Optimal compression in natural gas networks: a geometric programming approach

Sidhant Misra, Michael W. Fisher, Scott Backhaus, Russell Bent, Michael Chertkov, Feng Pan

***Abstract*—Natural gas transmission pipelines are complex systems whose flow characteristics are governed by challenging nonlinear physical behavior. These pipelines extend over hundreds and even thousands of miles with gas injected into the system at a constant rate. A series of compressors are distributed along the pipeline to boost the gas pressure to maintain system performance and throughput. These compressors consume a portion of the gas, and one goal of the operator is to control the compressor operation to minimize this consumption while satisfying pressure constraints at the gas load points. The optimization of these operations is computationally challenging, and many pipelines rely on the intuition and prior experience of operators to make these decisions. Here, we present a new geometric programming approach for optimizing compressor operation in natural gas pipelines. Using models of real natural gas pipelines, we show that the geometric programming algorithm consistently outperforms approaches that mimic existing state of practice.**

***Index Terms*—Natural Gas Network, Optimal Compression, Geometric Programming, Dynamic Programming,**

I. Introduction: History & Motivation

In recent years, worldwide natural gas reserves have expanded at a rapid pace. The invention and application of hydraulic fracturing in the US has enabled the economic capture of many sources of unconventional natural gas [1] while improved exploration techniques and increased offshore activity has led to increased conventional reserves in several countries. The increased availability and lower cost of gas in these regions are making it more attractive economically. In the US, the economic advantage of gas is pushing out coal (and to a lesser extent fuel oil) as a primary source of energy. In addition, the lower CO2 emissions from gas mitigate much of the uncertainty related to the future economic cost of carbon emissions. These properties make gas a very attractive bridge fuel to a low carbon economy, and this shift is already occurring in several regions of the US electric sector [2]. The high cost and long economic lifetime of the electrical generation assets acts to lock in this shift to a large degree.

The cost of the fuel is not the only advantage of natural gas over coal and fuel oil. From the planning and construction point of view, the physical footprint and total emissions of gas turbines is smaller than coal or fuel oil-based generation easing the difficulty of siting and permitting. From an operational perspective, gas turbines can quickly change their generation output in response to changes intermittent renewable generation such as wind. This ability to move quickly is also manifest in the ability to quickly start up a gas turbine from a cold condition (especially single-cycle gas turbines). The combination of these benefits is driving the higher penetration of gas turbines into the the electrical grid. The Independent System Operator of New England (ISO-NE) is a prime example. Over the two decades, the level of gas generation in ISO-NE has increased from 5% to 50% of total generation capacity [3]. However, the benefits of natural gas are not without some risk. As the level of natural gas-based generation increases, larger and perhaps more variable[1], natural gas loads will effect the operations of the large natural gas transmission pipelines that bring the gas from the sources to the generator and other gas loads. The impact is not just one-way. The finite capacity of these gas transmission pipelines will limit the availability of gas which will directly affect ability of natural gas generators to respond to grid operator control commands.

The majority of the distance between gas sources and gas generators and other loads is covered by large, high-pressure transmission pipelines. High pressure/density enables high throughput with the pressure drop driving the gas through the pipeline. As the pressure falls, the flow velocity increases (under constant mass flux) and the pressure then falls even faster. Gas compressors are used to maintain the throughput of pipeline and maintain the required pressure at the customer load points. Often these gas compressors are driven by gas engines that burn natural gas from the pipeline itself. Typical designs of transmission pipelines places compressors every $\sim$75-150 km. In large transmission pipelines that span 1000 km or more, compressors consume (burn) $\sim 2-5\%$ of the transmitted gas. This burn is a cost of transporting the gas, and who bears that cost affects the goals of the operational optimizations (discussed below). Complicating the domain, the bearer of this cost differs from country to country.

The difficulty and expense of building new or expanding large-scale infrastructure coupled with the increasing (and the potentially more time-variable) gas loads calls for improved optimization of pipeline operations. However, the goals of these optimizations must be aware of and developed within the regulatory, market and ownership frameworks of the pipelines.

S. Misra, M.W. Fisher and M. Chertkov are with Theory Division of LANL, Los Alamos, NM 87544

S. Misra is also with Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 02139

M.W. Fisher is also with Swarthmore College, 500 College Ave. Swarthmore, PA 19081

S. Backhaus is with MPA Division of LANL, Los Alamos, NM 87544

R. Bent and F.Pan are with DSA Division of LANL, Los Alamos, NM 87544

Manuscript received September 20, 2013

[1]Natural gas generation is often used to smooth the variability in renewable energy sources.

Here, we briefly review two existing frameworks that are at opposite ends of this regulatory/ownership spectrum. Norway presents a relatively simple framework. In Norway, gas sources, gas pipelines, and the sale of gas inside and outside the country is controlled by the government. Norway produces more gas than can be domestically consumed and has strong economic motivations to sell this excess to the rest of Europe. The demand for the gas (and the available Norwegian gas resource) is typically higher than the ability of Norway's pipeline network to transport the gas to markets at its border. To increase sales and revenues, the pipeline operator's primary objective is to increase the pipeline *throughput*, and the gas lost to compression offsets improvements in throughput making the optimal compression problem important in this context. The throughput on the Norwegian is complicated by the differing gas compositions required by the buyers of the gas and the differing compositions of the gas sources. See [4] for a discussion of this problem.

In the US, gas markets have been deregulated for many years [2]. The implication is that pipeline operators do not own sources of gas nor are they involved in sourcing and selling gas to consumers (LDC, industrial, or gas turbines). Instead, the pipeline operators are responsible for transporting the gas and maintaining and expanding the pipelines. Gas is sold in organized markets via bi-lateral arrangements between gas suppliers and consumers. In addition to securing the gas itself, the consumers (buyers) must have also purchased the right to move the gas though the pipeline from the gas sources to the gas load locations. It is the sale of these rights where pipeline owner/operators make their revenue, and reliably increasing the throughput of the pipeline can enable the owner/operator to secure additional revenue. Therefore, as with the case of Norway, the US pipeline operators have an interest in increasing the pipeline *throughput*. Gas lost to compression offsets improvements in throughput making the optimal compression problem important in this context.

Within these disparate pipeline ownership/operational frameworks, minimizing the cost of compression is an important problem whose solution will enable additional pipeline throughput. The key contribution of this paper is the development of a geometric programming (GP) based approach for optimizing the transport of natural gas. It offers similar computationally performance to existing algorithms [4] at the same time as being exact, with several advantages when considering natural extensions to the problem, including but not limited to stochastic gas draws, distributed control, risk mitigation, transient dynamics, and interdepedencies with power systems. The focus of this paper is to demonstrate GP's ability to match the performance of existing algorithms as to motivate its use in more complex settings where existing algorithms are not easily adapted.

