

COMBINATION OF NONLINEAR AND LINEAR OPTIMIZATION OF TRANSIENT GAS NETWORKS *

PIA BALES[†], BJÖRN GEIBLER, OLIVER KOLB[†], JENS LANG[†], ALEXANDER
MARTIN[†], AND ANTONIO MORSI[†]

Abstract. In this paper, we study the problem of technical transient gas network optimization, which can be considered as a minimum cost flow problem with a nonlinear objective function and additional nonlinear constraints on the network arcs. Applying an implicit box scheme to the isothermal Euler equation, we derive a mixed integer nonlinear program. This is solved by means of a combination of (i) a novel mixed integer linear programming approach based on piecewise linearization and (ii) a classical sequential quadratic program extended with a continuous treatment of combinatorial constraints. Numerical experiments for example problems as well as for a real-life application show that best optimal control solutions can be achieved by a combination of both approaches.

Key words. gas networks, optimal control, mixed integer programming, nonlinear programming

AMS subject classifications. 76N25, 90C11, 90C30, 90C90

1. Introduction. Nowadays, natural gas has become an increasingly important energy resource. To reach the customers, large quantities of natural gas are transported through transmission pipelines. Gas networks operate at high pressures and utilize a series of compressor stations to compensate the friction forces and to move the gas over long distances. The compressor stations fuel costs are the most significant operation costs of transmission pipelines. The optimal operating conditions that meet all contractual obligations of flow and pressure at various points along the pipeline and minimize fuel usage are determined by deciding which compressors need to be run and what are the best operating conditions for these compressors. The large extent of gas networks, their high complexity, transient nature and inherent nonlinearities make the optimization of their management a difficult computational task.

The problem of technical transient gas network optimization deals with the challenge of how to optimize the gas flow and how to operate the compressors cost-efficiently such that all customers' demands are satisfied. Since a gas network basically consists of a set of compressors and valves connected by pipes, it can be adequately modelled through a directed finite graph. The gas dynamics in pipes are described by the Euler equations – a set of well-known hyperbolic partial differential equations. Given a detailed nonlinear model of the pipeline hydraulics and compressor characteristics, a discretization of the Euler equations in space and time leads to a complex mixed integer nonlinear (and even non-convex) program (MINLP). In addition, combinatorial constraints are necessary to model discrete processes like switching compressors and valves on and off.

There are three main approaches available to solve MINLPs. (i) A first way is to use algorithms from nonlinear continuous optimization, e.g., sequential quadratic programming [19, 20] and interior point methods [2]. These solution strategies are often able to find good optimal controls in short running time, but combinatorial

*This work was supported by the DFG (Deutsche Forschungsgemeinschaft) under the grants LA1372/5-1 and MA1324/4-1.

[†]Department of Mathematics, Technische Universität Darmstadt, Schlossgartenstr. 7, 64289 Darmstadt, Germany, (bales@mathematik.tu-darmstadt.de, geissler@mathematik.tu-darmstadt.de, kolb@mathematik.tu-darmstadt.de, lang@mathematik.tu-darmstadt.de, martin@mathematik.tu-darmstadt.de, morsi@mathematik.tu-darmstadt.de).

constraints cannot be handled efficiently in general and usually have to be given externally [21]. Another drawback is that they only guarantee to find locally optimal solutions. (ii) A second opportunity is to apply general purpose global optimization algorithms such as the branch-and-reduce strategy, which is implemented in the well-known software BARON [18]. However, general globally optimal solvers are currently not able to solve transient gas network problems of reasonable size as was confirmed by our test runs. Reasons are that they cannot exploit the special structure of the underlying nonlinear equations and they are very limited in choosing the search space without additional information at hand. (iii) A third, less known technique is to use mixed integer linear programming (MILP). Once all nonlinearities have been linearized and binary variables have been introduced to reflect discrete switching processes, efficient algorithms are available to solve MILPs to global optimality. Such methods have already been successfully applied to steady state optimization of gas networks [10, 11] and more recently also to the transient case [12].

Other techniques to attack transient technical optimization include dynamic programming [3], simulated annealing [8] and hierarchical system theory [14]. The gas network optimization problem can also be formulated as a non-cooperative game, where compressors and sources are the players and communication is established through the network connectivity constraints. The solution is then given as a Nash-equilibrium found by an iterative algorithm [16, 17].

In this paper, we combine a global optimization algorithm from mixed integer linear programming and a classical optimization algorithm from sequential quadratic programming (SQP) extended by a continuous treatment of binary control variables. The main idea is to handle binary decisions by means of MILPs and to give attention to nonlinear physical laws within the SQP framework. We will see that by this combination very good solutions can be found, which could not have been obtained by using each of the methods solely. The paper is organized as follows. In Section 2, we derive our mathematical model. In Section 3, we present the MILP and explain how the nonlinearities can be approximated by piecewise linear functions to fit into a linear program. The nonlinear continuous SQP-based optimization is described in Section 4. In Section 5, we present numerical results for two academic example problems and one real-life application. We end with a summary and main conclusions in Section 6.

2. Mathematical Model. In the following we describe our mathematical model used to optimize transient gas flow in networks.

