) + \eta_{2,m} \sum_{\ell=1}^{N} A_{m,\ell}\lambda_\ell$. Because $\lambda_n > 0$ and $u_m = \min\{\hat{u}_m, \bar{u}_m\}$, it holds again that $\hat{u}_m > u_m = \bar{u}_m$, so $\lambda_m > 0$. Repeating this argument for all neighbors of neighbors of n, and so on until all nodes have been covered, one gets $\boldsymbol{\lambda} > \mathbf{0}$ and:

$$\mathbf{u} = \bar{\mathbf{u}}. \tag{7.15}$$

However, because $\mathcal{F}$ is assumed to be non-empty, Assumption 7 implies that $\mathbf{y}(\bar{\mathbf{u}}) \geq \mathbf{y}(\mathbf{u}')$ for $\underline{\mathbf{u}} \leq \mathbf{u}' \leq \bar{\mathbf{u}}$; hence:

$$\mathbf{y}(\bar{\mathbf{u}}) \geq \underline{\mathbf{y}}. \tag{7.16}$$

In other words, applying the maximum control effort makes the constraints on observed outputs hold. This concludes the proof for Case 1 because (7.14) and (7.15) contradict (7.16).

Case 2): Suppose there exists a $\boldsymbol{\lambda} \neq \mathbf{0}$ and $\mathbf{u} \in \mathcal{F}$, such that $(\mathbf{u}, \boldsymbol{\lambda})$ is an equilibrium for the proposed scheme. By plugging (7.10) into (7.11), we can express $\boldsymbol{\lambda}(t+1)$ as:

$$\begin{aligned}\boldsymbol{\lambda}(t+1) = \max\big\{\mathbf{0}, \mathrm{dg}(\boldsymbol{\eta}_3)\big(\mathbf{u}(t) + \mathrm{dg}(\boldsymbol{\eta}_1)\mathbf{f}(\mathbf{u}(t)) + \\ \mathrm{dg}(\boldsymbol{\eta}_2)\mathbf{A}\boldsymbol{\lambda}(t) - \bar{\mathbf{u}}\big)\big\}.\end{aligned}$$

The feasibility assumption provides $\mathbf{f}(\mathbf{u}) = \mathbf{0}$, and the equilibrium condition yields:

$$\begin{aligned}\boldsymbol{\lambda} &= \max\big\{\mathbf{0}, \mathrm{dg}(\boldsymbol{\eta}_3)\big(\mathbf{u} + \mathrm{dg}(\boldsymbol{\eta}_2)\mathbf{A}\boldsymbol{\lambda} - \bar{\mathbf{u}}\big)\big\} \\ &\leq \mathrm{dg}(\boldsymbol{\eta}_3)\,\mathrm{dg}(\boldsymbol{\eta}_2)\mathbf{A}\boldsymbol{\lambda} \end{aligned} \tag{7.17}$$

where the inequality stems from the fact that $\mathbf{u} - \bar{\mathbf{u}} \leq \mathbf{0}$ and $\mathrm{dg}(\boldsymbol{\eta}_2)\,\mathrm{dg}(\boldsymbol{\eta}_3)\mathbf{A}\boldsymbol{\lambda}(t) \geq \mathbf{0}$. Invoking the norm inequality on (7.17), we get $\|\boldsymbol{\lambda}\| \leq \|\,\mathrm{dg}(\boldsymbol{\eta}_2)\,\mathrm{dg}(\boldsymbol{\eta}_3)\mathbf{A}\|\|\boldsymbol{\lambda}\|$. Also, using the condition $\|\,\mathrm{dg}(\boldsymbol{\eta}_2)\,\mathrm{dg}(\boldsymbol{\eta}_3)\mathbf{A}\| < 1$ with $\boldsymbol{\lambda} \neq \mathbf{0}$ yields $\|\boldsymbol{\lambda}\| < \|\boldsymbol{\lambda}\|$, which is a contradiction.

Chapter 8

Optimal Power Flow under Dynamic Stability Considerations

8.1 Abstract

The dynamic response of power grids to small events or persistent stochastic disturbances influences their stable operation. The effect of network topology, line impedances, inertia and damping parameters on dynamic response is extensively being investigated. This paper studies the effect of the operating point on the linear time-invariant dynamics of power networks. For a variety of stability metrics, suitable observation matrices are identified to quantify stability via the $\mathcal{H}_2$-norm of the resulting system. We further put forth an optimal power flow formulation to yield a grid dispatch that optimizes the respective stability metrics. A semidefinite program (SDP) relaxation is employed to yield a computationally tractable convex problem. Numerical tests on the IEEE-39 bus system demonstrate that the SDP relaxation is exact yielding a rank-1 solution. The relative trade-off of different stability metrics versus the minimum generation cost is also studied.

8.2 Introduction

Power systems experience fluctuations of small magnitudes at all times, and are often exposed to large disturbances, as well. Stability, being the ability to withstand such disturbances, is naturally an extremely sought attribute. A diverse classification of power system stabililty studies can be made based on variables of interest, magnitude of disturbances, considered time-frame, and order and accuracy of component modeling; cf. [71], [62], [72], [35]. For large power transmission systems, rotor-angle stability has been widely investigated, where transient and dynamic stability studies investigate the effect of large and small disturbances, respectively.

Transient stability studies are primarily accomplished via time-series simulations or direct methods based on energy-type Lyapunov functions that characterize stability regions [35], [130]. Such approaches may not be practically scalable, and thus oftentimes confine transient stability analyses to a limited set of probable contingencies [55]. Transient stability is known to be strongly related to the operating point, and is actually studied accordingly. Attempts have also been made at formulating optimal power flow problems that yield operating points with guaranteed transient stability [55], [31].

For dynamic stability, the power network is oftentimes approximated by a linear time-invariant (LTI) system obtained upon linearizing the chosen dynamical model and the power flow equations at a given operating point. Thereafter, the system's dynamic stability is assessed by investigating the eigenvalues of the related *state matrix*; specifically ensuring that they have negative real parts. To capitalize the dependence of dynamic stability on the operating point, several works aim at either perturbing or obtaining operating points, while minimizing the *spectral abscissa*, i.e, the real-part of the eigenvalue closest to the imaginary axis [36], [142] [76]. The tractability of these approaches is often challenged by a need to iteratively compute eigenvalues, and the nonsmoothness of spectral abscissa with respect to state variables [97], [76]. To circumvent explicit eigenvalue computation, Lyapunov function-based approaches have also been proposed that involve bilinear matrix inequalities [97], [75]. Instead of obtaining operating points explicitly, several works incorporate Lyapunov function-based approach in designing stability-ensuring control mechanisms [120], [11].

For typical power system operating conditions, the LTI system used for rotor-angle dynamic stability analysis is found to be marginally stable with one pole at the origin [35], [98]. Such observations have motivated simpler modeling amenable to more physically insightful analysis, specifically, the classical second-order swing dynamics model [72]. With simplified modeling, stability metrics more systematic and insightful than the spectral abscissa have been proposed [95], [123]. The effect of network size, topology, and dynamic model parameters on diverse stability metrics has been widely investigated; and motivated resource planning and placement studies [123], [98], [118]. However, investigations on the effect of operating point on such stability metrics are relatively sparse. While the planning studies involving sophisticated metrics assume a fixed operating point, most works that aim at determining stability-improving operating points limit themselves to spectral abscissa [97], [142].

In this paper, we first demonstrate that for various practical metrics of dynamic stability, a prudent choice of the observation matrix can ensure that the $\mathcal{H}_2$-norm of the resulting LTI system is equivalent to the desired stability metric. Next, we formulate an SDP-based OPF to minimize the $\mathcal{H}_2$-norm. Finally, numerical tests on a benchmark system demonstrate that the SDP relaxation is exact. A trade-off between dynamic stability and generation-cost is also studied numerically.

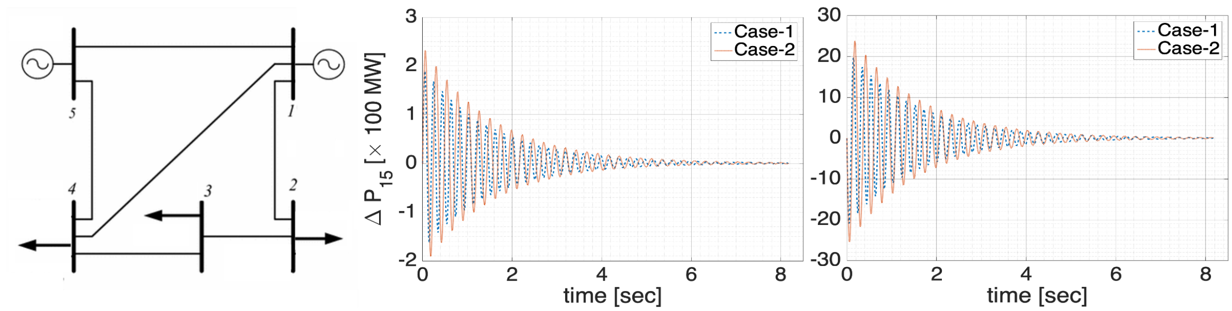

Figure 8.1: The left panel shows a modified PJM 5-bus power system with two generators and three loads. The loads were $s_2^d = s_3^d = 300 + j98.61$, and $s_4^d = 400 + 131.47$ MVA. The center and right panel show the the deviations in active power flow on line $(1, 5)$ obtained from LTI systems for impulse inputs on bus 1 and 5, respectively. Case-1 and 2 correspond to different operating points used for linearization.

8.3 Problem Statement and Motivation

Stability and security are key features to be considered during power system operation. Catering to the abstractness and evolving understanding of these notions themselves, there has been a concerted effort in streamlining stability-related definitions and metrics [71], [62]. Thus the studies aimed at improving system stability resort to various metrics based on the intended application. Popular industry metrics include event-based measures of frequency nadir, settling times, bus frequency deviations from the center of inertia, and network coherence; see e.g., [95]. System theoretic metrics invoked in analytical studies include the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms [98], [123], [120]. There is a rich and growing corpus of planning studies that optimize upon the aforementioned stability metrics while selecting network topologies and line impedances, placing virtual inertia and damping through inverter-interfaced resources, and controller design [98], [18].

Analytical studies rely on dynamical power system models linearized at a chosen point typically selected to be the steady-state operating point. Hence inevitably most metrics are affected by the choice of this operating point. Nevertheless, in lack of a lucid characterization of such dependence, most planning studies limit their formulations to nominal operating conditions or even the so termed flat voltage profile experienced by an energized yet unloaded network. Similarly, the studies pertaining to system operation oftentimes assess system stability assuming a known and fixed operating point. Before confidently adopting this practice, one needs to answer: *q1) How strong is the dependence of power system dynamic stability on the operating point?*

To emphasize on the pertinence of *q1)*, we provide an illustrative example on the PJM 5-bus power system. We evaluated the dynamic performance of the modified 5-bus system shown on the left panel of Fig. 8.1 for two dispatch instances serving identical load conditions, but differing in generation as:

Case C1) $s_1^g = 0 + j388$ MVA, $s_2^g = 1013 + j68$ MVA;

Case C2) $s_1^g = 500 + j382$ MVA, $s_2^g = 518 + j122$ MVA.

The resulting voltages were determined by solving the AC power flow equations, and the respective second-order LTI systems were obtained at the two operating points. The center and right panels of Figure 8.1 show the deviation from the nominal active power flow on the line connecting buses 1 and 5 for an impulse input at these two buses. Two interesting observations are in order. First, for a fixed load, changing the generator dispatch may yield non-trivial differences in dynamic performance; the oscillations under *C1)* have lower amplitude than those under *C2)*. Second, an impulse power injection at bus 5 induces oscillations of much higher amplitude compared to an impulse at bus 1. This corroborates the need for tactful selection of a stability metric.