The remainder of this manuscript is organized as follows. Section II gives an overview of the pipeline gas flow equations and formulates the Optimal Gas Flow (OGF) problem. Section III describes our approach to GP for gas pipelines and discusses the formulation on both line and tree-graph networks. For comparison, we also formulated a Dynamic Programming (DP) approach to the same problems. Section IV describes the implementation of the GP and DP algorithms as well as a greedy algorithm that is intended to represent how many US pipelines are operated today. This section also compares the results of applying these approaches to a model of the Belgian natural gas network and the Transco pipeline network in the US [5]. Finally, Section V provides some conclusions and a discussion of the paths forward for both the steady-state gas flow problem and the time variable flow (line-packing) problem.

II. Technical Introduction

A. Gas Flow Equations: Individual Pipe

To introduce notation and the fundamental physics of gas systems, we first consider the flow of a compressible gas in a single section of pipe. Transmission pipelines are typically 16-48 inches in diameter and operate at high pressures and mass flows, e.g. 200 to 1500 pounds per square inch (psi) and moving millions of cubic feet per day [6], [7]. Under these highly turbulent conditions, the pressure drop and energy loss due to shear is represented by a phenomenological friction factor, and the resulting gas flow model is a partial differential equation (PDE) with one spatial dimension x (along the pipe axis) and one time dimension [8]–[10]:

$$\partial_t \rho + \partial_x (u\rho) = 0, \tag{1}$$

$$\partial_t(\rho u) + \partial_x(\rho u^2) + \partial_x p = -\frac{\rho u|u|}{2d} f - \rho g \sin\alpha, \tag{2}$$

$$p = \rho Z R T. \tag{3}$$

Here, u, p, ρ are velocity, pressure, and density at the position, x. Equations (1,2,3) represent mass conservation, momentum balance and the ideal gas thermodynamic relation, respectively. The first term on the rhs of Eq. (2) represents the friction losses created in the pipe of diameter d with friction factor f. The second term on the rhs of Eq. (2) accounts for the gain or loss of momentum due to gravity g if the pipe is tilted by angle α. In Eq. (3), Z is the gas compressibility factor, T is the temperature, and R is the gas constant. We have also assumed (for simplicity) that the temperature does not change significantly along the pipe.

When gravity and gas inertia can be ignored (which are standard approximations for pipes operating near normal conditions), Eqs. (1,2,3) are rewritten in terms of the pressure p and the mass flux $\phi = u\rho$:

$$\partial_t p = -ZRT\partial_x \phi, \tag{4}$$

$$\partial_x p^2 = -\frac{fZRT}{d}\phi|\phi|, \tag{5}$$

We note that this system of equations leads to the following nonlinear diffusion equation:

$$\partial_t p^2 = \frac{d}{f}\frac{p}{|\phi|}\partial_x^2 p^2. \tag{6}$$

If the flow into and out of the pipe at the two ends balance such that the total mass of gas in the pipe does not change, the flow is steady and Eqs. (4,5) can be solved:

$$\phi = \text{const}, \quad p_{in}^2 - p(x)^2 = a\frac{x}{L}\phi|\phi|, \quad a \equiv \frac{fZRTL}{d}. \tag{7}$$

Here , $0 \le x \le L$, and L is the length of the pipe. The constant a characterizes the pressure drop due to flow in the pipe and is the only important pipe parameter in the steady-state model.

B. Steady Gas Flow over Network

Here we discuss how to apply the solution in Eq. (7) to the case of a steady flow over network. First, the direction of flow in each segment must be known to resolve the sign of the term $\phi|\phi|$. In the case of tree networks (see below), this direction is known if all of the injections (positive and negative) into the network are known. In meshed networks, this procedure is not so straight forward. One approach sets the directions arbitrarily initially. These directions are modified using iterative or enumerative approaches until a consistent and feasible solution is found. In the remainder of this discussion, we will assume that the flow directions are known.

To continue the discussion, we consider a Gas Flow (GF) network without compressors which is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with edges $\mathcal{E}$ and vertexes $\mathcal{V}$. A solution of the steady gas flow problem consists of finding a set of positive node pressures $p = (p_i \ge 0 | i \in \mathcal{V})$ and edge flows $\phi = (\phi_{ij} > 0 | (i,j) \in \mathcal{E})$ corresponding to a given set of gas injections $q = (q_i | i \in \mathcal{V})$, i.e.:

$$\forall (i,j) \in \mathcal{E}: \quad p_i^2 - p_j^2 = a_{ij}\phi_{ij}^2, \tag{8}$$

$$\forall i \in \mathcal{V}: \quad q_i = \sum_{j:(i,j)\in\mathcal{E}} \phi_{ij} - \sum_{j:(j,i)\in\mathcal{E}} \phi_{ji}. \tag{9}$$

We note that the GF problem in Eqs. (8, 9) can be restated as a convex optimization [11], [12]

In the steady-state model, the injections are balanced, i.e. $\sum_{i\in\mathcal{V}} q_i = 0$. There is one more node than there are edge equations in (8). Therefore, the pressure must be fixed at one of the nodes. Depending on the structure of the GF network and the gas injections, there may be no physical solution to the GF problem, i.e. the solution may have one or more $p_i^2 < 0$. In reality, the GF network cannot support the imposed gas injections and resulting edge flows ϕ_{ij} without boosting the pressure with gas compressors.

Next, the GF problem is formulated with compressors placed along edges (i,j) at a relative locations $r \in (0,1)$ (see Fig. 1). Let p_i and p_j be the pressures at nodes i and j, respectively. Assuming positive flow from i to j, the compressor inlet pressure is $p_i^2 - r a_{ij}\phi_{ij}^2$, and the compressor outlet pressure is $p_j^2 + (1-r)a_{ij}\phi_{ij}^2$. Equation (8) generalizes to

$$\forall (i,j) \in \mathcal{E}: \quad \alpha_{ij}^2 = \frac{p_j^2 + (1-r)a_{ij}\phi_{ij}^2}{p_i^2 - r a_{ij}\phi_{ij}^2}, \tag{10}$$

where α_{ij} is the ratio of the compressor outlet and inlet pressures along edge (i,j), i.e. the compression ratio (see Fig. 1). α_{ij} is the main control input to the GF network. For edges without compressors, $\alpha_{ij} = 1$, and Eq. (10) reduces to Eq. (8). Although a compressor has been added, the flow balance in Eq. (9) remains the same.

