Gas Network Optimization: A comparison of Piecewise Linear Models

Carlos M. Correa-Posada[a,b,*], Pedro Sánchez-Martín[b]

[a] *XM, Compañía de Expertos en Mercados, Colombian Power System Operator, Medellín, Colombia.*
[b] *Technological Research Institute (IIT), ICAI School of Engineering, Comillas Pontifical University, Madrid 28015, Spain.*

Abstract

Gas network optimization manages the gas transport by minimizing operating costs and fulfilling contracts between consumers and suppliers. This is an NP-hard problem governed by non-convex and nonlinear gas transport functions that can be modeled by mixed integer linear programming (MILP) techniques. Under these methods, piecewise linear functions describe nonlinearities and binary variables avoid local optima due to non-convexities. This paper compares theoretically and computationally basic and advanced MILP formulations for the gas network optimization in dynamic or in steady-state conditions. Case studies are carried out to compare the performance of each MILP formulation for different network configurations, sizes and levels of complexity. In addition, since the accuracy of linear approximations significantly depends on the number and location of linear segments, this paper also proposes a goal programming method to construct a-priori the piecewise linear functions. This method is based on the minimization of the mean squared error of each approximation subject to predefined error goals.

Keywords: Computation, Gases, Mathematical modelling, Nonlinear dynamics, Optimisation, Transport processes

1. Introduction

The aim of gas network optimization is to manage the gas transport minimizing operating costs and fulfilling contracts between consumers and suppliers. The relevance and complexity of this kind of problems is continuously increasing. The way gas networks have been traditionally planned and operated is changing (Wang and Xu, 2014). Nowadays planners and dispatchers are facing larger transport networks, unceasing growth in production and consumption,

[*] Corresponding author. Tel: +34 91 542-2800 ext. 2349
Email addresses: cmcorrea@xm.com.com; alomariox@gmail.com (Carlos M. Correa-Posada), psanchez@upcomillas.es (Pedro Sánchez-Martín)

market liberalization processes and higher levels of interdependency between energy networks. Usually, steady-state representations of the system, based on average values, are sufficient for the long- and midterm planning. However, the gas dispatch in the short term entails the consideration of system responses to changes in demand, supply or equipment availability (Mahgerefteh et al., 2006a). This situation gives rise to dynamic or inter-temporal relationships in the optimization problem.

The solution of mathematical formulation for systems represented either dynamically or in steady state represents a major challenge. The optimization problem besides being NP-hard (Vielma, 2013) is nonlinear and non-convex. Different methods are proposed in the literature to solve models of this nature. Particularly nonlinear models are applied because nonlinearities can be represented in their original form. E.g., Selot et al. (2008) uses mixed integer nonlinear programming to tackle the steady-state case; Ehrhardt and Steinbach (2005) applies sequential quadratic programming to solve the system dynamics; and Chaudry et al. (2008) employs sequential linear programming to study the behavior of an integrated gas and power system. Nevertheless, the main drawback of nonlinear approaches is that, when the problem is non-convex as the gas network optimization, no efficient method is available to derive global optimality (Geißler et al., 2012). One possible solution to overcome this adversity is to approximate nonlinear functions by piecewise linear segments utilizing mixed integer linear programming (MILP). For instance, Möller (2004) applies MILP for the steady-state optimization of gas flows and Mahlke et al. (2010), Domschke et al. (2011) use it for the dynamic situation. Nowadays, it is possible to state that MILP techniques are mature because they are fast, robust, and are able to solve problems with up to hundred thousands of variables (Geißler et al., 2012). Even so, constructing efficient MILP formulations is not a trivial task as certain attributes can significantly reduce the solver effectiveness.

The computational performance of an MILP model is impacted by its strength, size and the effect of branching on the formulation to satisfy integrality constraints (see Vielma (2013), Morales-Espana et al. (2012) and references therein). Fortunately, an MILP model can be constructed so that its linear programming (LP) relaxation automatically satisfies integrality. These formulations are denoted as *locally ideal* by Padberg (2000) and are strong from both the LP relaxation and integer feasibility perspective. Several locally ideal MILP formulations used to piecewisely linearize a nonlinear function have been proposed in the literature. A unified framework can be found in Vielma et al. (2010). These formulations differ from each other in the number of variables, constraints and the way nonlinear functions are computed. Although comparisons of some of these proposals have been previously carried out by Geißler et al. (2012), Vielma et al. (2010), Croxton et al. (2003), Keha et al. (2004) for different problems and case studies, their conclusions suggest that the performance of each technique is influenced by the structure, data and nature of the problem.

Accordingly, the main contributions of this paper are:

- Theoretical and computational comparisons of locally ideal piecewise MILP formulations to approximate the general flow equation in the gas network optimization. Benefits and drawbacks of each modeling technique are analyzed qualitatively and quantitatively from the point of view of computational efficiency, implementation requirements and applicability.

- Categorical conclusions are presented on which formulation performs best to linearly approximate the flow equation for the gas network optimization problem in dynamic or steady state conditions based on the development of different case studies.
- Since the accuracy of each linear approximation depends on how piecewise segments are designed, this paper also contributes with a goal programming method to construct a-priori the linear functions by minimizing the error of each approximation.

The reminder of the paper is organized as follows. Section 2 presents a practical formulation for the gas network optimization based on operating conditions. The problem to be solved is the short-term dispatch for both dynamic and steady states. Then, Section 3 reviews locally ideal MILP models that can be employed to approximate the proposed optimization problem. Furthermore, Section 4 formulates the goal programming method to construct a-priori the piecewise linear functions that minimize the error of the approximation. Section 5 analyzes the theoretical properties of each MILP approximation and illustrates their computational performance for the gas network optimization. Case studies are developed for three system configurations of different sizes and ten demand profiles. Finally, main conclusions and further work are drawn in Section 6 and the corresponding nomenclature is listed at the end.

2. Dynamic and Steady Gas Network Optimization

This section develops a practical formulation for the short-term gas network optimization. This problem represents the transportation process of natural gas from wells (or regasification terminals) to consumers. The gas flow modeling receives specific attention due to the nonlinear and non-convex functions driving the equations. Basically, gas is transported through pipelines and because of the friction with pipe walls some gas pressure is lost. Then, this pressure gradient is compensated by compressors guaranteeing the continuity of the service.

2.1. Objective function

The objective faced by a neutral operator is to route the gas flow through the network in order to meet demand in accordance with contractual obligations (Midthun, 2007). To this end, (*i*) production costs, (*ii*) storage costs and (*iii*) non-served gas costs are minimized as follows:

$$\min \sum_{t\in\mathcal{T}} \left(\sum_{w\in\mathcal{W}} \underbrace{C_w^G pg_{wt}}_{i} + \sum_{s\in\mathcal{S}} \underbrace{C_s^S fs_{st}^{out}}_{ii} + \sum_{n\in\mathcal{N}} \underbrace{C_n^D ng_{nt}}_{iii} \right). \tag{1}$$

Alternatively, or additionally, terms could be included straightforward in the objective function representing for example fuel consumed by compressors or deviations from contracted pressures or line pack goals.