Power systems are traditionally dispatched by solving a rendition of the optimal power flow (OPF) problem, which aims at minimizing the cost of generation while serving the anticipated loads subject to network constraints. Since the dynamic performance of a power systems can be significantly affected by the choice of the operating point decided by the OPF, one naturally asks: *q2) How to include dynamic stability considerations in OPF formulations?* Aiming to address *q1)-q2)*, this work articulates a model that captures the dependence of dynamic stability on the steady-state operating point of a power system. It subsequently puts forth a tractable OPF based on a stability-cognizant semidefinite program (SDP) relaxation, which allows operators to systematically study the cost-stability trade-off. The aforesaid tasks are undertaken while carefully selecting the *observation matrices*, which facilitate the use of $\mathcal{H}_2$-norm to represent metrics of direct interest to industry practice.

8.4 Grid Modeling and Stability Metrics

This section reviews a linearized model of power system dynamics, defines stability metrics, and establishes their connection with the $\mathcal{H}_2$-norm of a linear dynamical system.

8.4.1 Algebraic Power System Model

A power transmission network with $N+1$ buses can be represented by an undirected connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, whose nodes $n \in \mathcal{N} := \{0, 1, \dots, N\}$ correspond to buses, and edges $\ell = (m, n) \in \mathcal{L}$ to transmission lines. Let $\mathbf{v}$ and $\mathbf{i}$ denote $(N+1)$-length vectors collecting respectively the complex voltage and injection current at bus $n \in \mathcal{N}$. Although phasors in $(\mathbf{v}, \mathbf{i})$ are expressed in rectangular (Cartesian) coordinates, we also denote the voltage magnitude and phase at bus n as V_n and θ_n, respectively. Given line impedances and shunt susceptances, one can derive the $(N+1) \times (N+1)$ bus admittance matrix $\mathbf{Y}$, satisfying the Ohm's law $\mathbf{i} = \mathbf{Y}\mathbf{v}$.

Set $\mathcal{N}$ constitutues a collection of generation, load, and zero-injection buses. In high-voltage transmission systems, generator buses primarily host synchronous machines. Load buses however oftentimes denote the connection to sub-transmission networks, which indeed host numerous medium-sized synchronous generators and motors, alongside other types of loads. We therefore collectively refer to generation and load buses as synchronous buses, collect them in $\mathcal{S} \subset \mathcal{N}$, and index them by $n = 0, \ldots, S$. Non-synchronous buses form set $\bar{\mathcal{S}} = \mathcal{N} \setminus \mathcal{S}$ and are indexed by $n = S+1, \ldots, N$. For synchronous buses, voltage $v_n = V_n \angle \theta_n$ captures the *external* bus voltage; the *internal* or Thevenin-equivalent voltage is denoted by $e_n = E_n \angle \delta_n$. Complex vector $\mathbf{e}$ collects all $\{e_n\}_{n=1}^S$. Each internal bus is connected to its associated external bus n via reactance $x_n > 0$. Define $\mathbf{Y}_\mathcal{S} = \mathrm{dg}(\{1/(jx_n)\})$ to express the Ohm's law using matrix-vector partitions as [66]

$$\begin{bmatrix} \mathbf{Y}_{\mathcal{S},\mathcal{S}} & \mathbf{Y}_{\mathcal{S},\bar{\mathcal{S}}} \\ \mathbf{Y}_{\mathcal{S},\bar{\mathcal{S}}}^\top & \mathbf{Y}_{\bar{\mathcal{S}},\bar{\mathcal{S}}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_\mathcal{S} \\ \mathbf{v}_{\bar{\mathcal{S}}} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_\mathcal{S}(\mathbf{e} - \mathbf{v}_\mathcal{S}) \\ \mathbf{0} \end{bmatrix} \tag{8.1}$$

Eliminating $\mathbf{v}_{\bar{\mathcal{S}}}$ (a process also known as *Kron reduction* in the electrical circuit or graph theory parlance) provides

$$\mathbf{v}_\mathcal{S} = \mathbf{\Gamma} \mathbf{Y}_\mathcal{S} \mathbf{e}. \tag{8.2}$$

where matrix $\mathbf{\Gamma} := (\mathbf{Y}_{\mathcal{S},\mathcal{S}} + \mathbf{Y}_\mathcal{S} - \mathbf{Y}_{\mathcal{S},\bar{\mathcal{S}}} \mathbf{Y}_{\bar{\mathcal{S}},\bar{\mathcal{S}}}^{-1} \mathbf{Y}_{\mathcal{S},\bar{\mathcal{S}}}^\top)^{-1}$ is introduced to declutter notation and is known to exist [44].

Equation (8.2) lays a fundamental connection between traditional OPF formulations on one hand, and dynamic stability studies on the other. Let us be more specific and commence with OPF formulations. From the power flow equations [58, Ch. 6], the compex power $p_n + jq_n$ injected at bus $n \in \mathcal{N}$ relates to $\mathbf{v}$ as

$$p_n = \mathbf{v}^H \mathbf{M}_{p_n} \mathbf{v} \tag{8.3a}$$

$$q_n = \mathbf{v}^H \mathbf{M}_{q_n} \mathbf{v} \tag{8.3b}$$

where $\mathbf{M}_{p_n}$ and $\mathbf{M}_{q_n}$ are $(N+1) \times (N+1)$ Hermitian matrices [69].

Although OPF formulations relate power injections to external voltages $\mathbf{v}$, dynamic stability studies tend to eliminate external voltages and derive power injections at synchronous buses using $\mathbf{e}$ [66]. This is accompished by recognizing that the active power injection p_n can be interpreted as the active power flow over reactance x_n across the internal voltage e_n and external voltage v_n, so that

$$p_n = \frac{E_n V_n}{x_n} \sin(\delta_n - \theta_n) = \frac{1}{x_n} \operatorname{Im}\{e_n v_n^*\}. \tag{8.4}$$

Substituting v_n from (8.2) in (8.4), one obtains

$$\begin{aligned} p_n = \sum_{m=0}^{S} \frac{E_n E_m}{x_n x_m} \big[\operatorname{Re}\{\Gamma_{nm}\} \cos(\delta_n - \delta_m) \\ - \operatorname{Im}\{\Gamma_{nm}\} \sin(\delta_n - \delta_m) \big]. \end{aligned} \tag{8.5}$$

Owing to the observation that $|\operatorname{Re}(\Gamma_{nm})| \ll |\operatorname{Im}(\Gamma_{nm})|$, the assumption of a lossless transmission system is oftentimes invoked to obtain

$$p_n = \sum_{m=0}^{S} \frac{E_n E_m}{\gamma_{nm}} \sin(\delta_n - \delta_m). \tag{8.6}$$

where $\gamma_{nm} := -(x_n x_m)/\operatorname{Im}(\Gamma_{nm}) > 0$ serves as the *effective reactance* between the synchronous machines connected to bus m and n [66]. By symmetry, it holds that $\gamma_{nm} = \gamma_{mn}$. We next use (8.6) to build an LTI model for power system dynamics, and leave (8.3) aside for later OPF formulations.

8.4.2 Dynamic Power System Model

This section reviews the angular dynamics of the equivalent synchronous machines connected to the buses in $\mathcal{S}$. Let the synchronous machine connected to bus $n \in \mathcal{S}$ be associated with a frequency $\omega_n = \dot{\delta_n}$, inertia constant $M_n > 0$, and damping coefficient $D_n > 0$; see [72] for details. Recall that the complex phasor representation $E_n \angle \delta_n(t)$ corresponds to a sinusoidal voltage waveform $\sqrt{2} E_n \cos(\omega t + \delta_n(t))$, where ω is the nominal angular frequency, e.g., $\omega = 2\pi \cdot 60$ rad/sec in the US. The angular frequency ω_n is the deviation from the nominal, which ideally should be close to zero. The swing dynamics for the synchronous machine connected to bus n dictate [72]

$$M_n \dot{\omega}_n + D_n \omega_n = p_n^g - p_n - p_n^d \tag{8.7}$$

where p_n^g denotes the mechanical power input; p_n is the electric power flowing from bus n to the grid; and p_n^d is the power demand at bus n. While the swing equation (8.7) provides individual machine dynamics, the power injection p_n couples the system of $S+1$ machines via (8.6). Denote the vectors of frequencies and angles as $\boldsymbol{\omega}$ and $\boldsymbol{\delta}$, respectively. Further, let the state tuple $(\boldsymbol{\omega}^0, \boldsymbol{\delta}^0)$ correspond to an equilibrium of (8.7), that is $\dot{\omega}_n = 0$. We are particularly interested in frequency-synchronized solutions of (8.7), implying $\boldsymbol{\omega}^0 = \omega^0 \mathbf{1}$. Because $\sum_{n=0}^{S} p_n = 0$ from (8.6), one obtains

$$\omega^0 = \frac{\sum_{n=0}^{S} p_n^{in}}{\sum_{n=0}^{S} D_n}$$

where $p_n^{in} := p_n^g - p_n^d$ is the net power input at bus n.

We assume for convenience $\sum_{n=0}^{S} p_n^{in} = 0$, yielding $\omega^0 = 0$ and a constant vector $\boldsymbol{\delta}^0$ at equilibrium. If the assumption is violated, one can seamlessly resort to a change of variables

$\tilde{\omega}_n = \omega_n - \omega^0$, $\tilde{p}_n^{in} = p_n^{in} - \omega^0 D_n$, and $\tilde{\delta}_n(t) = \delta_n(t) - \omega^0 t$. Proceeding with $\omega^0 = 0$, we note from (8.7) and (8.6) that the angles $\boldsymbol{\delta}^0$ at equilibrium are such that

$$p_n^{in} = p_n = \sum_{m=0}^{S} \frac{E_n E_m}{\gamma_{nm}} \sin(\delta_n^0 - \delta_m^0). \tag{8.8}$$

We next obtain an LTI by linearizing (8.7) at the equilibrium. To this end, assume constant voltage magnitudes and consider small perturbations in (p_n^{in}, p_n) to $(p_n^{in} + \Delta p_n^{in}, p_n + \Delta p_n)$ corresponding to angles $\delta_n^0 + \Delta\delta$. Linearizing (8.6) yields

$$\Delta \mathbf{p} \simeq \mathbf{L}_{\boldsymbol{\delta}^0} \Delta \boldsymbol{\delta} \tag{8.9}$$

where $\Delta \mathbf{p}$ and $\Delta \boldsymbol{\delta}$ collect respectively $\{\Delta p_n\}_{n=0}^S$ and $\{\Delta\ \delta_n\}_{n=0}^S$, while the Jacobian matrix $\mathbf{L}_{\boldsymbol{\delta}^0}$ is evaluated at $\boldsymbol{\delta} = \boldsymbol{\delta}^0$ and has the entries

$$[\mathbf{L}_{\boldsymbol{\delta}^0}]_{n,m} = \begin{cases} \sum_{k \neq n} \frac{E_n E_k}{\gamma_{nk}} \cos(\delta_n^0 - \delta_k^0) & , \ m = n \\ \\ -\frac{E_n E_m}{\gamma_{nm}} \cos(\delta_n^0 - \delta_m^0) & , \ m \neq n. \end{cases} \tag{8.10}$$

Note that since $\gamma_{nm} > 0$, and for admissible power system operating points, it holds that $|\delta_n^0 - \delta_m^0| < \pi/2$, the Jacobian $\mathbf{L}_{\boldsymbol{\delta}^0}$ is in fact a Laplacian matrix [56].