Standard "box" constraints are imposed on the compression ratios and on the pressures to reflect the engineering limits of the compressors and pipes that form the network and to ensure

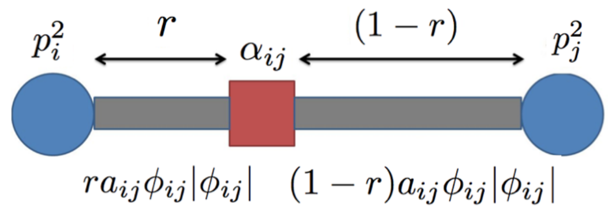

Fig. 1. Nodes (blue circles), edges (grey line) and compressor (red square) for the gas flow equation Eq. (10) and the line graph OGF. The compressor is at relative location r along the edge. In the line graph OGF, the compressor is assumed be be adjacent to node i, i.e. $r = 0$. The expressions below the edge are the drops in the square pressures before and after the compressor with compression ratio α_{ij}.

that pressures remain high enough to drive gas flow through the network:

$$\forall i \in \mathcal{V}: \quad 0 \le \underline{p}_i \le p_i \le \overline{p}_i, \tag{11}$$

$$\forall (i,j) \in \mathcal{E}: \quad 1 \le \alpha_{ij} \le \overline{\alpha}_{ij}. \tag{12}$$

For edges where there is no compressor, $1 = \alpha_{ij} = \overline{\alpha}_{ij}$.

C. Optimization Problem: Optimum Gas Flow (OGF)

In the GF model above, the only source of cost is the energy required to run the compressors (ij) at compression ratio α_{ij} and mass flux ϕ_{ij}. We adopt the simplest model of this cost [13] (see also [14] for more general discussion of compression cost):

$$C = \sum_{(i,j)\in\mathcal{E}} \frac{c_{ij}\phi_{ij}}{\eta_{ij}} \left(\alpha_{ij}^m - 1\right), \tag{13}$$

where c_{ij} is a constant which may depend on the compressor, $0 < m = (\gamma - 1)/\gamma < 1$, and γ is the gas heat capacity ratio. η_{ij} is the efficiency factor measuring the ratio of the useful power transferred to the gas flow to the shaft power required to run the compressor. Here, we assume η_{ij} is constant. (Note that more realistic modeling should account for the dependence of η_{ij} on the ratio of the compressor motor speed to the speed of the flow, however, we stick here to the simplest plausible model [14].)

In the remainder of the manuscript we focus on solving the OGF problem, i.e. minimizing the cost (13) over $\alpha = (\alpha_{ij} | (i,j) \in \mathcal{E})$ within the feasibility range defined by Eqs. (11,12).

III. Solving OGF on Network without Cycles

A. Geometric Programming (GP)

The major contribution of this manuscript is the formulation of exact and efficient solution of the OGF problem on a gas network without cycles. The approach, described below in two steps, is based on Geometric Programming (GP). See [15] and references therein for a comprehensive discussion of GGPs. For simplicity, we first consider a line graph without constraints on the compression ratios. Subsequently, we will quickly move to a tree graph that includes compression ratio constraints.

Line graph:

First, we consider a line graph with $n+1$ nodes numbered $0, 1, \dots, n$ and n edges/compressors numbered $1, 2, \dots, n$. In the steady-state model considered here, the flow on each edge $(i, i+1)$ is computed by summing over all of the injections prior to that edge, i.e. $\phi_{i,i+1} = \sum_{k=0\dots i} q_k$. (We note that this simple approach does not work in the non-steady case.) For sake of this discussion, we assume that all $\phi_{i,i+1}$ are directed $i \to i+1$ and that compressor i with compression ratio α_i is located on the edge $(i-1, i)$ right after node $i-1$, i.e, $r = 0^+$ in Eq (10) and Fig. 1. The OGF can be stated as the following optimization problem:

$$\begin{aligned} \min_{\alpha,\beta} \quad & \textstyle\sum_i d_i(\alpha_i^m - 1) \qquad (14)\\ \text{s.t.} \quad & \beta_{i-1}\alpha_i - \delta_i = \beta_i, \\ & \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i, \end{aligned}$$

where $\beta_i = p_i^2$ is the square of the pressure at node i, $\delta_i = a_{i-1,i}\phi_{i-1,i}^2 \ge 0$ is the loss of pressure squared that occurs between the outlet of compressor i and node i. Note that this drop in pressure is known since $\phi_{i-1,i}$ is known. We also consider the quantities $d_i = c_{ij}\phi_{i,j}/\eta_{ij}$ to be constants.

Eliminating the α_i's by expressing them in terms of the β_i's and substituting the result back into Eq. (14), one arrives at

$$\begin{aligned} \min_{\beta} \quad & \textstyle\sum_i d_i(\beta_i\beta_{i-1}^{-1} + \delta_i\beta_{i-1}^{-1})^m \qquad (15)\\ \text{s.t.} \quad & \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i, \end{aligned}$$

where we have dropped the constant d_i term from Eq. (14). Equation (15) is an instance of what is called a Generalized Geometric Program (GGP). A GGP can be transformed into a convex optimization problem—a process we summarize below.

By introducing the extra variables t_i, the GGP OGF is recast in the following equivalent formulation:

$$\begin{aligned} \min_{\beta,t} \quad & \sum_i d_i t_i^m \\ \text{s.t.} \quad & \beta_{i+1}\beta_i^{-1} + \delta_{i+1}\beta_i^{-1} \le t_i, \\ & \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i. \end{aligned}$$

The equivalence to Eq. (15) follows because, at optimality, we must have $\beta_i\beta_{i-1}^{-1} + \delta_i\beta_{i-1}^{-1} = t_i$. Then we introduce $z_i = \log\beta_i$ and $u_i = \log t_i$, and the equivalent formulation becomes

$$\begin{aligned} \min_{u,z} \quad & \sum_i d_i e^{mu_i} && (16)\\ \text{s.t.} \quad & \log\left(e^{z_i - z_{i-1} - u_i} + \delta_i e^{-z_{i-1}-u_i}\right) \le 0, && (17)\\ & \log\underline{\beta}_i \le z_i \le \log\bar{\beta}_i, \quad \forall i. && (18) \end{aligned}$$

It can be verified that Eq. (16) - (18) constitutes a convex optimization problem and can be solved efficiently.

Tree Graph.

Using the guidance provided by formulation for a simple line graph network, we now consider a tree network and give a complete description of our OGF solution method. As with the line network, the first step is to determine the ϕ_{ij} from the q_i at each node. In particular, removing any edge $(i,j) \in \mathcal{E}$ splits $\mathcal{G}$ into two disjoint graphs $\mathcal{G}_i$ and $\mathcal{G}_j$. The flow on (i,j) can then be computed as

$$\phi_{ij} = \sum_{i\in\mathcal{G}_i} q_i = -\sum_{i\in\mathcal{G}_j} q_i. \qquad (19)$$

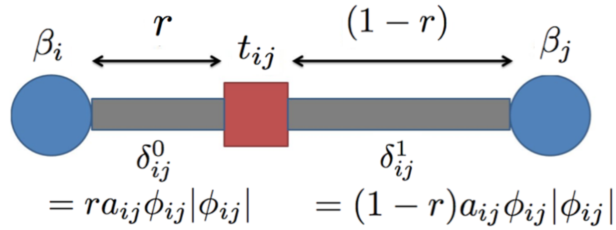

Fig. 2. Pipeline segment configuration for the the tree graph OGF. Color coding of the components is the same as in Fig. 1. δ_{ij}^0 and δ_{ij}^1 are the drop in the squared pressure β from node i to the compressor inlet and from the compressor outlet to node j, respectively. The compression ratio is t_{ij}.