2.2. Operating constraints

Gas production and nodal pressures are bounded by either physical characteristics or contracted amounts according to

$$\underline{W}_{wt} \leq pg_{wt} \leq \overline{W}_{wt} \quad \forall w, t, \tag{2}$$

$$\underline{P}_n \leq p_{nt} \leq \overline{P}_n \quad \forall n, t. \tag{3}$$

Inventory constraints and operative levels are formulated to represent the operation of gas storages as

$$\underline{S}_s \leq sl_{st} = sl_{s,t-\triangle t} + fs^{in}_{st} - fs^{out}_{st} \leq \overline{S}_s \quad \forall s, t. \tag{4}$$

Likewise, in- and out-flow rates limit the amount of gas that can flow in and out of a storage

$$fs^{in}_{st} \leq IR_s, \quad fs^{out}_{st} \leq OR_s \quad \forall s, t. \tag{5}$$

2.3. Network constraints

The following equation models the mass balance in the transportation network, i.e., what flows into a node must also flow out of the node.

$$\sum_{m \to n} \left(q^{out}_{nm,t} - q^{in}_{nm,t}\right) + \sum_{w \to n} pg_{wt} + \sum_{s \to n} \left(fs^{out}_{st} - fs^{in}_{st}\right) + ng_{nt} = L_{nt} \, \forall n, t. \tag{6}$$

Compressors are used to increase the pressure in a node. This paper borrows the simplified compressor modeling from Midthun (2007) to limit how much the gas can be compressed in a node according with a compressor factor Γ.

$$p_{mt} \leq \Gamma_{nm} \cdot p_{nt} \quad \forall (n, m) \in \text{Compressor}. \tag{7}$$

Regarding the dynamic of the gas transport in a pipe, it is described by a non-linear and non-convex system of partial differential equations (PDEs)(Mahgerefteh et al., 2006b): the continuity equation (8) representing the principle of conservation of mass; the momentum equation (9) describing the sum of all forces acting on the gas particles; and the energy equation (10) corresponding to the law of preservation of energy.

$$\frac{\partial(\rho v)}{\partial x} + \frac{\partial \rho}{\partial t} = 0, \tag{8}$$

$$\frac{\partial p}{\partial x} + G\rho \frac{\partial H}{\partial x} + \frac{F \rho v \left|v\right|}{2D} + \frac{\partial \rho v}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} = 0, \tag{9}$$

$$\frac{\partial}{\partial t}\left[\rho\left(u + \frac{v^2}{2}\right)\right] + \frac{\partial}{\partial x}\left[(\rho v)\left(H + \frac{v^2}{2}\right)\right] = \rho e - \rho v G \sin\alpha \tag{10}$$

These equations can be expressed in terms of pressure and volumetric flow rate, which are the variables measured in real systems. This transformation is carried out by using the thermodynamic state equation of gas (11) and considering that the volumetric flow rate in standard conditions is calculated as $\widetilde{q} = \frac{\pi}{4}\frac{D^2 \rho v}{\rho_0}$ (Menon, 2005, Michels and Nkeng, 1997).

$$p = \rho RTZ(p, T). \tag{11}$$

Real operating conditions of gas transmission pipelines allow the adoption of some simplification hypotheses required to be able to formulate a tractable optimization model for large transport networks. Regarding the momentum equation (9), the second term $G\rho\frac{\partial H}{\partial x}$, representing the force of gravity influenced by the slope of the pipeline, can be neglected by assuming horizontal pipelines (Dorin and Toma-Leonida, 2008). Moreover, the last two terms $\frac{\partial \rho v}{\partial t}$ and $\frac{\partial(\rho v^2)}{\partial x}$, describing the inertia and kinetic energy respectively, can be simplified since they contribute with less than 1% to the solution of the equation under normal conditions (Dorin and Toma-Leonida, 2008). With respect to the energy equation (10), it can be neglected by assuming a constant temperature (Domschke et al., 2011). Dorin and Toma-Leonida (2008) shows that flow deviations rank under 2% in comparison with values calculated without this assumption. Finally, the compressibility factor $Z(p, T)$ from the state equation (11), which indicates the deviation of a real gas from the ideal gas as a function of pressure and temperature, can be tuned and treated as a constant for transport networks as shown in Midthun (2007). Under these assumptions, PDEs result in the continuity equation (8) and a reduced momentum equation composed by two terms: the pressure gradient $\frac{\partial p}{\partial x}$ and the friction force $\frac{F\rho v|v|}{2D}$. The friction factor F is a parameter that can be calculated by the Colebrook-White equation presented in Menon (2005).

PDEs must be discretized in time and space in order to obtain a system of differential algebraic equations. By applying a finite difference scheme, the planning horizon is divided equidistantly in $\triangle t$ time steps and the pipe is divided in segments of length $\triangle x$. Figure 1 depicts how pressures (p_n, p_m) and in- and outflow $(q^{in}_{nm}, q^{out}_{nm})$ variables are defined at each pair of nodes while average pipe pressure $(\widetilde{p}_{nm})$, flow $(\widetilde{q}_{nm})$ and mass $(\widetilde{m}_{nm})$ are assumed to be at the middle of the pipe. In practice, time periods and cell's lengths can be as short as required allowing the model to be solved with sufficient accuracy.

Finally, simplified and discretized continuity and momentum equations, expressed in terms of pressure and volumetric flow rates, result in (12) and (13) respectively.

$$q^{out}_{nm,t} - q^{in}_{nm,t} = K_1 \left(\widetilde{p}_{nm,t-\triangle t} - \widetilde{p}_{nm,t}\right) \ \forall (n, m), t, \tag{12}$$

$$\operatorname{sgn}(\widetilde{q}_{nm,t})\widetilde{q}^{2}_{nm,t} = K_2 \left(p^2_{nt} - p^2_{mt}\right) \ \forall (n, m), t, \tag{13}$$

where

$$\widetilde{p}_{nm,t} = \frac{p_{n,t} + p_{m,t}}{2} \quad \forall (n, m), t, \tag{14}$$

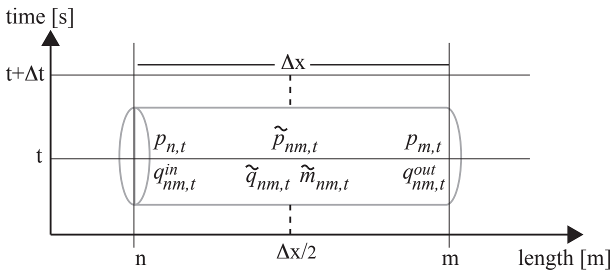

Figure 1: Time and space discretization

$$\widetilde{q}_{nm,t} = \frac{q^{out}_{nm,t} + q^{in}_{nm,t}}{2} \quad \forall(n,m), t, \tag{15}$$

and constants $K_1 = \frac{\pi}{4}\frac{X_{nm}D^2_{nm}}{RTZ\rho_0}$ and $K_2 = \left(\frac{\pi}{4}\right)^2 \frac{D^5_{nm}}{X_{nm}F_{nm}RTZ\rho_0^2}$. Because the continuity equation (12) allows the system to inject and withdraw different flow quantities in a pipe, a certain amount of gas mass, known as line pack, can be stored into the pipe. The line pack is defined as

$$\widetilde{m}_{nm,t} = K_1 \widetilde{p}_{nm,t} \quad \forall(n,m), t, \tag{16}$$

and the equation (12) represents the principle of conservation of mass in time. Usually, minimum levels of line pack are set for the last optimization period procuring for conditions for the next day's operation. The obtained momentum equation (13) corresponds to the general flow equation (or fundamental flow equation) utilized also for steady-state isothermal flows (Menon, 2005). This equation nonlinearly relates the gas flow rate with gas properties, pipe diameter, length, friction factor and upstream and downstream pressures. The sign function (sgn) represents the possibility to model bidirectional flows, i.e., flows that can be positive or negative depending on the pressure gradient. Figure 2 illustrates these nonlinear and non-convex features. The general flow equation can be replaced by other equations with similar structures that can be more accurate for certain flows and pipes, such as Weymouth, Panhandle A or Panhandle B (Menon, 2005, More, 2006).