Network Model. We model a gas network by means of a directed finite graph $G = (\mathbb{V}, \mathbb{E})$. The set \mathbb{E} of edges, here also called segments, is partitioned into the set \mathbb{E}_C of compressors, the set \mathbb{E}_V of valves, and the set \mathbb{E}_P of pipes. The set \mathbb{V} of nodes consists of the set \mathbb{V}_P of intersection points of the segments, the set \mathbb{V}_S of sources, and the set \mathbb{V}_D of sinks. Sources are considered as gas delivering points and sinks reflect gas demands, specified by the quantities flow and pressure. We denote by δ_ν^- (δ_ν^+) the set of all indices of edges $e_i \in \mathbb{E}_P, i \in \mathbb{N}$, outgoing (ingoing) from (to) the node $\nu \in \mathbb{V}$. To allow description of physical state variables at the ends of edges, each node $\nu \in \mathbb{V}$ is associated with a family of intermediate vertices $\{\nu_i\}, i \in \delta_\nu^+ \cup \delta_\nu^-$, see Fig. 2.1. We shall use $e_i = \nu_i w_i$ to model a directed pipe in the gas network.

Gas Flow in Pipes. The gas flow in a pipe is governed by the system of Euler equations supplemented by a suitable equation of state. Since pipes in Germany are typically at least one meter beneath the ground, we can assume a nearly constant temperature

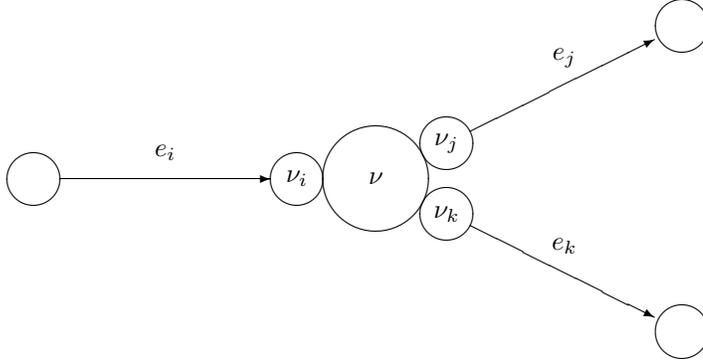


Fig. 2.1: Network Model. Suppose three directed pipes e_i , e_j , e_k , and one node ν representing an intersection are given as shown. Then, we have $\delta_\nu^+ = \{i\}$ for the set of indices of ingoing edges and $\delta_\nu^- = \{j, k\}$ for the set of indices of outgoing edges. The node ν is associated with the family of intermediate nodes $\{\nu_i, \nu_j, \nu_k\}$, where values of physical state variables are described.

$T = \bar{T}$. In such a situation, isothermal flow is an appropriate model. Taking into account a non-ideal gas behaviour, the Euler equations reduce to the continuity and the momentum equation, together with the equation of state. On each pipe $e \in \mathbb{E}_P$ of the network, we have for $t > t_0$

$$(2.1) \quad \partial_t \rho + \partial_x(\rho v) = 0,$$

$$(2.2) \quad \partial_t(\rho v) + \partial_x(\rho v^2) + \partial_x p = -g\rho \partial_x h - \frac{\lambda}{2D}\rho|v|v,$$

$$(2.3) \quad \rho = \frac{p}{z(p)R_0\bar{T}},$$

where (ρ, v, p) is the state vector consisting of the density, the flow velocity and the pressure of the gas, respectively. The two terms on the right-hand side of (2.2) describe the influence of gravity and friction. Here, g is the acceleration constant, $\partial_x h$ is the slope of the pipe, λ is the pipe friction value, and D is the diameter of the pipe. In our practical computations, all pipes are nearly horizontal. So, we will neglect the gravity term.

The friction factor λ is implicitly given by the Prandtl-Colebrook law,

$$(2.4) \quad \frac{1}{\sqrt{\lambda}} = -2 \log_{10} \left(\frac{2.51}{Re \sqrt{\lambda}} + \frac{k}{3.71 D} \right)$$

with the Reynolds number $Re = D\rho|v|/\eta$, where η is the dynamic viscosity of the gas, and with the roughness k of the pipe. The compressibility factor $z(p)$ in (2.3) is

defined by

$$(2.5) \quad z(p) = 1 + 0.257 \frac{p}{p_c} - 0.533 \frac{p T_c}{\bar{T} p_c},$$

where p_c and T_c are the pseudo-critical pressure and temperature, respectively. This formula from the American Gas Association works quite well for pressures up to 70bar. Due to the constant temperature \bar{T} , we get $z(p) = 1 + \alpha p$ with $\alpha < 0$, leading to $0 < z(p) < 1$. Typical values in gas networks are $p_c = 46.4512\text{bar}$, $T_c = 192.033\text{K}$, $\bar{T} = 283.15\text{K}$, which gives $\alpha = -0.00224928/\text{bar}$. Finally, R_0 in (2.3) is the normalized gas constant.