For the line graph, we did not consider constraints on the compression ratio. Equation (14) shows one possible consequence of this choice. Specifically, decompression (i.e. $\alpha < 1$) could actually be encouraged. In practice, this should not occur, and we eliminate this potentially negative cost by putting a floor of 1 on the term in parenthesis in Eq. 14. Adding the $\bar{\alpha}$ constraint and the modified compression cost function, the tree-graph version of the OGF becomes

$$\begin{aligned} \min \quad & \sum_{(i,j)\in\mathcal{E}} d_{ij}\max\{\alpha_{ij}^m, 1\} && (20)\\ \text{s.t.} \quad & \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i \in \mathcal{V} && (21)\\ & \alpha_{ij} \le \bar{\alpha}_{ij}, \quad \forall (i,j) \in \mathcal{E} && (22) \end{aligned}$$

For the case of a tree, we use a more general system representation by allowing a compressor to appear anywhere along edge (i,j) (see Fig. 2). For each $(i,j) \in \mathcal{E}$, let γ_{ij} denote the pressure squared immediately after the compressor and let β_{ij} denote the pressure squared immediately before the compressor. Let ϕ_{ij} be the positive flow from i to j. Denote by $\delta_{ij}^0 \ge 0$ the drop in pressure squared between the node i and the compressor on edge (i,j) and by $\delta_{ij}^1 \ge 0$ the drop in pressure between the point right after the compressor and node j. Refering to Eq (10), these quantities are given by $\delta_{ij}^0 = ra_{ij}\phi_{ij}^2$ and $\delta_{ij}^1 = (1-r)a_{ij}\phi_{ij}^2$.

Therefore,

$$\beta_i - \delta_{ij}^0 = \beta_{ij}, \qquad (23)$$
$$\gamma_{ij} - \delta_{ij}^1 = \beta_j, \qquad (24)$$

and the compression ratio is now given by

$$\alpha_{ij} = \frac{\gamma_{ij}}{\beta_{ij}}. \qquad (25)$$

Substituting Eqs. (23,24,25) into the tree-graph OGF in

Eqs. (20,21,22), we obtain the equivalent formulation

$$\min \quad \sum_{(i,j)\in\mathcal{E}} d_{ij}\max\{(\beta_j+\delta^1_{ij})^m(\beta_i-\delta^0_{ij})^{-m},1\} \tag{26}$$
$$\text{s.t.} \quad \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i\in\mathcal{V} \tag{27}$$
$$\frac{\beta_j+\delta^1_{ij}}{\beta_i-\delta^0_{ij}} \le \bar{\alpha}_{ij}, \quad \forall (i,j)\in\mathcal{E} \tag{28}$$

Similar to the line-graph OGF, we introduce the extra variable t_{ij} and rewrite the tree-graph OGF as (see Fig. 2)

$$\min \quad \sum_{(i,j)\in\mathcal{E}} d_{ij}t^m_{ij} \tag{29}$$
$$\text{s.t.} \quad \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i\in\mathcal{V} \tag{30}$$
$$\max\{(\beta_j+\delta^1_{ij})^m(\beta_i-\delta^0_{ij})^{-m},1\} \le t_{ij}, \tag{31}$$
$$t_{ij}\le\bar{\alpha}_{ij}, \quad \forall (i,j)\in\mathcal{E}. \tag{32}$$

The cost function is an increasing function of t_{ij} for all $(i,j)\in\mathcal{E}$ which guarantees that, at optimality, Eq. (31) is replaced by an equality and t_{ij} is the compression ratio of the compressor. The tree-graph OGF above is equivalent to the following GP:

$$\min \quad \sum_{(i,j)\in\mathcal{E}} d_{ij}t^m_{ij} \tag{33}$$
$$\text{s.t.} \quad \underline{\beta}_i \le \beta_i \le \bar{\beta}_i, \quad \forall i\in\mathcal{V} \tag{34}$$
$$1\le t_{ij}\le\bar{\alpha}_{ij}, \tag{35}$$
$$\beta_j\beta_i^{-1}t_{ij}^{-1}+\delta^1_{ij}\beta_i^{-1}t_{ij}^{-1}+\delta^0_{ij}\beta_i^{-1}\le 1, \quad \forall (i,j)\in\mathcal{E} \tag{36}$$

As with the line-graph OGF, this can be reduced to a convex optimization problem by introducing variables which are the $\log$ of the original variables. Letting $\hat{t}_{ij}=\log t_{ij}$ and $\hat{\beta}_i=\log\beta_i$, we arrive at the following convex, tree-graph OGF formulation:

$$\min \quad \log\left(\sum_{(i,j)\in\mathcal{E}} d_{ij}e^{m\hat{t}_{ij}}\right), \quad \forall i\in\mathcal{V} \tag{37}$$
$$\text{s.t.} \quad \log(\underline{\beta}_i)\le\hat{\beta}_i\le\log(\bar{\beta}_i) \tag{38}$$
$$0\le\hat{t}_{ij}\le\log(\bar{\alpha}_{ij}), \tag{39}$$
$$\log\left(e^{\hat{\beta}_j-\hat{\beta}_i-\hat{t}_{ij}}+\delta^1_{ij}e^{-\hat{\beta}_i-\hat{t}_{ij}}+\delta^0_{ij}e^{-\hat{\beta}_i}\right)\le 0, \quad \forall (i,j)\in\mathcal{E}. \tag{40}$$

This is the final version of our Geometric Programming formulation of the OGF.

Recall that the cost function in our original tree graph OGF formulation in Eqs. (20,21,22) is motivated by not wanting to reward decompression in the cost function. However, the formulation in Eqs. (20,21,22) (and all of its reformulations) does allow for unrewarded decompression. In practice, decompression can be implemented by simple procedures (such as a throttling valve) and is usually not associated with any significant cost. It is possible that allowing zero-cost decompression could lead to a lower overall global cost of compression when compared to the case where we do not allow decompression (i.e., $\underline{\alpha}_{ij}=1$ for all edges). However, for the sake of remaining consistent with the current practices and to accommodate circumstances where it is not be possible to implement decompression, we modify our OGF formulation in Eqs. (20,21,22) by adding the constraints $\underline{\alpha}_{ij}=1$ for all edges and propose a signomial programming approach to deal with these additional constraints.