In short, a practical and simplified gas network optimization model considering the system dynamics is given then by equations (1)-(7), (12) - (16).

For the steady-state case, the only difference is that the mass flow is assumed as constant in space, this is, $\frac{\partial q}{\partial x} = 0$. Consequently the continuity equation is dropped; the gas flowing into the pipe is the equal to the gas flowing out of it (equation (15) neglected) ; and the general flow equation can be rewritten as

$$\operatorname{sgn}(\widetilde{q}_{nm,t})\widetilde{q}^2_{nm,t} = K_2\left(ps_{nt} - ps_{mt}\right) \ \forall n, m, t, \tag{17}$$

where a simple change of variable $p^2_n = ps_n$ is carried out because now the pressure squared appears only in this equation (De Wolf and Smeers, 2000). Therefore, the steady-state formulation is given by equations (1)-(7) and (17).

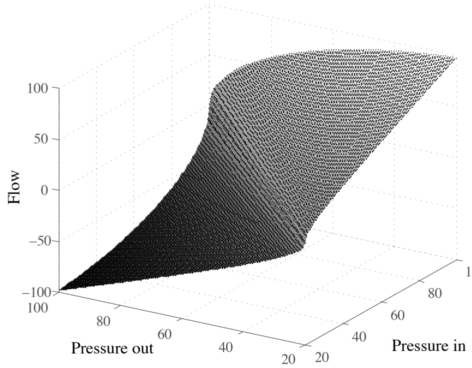

Figure 2: General flow equation function

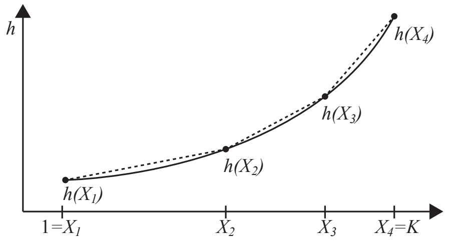

Figure 3: Approximation of a nonlinear separable function

Notice that both dynamic and steady-state models are nonlinear and non-convex because of general flow equations (13) and (17) respectively. The next section reviews piecewise MILP formulations that can be used to approximate these equations.

3. Piecewise Linear Approximations of the General Flow Equation

General flow equations proposed to model the gas flow in dynamic and steady-state conditions ((13) and (17) respectively) are separable functions constituted by nonlinear and non-convex univariate functions of the form $h(x) = x^2$ (Padberg, 2000). A nonlinear univariate function $h : D \to R$ defined on an interval $D := [A, B]$ can be piecewisely approximated as indicated in Figure 3. Each linear segment $P \in \mathcal{P} := \{1, \ldots, k-1\}$ is obtained by dividing the interval in a grid of points $A = X_1 \leq X_2 \leq \ldots \leq X_k = B$ with corresponding function values $h(x_i), i \in V := \{1, \ldots, k\}$.

Mainly three different strategies have been proposed to formulate piecewise linear models. They can be categorized according to the used given to binary variables and the way nonlinearities are approximated, as shown in Figure 4. The first two categories, convex combination and multiple-choice methods, formulate binary variables to select the segment in which the function lies. The

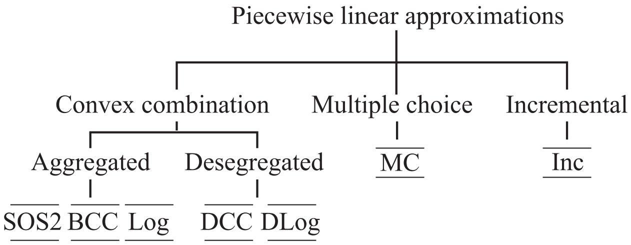

Figure 4: Classification of piecewise linear models

difference between them is that convex combination methods calculate the function value as a convex combination of the neighboring nodes and multiple-choice model does it according to the segment slope and intercept. Convex combination models are divided in two approximations: aggregated and desegregated methods. They differ in that desegregated proposals can effectively handle discontinuous functions. The third category is the incremental model. There, the function is calculated as the sum of incremental quantities in each segment and binaries represent segments used for the calculation. All of these methods are reviewed next for the case of univariate and separable functions as they can be applied to approximate general flow equations formulated in Section 2.

3.1. Convex combination models

In this approximation the nonlinear function is computed as a convex combination of at most two consecutive points of the partitioned interval.

3.1.1. Aggregated convex combination models (SOS2, BCC, Log)

These formulations introduce positive variables λ_i for each grid point X_i. The nonlinear function is represented as

$$h(x) \approx \sum_{i \in V} h(X_i)\lambda_i, \quad x = \sum_{i \in V} X_i \lambda_i, \quad \sum_{i \in V} \lambda_i = 1. \tag{18}$$

Additionally, λ-variables must fulfill the condition that at most two λs can be positive, and if two are positive, they must be consecutive. This condition is known as special ordered set of type two (SOS Type 2) (Möller, 2004). A first approximation to impose SOS Type 2 condition is to enforce nonlinearities algorithmically, directly in the branch-and-bound algorithm, by branching on the set of λ-variables. This implicit modeling technique has been proposed by Beale and Tomlin (1970) and is referred in this paper as *SOS2*.

A second way is to model SOS Type 2 condition explicitly via additional binary variables y_p, defined for $P \in \mathcal{P}$ segments of the interval. The idea is that when $y_p = 1$, only the λ-variables belonging to the segment p can take a value different than zero, as follows:

$$\sum_{p \in \mathcal{P}} y_p = 1, \tag{19}$$

$$\sum_{z=l+1}^{k-1} y_z \leq \sum_{z=l+1}^{k} \lambda_z \leq \sum_{z=l}^{k-1} y_z \ \forall l = 2..k-1, \tag{20}$$

$$\lambda_1 \leq y_1, \quad \lambda_k \leq y_{k-1}. \tag{21}$$

This method is referenced as the *basic convex combination (BCC)* model. It is composed by equations (18)- (21) and corresponds to the contribution made by Padberg (2000) to the traditional formulation developed by Dantzig (1960). Recently, Vielma and Nemhauser (2011) proposed a formulation that reduces logarithmically the number of binary variables of the BCC model. The idea is to impose the SOS Type 2 condition by fixing to zero disjoints sets of λ-variables in each side of a series of branching dichotomies. A family of dichotomies $\{\mathrm{L}_s,\mathrm{R}_s\}_{s\in S}$ is indexed by a finite set S, with L_s and R_s $\subset V(\mathcal{P})$ and the number of dichotomies given by $S(\mathcal{P}) := \{1,..,\lceil \log_2 |\mathcal{P}| \rceil\}$ (for the univariate case). For example, for the function depicted in Figure 3, $\mathcal{P} = \{[x_1,x_2],[x_2,x_3],[x_3,x_4]\}$ and the SOS Type 2 condition is forced by the following dichotomies: ($\lambda_3 = \lambda_4 = 0$ or $\lambda_1 = 0$) and ($\lambda_4 = 0$ or $\lambda_1 = \lambda_2 = 0$). Instead of formulating (19)-(21), this proposal models the following constraints with logarithmic number of binary variables:

$$\sum_{i\in L_s} \lambda_i \leq y_s, \quad \sum_{i\in R_s} \lambda_i \leq (1-y_s) \quad \forall s \in S. \tag{22}$$

The method developed by Vielma and Nemhauser (2011) is denoted as the *logarithmic (Log)* model and corresponds to equations (18) and (22). For more details about how to construct the dichotomies consult the cited reference.