Let the diagonal matrices $\mathbf{M}$ and $\mathbf{D}$ carry on their main diagonals the inertia and damping coefficients, respectively. Linearizing (8.7) for all buses in $\mathcal{S}$ provides

$$\mathbf{M}\dot{\boldsymbol{\omega}} + \mathbf{D}\boldsymbol{\omega} + \mathbf{p} + \mathbf{L}_{\boldsymbol{\delta}^0} \Delta \boldsymbol{\delta} = \mathbf{p}^{in} + \Delta \mathbf{p}^{in} \tag{8.11}$$

where the input and injected powers cancel out due to (8.8). Note that $\boldsymbol{\omega} = \dot{\boldsymbol{\delta}}^0 + \Delta\dot{\boldsymbol{\delta}} = \Delta\dot{\boldsymbol{\delta}}$. Therefore, the LTI system (8.11) involves the deviations of states and inputs from their nominal values. For notational brevity, we henceforth omit Δ's, while the absolute and differential quantities will be discernible from context. To study a power system under small-signal disturbances, the linearized dynamics of (8.7) can be compactly expressed using (8.9) as [56]

$$\begin{bmatrix} \dot{\boldsymbol{\delta}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{L}_{\boldsymbol{\delta}^0} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix}}_{\mathbf{A}:=} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\omega} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix}}_{\mathbf{B}:=} \mathbf{p}^{in}. \tag{8.12}$$

Proceeding to the observables of interest, suppose one seeks K_θ outputs relating to angles, and K_ω outputs relating to frequencies. Because the relevant observables on power system angles and frequencies are disparate, it is customary to impose a block-diagonal structure on the observation matrix as

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} \end{bmatrix}}_{\mathbf{C}:=} \begin{bmatrix} \boldsymbol{\delta} \\ \boldsymbol{\omega} \end{bmatrix} \tag{8.13}$$

where $\mathbf{C}_{11} \in \mathbb{R}^{K_\delta \times (S+1)}$ and $\mathbf{C}_{22} \in \mathbb{R}^{K_\omega \times (S+1)}$. The specific choices for $\mathbf{C}$ that yield some commonly sought power system stabiity metrics are detailed in Section 8.4.4. For now, given matrix $\mathbf{C}$, we expound on quantifying the stability of the LTI system (8.12)–(8.13).

For LTI systems, the $\mathcal{H}_2$-norm is a commonly used measure for quantifying stability. Given the LTI system H defined by matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ via (8.12)–(8.13), its $\mathcal{H}_2$-norm may be computed as $\|H\|_{\mathcal{H}_2}^2 = \mathrm{Tr}(\mathbf{B}^\top \mathbf{Q} \mathbf{B})$, where $\mathbf{Q}$ is the observability Grammian matrix $\mathbf{Q} \succeq \mathbf{0}$. Matrix $\mathbf{Q}$ is defined as

$$\mathbf{Q} := \int_0^\infty e^{\mathbf{A}^\top t} \mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} \, \mathrm{d}t. \tag{8.14}$$

In practice $\mathbf{Q}$ is computed by solving the Lyapunov equation $\mathbf{A}^\top \mathbf{Q} + \mathbf{Q}\mathbf{A} = -\mathbf{C}^\top \mathbf{C}$. The $\mathcal{H}_2$-norm is generally defined only for stable systems. Owing to matrix $\mathbf{L}_{\boldsymbol{\delta}^0}$ being a Laplacian, matrix $\mathbf{A}$ in (8.12) has $[\mathbf{1}^\top \; \mathbf{0}^\top]^\top$ in its nullspace. Nevertheless, all other poles of the system are strictly in the negative complex half-plane, hence resulting in a marginally-stable system. Furthermore, the marginally stable eigenvector of $\mathbf{A}$ corresponds to a constant shift in angles $\boldsymbol{\delta}$. Since angles in power systems are anyways defined with respect to a reference, a uniform shift is often non-detectable. Thus, the next assumption holds in practice

Assumption 8. The outputs are chosen such that $\mathbf{C}_{11}\mathbf{1} = \mathbf{0}$.

Thanks to Assumption 8, despite the system (8.12)–(8.13) being marginally stable, the integral (8.14) remains finite, resulting in a finite $\mathcal{H}_2$-norm [98]. We next characterize the dependence of $\|H\|_{\mathcal{H}_2}$ on the power system operating point, and subsequently, relate $\|H\|_{\mathcal{H}_2}$ to practical power system stability metrics.

8.4.3 Relating $\mathcal{H}_2$-norm to the Operating Point

As evident from (8.14), matrix $\mathbf{Q}$ depends on matrix $\mathbf{A}$ and hence $\mathbf{L}_{\boldsymbol{\delta}^0}$, which depends on the power system operating point via the internal voltages $\mathbf{e}$ [cf. (8.10)]. Aiming for a more stable power system operation, one may attempt finding an OPF dispatch that in addition to minimizing generation costs, it yields a relatively small norm $\|H\|_{\mathcal{H}_2}$. Since dealing with such task for a general transmission network may be challenging, we resort to a widely adopted assumption to render our analysis tractable.

Assumption 9. Damping coefficients are identical at all buses, that is $D_n = d$ for all $n \in \mathcal{S}$.

Under Assumption 9, the $\mathcal{H}_2$-norm of the linearized swing dynamics simplifies as [98]

$$\|H\|^2_{\mathcal{H}_2} = \frac{1}{2d}\left(\mathrm{Tr}(\mathbf{C}_{11}^\top \mathbf{C}_{11}\mathbf{L}^\dagger_{\boldsymbol{\delta}^0}) + \mathrm{Tr}(\mathbf{C}_{22}^\top \mathbf{C}_{22}\mathbf{M}^{-1})\right) \tag{8.15}$$

where $\mathbf{L}^\dagger_{\boldsymbol{\delta}^0}$ is the pseudoinverse of $\mathbf{L}_{\boldsymbol{\delta}^0}$. Equation (8.15) eatablishes a direct relation between the $\mathcal{H}_2$-norm and the power system operating point. Capitalizing this closed-form relation, Section 8.5 proposes a novel tractable OPF formulation that minimizes $\|H\|_{\mathcal{H}_2}^2$. However, justifying the choice of $\mathcal{H}_2$-norm as the stability metric to be optimized is due. We address this in two steps: Section 8.4.4 introduces choices of matrix $\mathbf{C}$ that yield metrics of practical interest as the output norm $\|\mathbf{y}\|_2^2$; and Section 8.4.5 lists interpretations of $\|H\|_{\mathcal{H}_2}^2$-norm as a measure of $\|\mathbf{y}\|_2^2$ for different inputs $\mathbf{p}^{in}$ to the LTI system H.

Remark 8.1. In this work, the load model ascribes inertia and damping attributes to all loads, thus imparting the convenience of identical mathematical treatment for loads and generators [66], [98]. Another popular transmission system-level model assumes frequency-dependent loads with $M_n = 0$ and $D_n > 0$; see e.g., [118], [147]. The resulting first-order load model results in fewer system states; specifically for N_g generators, the corresponding LTI system has $S + N_g + 1$ states, compared to the $2S + 2$ states in (8.12), [118]. Fortunately, the frequency-dependent load model also enjoys a closed-form expression for $\mathcal{H}_2$-norm similar to (8.15) [61]. Thus, the approach proposed in this work can be readily incorporated is such settings.

8.4.4 Network Stability Metrics

As noted in Section 8.4.2, the LTI system H is asymptotically stable, except for one marginally stable mode. The stability guarantee however is valid only for small disturbances that do not jeopardize the assumptions for linearization. Thus, it is desirable that the angular differences across lines and nodal frequencies exhibit small deviations from their nominal values. One common metric of stability is *network coherence*, which quantifies how much voltage angles differ from their network average [123] [9]

$$f_c(t) := \sum_{m=0}^{S} \left(\delta_m(t) - \frac{1}{S+1} \sum_{n=0}^{S} \delta_n(t) \right)^2 . \tag{8.16}$$

The metric $f_c(t)$ can be expressed as $\|\mathbf{y}(t)\|_2^2$ in (8.13) by selecting $\mathbf{C}_{11} = \mathbf{I} - \frac{1}{S+1}\mathbf{1}\mathbf{1}^\top$ and $\mathbf{C}_{22} = \mathbf{0}$.

In a large multi-area power system, an area-wise coherency may be more practical. Such metric can be captured through $\|\mathbf{y}(t)\|_2^2$ by choosing a block-diagonal $\mathbf{C}_{11}$. Furthermore, a metric for voltage angle differences across all lines can be obtained by selecting $\mathbf{C}_{11}$ as the (non-weighted) grid Laplacian matrix. Any metric related exclusively on voltage angles will be henceforth referred to as $f_\delta(t)$.

In addition to stability metrics on voltage angles, metrics of practical interest can be imposed on angular frequencies as well. One such important metric is the deviation of frequencies from their *center of inertia* (COI) with the latter defined as [72]

$$\bar{\omega}(t) := \frac{\sum_{n=0}^{S} M_n \omega_n(t)}{\sum_{n=0}^{S} M_n}. \tag{8.17}$$

The departure of frequencies from the center of inertia is a measure of deviation from *synchrony*, and may be defined as [95]

$$f_s(t) := \sum_{n=0}^{S} (\omega_n(t) - \bar{\omega}(t))^2. \tag{8.18}$$

This metric can be made equal to $\|\mathbf{y}(t)\|_2^2$ by choosing $\mathbf{C}_{11} = \mathbf{0}$ and $\mathbf{C}_{22} = \mathbf{I} - \frac{1}{\mathbf{1}^\top \mathbf{M}\mathbf{1}}\mathbf{1}\mathbf{1}^\top \mathbf{M}$. Different from characterizing synchrony, a simpler stability metric could be the sum of squared frequency excursions obtained by selecting $\mathbf{C}_{22} = \mathbf{I}$. Metrics on angular frequencies will be henceforth denoted by $f_\omega(t)$.

As evident from (8.15), if the stability criterion of interest relates solely to frequencies (that is $\mathbf{C}_{11} = \mathbf{0}$), the $\mathcal{H}_2$-norm is independent of $\mathbf{L}_{\boldsymbol{\delta}^0}$ and depends only on the inertia and damping parameters. Nonetheless, in such cases the operating point still affects frequency dynamics in ways not necessarily captured by the $\mathcal{H}_2$-norm. Basic linear system theory dictates that the step response of voltage frequencies coincides with the impulse response of voltage angles. Therefore, if one is interested in studying a stability metric $f_\omega(t)$ obtained by choosing $(\mathbf{C}_{11}, \mathbf{C}_{22}) = (\mathbf{0}, \mathbf{C}_\omega)$ under a step disturbance, it suffices to choose $(\mathbf{C}_{11}, \mathbf{C}_{22}) = (\mathbf{C}_\omega, \mathbf{0})$ and analyze the resulting impulse response of $\|\mathbf{y}(t)\|_2^2$. Such reformulation enables us to study the effect of the operating point and network parameters on the power network frequency dynamics as well.