Note that adding a lower bound of 1 on the compression ratios is same as $1\le\alpha_{ij}=\frac{\beta_j+\delta^1_{ij}}{\beta_i-\delta^0_{ij}}$ which after rearranging the terms becomes $\beta_i-\beta_j\le\delta^0_{ij}+\delta^1_{ij}$. Following the exact same steps as in the derivation of the GP OGF, we get to the following optimization problem

$$\min \quad \log\left(\sum_{(i,j)\in\mathcal{E}} d_{ij}e^{m\hat{t}_{ij}}\right) \tag{41}$$
$$\text{s.t.} \quad \log(\underline{\beta}_i)\le\hat{\beta}_i\le\log(\bar{\beta}_i), \quad \forall i\in\mathcal{V} \tag{42}$$
$$0\le\hat{t}_{ij}\le\log(\bar{\alpha}_{ij}), \tag{43}$$
$$\log\left(e^{\hat{\beta}_j-\hat{\beta}_i-\hat{t}_{ij}}+\delta^1_{ij}e^{-\hat{\beta}_i-\hat{t}_{ij}}+\delta^0_{ij}e^{-\hat{\beta}_i}\right)\le 0, \tag{44}$$
$$\hat{\beta}_i\le\log(e^{\hat{\beta}_j}+\delta_{ij}), \quad \forall (i,j)\in\mathcal{E} \tag{45}$$

where $\delta_{ij}=\delta^0_{ij}+\delta^2_{ij}$. The challenge with this formulation is that the constraints in Eq. (45) are non-convex. However, this formulation is almost a GP, and we propose a signomial programming approach. Signomial programming is an iterative descent method, where, in each iteration, the non-convex constraints are linearized and the resulting GP is solved to perform one descent step. The iterations of the algorithm are described below.

Signomial Programming iteration

1. The constraints Eq. (45) are linearized to obtain
$$\hat{\beta}_i\le\log\left(e^{\hat{\beta}^{(t)}_j}+\delta_{ij}\right)+\frac{e^{\hat{\beta}^{(t)}_j}}{e^{\hat{\beta}^{(t)}_j}+\delta_{ij}}(\hat{\beta}_j-\hat{\beta}^{(t)}_j)+\epsilon, \tag{46}$$
where a small tolerance parameter $\epsilon>0$ is added to act as a trade-off between speed of convergence and accuracy.
2. Solve the Geometric Program that results from Eqs. (41)-(45) by replacing the constraints in Eq. (45) withEq. (46) to obtain the new iterates at iteration number $t+1$.

In practice, if the network consists of a mixture of edges where decompression can be performed and edges where decompression cannot be performed, then the signomial program only needs to linearize the Eq. (45) constraints only for edges that do not allow decompression. Steps 1 and 2 are repeated until a certain stopping criterion is reached.

B. Dynamic Programming (DP)

For comparison of both the formulation and the numerics, we describe a Dynamic Programming (DP) approach to solving the OGF. The DP approach is not new. It was pioneered by [13] and has a long history, see e.g. [4] for extended bibliography. The DP approach exploits the separability of the cost function in Eq. (20) over the edges as well as the tree structure of the underlying graph by calculating the "cost-to-go" functions recursively from the leaves upwards.

Specifically, choose the node where the pressure is fixed to be the root of the tree denoted by r. At each node i, we have a cost-to-go function $J_i(\beta_i)$ which is a function of the squared pressure at that node. The DP algorithm proceeds as:

1. *Initialization*. Set $\mathcal{S} = \mathcal{V}$, i.e. the set of all nodes. For each node i that is a leaf of the tree $\mathcal{G}$ set
$$J_i(\beta_i) = \begin{cases} 0, & \underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i \\ \infty, & \text{otherwise} \end{cases}$$
Remove all the leaves from $\mathcal{S}$.
2. Repeat the following steps while $\mathcal{S}$ is non-empty:
   (a) Pick a node $i \in \mathcal{S}$ such that all its children have been removed from $\mathcal{S}$.
   (b) Let $v_1, \ldots, v_k$ denote the children of i. Determine the value of the cost-to-go function $J_i(\beta_i)$ for each $\underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i$ as follows.
   * For each choice of compression ratio on the edges $(i, v_1), \ldots, (i, v_k)$, compute the quantity
   $$L(\alpha_1, \ldots, \alpha_k) = \sum_{j=1}^{m} d_{iv_j} \alpha_{iv_j}^m + J_{v_j}(\beta_{v_j}),$$
   where β_{v_j} is the implied squared pressure at v_j for the choice of α_{iv_j} above, i.e.
   $$\beta_{v_j} = \begin{cases} (\beta_i - \delta^0_{iv_j})\alpha_{iv_j} - \delta^1_{iv_j} & \text{if } \phi_{iv_j} > 0, \\ \beta_i + \delta^0_{iv_j} + \delta^1_{iv_j}, & \text{otherwise.} \end{cases} \tag{47}$$
   * Set
   $$J_i(\beta_i) = \begin{cases} \min L(\alpha_1, \ldots, \alpha_k) & \text{if } \underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i \\ \infty & \text{otherwise} \end{cases} \tag{48}$$
   * Remove i from $\mathcal{S}$.
3. *Traceback*. Fix the root sqaured pressure $\beta_r = \beta_0$ where β_0 is the given squared pressure. Set $\mathcal{S} = \mathcal{V}$ to be the set of all nodes. Remove the root r from $\mathcal{S}$. Repeat the following while $\mathcal{S}$ is non-empty.
   (a) Pick $i \in \mathcal{S}$ such that its parent has been removed from $\mathcal{S}$.
   (b) Find the implied pressure β_i at i by using the optimal choice of α's in the optimization Eq. (48) and using Eq. (47).
   (c) Remove i from $\mathcal{S}$.

The squared pressures β_i obtained in Step 3 are optimal. The optimal value is given by the root cost-to-go function $J_r(\beta_r)$. In practice for implementation, one needs to discretize the space $\underline{\beta}_i \leq \beta_i \leq \bar{\beta}_i$ for each $i \in \mathcal{V}$ and the space $1 \leq \alpha_{ij} \leq \bar{\alpha}_{ij}$ for each edge $(i, j) \in \mathcal{E}$ which has a compressor.

IV. Experiments

A. Implementation

The dynamic program and greedy algorithm (described below) were implemented in C++. The geometric programming algorithm is implemented in python using CVXOPT [16]. The results were obtained on an Intel 80386 (i386) 32-bit processor running Linux 11.10 . The first step for all the algorithms is computing the flow on each edge of the tree networks using Eq. (19). For each node, n, in the tree, we first compute the sum of the injections that occur on nodes that have an ancestor that is n. Second, working from the root of the tree towards the leaves, the inflow to a node is distributed to the child edges of that node by weighting with the sum of loads beneath each edge.

Next we solve the OGF without allowing decompression, i.e. $\underline{\alpha}_{ij} = 1$. We compare the signomial program results with the dynamic program results and with results from a "greedy compression" scheme. We believe the greedy compression algorithm is reasonable representation of the intuition of many of the trained operators discussed above.