3.1.2. Desegregated convex combination models (DCC, DLog)

These methods define “desegregated weights” λ_i^L and λ_i^R to each left and right end-points of the ith segment. The idea is that only when a binary variable $y_p = 1$, λ_i^L and λ_i^R for $i \in P$ can be different than zero. The nonlinear function is approximated as

$$h(x) \approx \sum_{i\in V} \left[h(X_{i-1})\lambda_i^L + h(X_i)\lambda_i^R\right], \tag{23}$$

$$x = \sum_{i\in V} \left[X_{i-1}\lambda_i^L + X_i\lambda_i^R\right]. \tag{24}$$

Again, the SOS Type 2 condition must be fulfilled for λ-variables. Accordingly, Sherali (2001) proposes to formulate

$$\sum_{p\in\mathcal{P}} y_p \ = \ 1, \quad \lambda_i^L + \lambda_i^R = y_i \ \forall i \in P. \tag{25}$$

This model is referred as *desegregated convex combination (DCC)* model and is composed by equations (23)-(25). Similarly to the Log model construction, Vielma and Nemhauser (2011) proposes also a formulation that reduces logarithmically the number of binary variables of DCC. The idea is to use a Gray code to define intervals of the piecewise linear function. Each segment in $\mathcal{P}$ is identified with a binary vector $\{0{,}1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$ through an injective function $B : \mathcal{P} \rightarrow \{0{,}1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$, where vectors $B(P)$ and $B(P+1)$ differ in at most one component for all $P \in \mathcal{P}$. As a result, the model uses $\lceil \log_2 |\mathcal{P}| \rceil$ binary variables $y \in \{0{,}1\}^{\lceil \log_2 |k-1| \rceil}$ to force the SOS Type 2 condition when $y = B(\mathcal{P})$. Instead of formulating (25), this proposal models

$$\sum_{i \in P} \lambda_i^L + \lambda_i^R = 1, \tag{26}$$

$$\sum_{p \in \mathcal{P}^+(B,l)} \sum_{i \in V(P)} \lambda_i \leq y_l, \quad \sum_{p \in \mathcal{P}^0(B,l)} \sum_{i \in V(P)} \lambda_i \leq (1 - y_l) \quad \forall l \in L(\mathcal{P}), \tag{27}$$

where $\mathcal{P}^+(B,l) := \{P \in \mathcal{P} : B(P)_l = 1\}$, $\mathcal{P}^0(B,l) := \{P \in \mathcal{P} : B(P)_l = 0\}$ and $L(\mathcal{P}) := \{1, .., \lceil \log_2 |\mathcal{P}| \rceil\}$. The *logarithmic desegregated convex combination (DLog)* model developed by Vielma and Nemhauser (2011) is composed then by equations (23), (24), (26) and (27). For more details about how to construct injective functions see Vielma and Nemhauser (2011) and Geißler et al. (2012).

3.2. Multiple choice model (MC)

The formulation studied in Balakrishnan and Graves (1989), among others, introduces continuous δ_p and binary y_p variables for $P \in \mathcal{P}$. Continuous variables represent the total load of the segment P if the function value lies in this segment. Also, binary variables indicate the chosen segment. If each linear function has the form of $g(x_p) = M_p x_p + B_p$ the nonlinearity is approximated as

$$h(x) \approx \sum_{p \in \mathcal{P}} M_p \delta_p + B_p y_p, \tag{28}$$

$$y_p X_i \leq \delta_p \leq y_p X_{i+1} \; \forall i \in P, P \in \mathcal{P} \tag{29}$$

$$\sum_{p \in \mathcal{P}} y_p \leq 1, \quad \sum_{p \in \mathcal{P}} \delta_p = x. \tag{30}$$

3.3. Incremental model (Inc)

Similarly as MC, Markowitz and Manne (1957) formulates continuos δ_p and binary y_p variables for each $P \in \mathcal{P}$. In this case, continuous variables represent the portion (or load) of each segment. Binary variables force that if an interval is chosen, then all intervals to its left must be completely used. The nonlinear function is approximated as

$$h(x) \approx h(x_1) + \textstyle\sum_{i \in \mathcal{P}} \left(h(x_{i+1}) - h(x_i)\right) \delta_i, \tag{31}$$

$$x = x_1 + \textstyle\sum_{i \in \mathcal{P}} \left(x_{i+1} - x_i\right) \delta_i, \tag{32}$$

$$\delta_{i+1} \leq y_i, \; \forall i \in \mathcal{P} - 1 \quad y_i \leq \delta_i, \; \forall i \in \mathcal{P} - 1, \tag{33}$$

$$0 \leq \delta_i \leq 1 \; \forall i \in \mathcal{P}. \tag{34}$$

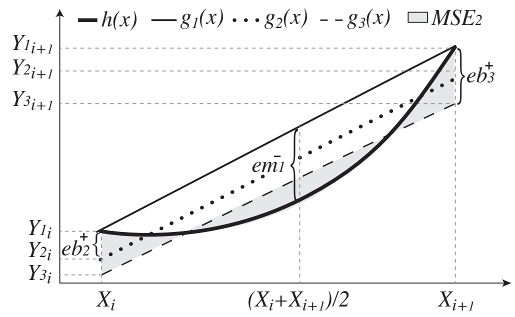

Figure 5: Linear approximations

Equations (33) are the so-called filling conditions: if $\delta_i > 0$ with $2 \leq i \leq k-1$ then $\delta_j = 1$ for $1 \leq j < i$.

4. Error Estimation

Commonly, a nonlinear function $h(x)$ is linearly approximated by straight line segments connecting points $(x_i, h(x_i))$ placed over the function, as shown in Figure 3. These lines always lie above the curve for functions of the form $h(x) = x^2$, such as those in the general flow equation. For these approximations, the maximum linear interpolation error $E = \max(eb_i^+, eb_i^-, em_i^+, em_i^-)$ calculated as the maximum deviation between the straight line $g(x)$ and the function $h(x)$ at vertices (eb_i^+, eb_i^-) and midpoints (em_i^+, em_i^-), corresponds to the evaluation of $|h(x) - g(x)|$ at the midpoint of the segment. This kind of representations is accurate when the function value lies in the neighborhood of the endpoints but it lacks of accuracy when the function lies near the midpoint. Under this assumption, the error in the piecewise linear fit decreases as $x_{i+1} - x_i$ decreases and the number of data points increases. However, as shown in Section 5, more data points mean more variables and more constraints, which affects the performance of the formulation. Such systematic errors can be minimized by defining line segments between points (x_i, y_i), where y_i is not necessary equal to $h(x_i)$. This situation is evidenced by the example presented in Figure 5. Besides the traditional linearization $g_1(x)$, other two possible approximations $g_2(x)$ and $g_3(x)$ are depicted. Both the error E and mean squared error MSE between $h(x)$ and $g_1(x)$ are greater than $h(x)$ and $g_2(x)$ and $g_3(x)$. However, while $g_2(x)$ has lower interpolation errors $g_3(x)$ offers a lower mean squared error.