In practical gas network calculations, the gas flow rate q under norm conditions is considered in addition to the pressure p . We have $q = A\rho v/\rho_0$ with the cross-sectional area A of the pipe and the norm density ρ_0 . Replacing the density ρ and the velocity v in (2.1)-(2.2) and defining $C_0 = R_0\rho_0\bar{T}/A$, we get the following system of equations

$$(2.6) \quad \partial_t \left(\frac{p}{z(p)} \right) + C_0 \partial_x q = 0,$$

$$(2.7) \quad \partial_t q + C_0 \partial_x \left(\frac{z(p)}{p} q^2 \right) + \frac{A}{\rho_0} \partial_x p = -\frac{C_0}{2D} \frac{z(p)}{p} \lambda(|q|) q|q|.$$

To solve these equations numerically, it is important to study their flow characteristics. Setting $P = p/z(p) = p/(1 + \alpha p)$ and reformulating (2.6)-(2.7) in terms of the vector $u = (P, q)$ as $\partial_t u + \partial_x F(u) = Q(u)$, a short calculation yields the eigenvalues of the Jacobian $\partial_u F(u)$. We obtain $\lambda_{1/2} = v \pm c(p)$, where c is the speed of sound defined by $c^2 = \partial_p p$. From (2.3), we have $c(p) = z(p)\sqrt{R_0\bar{T}}$.

The system of differential equations (2.6)-(2.7) has to be completed by initial, boundary, and coupling conditions across the whole network. Suppose initial data $p(x, t_0) = p^0(x)$ and $q(x, t_0) = q^0(x)$ are given. Admissible boundary values must be chosen in accordance to the characteristics [7]. In our application, we make either the gas flow rate q or the pressure p available at sources and sinks. At each node $\nu \in V$ with ingoing pipes $e_i, i \in \delta_\nu^+$, outgoing pipes $e_j, j \in \delta_\nu^-$, and a family of intermediate vertices $\{\nu_k\}, k \in \delta_\nu^- \cup \delta_\nu^+$, we enforce conservation of mass,

$$(2.8) \quad \sum_{i \in \delta_\nu^+} q(\nu_i, t) = \sum_{j \in \delta_\nu^-} q(\nu_j, t),$$

and consistency of the pressure,

$$(2.9) \quad p(\nu_i, t) = p(\nu_j, t) \quad \text{for all } i \in \delta_\nu^+, j \in \delta_\nu^-.$$

Condition (2.8) is known as Kirchoff's law and is often referred to as Rankine-Hugoniot condition at a node [7].

Gas networks are operated in the subsonic flow region, that is, $|v| < c$. Usually, we even observe $|v| \ll c$ in practically relevant situations. We thus can conclude that although the characteristics are solution-dependent, they do not change their sign, that is, the information directions are maintained. In this case, implicit box schemes are known to work very effectively. Box schemes, originally introduced by Wendroff [22], have been used for several years. They are conservative schemes, i.e., they guarantee exact conservation of physical quantities at the level of the box.

Nonphysical oscillations, often caused by parasitic solution components in standard finite difference or finite volume approximations, are avoided. Box schemes are stable under mild conditions or even unconditionally stable and therefore allow large time steps, while they are as easy to program as explicit methods.

The basic idea of box schemes is to locate the degrees of freedom at the centre of the nodes instead at the centre of the edges as in finite volume schemes. We consider a sequence of discrete time points $t_0 < t_1 < \dots < t_N$. Let $p_{\nu_i}^n$, $p_{w_i}^n$, $q_{\nu_i}^n$, and $q_{w_i}^n$ be grid functions at time t_n and the ends of pipe $e_i = \nu_i w_i$. Then, our box scheme on e_i reads

$$(2.10) \quad \frac{P_{w_i}^{n+1} + P_{\nu_i}^{n+1}}{2\tau_n} - \frac{P_{w_i}^n + P_{\nu_i}^n}{2\tau_n} + C_0 \frac{q_{w_i}^{n+1} - q_{\nu_i}^{n+1}}{h_i} = 0$$

$$(2.11) \quad \frac{q_{w_i}^{n+1} + q_{\nu_i}^{n+1}}{2\tau_n} - \frac{q_{w_i}^n + q_{\nu_i}^n}{2\tau_n} + \frac{C_0}{h_i} \left(\frac{(q_{w_i}^{n+1})^2}{P_{w_i}^{n+1}} - \frac{(q_{\nu_i}^{n+1})^2}{P_{\nu_i}^{n+1}} \right) + \frac{A}{\rho_0} \frac{p_{w_i}^{n+1} - p_{\nu_i}^{n+1}}{h_i} =$$

$$- \frac{C_0}{2D} \left(\frac{\lambda(|q_{w_i}^{n+1}|) q_{w_i}^{n+1} |q_{w_i}^{n+1}|}{2P_{w_i}^{n+1}} + \frac{\lambda(|q_{\nu_i}^{n+1}|) q_{\nu_i}^{n+1} |q_{\nu_i}^{n+1}|}{2P_{\nu_i}^{n+1}} \right).$$

Here, $P_{\nu}^n = P(p_{\nu}^n) = p_{\nu}^n / z(p_{\nu}^n)$ for all ν , $h_i = |e_i|$ is the box length, and $\tau_n = t_{n+1} - t_n$ is the time step size. The scheme is symmetric in space and first order. For scalar conservation laws, it is stable for $\tau_n \geq kh$, where the constant $k > 0$ depends on the flux function and h is the uniform box length, and converges to the entropy solution [5]. Finally, to built up the fully coupled grid equations, the coupling conditions (2.8) and (2.9) are discretized at $t = t_{n+1}$.