1) Signomial Programming: The geometric program iterations of the signomial program are solved using CVXOPT with a trust region of ϵ about the previous solution. The initial point for this iterative method is the solution of the geometric program with no additional edge constraints (which allows for decompression). The signomial program interations are terminated when the difference in the norms of the solution vectors from one iteration to the next is less than a specified tolerance $\delta > 0$. As remarked earlier, smaller tolerance parameters ϵ and δ lead to higher accuracy but longer runtimes.

2) Dynamic Programming: The DP was solved using our own code develop according to the algorithm in Section III-B. The number of bins for the α's and β's are specified as inputs. Finer discretization leads to higher accuracy but with longer runtime. DP runtimes increase exponentially with the number of compressors, while signomial programming runtimes do not.

3) Greedy Compression: A third "greedy compression" algorithm was implemented for comparison with the DP and GP. Although exact representation of operator behavior is beyond the scope of this manuscript, we believe the greedy compression algorithm to be a reasonable representation of the day-to-day practice of operators of many natural gas transmission pipelines [17]. Greedy compression is a simple scheme which uses local observations to decide when to compress. The simple rule it tries to adhere to is: whenever the pressure falls below the lower bound, boosts the pressure to the maximum value allowed by the local constraints. This simple rule by itself is not enough to implement the algorithm because of the complications arising from pressure inconsistencies, i.e., it does not by itself always produce a feasible compression ratio configuration in the network. In what follows, we discuss how to address this issue in more detail and implement it for the Transco pipeline model [5].

For the purposes of this discussion we assume that all $\bar{\beta}_i = \overline{\beta}$ and $\underline{\beta}_i = \underline{\beta}$ since this is what we actually implement for the Transco model. Recall that we always pick the root node to be a significant source. We first set the pressure at the root node to the maximum $\overline{\beta}$ and traverse downstream from the root updating the square pressure at other nodes using Equation 8. This of course leads to several node pressures which are below the lower bound. For reasons that will become clear, we first classify such infeasible nodes into those having one inflow and those having multiple inflows. For those with a single inflow, we set the output pressure of the nearest upstream compressor to the maximum possible value subject to the upper bound on pressure and compression ratio. In some cases the pressure at the infeasible node is significantly below the allowed lower bound such that multiple compressors upstream must be turned on to boost the pressure into the feasible range. In this case, we try to implement the compressor ratio configurations in a "greedy" manner described above, which however sometimes needs to be specialized to our given network. We omit the details of this for brevity.

Next, we address the case of nodes where there are multiple inflows. Consider such a node q. The issue with such nodes is that if we follow the method described in the previous case for each such inflow, then the implied pressure at q from each of these paths don't always match leading to an infeasible configuration of compression ratios. Additionally, such inconsistencies might have already been introduced by the algorithm in one of its previous steps. We need to set the compression ratios of the nearest upstream compressors such that the implied pressure at q is consistent. Suppose $e_1, \ldots, e_k$ are the edges adjacent and immediately upstream of q. Let the implied pressure squared from upstream e_i be $\beta_q(i)$. As remarked, they are not necessarily equal. We handle the following two cases separately.

(i) At least one $\beta_q(i)$ is below the minimum pressure $\underline{\beta}$:
In this case denote by Δ_i the pressure drop from the outlet of the nearest compressor along upstream e_i and node q. Assume without loss of generality that $\Delta_1 \geq \Delta_2 \geq \ldots \geq \Delta_k$. We first want to decide on one of these compressors at which we set the outlet pressure to the maximum allowed value. As it turns out, we must choose the compressor on path e_1 for this purpose. This is because a quick calculation shows that if we choose another compressor, say along path e_2 and set the output pressure of that compressor to $\bar{\beta}$, then the implied pressure at the output of compressor along path e_1 is $\bar{\beta} + \Delta_1 - \Delta_2$ which can be greater than $\bar{\beta}$ leading to infeasibility. So, we set the output pressure of the compressor along e_1 to the maximum allowed pressure and set the other compressors such that the implied pressure at q match.

(ii) No $\beta_q(i)$ is outside the pressure bounds:
In this case we set $\beta_q = \max\{\beta_q(i)\}$ to be the target pressure at q and set the compressors upstream such that the implied pressure squared at q all match and are equal to β_q. This choice of β_q ensures that we do not have to decompress at any of the upstream compressors.

B. Models

We consider two natural gas pipeline networks to test our algorithms–the Belgian gas network [12] and the Transco gas network [5] in the Eastern US. Both networks are nearly tree like. The minor amount of looping in each network was reduced to a tree topology by breaking the loops locations where the flow is expected to be relatively low. For both test cases, a root node is selected and the square pressure at the root is set to $\overline{\beta}$.

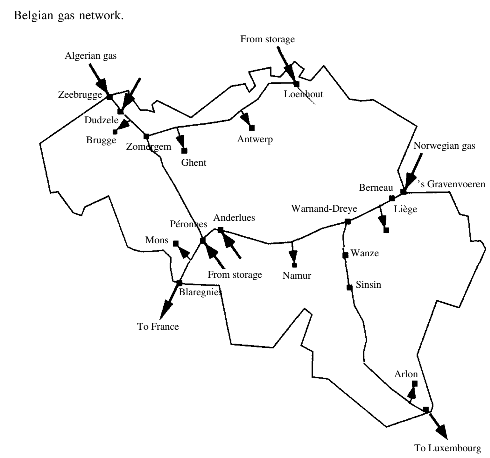

Fig. 3. Schematic representation of the Belgian gas transmission network.

1) Belgian Gas Network: Before comparing the algorithms discussed above on large pipeline networks, we test the accuracy on a small test case of the Belgian gas network (see Fig. 3) and compare our results to those in [12]. The Belgian network only contains 20 nodes and 2 compressors. Both signomial and dynamic programming are used to solve for the optimum steady-state compression. For dynamic programming, 1000 pressure bins and 1000 α bins are used. For the signomial programming, ϵ was set to 10^{-3} and the tolerance δ was set to 10^{-6}. Using the same pressure and compression limits as in [12], the fractional difference between our optimal compression costs and those in [12] is only $\sim 5 \times 10^{-4}$. Our pressure profiles at optimal compression ratio also agreed with the results in [12].

To test for the effect of allowing decompression, we compare signomial programming without decompression and pure geometric programming (which does allow decompression). The fractional difference in optimal costs is $\sim 10^{-2}$ with the geometric programming cost less than for signomial programming. For this small test case, the additional freedom of decompression slightly decreases the total cost of compression.

2) Williams Transco Pipeline: The Willams Transcontinental (Transco) pipeline (see Fig. 4and [5]) is our second test case and represents a large pipeline network. The Tranco pipeline extends northeast from gas sources in and around the Gulf of Mexico to load centers in New York and New Jersey. The structure of the pipeline near to the sources is tree like, however, the details of the gas injections and withdrawls is quite complicated. Therefore, we choose to test our algorithms on the northern half of the pipeline extending from South Carolina up to the load centers in New Jersey and New York and additional sources in Pennsylvania. We partition a few small loops near the end of the pipeline to achieve a tree-like structure. In spite of reducing the scale of the Transco model, it still consists of 98 nodes and 31 compressors.