By using goal programming, an optimization problem is formulated below to calculate a-priori the pair of points (x_i, y_i) that define a predefined number of straight line segments. The idea is to minimize the MSE limiting the maximum interpolation error E between straight line segments and the function at vertices and midpoints according with an error goal G:

$$\min \quad MSE \tag{35}$$

$$\text{s.t.} \quad E \leq G. \tag{36}$$

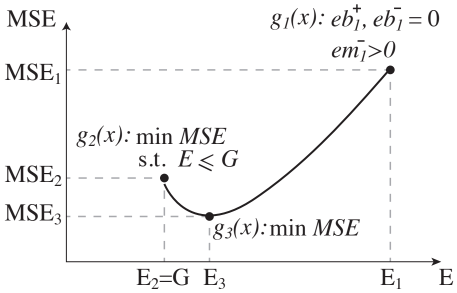

Figure 6: Mean squared error (MSE) vs Maximum interpolation error (E)

Figure 6 illustrates the tradeoff between MSE and E continuing with the functions depicted in Figure 5. The maximum MSE (MSE_1) corresponds to the traditional approximation $g_1(x)$ in which the error interpolation E_1 is null at vertices but is large at the middle. The minimum MSE (MSE_3) corresponds to the straight line segment $g_3(x)$. This option deviates more at vertices than $g_2(x)$ ($E_3 > E_2$) which gives up in the MSE in order to guarantee a given error goal (G). In this example $g_2(x)$ would be the solution to the formulation given by (35)-(36). This optimization problem is generalized for p segments and i vertices as follows:

$$\min\, z = \sum_{p \in \mathcal{P}} MSE_p, \tag{37}$$

where

$$MSE_p = \int_{x_i}^{x_{i+1}} \left[h(x) - g_p(x_i)\right]^2 dx, \quad \forall p, i \in \mathcal{P}, \tag{38}$$

$$h(x) = Kx^2, \tag{39}$$

$$g_p(x_i) = m_p x_i + b_p \quad \forall p, i \in \mathcal{P}, \tag{40}$$

$$m_p = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \quad \forall p, i \in \mathcal{P}, \tag{41}$$

$$b_p = y_i - m_p x_i \quad \forall p, i \in \mathcal{P}. \tag{42}$$

This objective is subject to ordering constraints

$$x_1 \leq x_{i+1} .. \leq x_k \quad \forall i \in \mathcal{P}, \tag{43}$$

$$y_1 \leq y_{i+1} .. \leq y_k \quad \forall i \in \mathcal{P}, \tag{44}$$

where $x_1 = A$, $x_k = B$ to ensure that the total function is covered. Finally, the error goal is ensured by

$$eb_i^+, eb_i^-, em_i^+, em_i^- \leq G \quad \forall i \in \mathcal{P}, \tag{45}$$

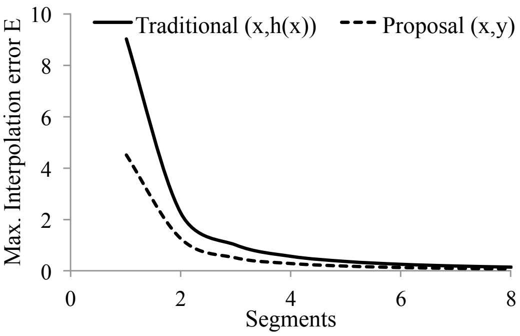

Figure 7: Comparison of interpolation error E

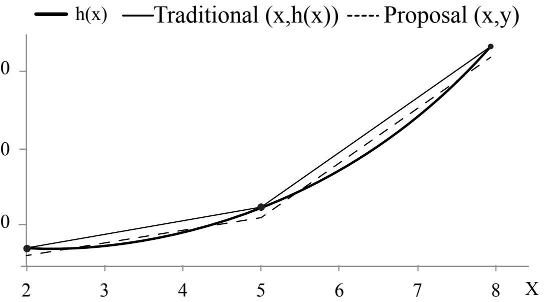

Figure 8: Comparison of linear approximations for two segments

where

$$eb_i^+ - eb_i^- = Kx_i^2 - m_p x_i - b_j \quad \forall p, i \in \mathcal{P}, \tag{46}$$

$$em_i^+ - em_i^- = K(\frac{x_{i+1}+x_i}{2})^2 - m_p(\frac{x_{i+1}+x_i}{2}) - b_p \quad \forall p, i \in \mathcal{P}. \tag{47}$$

Equations (45)-(47) are formulated generically for concave or convex functions. The optimization problem defined by (37)-(47) can be easily run iteratively to find the lower number of segments P that satisfies the error goal G. Given the nonlinear behavior of this model, a good start solution must be provided. A feasible alternative is to start from the traditional approximation in which the points correspond to $(x_i, h(x_i))$.

Figures 7 and 8 illustrate the benefits of the proposal with respect to the traditional approach for the function $h(x) = x^2,\ x \in [2, 8]$. Initially, Figure 7 shows how errors for different number of segments are reduced at least by a half without increasing the number of segments. Then, Figure 8 presents the adjustment of proposed straight lines when the function is linearized by two segments compared with the traditional approach.

5. Comparison of Piecewise Linear Formulations

This section compares MILP models for piecewise linear functions reviewed in Section 3. Initially, a theoretical analysis is developed in order to study the properties of tightness and compactness for each MILP approximation. Thereupon, the computational performance of each model is illustrated when linear methods are applied for the gas network optimization in both dynamic and steady-state conditions.

5.1. Theoretical analysis

The tightness and compactness of each alternative provide an idea of the efficiency of each formulation. The tightness represents how strength is the LP relaxation of each method. This attribute defines the search space (relaxed feasible region) that the solver requires to explore in order to determine the (optimal integer) solution (Morales-Espana et al., 2012). For an univariate and separable function, it is possible to state that:

- All formulations from Section 3 are locally ideal, i.e., all vertices of the corresponding LP relaxation comply with integrality requirements (Padberg, 2000). This theorem has been proved by Padberg (2000) for BCC and Inc models; by Sherali (2001) for DCC; by Keha et al. (2004) for SOS2; by Croxton et al. (2003) for MC and by Vielma et al. (2010) for Log and Dlog.

- All formulations from Section 3 are sharp (convex hull), i.e., LP relaxations of all formulations approximate the feasible region by its lower convex envelope, which is the tightest convex underestimator of the function. This statement is proved in Vielma et al. (2010) by demonstrating that any locally ideal formulation is sharp.