8.4.5 Relating Stability Metrics to $\mathcal{H}_2$-norm

We so far have related different stability metrics $f_\delta(t)$ and $f_\omega(t)$ to the output norm $\|\mathbf{y}(t)\|_2^2$, and realized that the $\mathcal{H}_2$-norm depends on the operating point only if $\mathbf{C}_{11} \neq \mathbf{0}$. Since the popular choices of stability metrics for angles and frequencies are distinct, we seek setups with $\mathbf{C}_{11} \neq \mathbf{0}$ and $\mathbf{C}_{22} = \mathbf{0}$. This section establishes the relation between different stability metrics and the $\|H\|_{\mathcal{H}_2}^2$-norm, while respecting the aforementioned restriction on the structure of $\mathbf{C}$. With a suitable choice of $\mathbf{C}$, the $\mathcal{H}_2$-norm can be shown to be equivalent to:

A1) The expected steady-state value of an angle-based stability metric for unit-variance stochastic white noise input $\mathbf{p}^{in}(t)$

$$\|H\|_{\mathcal{H}_2}^2 := \lim_{t\to\infty} \mathbb{E}\left[\|\mathbf{y}(t)\|_2^2\right] = \lim_{t\to\infty} \mathbb{E}\left[f_\delta(t)\right];$$

A2) The sum of time-integrals of a angle-based stability metric $f_\delta^n(t)$ for unit-impulse disturbances $\mathbf{p}_n^{in}(t) = \mathbf{e}_n\delta(t)$[1] per bus $n \in \mathcal{S}$

$$\|H\|_{\mathcal{H}_2}^2 = \sum_{n=0}^{S} \int_0^\infty f_\delta^n(t)\,\mathrm{d}t$$

A3) The sum of time-integrals of a frequency-based stability metric $f_\omega^n(t)$ for unit-step disturbances $\mathbf{p}_n^{in}(t) = \mathbf{e}_n u(t)$ per bus $n \in \mathcal{S}$

$$\|H\|_{\mathcal{H}_2}^2 = \sum_{n=0}^{S} \int_0^\infty f_\omega^n(t)\,\mathrm{d}t.$$

Having established the equivalence of various stability metrics to the $\mathcal{H}_2$-norm of a dynamical system, we embark on posing an OPF that maximizes power system stability via $\mathcal{H}_2$-norm minimization.

8.5 Stability-Cognizant OPF Formulation

Given load demands, this section develops an OPF to find an operating point corresponding to minimum $\|H\|_{\mathcal{H}_2}^2$. It is worth noting that an admissible OPF formulation should enforce the traditional engineering limits on bus voltages and line flows of the full power system graph $\mathcal{G}$, while limiting the $\|H\|_{\mathcal{H}_2}^2$ computation on the Kron-reduced graph.

From the algebraic model of Section 8.4.1, the bus power injections are given by (8.3). The squared voltage magnitudes can be expressed as

$$|v_n|^2 = \mathbf{v}^H \mathbf{M}_{v_n} \mathbf{v} \tag{8.19}$$

[1] Distinct from $\delta_n(t)$ that denotes angle of machine n, the notation $\delta(t)$ denotes an impulse; the notation e_n is for n-the entry of internal voltage vector $\mathbf{e}$, while $\mathbf{e}_n$ denotes n-th canonical vector.

where $\mathbf{M}_{v_n} := \mathbf{e}_n\mathbf{e}_n^\top$. Squared line current magnitudes can similarly be expressed as quadratic functions of $\mathbf{v}$. If y_{mn} is the series impedance of line $(m,n) \in \mathcal{L}$, the current flowing on this line, denoted as $\tilde{i}_{mn}$ is $\tilde{i}_{mn} = (v_m - v_n)y_{mn}$, and thus,

$$|\tilde{i}_{mn}|^2 = \mathbf{v}^H\mathbf{M}_{i_{mn}}\mathbf{v} \tag{8.20}$$

where $\mathbf{M}_{i_{mn}} := |y_{mn}|^2(\mathbf{e}_m - \mathbf{e}_n)(\mathbf{e}_m - \mathbf{e}_n)^\top$.

The active power injected into bus n can be decomposed into a dispatchable generation p_n^g and fixed demand p_n^d as $p_n = p_n^g - p_n^d$. Flexible demands at bus n, if any, can also be incorporated in p_n^g. The reactive power injected into bus n is decomposed similarly as $q_n = q_n^g - q_n^d$. Recall that $p_n^g = p_n^d = q_n^g = q_n^d = 0$ for all buses $n \in \bar{\mathcal{S}}$. Given the demands at all buses $\{p_n^d, q_n^d\}_{n\in\mathcal{S}}$, the OPF problem aims at optimally dispatching generators $\{p_n^g, q_n^g\}_{n\in\mathcal{S}}$ while meeting resource and network limits. The OPF can be formulated as

$$\begin{aligned}
\min \quad & \|H\|_{\mathcal{H}_2}^2 && && \text{(S-OPF)}\\
\text{over} \quad & \mathbf{v}, \mathbf{e}, \mathbf{L}_{\boldsymbol{\delta}^0}, \{p_n^g, q_n^g\}_{n\in\mathcal{S}} && && \\
\text{s.to} \quad & (8.2),\ (8.10),\ (8.15),\ \text{and} && && \\
& \mathbf{v}^H\mathbf{M}_{p_n}\mathbf{v} = p_n^g - p_n^d, && \forall\ n \in \mathcal{S} && \text{(8.21a)}\\
& \mathbf{v}^H\mathbf{M}_{q_n}\mathbf{v} = q_n^g - q_n^d, && \forall\ n \in \mathcal{S} && \text{(8.21b)}\\
& \underline{p}_n^g \leq p_n^g \leq \bar{p}_n^g, && \forall\ n \in \mathcal{S} && \text{(8.21c)}\\
& \underline{q}_n^g \leq q_n^g \leq \bar{q}_n^g, && \forall\ n \in \mathcal{S} && \text{(8.21d)}\\
& \underline{v}_n \leq \mathbf{v}^\top\mathbf{M}_{v_n}\mathbf{v} \leq \bar{v}_n, && \forall\ n \in \mathcal{N} && \text{(8.21e)}\\
& \mathbf{v}^H\mathbf{M}_{i_{mn}}\mathbf{v} \leq \bar{i}_{mn}, && \forall\ (m,n) \in \mathcal{L} && \text{(8.21f)}
\end{aligned}$$

Constraints (8.21a)–(8.21b) enforce the power flow equations at buses with non-zero injection. Constraints (8.21c)–(8.21d) impose generation limits. Constraints (8.21e) confine squared voltage magnitudes within given ranges and constraint (8.21f) limits squared current magnitudes according to line ratings.

Since the non-convexity of (S-OPF) makes it computationally difficult to solve, we seek tractable convex relaxations next. To tackle non-convexity from quadratic constraints such as (8.21a)-(8.21b) and (8.21e)-(8.21f), SDP relaxations have yielded promising results [8], [80]. However, non-convex constraints (8.10) and (8.15) are novel to the proposed formulation, and thus do not have tractable prior reformulations.

Interestingly, although (8.10) provides a complicated nonlinear dependence of matrix $\mathbf{L}_{\boldsymbol{\delta}^0}$ on the magnitude and angles of $\mathbf{e}$, the constraint translates to a convenient reformulation. Specifically, let us introduce the *lifted* matrix variable $\mathbf{E} := \mathbf{e}\mathbf{e}^H$. Thanks to $\mathbf{E}$, constraint (8.10) is equivalent to

$$[\mathbf{L}_{\boldsymbol{\delta}^0}]_{n,m} = \begin{cases} \sum_{k\neq n} \frac{\mathrm{Re}\{E_{nk}\}}{\gamma_{nk}} & , \ m = n \\ \\ -\frac{\mathrm{Re}\{E_{nm}\}}{\gamma_{nm}} & , \ m \neq n. \end{cases} \tag{8.22}$$

Constraint (8.22) relates $\mathbf{L}_{\boldsymbol{\delta}^0}$ to the lifted variable $\mathbf{E}$ via linear equality constraints.

To deal with the $\mathcal{H}_2$-norm of the dynamical system appearing in the objective of (S-OPF), heed that the second summand in (8.15) and the scaling factor $1/2d$ are constants, and hence can be ignored when optimizing (S-OPF). We indeed seek to minimize $\mathrm{Tr}(\mathbf{C}_{11}^\top \mathbf{C}_{11} \mathbf{L}_{\boldsymbol{\delta}^0}^\dagger)$. The ensuing result further simplifies the objective of (S-OPF).

Lemma 8.2. *For $\mathbf{W} := \mathbf{C}_{11}^\top \mathbf{C}_{11}$, let $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0}$ be the $S \times S$ matrices obtained after removing* *the first row and column from* $\mathbf{W}$ *and* $\mathbf{L}_{\boldsymbol{\delta}^0}$*, respectively. If Assumption 8 holds, then*

$$\mathrm{Tr}(\mathbf{W}\mathbf{L}_{\boldsymbol{\delta}^0}^{\dagger}) = \mathrm{Tr}(\tilde{\mathbf{W}}\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0}^{-1}).$$

Using Lemma 8.2 and the reformulation in (8.22), the SDP-relaxation of (S-OPF) can be posed as

$$\begin{aligned}
\min \quad & \mathrm{Tr}(\mathbf{X}) && && \text{(SR-OPF)}\\
\text{s.to} \quad & (8.21c) - (8.21d),\ (8.22),\ \text{and} && && \\
& \mathrm{Tr}(\mathbf{M}_{p_n}\mathbf{V}) = p_n^g - p_n^d, && \forall\, n \in \mathcal{S} && (8.23a)\\
& \mathrm{Tr}(\mathbf{M}_{q_n}\mathbf{V}) = q_n^g - q_n^d, && \forall\, n \in \mathcal{S} && (8.23b)\\
& \underline{v}_n \le \mathrm{Tr}(\mathbf{M}_{v_n}\mathbf{V}) \le \bar{v}_n, && \forall\, n \in \mathcal{N} && (8.23c)\\
& \mathrm{Tr}(\mathbf{M}_{i_{mn}}\mathbf{V}) \le \bar{i}_{mn}, && \forall\, (m,n) \in \mathcal{L} && (8.23d)\\
& \mathbf{V}_{\mathcal{S},\mathcal{S}} = \mathbf{\Gamma}\mathbf{Y}_{\mathcal{S}}\mathbf{E}\mathbf{Y}_{\mathcal{S}}^H\mathbf{\Gamma}^H, && && (8.23e)\\
& \begin{bmatrix} \mathbf{X} & \tilde{\mathbf{W}}^{1/2} \\ \tilde{\mathbf{W}}^{1/2} & \tilde{\mathbf{L}}_{\boldsymbol{\delta}^0} \end{bmatrix} \succeq 0, && && (8.23f)\\
& \mathbf{V} \succeq 0 && && (8.23g)
\end{aligned}$$

where $\mathbf{V}_{\mathcal{S},\mathcal{S}}$ is the submatrix of $\mathbf{V}$ obtained from the first $S+1$ rows and columns; and $\tilde{\mathbf{W}}^{1/2}$ is the square root of matrix $\mathbf{W}$. The linear matrix inequality (LMI) in (8.23f) is equivalent to $\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0} \succ 0$ and $\mathbf{X} \succeq \tilde{\mathbf{W}}^{1/2}\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0}^{-1}\tilde{\mathbf{W}}^{1/2}$. The former holds by the definition of matrix $\mathbf{L}_{\boldsymbol{\delta}^0}$ via (8.22). For the latter, it is not hard to show that it is satisfied with equality, so that $\mathbf{X} = \tilde{\mathbf{W}}^{1/2}\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0}^{-1}\tilde{\mathbf{W}}^{1/2}$ at optimality. Then from Lemma 8.2, minimizing $\mathrm{Tr}(\mathbf{X})$ in (SR-OPF) is equivalent to minimizing $\mathrm{Tr}(\mathbf{W}\mathbf{L}_{\boldsymbol{\delta}^0}^{\dagger})$. The relaxed problem (SR-OPF) is termed exact if the optimal $\mathbf{V}$ is rank-1.