The GP and DP algorithms only constrain the pressure at the nodes. In order to maintain allowable pressures along the

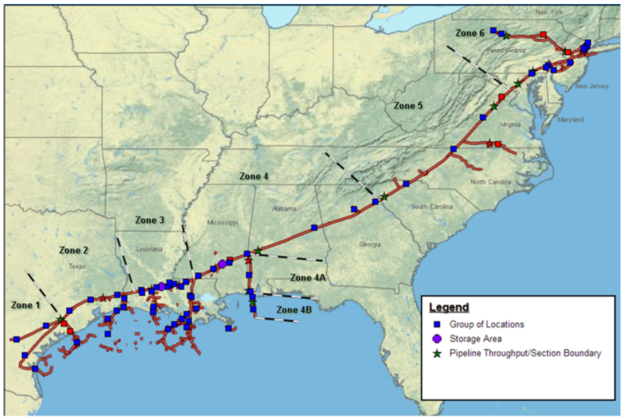

Fig. 4. Schematic representation of the Transco gas transmission network. Small loops in the load centers near the northern end of the pipeline were partitioned to create a tree structure. For this work, only the northern half of the pipeline was modeled, starting from the southern border of South Carolina.

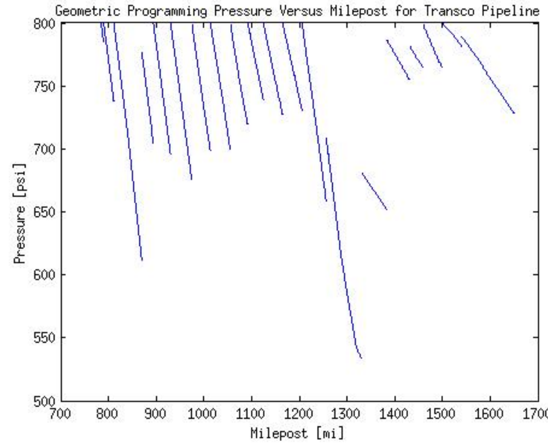

Fig. 5. Gas pressure versus milepost for the signomial programming solution for the Transco pipeline.

entirety of the pipeline, we model each compressor as being attached to very short runs of inlet and outlet pipeline that attach to nodes with zero gas injections. These short runs of inlet and outlet pipes put nodes very close to the compressor and keep the inlet and outlet square pressures from violating $\underline{\beta}$ or $\overline{\beta}$. The minimum and maximum pressures are set to 500 psi and 800 psi, respectively, as suggested by plots of operational data over this section of pipeline [18].

Using inflow and injection data from December 29, 2012 we compare results for signomial programming, dynamic programming, and greedy compression. We note that this data corresponds to near peak load conditions on the Transco pipeline. For the dynamic programming, 1000 pressure bins and 400 α bins were used. For the signomial programming, ϵ is set to 10^{-2} and the tolerance δ is set to 10^{-3}. The fractional difference in optimal costs between signomial programming and DP is $\sim\ 3 \times 10^{-5}$. The greedy compression optimal cost is 5.4% higher than the two other methods demonstrating the benefits of a global optimization approach. The fractional difference between the optimal costs for signomial programming without decompression and pure geometric programming (which does allow decompression) is only $\sim 1 \times 10^{-7}$. Therefore, decompression is not a significant degree of freedom in all cases.

Figures 5, 6, and 7 show plots of the pressure as a function of distance along the pipeline for the signomial program, dynamic program, and greedy compression approaches, respectively. The signomial and dynamic programming show negligible differences, while the greedy compression algorithm has a very different pressure profile. It is interesting to note that that although the greedy algorithm runs fewer compressors (9) in comparison to signomial or dynamic programming (19), the cost of compression higher for the greedy algorithm. A likely cause for this difference is the lower average gas density, and therefore higher gas velocities and larger pressure drops, in the greedy compression case.

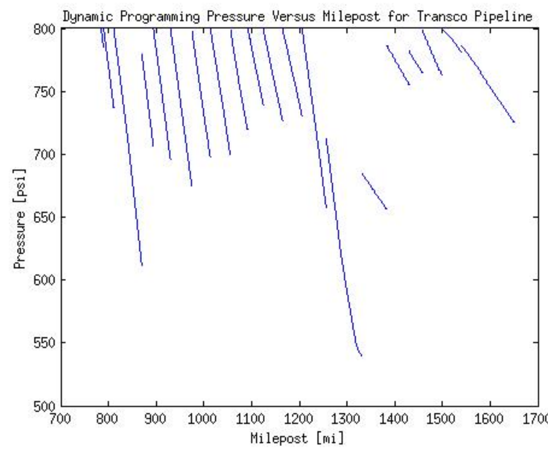

Fig. 6. Gas pressure versus milepost for the dynamic programming solution for the Transco pipeline.

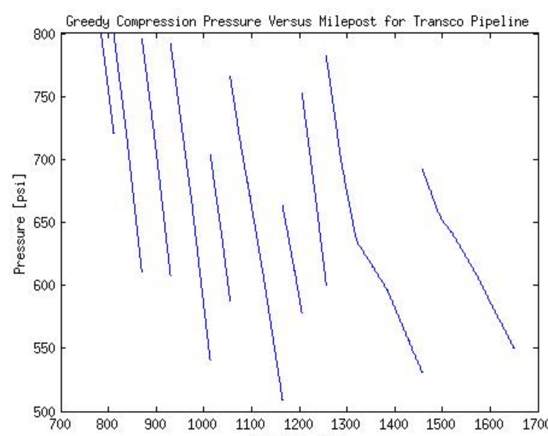

Fig. 7. Gas pressure versus milepost for the greedy compression algorithm solution for the Transco pipeline.

V. Path Forward

The main contribution of this manuscript consists in exactly solving the steady-state Optimal Gas Flow (OGF) problem (also called Fuel Cost Minimization Problem in the literature [4], [14]) with a Geometric Programming (GP) approach [15]—a new approach for this application. For loop-free gas pipeline networks, the GP approach turns the OGF problem into a convex optimization allowing for exact and efficient (polynomial time) solution. A significant advantage of the GP method over the traditional Dynamic Programming approach [13], [19] derives from not having to discretize the node pressure and compression ratio variables. The GP approach also scales well, even in networks with a high degree of branching.