- All formulations from Section 3 give the same LP bound. This theorem is proven by Croxton et al. (2003), Vielma et al. (2010) from definitions of sharpness and convex envelopes.

Nevertheless, while sharp formulations are the strongest with regards to LP relaxation bounds, they can be weak with respect to finding optimal or good quality integer feasible solutions (Vielma, 2013). As a consequence, the compactness property (problem size) must be also analyzed. The size of the formulation is useful to estimate the solution speed of the LP relaxation. Table 1 presents the number of constraints and variables required for each MILP method to approximate an univariate and separable nonlinear function, in terms of the number of vertices V and linear segments $\mathcal{P}$. Specifically, Figure 9 illustrates the number of binary variables formulated by each method. Notice that great savings offered from logarithmic models are relevant for large number of segments $\mathcal{P}$. This observation can also be extended to the number of constraints and in general to the formulation size.

Nonetheless, theoretical comparisons are not enough to conclude which method performs best for the gas network optimization. Given the complexities of state

Table 1: Sizes of piecewise formulations for univariate functions

Model	Constraints	Cont. var.	Binary var.
SOS2	3	V	0
BCC	$6+2(\mathcal{P}-2)$	V	$\mathcal{P}$
Log	$3+2\lceil\log_2\|\mathcal{P}\|\rceil$	V	$\lceil\log_2\|\mathcal{P}\|\rceil$
DCC	$3+\mathcal{P}$	$2\mathcal{P}$	$\mathcal{P}$
DLog	$3+2\lceil\log_2\|\mathcal{P}\|\rceil$	$2\mathcal{P}$	$\lceil\log_2\|\mathcal{P}\|\rceil$
MC	$3+2\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
Inc	$2\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}-1$

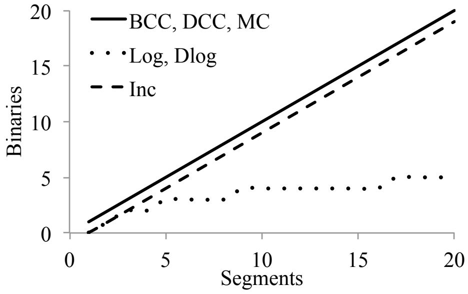

Figure 9: Number of binary variables

of the art solvers, it is hard to predict with high accuracy the performance of each formulation. As a result, computational comparisons presented next aims to provide additional information in this regard.

5.2. Computational analysis

In this section the general flow equation for gas network optimization problems is piecewisely approximated by MILP models presented in Section 3. For dynamic conditions three univariate functions, linked by the general flow equation (13), are individually approximated. The flow squared ($\widetilde{q}^2_{nm,t}$) is linearized for each pipe and each period. Similarly, in- and outlet pressure squared (p^2_{nt}, p^2_{mt}) are approximated for each node and period. On the other hand, for the steady state formulation only one univariate function (the flow squared $\widetilde{q}^2_{nm,t}$) is linearized according with (17). Different case studies illustrate the computational performance of each piecewise MILP formulation for both dynamic and steady-state systems:

- Case 1 (from Liu et al. (2009)): 7 nodes, 6 pipelines, 2 sources, 3 loads, 1 compressor and 1 storage.
- Case 2 (from Li et al. (2003)): 14 nodes, 12 pipelines, 1 source, 5 loads and 4 compressors.
- Case 3 (Belgian network from De Wolf and Smeers (2000)): 20 nodes, 24 pipelines, 2 sources, 8 loads, 3 compressors and 4 storages.

Table 2: Metrics for the dynamic system - Case 3. Percentage increase with respect to Inc.

	Inc	**SOS2**	**BCC**	**Log**	**DCC**	**DLog**	**MC**
Const.	4 090	-36%	25%	0%	0%	0%	36%
Real	3 189	15%	15%	15%	46%	46%	0%
Binary	984	-100%	50%	-24%	50%	-24%	50%
Nonzero	12 103	-8%	77%	26%	42%	79%	42%
Nodes	140	790%	15 507%	5 787%	44 260%	10 920%	40 809%
Iterations	32 045	134%	6 941%	2 971%	10 997%	8 797%	62 332%

All system data are available online at https://db.tt/OhrvNg8h. Case studies are solved in hourly periods for 10 load profiles and different optimization horizons. For the dynamic system, periods of 24 hours are set for Cases 1 and 2 and 12 hours are optimized for Case 3. For the steady state, Cases 1 and 2 solve the dispatch for 1 week and Case 3 does it for 2 weeks. Linear segments ranging from 3 to 8 are modeled to guarantee a maximum error of 1% in the approximation of each variable (pressures and flows), according with their operative ranges. Experiments are run on an Intel-i5 3.2-GHz personal computer with 8 GB of RAM memory by using CPLEX 12.5. Case studies are solved until they hit the time limit of 1 hour or until they reach an optimality tolerance of 1E-3 by using solvers' defaults.

Computationally, the tightness of each method is evaluated by the integrality gap (Morales-Espana et al., 2012). This parameter is defined as $(Z_{MILP} - Z_{LP})/Z_{MILP}$, where Z_{LP} is the optimal value of the (initial) relaxed LP problem and Z_{MILP} is the integer solution found when the MILP problem is completely solved. Nonetheless, the integrality gap is equal for all formulations reviewed in this paper, so they are all equally tight. All approximations give the same LP bound, as shown in Section 5.1, and result in the same optimal solution because they describe the same MILP problem. However, as discussed in Vielma (2013), equality in the LP relaxation bound does not necessarily imply equality on the LP bound obtained at the root node by CPLEX as this includes preprocessing and cuts. On the other hand, the compactness and the effect of the branch-and-bound are analyzed by means of the metrics shown in Tables 2 and 3 for specific case studies. The problem size (compactness) is measured by the number of constraints (Const.), continuous variables (Real), binary variables (Binary) and nonzero elements. Additionally, the number of explored nodes and iterations required to find the optimal solution represent the branching process. According to Morales-Espana et al. (2012), the number of nodes must be carefully taken into account because formulations more compact may be able to explore more nodes in shorter times than larger formulations because LP problems are solved faster.

Tightness and compactness metrics are utilized to explain solve times obtained for each formulation. Figures 10 and 11 use box-and-whisker plots to graphically depict through quartiles solve times (in logarithmic scale) for dynamic and steady-state networks respectively. Also, a black circle indicates the

Table 3: Metrics for the steady-state system - Case2. Percentage increase with respect to Inc.

	Inc	**SOS2**	**BCC**	**Log**	**DCC**	**DLog**	**MC**
Const.	24 865	-16%	32%	16%	16%	16%	49%
Real	19 824	20%	20%	20%	41%	41%	0%
Binary	4 032	-100%	100%	0%	100%	0%	100%
Nonzero	64 849	2%	39%	27%	45%	57%	45%
Nodes	26 168	3%	26%	28%	18%	24%	13%
Iterations	477	-1%	46%	-3%	3%	3%	3%

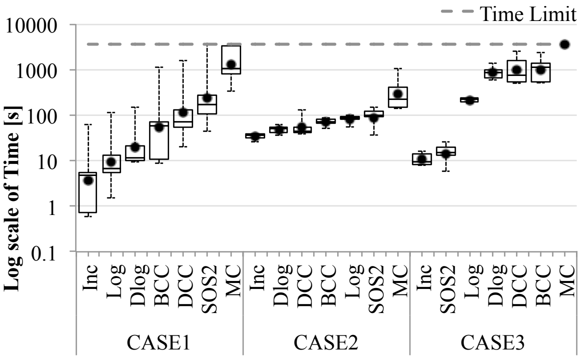

Figure 10: Solve times for the dynamic state

geometric mean for each case study. This mean is utilized because of its robustness for outliers and its good performance for sets of ratios (Morales-Espana et al., 2012).