Even though dynamic stability is a desirable attribute, a typical OPF targets at minimizing the cost of generation. Assuming (c_n^p, c_n^q) are the coefficients for generation cost for bus n, a typical OPF reads as

$$\begin{aligned} \min \quad & \sum_{n \in \mathcal{S}} (c_n^p p_n^g + c_n^q q_n^g) \\ \text{s.to} \quad & (8.21c) - (8.21d), (8.23a) - (8.23d), \text{and } (8.23g). \end{aligned} \tag{OPF-C}$$

With increasing stability considerations, an OPF balancing the trade-off between the conventional OPF of (OPF-C) and the stability-enhancing OPF of (SR-OPF) could be pursued. Such a problem may be posed as the bi-objective OPF

$$\begin{aligned} \min \quad & \lambda \operatorname{Tr}(\mathbf{X}) + (1-\lambda) \left(\sum_{n \in \mathcal{S}} (c_n^p p_n^g + c_n^q q_n^g) \right) \\ \text{s.to} \quad & (6.6e) - (6.6f), (8.22), \text{and } (8.23a) - (8.23g) \end{aligned} \tag{λ-OPF}$$

for some parameter $\lambda \in (0, 1)$. Alternatively, one may first solve (OPF-C) and obtain a minimum generation cost $f^\star$. Subsequently, a more stable generation dispatch can be found by solving (SR-OPF) after appending the additional constraint of $\sum_{n \in \mathcal{S}} (c_n^p p_n^g + c_n^q q_n^g) \leq (1+\epsilon) f^\star$ for a desired margin ϵ.

8.6 Numerical Tests

The developed OPF formulation with dynamic stability consideration was numerically evaluated on the IEEE 39-bus system using the MATLAB-based optimization toolbox YALMIP alongside the SDP solver Mosek [78], [90]. There were three main objectives: *First*, to study the exactness of the SDP relaxation in (SR-OPF); *Second*, to numerically evaluate

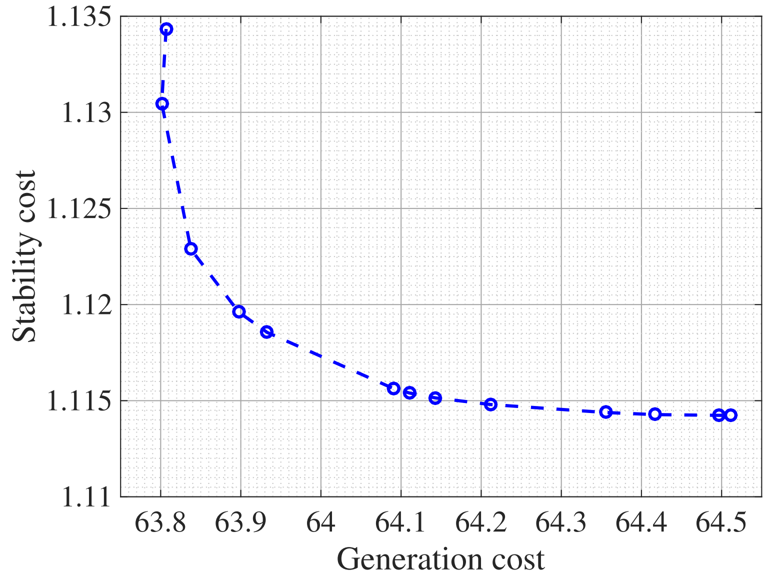

Figure 8.2: Generation cost and stability trade-off for bi-objective OPF.

the trade-off between generation cost and stability; and *Third*, to quantify the extent to which dynamic stability could be affected by generation dispatch. The network parameters, generator limits, and nominal loads for the IEEE 39-bus system were fetched from the MATPOWER `casefile` [153]. The 39-bus system features $|\bar{\mathcal{S}}| = 10$ zero-injection buses, and 10 generators. Treating the loads as non-flexible, the corresponding parameters $\underline{p}_n^g, \bar{p}_n^g, \underline{q}_n^g, \bar{q}_n^g, c_n^p$ and c_n^q were set to zero. Since the benchmark system has identical cost curves for all generators, a uniform active power cost $c_n^p = 1$ was used for all generators. The stability index for network coherence given by (8.16) was chosen for the tests.

The SDP relaxation of (OPF-C) is generally known to be exact, particularly with a penalty on reactive power generation [80]. Thus a uniform reactive power cost $c_n^q = 0.1$ was used for all generators. To numerically determine if an obtained matrix $\mathbf{V}$ is rank-1, let us define the indices

$$\eta_1 = \frac{\lambda_1}{\lambda_0} \text{ and } \eta_2 = \frac{\sum_{n=1}^{N} \lambda_n}{\lambda_0},$$

where $0 \leq \lambda_N \leq \ldots \lambda_1 \leq \lambda_0$ are the eigenvalues of $\mathbf{V}$. While theoretically, a rank-1 $\mathbf{V}$ requires $\eta_1 = \eta_2 = 0$, numerically we seek small values for these indices. Solving (OPF-C) with the aforementioned cost coefficients and nominal loads on the 39-bus system yields $\eta_1 = 5 \cdot 10^{-6}$ and $\eta_2 = 2 \cdot 10^{-5}$; while for (SR-OPF) we obtained $\eta_1 = 6 \cdot 10^{-4}$ and $\eta_2 = 7 \cdot 10^{-4}$. The small values of η_1 and η_2 indicate that similar to (OPF-C), a minimizer of (SR-OPF) features nearly rank-1 $\mathbf{V}$, rendering the relaxation exact. For the remaining tests to be presented next, indices η_1 and η_2 of the same order were obtained.

Proceeding to the second objective of assessing the generation cost versus stability trade-off, a Pareto optimality analysis was carried out by varying $\lambda \in [0,1]$ and solving (λ-OPF); see Figure 8.2. The stability cost in Figure 8.2 is given by $\mathrm{Tr}(\tilde{\mathbf{W}}\tilde{\mathbf{L}}_{\boldsymbol{\delta}^0}^{-1})$ while the generation cost includes the total (re)active power cost. The Pareto analysis at the nominal load level shows that an improvement of 1.77% in $\|H\|_{\mathcal{H}_2}^2$ can be obtained by increasing the generation cost by 1.11%. Decomposing the generation cost interestingly reveals that the increase in active power generation cost was just 0.49%. Moreover, not seeking the most stable operating point, a significant improvement in stability can be attained with a marginal increase in active power cost. For instance, a reduction of 1.69% in $\|H\|_{\mathcal{H}_2}^2$ comes with an increase of 0.18% in active power cost. These observations motivate future exploration of the possibility of improving dynamic stability via reactive power dispatch, thus limiting the financial impact.

We next conducted a set of numerical tests with 50 random generation costs and load levels. The goal of these tests was to emphasize on the range of values for generation cost and stability metrics that can be obtained from (SR-OPF) and (OPF-C). The random active and reactive generation costs were drawn independently from a uniform distribution on $[1,2]$ and $[0.1,0.2]$, respectively. For generating 50 load instances, the nominal loads at all buses of the IEEE 39-bus system were scaled independently and uniformly at random within $[0.8,1.2]$. The values of the stability metric and generation cost obtained for these random instances are shown in Figure 8.3. The impact on the stability metric obtained by a change of OPF objective demonstrates the significant room for enhancing dynamic system stability with an informed expense on generation cost. Specifically, the tests on the random OPF instances demonstrate a potential reduction in $\|H\|^2_{\mathcal{H}_2}$ by up to 5.7% by transiting from (OPF-C) to (SR-OPF).

8.7 Conclusions

A detailed model capturing the dependence of power system small-signal dynamics on the operating point has been put forth. Thereafter, a novel OPF formulation for minimizing the $\mathcal{H}_2$-norm, capable of representing practical stability metrics, has been proposed. To garner global optimality and scalability, the proposed OPF relies on SDP relaxation, which can be conveniently integrated with classical OPF formulations that minimize generation costs. Thanks to such integration, an insightful investigation of the trade-off between the chosen stability metric and generation cost was carried out for the IEEE 39-bus benchmark system. The numerical tests interestingly reveal that despite relying on SDP relaxation, the proposed stability-improving OPF yields nearly rank-1 minimizers, rendering the relaxation exact. The numerical tests further exemplify that the developed OPF can be used to obtain an optimally stable operating point based on the acceptable deviation from least generation cost dispatch. This work sets the foundations for interesting research directions. Improving small-signal stability via reactive power dispatch is a lucrative next step, mainly due to its limited monetary requirements. Expanding the proposed framework to include diverse stability metrics and more detailed generator models would further help to expand its applicability.

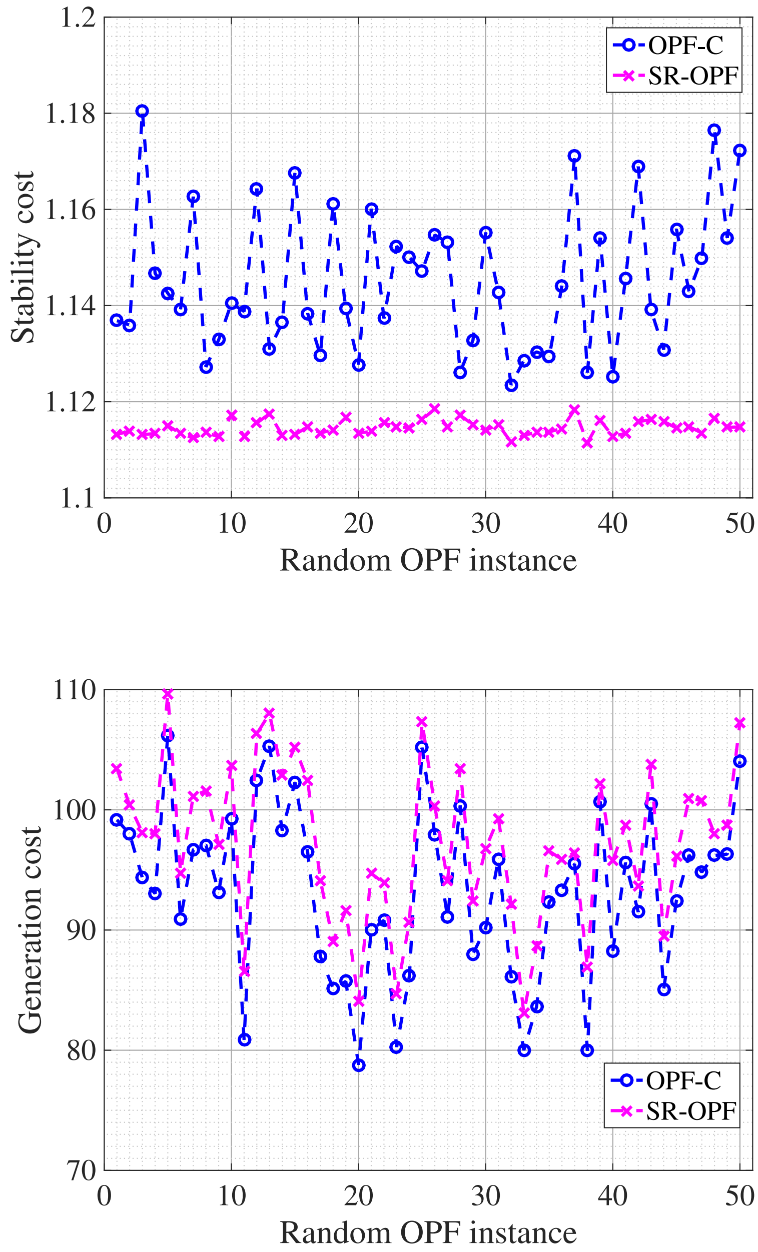

Figure 8.3: The stability metric and generation cost obtained by solving the stability-promoting (SR-OPF) and the minimum generation cost (OPF-C) for 50 random load scenarios.

Chapter 9

Summary: Contributions and Future Work

Aimed at improving the state of the art in energy network management, this dissertation presents the following exciting developments.