The nature of the OGF problem discussed in the manuscript as well as practical considerations resulted in a number of assumptions: 1) the steady-state (balanced injections) nature of the gas flow, 2) additional compressor feasibility constraints, 3) uniform temperature distribution along the pipe, and 4) the reduction of network cycles to tree-like structures. However, the majority of these assumptions can be relaxed, which form natural extensions to the current work:

- Many modern gas networks contain no or very few cycles. Combining and extending currently separated (tree-like) systems into one larger and thus more reliable system will lead to the emergence of significantly meshed systems containing multiple cycles. The extension of the GP approach to the general case of networks with cycles constitutes an interesting challenge. Indeed, finding the flows and finding optimal compression rates — the two problems which became separable in the tree-network case – are now mutually dependent. However, this complication can be overcome. One promising approach consists in solving the OGF through multiple repetitions of the following two alternating steps – (1) finding compression ratios given the flows (where the GP applies directly), and (2) finding flows given compression ratios. Another approach is to apply the log-change of variables (leading to the convex optimization in the tree case) followed by relaxation of the new non-convex, cycle-related constraints.
- Equation (7) describes the case of balanced flows, i.e. $\sum_{i\in\mathcal{V}} q_i = 0$. However, this strict balance does not need to be hold on the scale of minutes or even hours. When the system is not balanced, the gas pressure changes leverage the natural storage capacity of pipelines, i.e. linepack. Exactly accounting for this effect within the basic model described by Eqs. (4,5) requires solving a system of coupled PDEs over all pipes of the network [9] [20], a problem which does not scale well. To achieve a computationally tractable approach, we plan to approximate Eqs. (4,5) by a linearized version of the nonlinear diffusion equation in Eq. (6). When temporal evolution of sources and sinks is sufficiently slow (so that one can ignore sound-wave-like transients), the (linearized) diffusive approximation will allow explicit solution for the spatiotemporal and flow dependence of the pressure, i.e. an approximate solution for the time-dependent line pack and a generalization of Eq. (7). The result is a generalized OGF that extends what used to be instantaneous optimization into multi-stage optimization that accounts for the evolution of the gas injections over time. We believe the GP approach can be extended to include this temporal evolution.

The GP approach has advantages over DP not only because it scales well, but also because GP allows a fully distributed implementation based on local measurements of pressure and flows at the compressors and local communications between nearest-neighbor compressors. We plan to explore this distributed cyber-physical control [21], [22] to gas networks in future work.

Finally, this study is motivated by our interest in coupled energy infrastructures, in particular gas and power system networks. Future increases in stochasticity in one network is expected to have impacts across the other coupled networks. For example, one mitigation strategy for addressing intermittency of renewable generation, e.g. wind and solar, uses controls on gas turbines to “smooth” the intermittency. However, these gas turbines are loads on the gas network (often burning comparable amount of gas as all other consumers combined). Therefore, the uncertainty of electric generation translates into temporally fast but spatially long-correlated uncertainty of gas consumption. Future work will quantify these and other effects of such coupling with a focus on analyzing the stochasticity and correlations across coupled infrastructure networks and using this understanding to develop improved optimization and control of combined systems.

Acknowledgment

The authors would like to thank Conrado Borraz-Sánchez for fruitful discussions and references, Ben Williams of Willams Pipelines for providing customer maps of the Transco pipeline. The work at LANL was funded by the Advanced Grid Modeling Program in the Office of Electricity in the US Department of Energy and was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396.

References

[1] T. J. Considine, R. Watson, and S. Blumsack, “The economic impacts of the pennsylvania marcellus shale natural gas play: An update,” 2010.

[2] “The future of natural gas:mit energy initiative, http://mitei.mit.edu/system/files/NaturalGas_Report.pdf,” 2010.

[3] “Iso new england: Adressing gas dependence, http://www.iso-ne.com/committees/comm_wkgrps/strategic_planning_discussion/materials/natural-gas-white-paper-draft-july-2012.pdf, year=2012,.”

[4] C. Borraz-Sánchez, “Optimization methods for pipeline transportation of natural gas,” Ph.D. dissertation, Department of Informatics, University of Bergen, Norway, October 2010.

[5] “The william transco pipe line, http://www.1line.williams.com/Transco/index.html.”

[6] CRANE, “Flow of fluids: Through valves, fittings and pipe,” Crane Company, New York, Technical paper 410M, 1982.

[7] S. Mokhatab, W. A. Poe, and J. G. Speight, *Handbook of Natural Gas Transmission and Processing*. Houston: Gulf Professional Publishing, 2006.

[8] A. Osiadacz, *Simulation and analysis of gas networks*. Gulf Pub. Co., 1987. [Online]. Available: http://books.google.com/books?id=cMxTAAAAMAAJ

[9] A. Thorley and C. Tiley, "Unsteady and transient flow of compressible fluids in pipelinesa review of theoretical and some experimental studies," *International Journal of Heat and Fluid Flow*, vol. 8, no. 1, pp. 3 – 15, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0142727X87900440

[10] S. A. Sardanashvili, *Computational Techniques and Algorithms (Pipeline Gas Transmission) [in Russian]*. FSUE Oil and Gaz, I.M. Gubkin, Russian State University of Oil and Gas, 2005.

[11] J. J. Maugis, "Etude de rseaux de transport et de distribution de uide [in french]," pp. 243–248, 1977.

[12] F. Babonneau, Y. Nesterov, and J.-P. Vial, "Design and operations of gas transmission networks," *Operations Research*, 2012. [Online]. Available: http://or.journal.informs.org/content/early/2012/02/10/opre.1110.1001.abstract

[13] P. Wong and R. Larson, "Optimization of natural-gas pipeline systems via dynamic programming," *Automatic Control, IEEE Transactions on*, vol. 13, no. 5, pp. 475–481, 1968.

[14] S. Wu, R. Ros-Mercado, E. Boyd, and L. Scott, "Model relaxations for the fuel cost minimization of steady-state gas pipeline networks," *Mathematical and Computer Modelling*, vol. 31, no. 23, pp. 197 – 220, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0895717799002320

[15] S. Boyd, S.-J. Kim, L. Vanderberghe, and A. Hassibi, "A tutorial on geometric programming," *Optim Eng*, vol. 8, pp. 67–127, 2007.

[16] "http://cvxopt.org/."

[17] "Private communication with the leadership of the spectra energy corporation, http://www.spectraenergy.com/."

[18] "http://www.gaselectricpartnership.com/fbowdenWms020811.pdf."

[19] H. Lall and P. Percell, "A dynamic programming based gas pipeline optimizer," in *Analysis and Optimization of Systems*, A. Bensoussan and J. Lions, Eds. Springer, Berlin, Germany, 1990, vol. 57, pp. 123–132.

[20] T. Kiuchi, "An implicit method for transient gas flows in pipe networks," *International Journal of Heat and Fluid Flow*, vol. 15, no. 5, pp. 378 – 383, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0142727X94900515

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1 – 122, 2010.

[22] A. Nedić and A. Ozdaglar, *Cooperative distributed multi-agent optimization*. Cambridge University Press, 2010.