For all experiments Inc model presented by far the best performance. For the dynamic case, Inc solved on average 435% faster than Log, the second fastest formulation, and 9534% faster than MC, the slowest one. The poor performance of MC was more relevant in Cases 1 and 3 in which it was not even able to find an optimal solution after exploring the enormous quantity of 40 809% more nodes than Inc in 62 332% more iterations. From these results, it is relevant to highlight the unstable behavior of SOS2. This method behaved slowly for Cases 1 and 2 but was notably fast solving the Case 3. By analyzing results presented in Table 2, Inc and SOS2 executed better for Case 3 because were able to solve the problem by exploring fewer nodes with lower number of iterations than other formulations. This pattern remained for the Inc model explaining its well overall performance, but was very unsteady for SOS2. Even SOS2 yields the smallest formulation, it was not always the fastest model because usually required a lot of iterations to find the optimal solution. As discussed in Vielma et al. (2008), it seems that the reason for this functioning is more of an implementation issue than a property of the SOS2 formulation. As a result, Mahlke et al. (2010) develops a customized branch-and-cut algorithm able to handle more efficiently SOS Type 2 conditions improving SOS2 performance significantly.

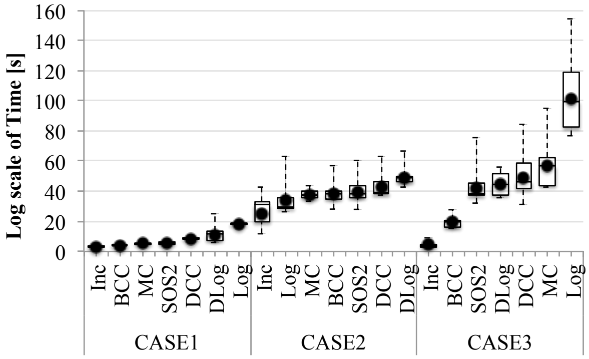

Figure 11: Solve times for the steady state

The pattern obtained in dynamic case studies for aggregated and desegregated convex combination formulations is discussed next. Usually Log and DLog approximations behaved better than BCC and DCC respectively because of their ability to reduce the number of binary variables. This saving is relevant for large number of linear segments and systems, as illustrated in Tables 2 and 3 and Figure 9 . On average, Log performed 209% better than DLog because the model besides being smaller (by formulating less continuous variables and nonzero elements) required less iterations to find the solution. With respect to BCC and DCC, their execution only differed on average 8%. While BCC is more compact with respect to the number of variables, DCC requires less constraints and nonzero elements as presented in Table 2.

Regarding the steady state optimization, Inc model also presented the best performance, especially for Cases 2 and 3. On average Inc solved 83% faster than BCC, the second fastest formulation, and 366% faster than Log, the slowest one. Solve times for Cases 1 and 2 were relatively similar. Table 3 evidences how the number of iterations for all formulations was close. As a consequence, differences on the functioning of each model were due to the time required to solve LP relaxations, which is highly influenced by the problem size. Unlike the dynamic optimization, for the steady state SOS2 had a stable performance being on average the third fastest method; and MC improved its positioning executing similarly to DCC.

To conclude, Table 4 presents a qualitative comparison of different formulations. Firstly, it categorizes the implementation difficulty for each method according with particular requirements and formulation size (in terms of number of variables and constraints) and lastly it highlights the most relevant features for each method.

6. Conclusions and Further Work

This paper compared theoretically and computationally piecewise MILP formulations capable of approximating the general flow equation for the gas net-

Table 4: Qualitative comparison of MILP formulations

Model	Implementation	Features
SOS2 (18)	Easiest implementation. Most compact model. Solver requires to support SOS Type 2 constraints.	No binaries. High solver dependency. Unstable performance for dynamic case.
BCC (18)-(21)	Easy implementation: no special requirements. More constraints.	Slow LP solve. Good performance for steady state.
Log (18)-(22)	Complex implementation: construction of dichotomies. Small formulation size.	Fast LP solve. Good for large $\mathcal{P}$. Good performance for dynamic cases but bad for steady state.
DCC (23)-(25)	Easy implementation: no special requirements. More variables.	Slow LP solve. Useful to model discontinuities.
DLog (23),(24), (26),(27)	Moderate implementation: construction of injective functions. Small formulation size.	Fast LP solve. Good for large $\mathcal{P}$ and to model discontinuities.
MC (28)-(30)	Easy implementation, but it needs slopes and intercepts. Big formulation size.	Slow LP solve. Bad performance for dynamic state.
Inc (31)-(34)	Easy implementation: no special requirements. Medium formulation size.	Best performance for dynamic and steady-state systems.

work optimization problem in both dynamic and steady-state conditions. To this end, a practical formulation was initially developed optimizing the gas network in the short term. Then, locally ideal piecewise MILP models that can be applied to linearize gas flows were reviewed. Furthermore, since the accuracy of each linear approximation significantly depends on the construction of the segments, this paper also proposed a goal programming method to optimally construct a-priori linear functions. Under this method the error of each piecewise approximation is notably reduced without introducing more linear segments. Theoretically, all piecewise linear models shared the same level of tightness since they are locally ideal. However, the difference between approximations is their compactness due to the way constraints are formulated. Computational results illustrated how the Inc model, based on incremental weights or loads, presented the best performance to solve the gas network optimization in both dynamic and steady-state conditions. This is a traditional formulation easy to implement and with stable performances. For dynamic systems three nonlinearities were individually linearized: the flow squared and in- and outlet pressure squared. For this condition Inc model was on average 435% faster than Log, the second fastest formulation, and 9534% faster than MC, the slowest one. Regarding the steady state, only the flow squared is linearly approximated. In this condition Inc was on average 83% faster than BCC, the second fastest formulation, and 366% faster than Log, the slowest one. For small case studies all models performed similarly. Nevertheless, for complex and large systems Inc formulation overcame by far other models. These conclusions are based on case studies developed for different network configurations, sizes and levels of complexity.

Just as this paper addressed the linearization of the general flow equation, further research could be focused in the approximation of other nonlinear functions that are also present in the gas network optimization. Initially, the modeling of gas storages may be improved by applying the same piecewise linear methods presented before since injection and withdrawal rates are nonlinearly related with the storage level by separable functions. Additionally, a closer approximation of compressors requires the modeling of non-separable functions composed by in- and outlet pressures. This kind of functions could be linearized through multi-variate techniques that result from the extension in n-dimensions of MILP formulations reviewed in Section 3. Initial formulations in this regard have been developed by Geißler et al. (2012), Mahlke et al. (2010), Möller (2004).

Nomenclature for the Gas Modeling

Indexes and Sets

Upper-case and greek letters are used for denoting parameters and sets. Lower-case letters denote variables and indexes.