Network flow problems

- *C-1):* Open questions on characterizing uniqueness of natural gas and water flow problems have been answered. The elegance of our results is that they apply for fairly general problem settings that include all practical network topologies, prevalent component models, and multiple pressure references within a network.
- *C-2):* There are several gas and water flow solvers with different benefits and limitations. In this work, we first propose novel convex relaxation-based flow solvers with performance guarantees. The formulations are readily extendable to optimal network flow settings, and our approach for obtaining exactness serves as a base for such settings as well. Having developed these novel solvers, we further develop techniques that bring together the benefits of the existing and new solvers. Extensive numerical tests on benchmark gas and water networks establish the efficacy of the proposed hybrid solvers.

Learning to optimize (L2O)

- *C-3):* We revisit the traditional supervised learning approach for deep neural networks (DNNs) that is witnessing increasing application in learning to optimize power systems. Arguing that the traditional approach is wasteful in not using additional physics-based knowledge in the form of optimality conditions and optimal dual variables, we propose a novel approach termed as sensitivity-informed DNNs (SI-DNN). The novel feature here is that this DNN is trained to match not only the minimizers, but also their sensitivities with respect to the optimal power flow problem parameters. Detailed numerical tests reveal an accuracy improvement of 2-3 orders of magnitudes compared to the traditional DNN training approach.

- *C-4):* Training SI-DNNs requires augmentation of the labeled training set with the true sensitivity of the minimizers with respect to the input parameters. Using classical perturbation analysis, we developed a rigorous, yet computationally cheap procedure to compute these sensitivities, when they exist. The proposed SI-DNN is compatible with a broad range of optimal power flow (OPF) solvers, including a non-convex quadratically constrained quadratic program, its semidefinite program (SDP) relaxation, and MATPOWER.

Network control for improved resiliency

- *C-5):* Network dispatchers often ensure operation safety by enforcing constraints while solving centralized dispatch problems. Unanticipated disruptive events that occur between two dispatch instances may change the system model and hence could cause violation of safety-critical constraints. In this context, we developed a novel communication-stingy approach with theoretical guarantees on recovering feasibility, under practically reasonable assumptions.

- *C-6):* According to this protocol, agents act on local readings unless their control resources have been depleted, in which case they send a beacon for assistance to peer agents. Our ripple-type scheme triggers communication locally only for the agents with saturated resources and it is proved to converge.

Stability improving network dispatch

- *C-7):* Power system dynamic stability is often assessed by linearizing the underlying dynamical model at a fixed operating point. Further, given power demands, an operating point is chosen by dispatching generators at minimum cost. We investigate the opportunity of deviating from a strictly cost-oriented network dispatch towards an stability-improving approach. Systematically developing a mathematical model that capture the needed inter-dependencies between the two studies, we propose a tractable SDP-relaxation based formulation that can be used to incorporate dynamic stability in OPF formulations.

The mathematical tools acquired via the works included in this dissertation have also guided our additional works in power distribution systems [116], [19]. The diverse selection of problems addressed in this dissertation alongside the fundamental tools incorporated predominantly from optimization, learning, and control theory set a solid platform for us to conduct exciting future research. Some immediate research directions include:

- *Optimal network flow:* Unlike power systems, where tremendous development has been made towards efficient OPF solvers, the tools for solving optimal gas and water flow problems still lack sophistication. The advancements in flow solvers provided in this dissertation motivates us to tackle optimal gas/water flow problem instances in future.

- *Harness L2O:* The SI-DNN approach developed for solving parametric OPF problems indeed has a general applicability in learning minimizers of arbitrary nonlinear optimization problems. We therefore plan to fully harness its scope by venturing out to other practical applications. Furthermore the underlying mathematical analysis can be extended to include sensitivities of optimal dual variables as well. This further sets a promising ground for problems affected by optimal dual variables such as predicting active constraints or locational prices in energy markets.

- *Exploring inter-network couplings:* In recent years, the opportunities and research interest in network inter-dependencies have been on the rise. With the modeling insights and reformulation techniques developed in this thesis, we next plan to explore the areas on gas-electric-water nexus.

Bibliography

[1] The future of natural gas: MIT energy initiative. Technical report, Massachusetts Institute of Technology, 2011. URL http://energy.mit.edu/wp-content/uploads/2011/06/MITEI-The-Future-of-Natural-Gas.pdf.

[2] Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating through a cone program. (submitted), 2020. URL https://arxiv.org/abs/1904.09043.

[3] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical programming*, 95(1):3–51, January 2003.

[4] F. Alvarruiz, F. Martinez-Alzamora, and A. M. Vidal. Improving the efficiency of the loop method for the simulation of water distribution systems. *J. of Water Resources Planning and Management*, 141(10):1–10, October 2015.

[5] Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Intl. Conf. on Machine Learning*, page 136–145, Sydney, NSW, Australia, 2017.

[6] Denis Assmann, Frauke Liers, and Michael Stingl. Decomposable robust two-stage optimization: An application to gas network operations under uncertainty. *Networks*, 74(1):40–61, July 2019.

[7] Alper Atamtürk and Vishnu Narayanan. Conic mixed-integer rounding cuts. *Mathematical Programming*, 122(1):1–20, March 2010.

[8] X. Bai, H. Wei, K. Fujisawa, and Y. Yang. Semidefinite programming for optimal power flow problems. *Intl. Journal of Electric Power & Energy Systems*, 30(6):383–392, 2008.

[9] Bassam Bamieh, Mihailo R. Jovanovic, Partha Mitra, and Stacy Patterson. Coherence in large-scale networks: Dimension-dependent limitations of local feedback. *IEEE Trans. Automat. Contr.*, 57(9):2235–2249, September 2012.

[10] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht. Smart*: An open data set and tools for enabling research in sustainable homes. In *Proc. Workshop on Data Mining Applications in Sustainability*, Beijing, China, August 2012.

[11] Stefanos Baros, Andrey Bernstein, and Nikos D. Hatziargyriou. Distributed conditions for small-signal stability of power grids and local control design. *IEEE Trans. Power Syst.*, 36(3):2058–2067, May 2021.

[12] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.*, 18(1):5595–5637, January 2017.

[13] Mohammadhafez Bazrafshan, Nikolaos Gatsis, Marcio Giacomoni, and Ahmad Taha. A fixed-point iteration for steady-state analysis of water distribution networks. In *Proc. IEEE Global Conf. on Signal and Information Process.*, pages 880–884, Anaheim, CA, November 2018.

[14] Peter Benner, Sara Grundel, Christian Himpe, Christoph Huck, Tom Streubel, and Caren Tischendorf. Gas network benchmark models. In *Differential-Algebraic Equations Forum.* Springer, Heidelberg, 2018.

[15] R. Bent, S. Blumsack, P. Van Hentenryck, C. Borraz-Sanchez, and M. Shahriari. Joint

electricity and natural gas transmission planning with endogenous market feedbacks. *IEEE Trans. Power Syst.*, 33(6):6397–6409, November 2018.

[16] A. Bernstein and E. Dall'Anese. Real-time feedback-based optimization of distribution grids: A unified approach. *IEEE Trans. Control of Network Systems*, 6(3):1197–1209, September 2019.

[17] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

[18] S. Bhela, D. Deka, H. Nagarajan, and V. Kekatos. Designing power grid topologies for minimizing network disturbances: An exact MILP formulation. In *Proc. IEEE American Control Conf.*, pages 1949–1956, Philadelphia, PA, jul 2019.

[19] Shuchismita Biswas, Manish K. Singh, and Virgilio A. Centeno. Chance-constrained optimal distribution network partitioning to enhance power grid resilience. *IEEE Access*, 9:42169–42181, March 2021.

[20] N. Bof, R. Carli, A. Cenedese, and L. Schenato. Asynchronous distributed camera network patrolling under unreliable communication. *IEEE Trans. Automat. Contr.*, 62(11):5982–5989, November 2017.

[21] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri. On the need for communication for voltage regulation of power distribution grids. *IEEE Trans. Control of Network Systems*, 6(3):1111–1123, September 2019.

[22] J Frederic Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Science & Business Media, New York, NY, 2000.

[23] Conrado Borraz-Sánchez, Russell Bent, Scott Backhaus, Hassan Hijazi, and Pascal Van

Hentenryck. Convex relaxations for gas expansion planning. *INFORMS Journal on Computing*, 28(4):645–656, August 2016.

[24] F. Borrelli, A. Bemporad, and M. Morari. Geometric algorithm for multiparametric linear programming. *Journal of Optimization Theory and Applications*, 118(3):515–540, September 2003.

[25] S. Bose, D.F. Gayme, K.M. Chandy, and S.H. Low. Quadratically constrained quadratic programs on acyclic graphs with application to power flow. *IEEE Trans. Control of Network Systems*, 2(3):278–287, September 2015.

[26] E. Castillo, A. J. Conejo, C. Castillo, R. Minguez, and D. Ortigosa. Perturbation approach to sensitivity analysis in mathematical programming. *Journal of Optimization Theory and Applications*, 128(1):49–74, January 2006.

[27] G. Cavraro and R. Carli. Local and distributed voltage control algorithms in distribution networks. *IEEE Trans. Power Syst.*, 33(2):1420–1430, March 2018.

[28] G. Cavraro, Manish K. Singh, and A. Bernstein. Emergency voltage regulation in power systems via ripple-type control. In *Proc. Mediterranean Conf. on Control and Automation*, Bari, Italy, June 2021.

[29] C. Chang, M. Colombino, J. Corté, and E. Dall'Anese. Saddle-flow dynamics for distributed feedback-based optimization. *IEEE Contr. Syst. Lett.*, 3(4):948–953, October 2019.

[30] S. Chen, Z. Wei, G. Sun, D. Wang, and H. Zang. Steady state and transient simulation for electricity-gas integrated energy systems by using convex optimisation. *IET Generation, Transmission Distribution*, 12(9):2199–2206, May 2018.

[31] Tianlun Chen, Albert Lam, Yue Song, and David J Hill. Robust transient stability constrained optimal power flow with power flow routers considering renewable uncertainties. (preprint), 2020. URL https://arxiv.org/abs/2006.00506.

[32] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn. Model-free primal-dual methods for network optimization with application to real-time optimal power flow. In *Proc. IEEE American Control Conf.*, pages 3140–3147, Denver, CO, July 2020.

[33] Yize Chen and Baosen Zhang. Learning to solve network flow problems via neural decoding. preprint, 2020. URL https://arxiv.org/abs/2002.04091.

[34] Yize Chen, Yuanyuan Shi, and Baosen Zhang. Input convex neural networks for optimal voltage regulation. (submitted), 2020. URL https://arxiv.org/abs/2002.08684.

[35] Joe H Chow and Juan J Sanchez-Gasca. *Power system modeling, computation, and control*. Wiley-IEEE Press, 2020.

[36] C.Y. Chung, Lei Wang, F. Howell, and P. Kundur. Generation rescheduling methods to improve power transfer capability constrained by small-signal stability. *IEEE Trans. Power Syst.*, 19(1):524–530, March 2004.

[37] D. Cohen, U. Shamir, and G. Sinai. Optimal operation of multi-quality water supply systems-II: The Q-H model. *Engineering Optimization*, 32(6):687–719, October 2000.

[38] M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc. Solving the pipe network analysis problem using optimization techniques. *Management Science*, 24(7): 747–760, March 1978.

[39] A. J. Conejo, E. Castillo, R. Minguez, and R. Garcia-Bertrand. *Decomposition Techniques in Mathematical Programming*. Springer, 2006.

[40] Claudia D'Ambrosio, Andrea Lodi, Sven Wiese, and Cristiana Bragalli. Mathematical programming techniques in water network optimization. *European J. of Operational Research*, 243(3):774 – 788, June 2015.