We note that a pipe can always be subdivided into smaller pieces of pipes in order to improve the approximation property of the proposed box scheme.

Valves and Compressors. Valves and compressors are modelled by segments of zero length. A valve is a control element which can be opened or closed. A compressor compensates for the pressure loss in pipes due to friction. Since the operation of a compressor is quite complex and often steered by empirical data, we use an idealized model for a compressor [12]. Let $\nu_i w_i = c \in \mathbb{E}_C$ and $\nu_j w_j = v \in \mathbb{E}_V$. Then, we have the following two conditions in each case,

$$(2.12) \quad q_{\nu_j} = q_{w_j} = 0, \quad \text{valve closed,}$$

$$(2.13) \quad q_{\nu_j} = q_{w_j}, \quad p_{\nu_j} = p_{w_j}, \quad \text{valve opened,}$$

$$(2.14) \quad q_{w_i} = q_{\nu_i} - d_c z(p_{\nu_i}) q_{\nu_i} \left(\left(\frac{p_{w_i}}{p_{\nu_i}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right), \quad p_{w_i} = p_{out}^c, \quad \text{compressor on,}$$

$$(2.15) \quad q_{\nu_i} = q_{w_i}, \quad p_{\nu_i} = p_{w_i}, \quad \text{compressor off.}$$

Here, γ is the isentropic exponent, d_c is a specific compressor constant, and p_{out}^c is the desired outgoing pressure. We note that a compressor is usually combined with a so-called bypass valve. If the compressor is on, then the bypass valve is closed and vice versa. All equations are discretized at $t = t_{n+1}$.

Optimization Task. The task is to route the gas through the network to satisfy the consumers' demands such that the fuel gas consumption of the compressors is

minimized. To keep the running costs at an acceptable level, all compressor stations have to work efficiently to transport the gas through the network. The costs of each compressor $\nu_i w_i = c \in \mathbb{E}_C$ are modelled by its entire fuel gas consumption,

$$(2.16) \quad F_c(p_{\nu_i}, p_{out}^c, q_{\nu_i}) = d_c z(p_{\nu_i}) q_{\nu_i} \left(\left(\frac{p_{out}^c}{p_{\nu_i}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right).$$

The fuel gas consumption is proportional to the compressor power $H_c(p_{\nu_i}, p_{out}^c, q_{\nu_i})$, i.e., $H_c = d_h F_c$ with a constant $d_h > 0$. We replace p_{out}^c in (2.14), using the formula for H_c , and take the set of all H_c as control variables. In real-world gas networks, it is necessary to bound the compressor power. Therefore, we have to choose the outgoing pressure p_{out}^c such that $0 \leq H_c \leq H_c^{max}$. Due to efficiency, it is also common to require $H_c^{min} \leq H_c$ whenever the compressor is on.

Our continuous optimization problem now reads as follows:

$$(2.17) \quad \min_{H_c \in H_c^{ad}, c \in \mathbb{E}_C} J(p, q) := \sum_{c \in \mathbb{E}_C} \int_{t_0}^{t_N} F_c(t) dt$$

subject to all continuous state equations describing the instationary behaviour of (p, q) . Here, $H_c^{ad} = \{H : H = 0 \text{ or } H_c^{min} \leq H \leq H_c^{max}\}$ is the set of admissible controls, which guarantees the boundedness of the compressor power for $c \in \mathbb{E}_C$. Observe that the objective function is neither convex nor concave. In order to get a fully discretized model, the objective function is approximated by the trapezoidal rule using the discrete time points t_n .

3. Mixed Integer Linear Programming Formulation. In this section, we derive a mixed integer linear programming (MILP) formulation for our gas network optimization problem.

Discrete processes. In a gas network, there are mainly two types of components which require the modelling of discrete processes: (i) a compressor is either switched off or works at least with a certain minimum power H_c^{min} and (ii) a valve can either be closed or opened, whereas its switching time is negligible. To reflect this behaviour in our model, we introduce binary variables $s_c^n \in \{0, 1\}$, $c \in \mathbb{E}_C$, and $s_v^n \in \{0, 1\}$, $v \in \mathbb{E}_V$, for each compressor and valve. If $s_c^n = 1$, the compressor is running at t_n , otherwise it is turned off. A valve is open at the same time iff $s_v^n = 1$.

Linearization. In the preceding section, we have introduced a symmetric implicit box scheme to discretize the isothermal Euler equations. The scheme allows large time steps, resulting in relatively few coupled time levels. This makes it attractive for setting up a MILP model. Since the discrete equations are still nonlinear, we have to approximate the nonlinear terms by piecewise linear functions, which will be described next.