$n, m \in \mathcal{N}$	Gas nodes, running from 1 to N.
$s \in \mathcal{S}$	Gas storage facilities, running from 1 to S.
$t \in \mathcal{T}$	Time periods, running from 1 to T.
$w \in \mathcal{W}$	Gas wells, running from 1 to W.

Constants

C^{G}_w,C^S_s,C^D_n	Gas production, storage and non-served cost [\$/Sm^3].
D_{ij}	Pipeline diameter [m].
F_{ij}	Pipeline friction coefficient [-].
G	Gravity force [m/s^2].
H	Pipeline height [m].
L_{nt}	Gas load in node n [Sm^3/s].
IR_s, OR_s	Injection/Withdrawal storage rates [Sm^3/s].
$\overline{P}_n$,$\underline{P}_n$	Max/Min nodal pressure [bar].
R	Specific gas constant [m^{3}bar/kgK].

$\overline{S}_s, \underline{S}_s$	Max/Min storage level [Sm^3].
T	Temperature [K].
$\overline{W}_{wt}$,$\underline{W}_{wt}$	Max/Min gas output of each well [Sm^3].
X_{nm}	Pipeline length (spatial coordinate x) [m].
Z	Compressibility factor [-].
α	Pipe angle with the horizontal [rad].
ρ	Density of gas [kg/m^3].
ρ_0	Density of gas at standard conditions [kg/m^3].
Γ_{nm}	Compression factor [-].

Variables

e	rate of heat transfer per unit time and unit mass of the gas [W/kg].
fs_{st}^{in},fs_{st}^{out}	Injection/Withdrawal storage flow [Sm^3/s].
$\widetilde{m}_{nm,t}$	Average line pack [Sm^3].
ng_{nt}	Non-served gas [Sm^3/s].
p_{nt}	Nodal pressure [bar].
ps_{nt}	Nodal pressure squared [bar].
$\tilde{p}_{nm,t}$	Average pipeline pressure [bar].
pg_{wt}	Gas production [Sm^3/s].
$q_{nm,t}^{in}$,$q_{nm,t}^{out}$	In- and outflow mass rate [Sm^3/s].
$\widetilde{q}_{nm,t}$	Average mass flow rate [Sm^3/s].
sl_{st}	Gas storage level [Sm^3].
u	Specific internal energy [J/kg].
v	flow velocity [m/s].

References

Balakrishnan, A., Graves, S. C., Mar. 1989. A composite algorithm for a concave-cost network flow problem. Networks 19 (2), 175–202.

Beale, E. M. L., Tomlin, J. A., 1970. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. OR 69 (447-454), 99.

Chaudry, M., Jenkins, N., Strbac, G., Jul. 2008. Multi-time period combined gas and electricity network optimisation. Electric Power Systems Research 78 (7), 1265–1279.

Croxton, K. L., Gendron, B., Magnanti, T. L., 2003. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. Management Science 49 (9), 1268–1273.

Dantzig, G. B., 1960. On the significance of solving linear programming problems with some integer variables. Econometrica, Journal of the Econometric Society, 30–44.

De Wolf, D., Smeers, Y., Nov. 2000. The gas transmission problem solved by an extension of the simplex algorithm. Management Science 46 (11), 1454–1465.

Domschke, P., Geißler, B., Kolb, O., Lang, J., Martin, A., Morsi, A., Nov. 2011. Combination of nonlinear and linear optimization of transient gas networks. INFORMS Journal on Computing 23 (4), 605–617.

Dorin, B. C., Toma-Leonida, D., 2008. On modelling and simulating natural gas transmission systems (part i). Journal of Control Engineering and Applied Informatics 10 (3), 27–36.

Ehrhardt, K., Steinbach, M. C., 2005. Nonlinear optimization in gas networks. Springer.

Geißler, B., Martin, A., Morsi, A., Schewe, L., 2012. Using piecewise linear functions for solving MINLPs. In: Mixed Integer Nonlinear Programming. Springer, p. 287–314.

Keha, A. B., de Farias, I. R., Nemhauser, G. L., Jan. 2004. Models for representing piecewise linear cost functions. Operations Research Letters 32 (1), 44–48.

Li, Q., An, S., Gedra, T. W., 2003. Solving natural gas loadflow problems using electric loadflow techniques. In: Proc. of the North American Power Symposium.

Liu, C., Shahidehpour, M., Fu, Y., Li, Z., Aug. 2009. Security-constrained unit commitment with natural gas transmission constraints. IEEE Transactions on Power Systems 24 (3), 1523–1536.

Mahgerefteh, H., Oke, A., Atti, O., Mar. 2006a. Modelling outflow following rupture in pipeline networks. Chemical Engineering Science 61 (6), 1811–1818.

Mahgerefteh, H., Oke, A. O., Rykov, Y., Aug. 2006b. Efficient numerical solution for highly transient flows. Chemical Engineering Science 61 (15), 5049–5056.

Mahlke, D., Martin, A., Moritz, S., Aug. 2010. A mixed integer approach for time-dependent gas network optimization. Optimization Methods and Software 25 (4), 625–644.

Markowitz, H. M., Manne, A. S., 1957. On the solution of discrete programming problems. Econometrica: journal of the Econometric Society, 84–110.

Menon, E. S., 2005. Gas pipeline hydraulics. Taylor & FRancis, Boca Raton, FL.

Michels, H. J., Nkeng, G. E., Dec. 1997. Simulation of transient pipeline flow by a reversed shock-tube technique. Chemical Engineering Science 52 (23), 4303–4316.

Midthun, K. T., 2007. Optimization models for liberalized natural gas markets. Norwegian University of Science and Technology, Faculty of Social Science and Technology Management, Department of Sociology and Political Science.

Möller, M., 2004. Mixed integer models for the optimisation of gas networks in the stationary case. Ph.D. thesis, TU Darmstadt.

Morales-Espana, G., Latorre, J. M., Ramos, A., 2012. Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment. IEEE Transactions on Power Systems, 1–1.

More, A. A., Aug. 2006. Analytical solutions for the colebrook and white equation and for pressure drop in ideal gas flow in pipes. Chemical Engineering Science 61 (16), 5515–5519.

Padberg, M., 2000. Approximating separable nonlinear functions via mixed zero-one programs. Operations Research Letters 27 (1), 1–5.

Selot, A., Kuok, L. K., Robinson, M., Mason, T. L., Barton, P. I., Feb. 2008. A short-term operational planning model for natural gas production systems. AIChE Journal 54 (2), 495–515.

Sherali, H. D., 2001. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. Operations Research Letters 28 (4), 155–160.

Vielma, J. P., 2013. Mixed integer linear programming formulation techniques.

Vielma, J. P., Ahmed, S., Nemhauser, G., 2010. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. Operations research 58 (2), 303–315.

Vielma, J. P., Keha, A. B., Nemhauser, G. L., 2008. Nonconvex, lower semicontinuous piecewise linear optimization. Discrete Optimization 5 (2), 467–488.

Vielma, J. P., Nemhauser, G. L., 2011. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Mathematical Programming 128 (1-2), 49–72.

Wang, M., Xu, Q., Jun. 2014. Optimal design and operation for simultaneous shale gas NGL recovery and LNG re-gasification under uncertainties. Chemical Engineering Science 112, 130–142.