[41] Daniel De Wolf and Yves Smeers. The gas transmission problem solved by an extension of the simplex algorithm. *Management Science*, 46(11):1454–1465, November 2000.

[42] D. Deka and S. Misra. Learning for DC-OPF: Classifying active sets using neural nets. In *IEEE PowerTech*, pages 1–6, Milan, Italy, June 2019.

[43] T. Ding, Y. Xu, Y. Yang, Z. Li, X. Zhang, and F. Blaabjerg. A tight linear program for feasibility check and solutions to natural gas flow equations. *IEEE Trans. Power Syst.*, 34(3):2441–2444, May 2019.

[44] F. Dorfler and F. Bullo. Kron reduction of graphs with applications to electrical networks. *IEEE Trans. Circuits Syst. I*, 60(1):150–163, January 2013.

[45] K. Dvijotham, M. Vuffray, S. Misra, and M. Chertkov. Natural gas flow solutions with guarantees: A monotone operator theory approach. preprint, 2015. URL http://arxiv.org/abs/1506.06075.

[46] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro. Learning optimal resource allocations in wireless systems. *IEEE Trans. Signal Processing*, 67(10):2775–2790, May 2019.

[47] Sylvan Elhay, Angus R. Simpson, Jochen Deuerlein, Bradley Alexander, and Wil H. A. Schilders. Reformulated co-tree flows method competitive with the global gradient algorithm for solving water distribution system equations. *Journal of Water Resources Planning and Management*, 140(12):1–10, December 2014.

[48] Sylvan Elhay, Jochen Deuerlein, Olivier Piller, and Angus R. Simpson. Graph partitioning in the analysis of pressure dependent water distribution systems. *J. of Water Resources Planning and Management*, 144(4):04018011, April 2018.

[49] Carlos Estrada, Cesar Gonzalez, Ricardo Aliod, and Jara Pano. Improved pressurized pipe network hydraulic solver for applications in irrigation systems. *J. of Irrigation and Drainage Engineering*, 135(4):421–430, August 2009.

[50] Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, December 1976.

[51] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. Predicting AC optimal power flows: Combining deep learning and Lagrangian dual methods. In *AAAI Conf. on Artificial Intelligence*, New York, NY, February 2020.

[52] D. Fooladivanda and J. A. Taylor. Optimal pump scheduling and water flow in water distribution networks. In *Proc. IEEE Conf. on Decision and Control*, pages 5265–5271, Osaka, Japan, December 2015.

[53] D. Fooladivanda and J. A. Taylor. Energy-optimal pump scheduling and water flow. *IEEE Trans. Control of Network Systems*, 5(3):1016–1026, September 2018.

[54] Armin Fügenschuh and Jesco Humpola. A unified view on relaxations for a nonlinear network flow problem. Technical report, Universität der Bundeswehr Hamburg, Germany, July 2013.

[55] D. Gan, R.J. Thomas, and R.D. Zimmerman. Stability-constrained optimal power flow. *IEEE Trans. Power Syst.*, 15(2):535–540, May 2000.

[56] B. Gentile, J.W. Simpson-Porco, F. Dorfler, S. Zampieri, and F. Bullo. On reactive

power flow and voltage stability in microgrids. In *Proc. IEEE American Control Conf.*, pages 759–764, Portland, OR, June 2014.

[57] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, New York, NY, 2001.

[58] A. Gómez-Expósito, A. J. Conejo, and C. Canizares, editors. *Electric Energy Systems, Analysis and Operation*. CRC Press, Boca Raton, FL, 2009.

[59] Neel Guha, Zhecheng Wang, Matt Wytock, and Arun Majumdar. Machine learning for AC optimal power flow. Climate Change Workshop at ICML 2019, 2019. URL https://arxiv.org/abs/1910.08842.

[60] S. Gupta, V. Kekatos, and M. Jin. Communication-limited inverter control using deep neural networks. In *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, pages 1–6, Tempe, AZ, November 2020.

[61] Tong Han and David J. Hill. $\mathcal{H}_2$-norm transmission switching to improve synchronism of low-inertia power grids. *IFAC-PapersOnLine*, 53(2):13299–13304, 2020. 21st IFAC World Congress.

[62] N. Hatziargyriou, J. V. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. J. Sanchez-Gasca, A. M. Stankovic, T. Van Cutsem, V. Vittal, and C. Vournas. Definition and classification of power system stability revisited extended. *IEEE Transactions on Power Systems*, pages 1–1, 2020. early access.

[63] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proc. IEEE Conf. on Decision and Control*, pages 3270–3285, Maui, HI, December 2012.

[64] IBM Corp. IBM ILOG CPLEX Optimization Studio CPLEX User's Manual, 2017. URL http://www.ibm.com.

[65] Pecan Street Inc. Dataport, 2018. URL https://dataport.cloud/.

[66] Takayuki Ishizaki, Aranya Chakrabortty, and Jun-Ichi Imura. Graph-theoretic analysis of power systems. *Proc. IEEE*, 106(5):931–952, May 2018.

[67] Y. Ji, R. J. Thomas, and L. Tong. Probabilistic forecasting of real-time LMP and network congestion. *IEEE Trans. Power Syst.*, 32(2):831–841, March 2017.

[68] Liu Jun and Yu Guoping. Iterative methodology of pressure-dependent demand based on EPANET for pressure-deficient water distribution analysis. *J. of Water Resources Planning and Management*, 139(1):34–44, January 2013.

[69] V. Kekatos, G. Wang, H. Zhu, and G. B. Giannakis. PSSE redux: Convex relaxation, decentralized, robust, and dynamic approaches. In Mo El-Hawary, editor, *Advances in Power System State Estimation*. Wiley, 2021.

[70] Bernhard Korte and Jens Vygen. *Combinatorial optimization*, volume 2. Springer, Heidelberg, 2012.

[71] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal. Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions. *IEEE Trans. Power Syst.*, 19(3):1387–1401, August 2004.

[72] Prabha Kundur. *Power system stability and control*. McGraw-Hill, New York, NY, 1994.

[73] Henning Lange, Bingqing Chen, Mario Berges, and Soummya Kar. Learning to solve AC optimal power flow by differentiating through holomorphic embeddings. (submitted), 2020. URL https://arxiv.org/abs/2012.096224.

[74] F. Li and R. Bo. Small test systems for power system economic studies. In *Proc. IEEE PES General Meeting*, pages 1–4, Providence, RI, July 2010.

[75] Peijie Li, Hua Wei, Bin Li, and Yude Yang. Eigenvalue-optimisation-based optimal power flow with small-signal stability constraints. *IET Gener. Transm. Dis.*, 7(5): 440–450, January 2013.

[76] Peijie Li, Junjian Qi, Jianhui Wang, Hua Wei, Xiaoqing Bai, and Feng Qiu. An sqp method combined with gradient sampling for small-signal stability constrained opf. *IEEE Trans. Power Syst.*, 32(3):2372–2381, May 2017.

[77] Qing Li, Seungwon An, and Thomas W Gedra. Solving natural gas loadflow problems using electric loadflow techniques. In *Proc. North American Power Symposium*, Rolla, MO, October 2003.

[78] J. Lofberg. A toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conf.*, 2004. URL http://users.isy.liu.se/johanl/yalmip/.

[79] Steven Low. Convex relaxation of optimal power flow – Part II: Exactness. *IEEE Trans. Control of Network Systems*, 1(2):177–189, June 2014.

[80] R. Madani, S. Sojoudi, and J. Lavaei. Convex relaxation for optimal power flow problem: Mesh networks. *IEEE Trans. Power Syst.*, 30(1):199–211, January 2015.

[81] Sindri Magnusson, Carlo Fischione, and Na Li. Optimal voltage control using event triggered communication. In *Proc. 10th ACM Int. Conf. Future Energy Syst. (e-Energy)*, pages 343–354, Phoenix, AZ, June 2019.

[82] Helena Mala-Jetmarova, Nargiz Sultanova, and Dragan Savic. Lost in optimisation of water distribution systems? A literature review of system operation. *Environmental Modelling & Software*, 93:209–254, 2017.

[83] A. Martinez-Mares and C. R. Fuerte-Esquivel. A unified gas and power flow analysis in natural gas and electricity coupled networks. *IEEE Trans. Power Syst.*, 27(4): 2156–2166, November 2012.

[84] A. Martinez-Mares and C. R. Fuerte-Esquivel. Integrated energy flow analysis in natural gas and electricity coupled systems. In *Proc. North American Power Symposium*, Boston, MA, August 2011.

[85] J. J. Maugis. Design and operation of gas transmission networks. *Etude de reseaux de transport et de distribution de fluide*, 11(2):243–248, 1977.

[86] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. *Mathematical Programming*, 10 (1):147–175, December 1976.

[87] S. Misra, M. W. Fisher, S. Backhaus, R. Bent, M. Chertkov, and F. Pan. Optimal compression in natural gas networks: A geometric programming approach. *IEEE Trans. Control of Network Systems*, 2(1):47–56, March 2015.

[88] Sidhant Misra, Marc Vuffray, and Michael Chertkov. Maximum throughput problem in dissipative flow networks with application to natural gas systems. preprint, 2015. URL `arXivpreprintarXiv:1504.02370`.

[89] George S Misyris, Andreas Venzke, and Spyros Chatzivasileiadis. Physics-informed neural networks for power systems. In *Proc. IEEE PES General Meeting*, pages 1–5, Montreal, Canada, August 2020.

[90] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. version 9.0., 2019. URL http://docs.mosek.com/9.0/toolbox/index.html.

[91] Cameron Nowzari and Jorge Cortés. Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica*, 68:237–244, June 2016.

[92] A. Ojha, V. Kekatos, and R. Baldick. Solving the natural gas flow problem using semidefinite program relaxation. In *Proc. IEEE PES General Meeting*, Chicago, IL, July 2017.

[93] A. Osiadacz. *Simulation and analysis of gas networks*. Gulf Publishing, 1987.

[94] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Optimal power flow using graph neural networks. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Process.*, pages 5930–5934, Barcelona, Spain, May 2020.

[95] F. Paganini and E. Mallada. Global analysis of synchronization performance for power systems: bridging the theory-practice gap. *IEEE Trans. Automat. Contr.*, 65(7):3007–3022, Jul 2020.

[96] Xiang Pan, Tianyu Zhao, and Minghua Chen. DeepOPF: Deep neural network for DC optimal power flow. In *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, pages 1–6, Beijing, China, October 2019.

[97] Parikshit Pareek and Hung D Nguyen. Small-signal stability constrained optimal power flow: A convexification approach. (preprint), 2020. URL https://arxiv.org/abs/1911.12001.

[98] Bala Kameshwar Poolla, Saverio Bolognani, and Florian Dörfler. Optimal placement of virtual inertia in power grids. *IEEE Trans. Automat. Contr.*, 62(12):6209–6220, December 2017.

[99] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[100] Roger Z Rios-Mercado and Conrado Borraz-Sanchez. Optimization problems in natural gas transportation systems: A state-of-the-art review. *Applied Energy*, 147:536–555, March 2015.

[101] Roger Z. Ríos-Mercado, Suming Wu, L. Ridgway Scott, and E. Andrew Boyd. A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research*, 117(1):217–234, November 2002.

[102] L. A. Rossman. EPANET 2 user's manual, 2000.

[103] Lewis A. Rossman, Robert M. Clark, and Walter M. Grayman. Modeling chlorine residuals in drinking-water distribution systems. *J. of Environmental Engineering*, 120(4):803–820, July 1994.

[104] C. B. Sanchez, R. Bent, S. Backhaus, S. Blumsack, H. Hijazi, and P. v. Hentenryck. Convex optimization for joint expansion planning of natural gas and power systems. In *Hawaii Intl. Conf. on System Sciences*, pages 2536–2545, Koloa, HI, January 2016.