The nonlinearities in the discrete continuity equation (2.10) result from the expressions $P(p_\nu^n) = p_\nu^n / (1 + \alpha p_\nu^n)$. The function P only depends on the pressure and can be approximated by piecewise linear interpolation. Let $p_\nu^n \in [p_{min}, p_{max}]$ and pressure nodes given with $p_{min} = p_0 < p_1 < \dots < p_l = p_{max}$. Then, the errors of the local interpolation

$$(3.1) \quad P(p_\nu^n) \approx P(p_j) + \frac{P(p_{j+1}) - P(p_j)}{p_{j+1} - p_j} (p_\nu^n - p_j), \quad p_\nu^n \in [p_j, p_{j+1}],$$

can be balanced over all subintervals by choosing appropriate inner nodes p_i . For example, setting $p_{min} = 30 \text{ bar}$, $p_{max} = 70 \text{ bar}$, and $p_1 = 48.5 \text{ bar}$ gives together with $\alpha = -0.00224928/\text{bar}$ (which is the value in our applications) a relative maximum interpolation error of less than 0.6%.

The discrete momentum equation (2.11) can be rewritten as

$$(3.2) \quad \frac{q_{w_i}^{n+1} + q_{v_i}^{n+1}}{2\tau_n} - \frac{q_{w_i}^n + q_{v_i}^n}{2\tau_n} + C_0(I_{w_i}^{n+1} + R_{v_i}^{n+1}) + \frac{A}{\rho_0} \frac{p_{w_i}^{n+1} - p_{v_i}^{n+1}}{h_i} = 0,$$

where the nonlinearities are given by functions I and R ,

$$(3.3) \quad I_v^n = I(p_v^n, q_v^n) = \frac{(q_v^n)^2}{P(p_v^n)} \left(\frac{1}{h_i} + \frac{\lambda(|q_v^n|)}{4D} \right),$$

$$(3.4) \quad R_v^n = R(p_v^n, q_v^n) = \frac{(q_v^n)^2}{P(p_v^n)} \left(\frac{\lambda(|q_v^n|)}{4D} - \frac{1}{h_i} \right).$$

Both functions depend on the pressure and the flow rate. Since $1/h_i$ is in general very small in comparison to $\lambda/(4D)$, the two functions are nearly identical. Introducing nonnegative bounds q_{min} and q_{max} as minimum and maximum flow rate, we use a triangulation of the rectangular domain $[p_{min}, p_{max}] \times [q_{min}, q_{max}]$ to construct piecewise linear approximations over triangles to I and R through interpolation in the mesh points. In order to achieve a desired maximum relative error with a nearly optimal number of mesh points, we combine a successive local insertion of new nodes with a common two-dimensional Delaunay mesh generator. Fig. 3.1 shows an illustrative example.

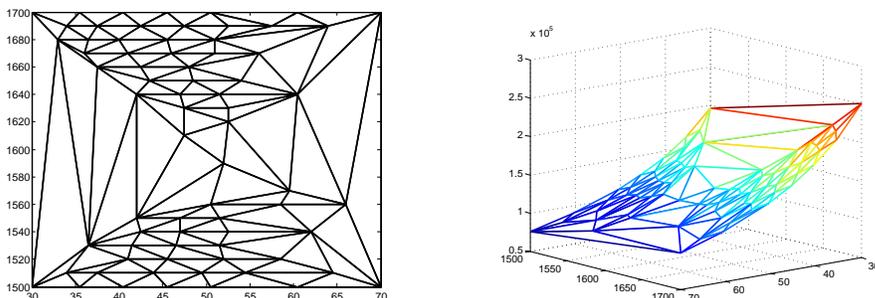


Fig. 3.1: Example: Two-dimensional adaptive Delaunay triangulation for the function $I(p, q)$ over the domain $[30 \text{ bar}, 70 \text{ bar}] \times [1.5 \times 10^6 \text{ m}^3/\text{h}, 1.7 \times 10^6 \text{ m}^3/\text{h}]$ with a maximum relative approximation error of 1%. The triangulation consists of 71 vertices and 126 triangles. Left: Mesh used to interpolate $I(p, q)$. Right: Surface plot for the piecewise linear interpolant ΠI of I .

It remains to discretize the trivariate function F_c in (2.16), describing the fuel gas consumption of a compressor. Analogously to the bivariate case, we apply an adaptive tetrahedralization. We note that all these mesh generations can be done a priori.

Piecewise linear functions in MILPs. We will now describe how general piecewise linear functions can be treated within a MILP framework.

Let $\phi : \Omega \subset \mathbb{R}^m \rightarrow \mathbb{R}$, $\mathbf{x} \mapsto \phi(\mathbf{x})$ be a continuous nonlinear function on a polygonally bounded domain Ω and $\Pi\phi$ its piecewise linear interpolant constructed through the approach described above. We assume that $\mathbb{T} = \{S^1, \dots, S^l\}$ is the set of m -simplices forming the underlying triangulation of Ω , which is a simplicial complex. For each simplex S^i , we denote with $\mathbf{v}_i^j \in \mathbb{R}^m$, $j = 0, \dots, m$, the set of its vertices. We assume the simplices to be ordered in such a way that two successive simplices, S^i and S^{i+1} , share at least one vertex and $\mathbf{v}_i^m = \mathbf{v}_{i+1}^0$ holds for all $i = 1, \dots, l-1$. Such a linear ordering of simplices and vertices is trivial for $m = 1$. In [1], an algorithm has been designed to find such an ordering for $m = 2$, the most complicated case. An example is presented in Fig. 3.2(a). For $m \geq 3$, we construct an appropriate ordering as follows:

- (S1) We choose two m -simplices S^1 and S^2 which have a common facet and number their vertices such that $\mathbf{v}_1^m = \mathbf{v}_2^0$ and $\mathbf{v}_1^0 = \mathbf{v}_2^m$ holds.
- (S2) We mark S^1 and S^2 as ordered.
- (S3) Since \mathbb{T} is a simplicial complex, we can find some unordered simplex S^i which has a common facet with some already ordered simplex S^j .
- (S4) Since $m \geq 3$, there is at least one common vertex of S^i and S^j which is neither \mathbf{v}_j^0 nor \mathbf{v}_j^m . Thus, we can extend the set of ordered simplices by S^i , changing only the vertex numbering for S^i and S^j .
- (S5) We repeat steps (S3) and (S4) until all simplices are ordered.

This leads to an appropriate ordering, where additionally $\mathbf{v}_l^m = \mathbf{v}_1^0$ holds.

Suppose such an ordering to be given. The so-called incremental method (sometimes also referred to as δ -method) [9, 23] can then be used to describe the interpolant $\Pi\phi$ in terms of linear constraints. Introducing nonnegative variables δ_i^j and binary variables w_i for $i = 1, \dots, l$ and $j = 1, \dots, m$, we add the following constraints to our MILP:

$$(3.5) \quad \mathbf{x} = \mathbf{v}_1^0 + \sum_{i=1}^l \sum_{j=1}^m \left(\mathbf{v}_i^j - \mathbf{v}_i^0 \right) \delta_i^j$$

$$(3.6) \quad \Pi\phi(\mathbf{x}) = \phi(\mathbf{v}_1^0) + \sum_{i=1}^l \sum_{j=1}^m \left(\phi(\mathbf{v}_i^j) - \phi(\mathbf{v}_i^0) \right) \delta_i^j$$

$$(3.7) \quad \sum_{j=1}^m \delta_i^j \leq 1 \quad \text{for } i = 1, \dots, l$$

$$(3.8) \quad \sum_{j=1}^m \delta_{i+1}^j \leq w_i \quad \text{for } i = 1, \dots, l-1$$

$$(3.9) \quad w_i \leq \delta_i^m \quad \text{for } i = 1, \dots, l-1$$

$$(3.10) \quad \delta_i^j \geq 0 \quad \text{for } i = 1, \dots, l \text{ and } j = 1, \dots, m$$

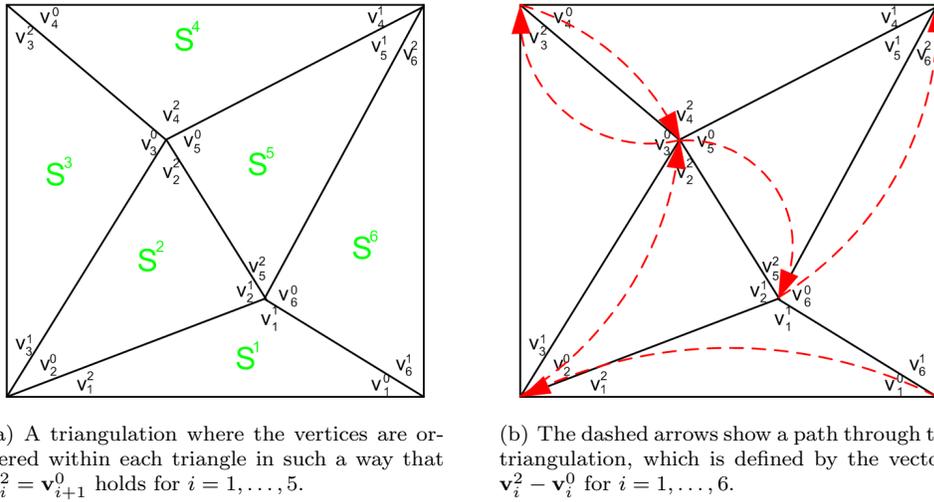
$$(3.11) \quad w_i \in \{0, 1\} \quad \text{for } i = 1, \dots, l.$$

The first constraint is used to express a certain point $x \in \mathbb{T}$ as a sum of vectors pointing along the edges of the triangulation, multiplied by their associated δ -variables. As a direct consequence, the piecewise linear function $\Pi\phi$ can be described through (3.6). The other constraints guarantee that the choice of all δ_i^j is admissible. For this, the so-called filling condition has to be satisfied: If a variable δ_{i+1}^j associated to the $(i+1)$ th simplex is positive for some $1 \leq j \leq m$, then all δ_j^m with $j < i$ have to equal

one. Observe that the vectors $\mathbf{v}_1^m - \mathbf{v}_1^0, \dots, \mathbf{v}_l^m - \mathbf{v}_l^0$ build a path along all simplices of the triangulation, see Fig. 3.2(b).

The incremental method is superior to the standard textbook approach as described, e.g., in [13]. There, convex combinations of variables associated to the vertices are used to describe a point in the triangulation. The advantage is based on the property that the polytope defined by (3.7)–(3.10), together with the nonnegativity constraints $w_i \geq 0$ for $i = 1, \dots, l$, is integral [15, 23]. This means that all vertices of the polytope, described by inequalities (3.7)–(3.10), are integral, even if we relax the integrality condition (3.11) on the w variables.

A third way to incorporate piecewise linear functions into a MILP is to use SOS branching, which was successfully implemented for gas network optimization in [11, 12]. Unfortunately, up to now, it is only suitable when uniform grids are applied.



(a) A triangulation where the vertices are ordered within each triangle in such a way that $\mathbf{v}_i^2 = \mathbf{v}_{i+1}^0$ holds for $i = 1, \dots, 5$.

(b) The dashed arrows show a path through the triangulation, which is defined by the vectors $\mathbf{v}_i^2 - \mathbf{v}_i^0$ for $i = 1, \dots, 6$.

Fig. 3.2: Example: A triangulation with a vertex path inducing a linear ordering of all simplices.

It is straightforward to apply the above approach to the nonlinearities P_ν^n, I_ν^n , and R_ν^n arising from the discretization of the isothermal Euler equations. For the piecewise linear approximation of F_c in (2.16), we have to modify the incremental method in order to satisfy $\Pi F_c \geq H_c^{min}/d_h$ whenever the compressor $c = \nu_k w_k \in \mathbb{E}_C$ is on. To meet this requirement, we introduce auxiliary variables $p_{\nu_k}^{aux}, p_{w_k}^{aux} \geq 0$ and apply a slightly modified version of the incremental method to approximate F_c :

$$(3.12) \quad p_{\nu_k}(\mathbf{x}) = p_{\nu_k}^{aux} + p_{\nu_k}(\mathbf{v}_1^0) s_c + \sum_{i=1}^l \sum_{j=1}^3 \left(p_{\nu_k}(\mathbf{v}_i^j) - p_{\nu_k}(\mathbf{v}_i^0) \right) \delta_i^j$$

$$(3.13) \quad p_{w_k}(\mathbf{x}) = p_{w_k}^{aux} + p_{w_k}(\mathbf{v}_1^0) s_c + \sum_{i=1}^l \sum_{j=1}^3 \left(p_{w_k}(\mathbf{v}_i^j) - p_{w_k}(\mathbf{v}_i^0) \right) \delta_i^j$$

$$(3.14) \quad q_{\nu_k}(\mathbf{x}) = q_{\nu_k}(\mathbf{v}_1^0) s_c + \sum_{i=1}^l \sum_{j=1}^3 \left(q_{\nu_k}(\mathbf{v}_i^j) - q_{\nu_k}(\mathbf{v}_i^0) \right) \delta_i^j$$

$$(3.15) \quad \Pi F_c(\mathbf{x}) = F_c(p_{\nu_k}(\mathbf{v}_1^0), p_{w_k}(\mathbf{v}_1^0), q_{\nu_k}(\mathbf{v}_1^0))s_c + \\ + \sum_{i=1}^l \sum_{j=1}^3 \left(F_c(p_{\nu_k}(\mathbf{v}_i^j), p_{w_k}(\mathbf{v}_i^j), q_{\nu_k}(\mathbf{v}_i^j)) - F_c(p_{\nu_k}(\mathbf{v}_i^0), p_{w_k}(\mathbf{v}_i^0), q_{\nu_k}(\mathbf{v}_i^0)) \right) \delta_i^j$$

$$(3.16) \quad s_c \geq \sum_{j=1}^3 \delta_i^j \text{ for } i = 1, \dots, l$$

$$(3.17) \quad w_i \geq \sum_{j=1}^3 \delta_{i+1}^j \text{ for } i = 1, \dots, l-1$$

$$(3.18) \quad w_i \leq \delta_i^3 \text{ for } i = 1, \dots, l-1$$

$$(3.19) \quad \delta_i^j \geq 0 \text{ for } i = 1, \dots, l \text{ and } j = 1, \dots, 3$$

$$(3.20) \quad w_i \in \{0, 1\} \text{ for } i = 1, \dots, l.$$

If the compressor is switched off, equation (3.16) forces all δ -variables to equal zero. In this case, we get from (3.12) and (3.13) the identities $p_{\nu_k} = p_{\nu_k}^{aux}$ and $p_{w_k} = p_{w_k}^{aux}$. The auxiliary variables are needed to provide the additional flexibility necessary to ensure the consistence of the pressure (2.9) at each time point. If the compressor is running, i.e., $s_c = 1$, then both auxiliary variables should vanish. To this end, we introduce two additional constraints,

$$(3.21) \quad p_{\nu_k}^{max}(s_c - 1) + p_{\nu_k}^{aux} \leq 0,$$

$$(3.22) \quad p_{w_k}^{max}(s_c - 1) + p_{w_k}^{aux} \leq 0,$$

where $p_{\nu_k}^{max}$ and $p_{w_k}^{max}$ are the maximum pressure values at node ν_k and w_k , respectively. These values are given due to technical restrictions.

4. Nonlinear Continuous optimization. We want to compare optimal solutions obtained with our MILP formulation with those computed with a fully nonlinear continuous optimization approach. The latter requests an appropriate treatment of binary variables. This is described next. Afterwards, we will shortly explain the ingredients of our simulation tool and the optimizer used.