[105] Martin Schmidt, Denis Assmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucherif, Marc Pfetsch, Lars Schewe, Robert Schwarz, and Mathias Sirvent. Gaslib - A library of gas network instances. *Data*, 2(4):40, 2017.

[106] Anna Schwele, Christos Ordoudis, Jalal Kazempour, and Pierre Pinson. Coordination of power and natural gas systems: Convexification approaches for linepack modeling. In *Proc. IEEE PES PowerTech Conf.*, Milan, Italy, July 2019.

[107] Hanif D. Sherali, Shivaram Subramanian, and G.V. Loganathan. Effective relaxations and partitioning schemes for solving water distribution network design problems to global optimality. *J. of Global Optimization*, 19(1):1–26, January 2001.

[108] J. W. Simpson-Porco, F. Dorfler, and F. Bullo. Voltage stabilization in microgrids via quadratic droop control. *IEEE Trans. Automat. Contr.*, 62(3):1239–1253, March 2017.

[109] M. Singh, G. Cavraro, A. Bernstein, and V. Kekatos. Ripple-type control for enhancing resilience of networked physical systems. In *Proc. IEEE American Control Conf.*, New Orleans, LA, May 2021.

[110] M. K. Singh and V. Kekatos. On the flow problem in water distribution networks: Uniqueness and solvers. *IEEE Trans. Control of Network Systems*, 8(1):462–474, March 2021.

[111] M. K. Singh, S. Gupta, V. Kekatos, G. Cavraro, and A. Bernstein. Learning to optimize power distribution grids using sensitivity-informed deep neural networks. In *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, pages 1–6, Tempe, AZ, November 2020.

[112] M. K. Singh, V. Kekatos, and G. B. Giannakis. Learning to solve the ac-opf using sensitivity-informed deep neural networks. (submitted), 2021. URL https://arxiv.org/abs/2103.14779.

[113] Manish K. Singh and Vassilis Kekatos. Natural gas flow equations: Uniqueness and an MI-SOCP solver. In *Proc. IEEE American Control Conf.*, pages 2114–2120, Philadelphia, PA, July 2019.

[114] Manish K. Singh and Vassilis Kekatos. Natural gas flow solvers using convex relaxation. *IEEE Trans. Control of Network Systems*, 7(3):1283–1295, September 2020.

[115] Manish K. Singh and Vassilis Kekatos. Optimal scheduling of water distribution systems. *IEEE Trans. Control of Network Systems*, 7(2):711–723, June 2020.

[116] Manish K. Singh, V. Kekatos, and C.-C. Liu. Optimal distribution system restoration with microgrids and distributed generators. In *Proc. IEEE PES General Meeting*, Atlanta, GA, August 2019.

[117] Manish K. Singh, Sarthak Gupta, and Vassilis Kekatos. Machine learning for optimal inverter operation in distribution grids. In *Proc. Annual Conf. Information Sc. and Sys.*, pages 1–1, March 2021.

[118] Yue Song, David J. Hill, and Tao Liu. Network-based analysis of small-disturbance angle stability of power systems. *IEEE Trans. Control of Network Systems*, 5(3): 901–912, September 2018.

[119] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Trans. Signal Processing*, 66(20):5438–5453, October 2018.

[120] Ahmad F. Taha, Mohammadhafez Bazrafshan, Sebastian Adi Nugroho, Nikolaos Gatsis, and Junjian Qi. Robust control for renewable-integrated power networks considering input bound constraints and worst case uncertainty measure. *IEEE Trans. Control of Network Systems*, 6(3):1210–1222, September 2019.

[121] S. Taheri, M. Jalali, V. Kekatos, and L. Tong. Fast probabilistic hosting capacity analysis for active distribution systems. *IEEE Trans. Smart Grid*, 2020. (early access).

[122] S. Taheri, V. Kekatos, and H. Veeramachaneni. Strategic investment in energy markets: A multiparametric programming approach. *IEEE Trans. Smart Grid*, April 2020. URL https://arxiv.org/abs/2004.06483. (submitted).

[123] Emma Tegling, Bassam Bamieh, and Dennice F Gayme. The price of synchrony: Evaluating the resistive losses in synchronizing power networks. *IEEE Trans. Control of Network Systems*, 2(3):254–266, September 2015.

[124] A.R.D. Thorley and C.H. Tiley. Unsteady and transient flow of compressible fluids in pipelines – a review of theoretical and some experimental studies. *Intl. J. of Heat & Fluid Flow*, 8(1):3–15, March 1987.

[125] E. Todini and Lewis A. Rossman. Unified framework for deriving simultaneous equation algorithms for water distribution networks. *J. of Hydraulic Engineering*, 139(5): 511–526, May 2013.

[126] Ezio Todini. On the convergence properties of the different pipe network algorithms. In *Water Distribution Systems Analysis Symp.*, pages 1–16, Reston, VA, March 2006.

[127] B. Ulanicki, J. Kahler, and B. Coulbeck. Modeling the efficiency and power characteristics of a pump group. *J. of Water Resources Planning and Management*, 134(1): 88–93, 2008.

[128] Jakobus E. Van-Zyl, Dragan A. Savic, and Godfrey A. Walters. Operational optimization of water distribution systems using a hybrid genetic algorithm. *J. of Water Resources Planning and Management*, 130(2):160–170, 2004.

[129] Derek Verleye and El-Houssaine Aghezzaf. Optimising production and distribution operations in large water supply networks: A piecewise linear optimisation approach. *Intl. J. of Production Research*, 51(23-24):7170–7189, 2013.

[130] Thanh Long Vu and Konstantin Turitsyn. Lyapunov functions family approach to transient stability assessment. *IEEE Trans. Power Syst.*, 31(2):1269–1277, March 2016.

[131] M. Vuffray, S. Misra, and M. Chertkov. Monotonicity of dissipative flow networks renders robust maximum profit problem tractable: General analysis and application to natural gas flows. In *Proc. IEEE Conf. on Decision and Control*, pages 4571–4578, Osaka, Japan, December 2015.

[132] Shen Wang, Ahmad F. Taha, Nikolaos Gatsis, and Marcio Giacomoni. Geometric programming-based control for nonlinear, DAE-constrained water distribution networks. In *Proc. IEEE American Control Conf.*, pages 1470–1475, Philadelphia, PA, July 2019.

[133] Shen Wang, Ahmad F. Taha, Lina Sela, Marcio H. Giacomoni, and Nikolaos Gatsis. A new derivative-free linear approximation for solving the network water flow problem with convergence guarantees. *Water Resources Research*, 56(3), January 2020.

[134] W. Wang, N. Yu, Y. Gao, and J. Shi. Safe off-policy deep reinforcement learning algorithm for Volt-VAR control in power distribution systems. *IEEE Trans. Smart Grid*, 11(4):3008–3018, July 2020.

[135] Fei Wu, H. Nagarajan, A. Zlotnik, R. Sioshansi, and A. M. Rudkevich. Adaptive convex relaxations for gas pipeline network optimization. In *Proc. IEEE American Control Conf.*, pages 4710–4716, Seattle, WA, May 2017.

[136] S. Wu, L Ridgway Scott, and E Andrew Boyd. Towards the simplification of natural gas transmission networks. In *Proc. NSF Design and Manufacturing Grantees Conference*, Long Beach, CA, January 1999.

[137] Alinson S. Xavier, Feng Qiu, and Shabbir Ahmed. Learning to solve large-scale security-constrained unit commitment problems. *INFORMS Journal on Computing*, pages 1–18, October 2020. (early access).

[138] Qiuling Yang, Alireza Sadeghi, Gang Wang, Georgios B Giannakis, and Jian Sun. Robust PSSE using graph neural networks for data-driven and topology-aware priors. (submitted), 2020. URL https://arxiv.org/abs/2003.01667.

[139] Mausam Yatin Nandwani, Abhishek Pathak and Parag Singla. A primal dual formulation for deep learning with constraints. In *Proc. of Adv. Neural Inf. Process. Syst.*, pages 12157–12168, Vancouver, Canada, December 2019.

[140] A. S. Zamzam, E. Dall'Anese, C. Zhao, J. A. Taylor, and N. Sidiropoulos. Optimal water-power flow problem: Formulation and distributed optimal solution. *IEEE Trans. Control of Network Systems*, 6(1):37–47, March 2019.

[141] Ahmed Zamzam and Kyri Baker. Learning optimal solutions for extremely fast AC optimal power flow. In *Proc. IEEE Intl. Conf. on Smart Grid Commun.*, pages 1–6, Tempe, AZ, November 2020.

[142] Rafael Zarate-Minano, Federico Milano, and Antonio J. Conejo. An opf methodology to ensure small-signal stability. *IEEE Trans. Power Syst.*, 26(3):1050–1061, August 2011.

[143] Hui Zhang, Xin Cheng, Tinglin Huang, Haibing Cong, and Jinlan Xu. Hydraulic analysis of water distribution systems based on fixed point iteration method. *Water Resources Management*, 31(5):1605–1618, March 2017.

[144] L. Zhang, G. Wang, and G. B. Giannakis. Real-time power system state estimation and forecasting via deep unrolled neural networks. *IEEE Trans. Signal Processing*, 67 (15):4069–4077, August 2019.

[145] Ling Zhang, Yize Chen, and Baosen Zhang. A convex neural network solver for DCOPF

with generalization guarantees. (submitted), 2020. URL https://arxiv.org/abs/2009.09109.

[146] Qianzhi Zhang, Kaveh Dehghanpour, Zhaoyu Wang, Feng Qiu, and Dongbo Zhao. Multi-agent safe policy learning for power management of networked microgrids. (submitted), 2020. URL https://arxiv.org/abs/1907.02091v3.

[147] Changhong Zhao, Ufuk Topcu, Na Li, and Steven Low. Design and stability of load-side primary frequency control in power systems. *IEEE Trans. Automat. Contr.*, 59 (5):1177–1189, May 2014.

[148] Yue Zhao and Baosen Zhang. Deep learning in power systems. In Ali Tajer, Samir M. Perlaza, and H. Vincent Poor, editors, *Advanced Data Analytics for Power Systems*. Cambridge University Press, Cambridge, UK, May 2021.

[149] M. Zholbaryssov and A. D. Dominguez-Garcia. Convex relaxations of the network flow problem under cycle constraints. *IEEE Trans. Control of Network Systems*, 7(1): 64–73, March 2020.

[150] Q. Zhou, L. Tesfatsion, and C.-C. Liu. Short-term congestion forecasting in wholesale power markets. *IEEE Trans. Power Syst.*, 26(4):2185–2196, November 2011.

[151] X. Zhou, M. Farivar, Z. Liu, L. Chen, and S. Low. Reverse and forward engineering of local voltage control in distribution networks. *IEEE Trans. Automat. Contr.*, 66(3): 1116 – 1128, March 2021.

[152] H. Zhu and H. J. Liu. Fast local voltage control under limited reactive power: Optimality and stability analysis. *IEEE Trans. Power Syst.*, 31(5):3794–3803, September 2016.

[153] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas. MATPOWER: steady-state operations, planning and analysis tools for power systems research and education. *IEEE Trans. Power Syst.*, 26(1):12–19, February 2011.

[154] A. Zlotnik, M. Chertkov, and K. Turitsyn. Assesing risk of gas-shortage in coupled gas-electricity infrastructures. In *Hawaii Intl. Conf. on System Sciences*, Koloa, HI, January 2016.

[155] A. Zlotnik, L. Roald, S. Backhaus, M. Chertkov, and G. Andersson. Coordinated scheduling for interdependent electric power and natural gas infrastructures. *IEEE Trans. Power Syst.*, 32(1):600–610, January 2017.