Treatment of binary control variables. As introduced in section 2, the set of admissible controls for a compressor station is of the form $H_c^{ad} = \{0\} \cup [H_c^{\min}, H_c^{\max}]$. That is, for a running compressor station, the compressor drive unit has to generate a certain power $H_c \in [H_c^{\min}, H_c^{\max}]$, and we have $H_c = 0$ when the compressor is switched off. Therefore, the control of a compressor station includes a binary decision in every time step whether to switch it on or off. This switching process has to be modelled in the context of continuous optimization.

Not only in the context of continuous optimization, it is common practice to relax binary variables. We extend H_c^{ad} used in (2.17) to a relaxed set of admissible controls $\tilde{H}_c^{ad} = [0, H_c^{\max}]$. Since we cannot expect the optimal solution of the relaxed problem to be admissible for the original problem, we have to define a strategy how to process the computed information. The main idea of our algorithm is the following [6]: First, we add a variable penalty term, as shown in Fig. 4.1, to the cost function of each

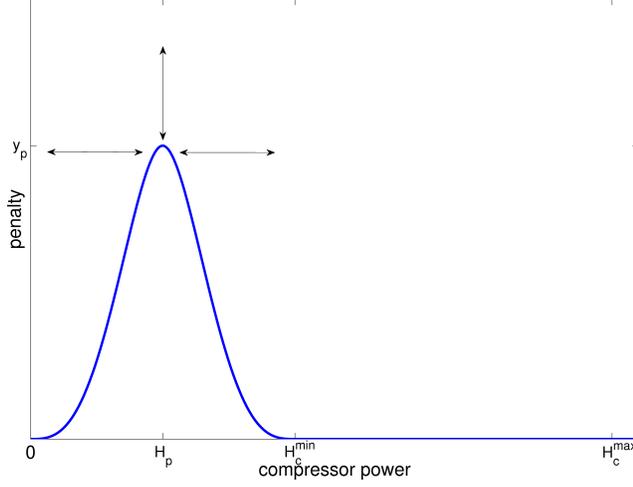


Fig. 4.1: Penalty function $F_p(H_c)$, which is added to the cost function of each compressor in order to penalize values of $H_c(t)$ within $(0, H_c^{min})$. Both the position of the peak H_p and its maximum value y_p can be varied.

compressor in every time step. The penalty term $F_p(H_c)$ is given by

$$(4.1) \quad F_p(H_c) = \begin{cases} y_p \left(2.5 \left(\frac{H_c}{H_p} \right)^3 - 1.5 \left(\frac{H_c}{H_p} \right)^5 \right) & \text{if } 0 \leq H_c \leq H_p \\ y_p \left(2.5 \left(\frac{H_c^{min} - H_c}{H_c^{min} - H_p} \right)^3 - 1.5 \left(\frac{H_c^{min} - H_c}{H_c^{min} - H_p} \right)^5 \right) & \text{if } H_p \leq H_c \leq H_c^{min} \\ 0 & \text{otherwise,} \end{cases}$$

where $H_p \in (0, H_c^{min})$ is the position of the peak of the penalty function and y_p the corresponding maximum value. These values have to be chosen for each compressor and each discrete time point t_n . Second, we keep switching decisions fixed for the next run of the optimizer as soon as the optimal solution of the relaxed problem is inside a Δ -region around the set of admissible controls. That is,

$$(4.2) \quad s_c^n = \begin{cases} 0 & \text{if } 0 \leq H_c(t_n) \leq \Delta \\ 1 & \text{if } H_c^{min} - \Delta \leq H_c(t_n) \leq H_c^{max}. \end{cases}$$

Concerning the variation of the penalty term, we apply the following strategy: After every run of the optimizer, the maximum value y_p is increased by the factor of 1.1. In addition, the position H_p of the peak of the penalty function is moved a certain amount in the direction of the current control value H_c . More precisely,

$$(4.3) \quad H_p = \begin{cases} H_p - \min(H_p - \Delta, \beta_l) & \text{if } H_c \leq H_p \\ H_p + \min(H_c^{min} - \Delta - H_p, \beta_r) & \text{if } H_c > H_p. \end{cases}$$

We use $\Delta = \beta_l = \beta_r = H_c^{min}/10$. This way, the control is supposed to be pushed into one of the Δ -regions in the next runs and the switching decision finally gets fixed.

Note that it is possible and might be even necessary that the position of the peak of the penalty function switches from one side of the current control value to the other.

Simulator and Optimizer. The main purpose of the simulator is to solve the system of discrete, nonlinear grid equations derived from the isothermal Euler equations, the coupling and boundary conditions, and the defining equations for compressors and valves in Section 2. For that, all binary variables s^n for the compressors and valves as well as the compressor powers H_c must be known. In order to allow a direct comparison, we use the same temporal and spatial discretization as for our MILP formulation. The nonlinear equations are solved by means of an adapted Newton’s method including the application of sparse matrix techniques and switching between simplified and full mode. The simulator is also used to check the admissibility, i.e., the compliance with given restrictions on the pressure and the flow rate, of the optimal control computed with the MILP approach. This allows to estimate the pollution effect of our linearizations.

The simulator is integrated into a black-box continuous optimizer that demands for function and gradient evaluations of the objective functional and the constraints. A single run of the simulator delivers a certain value for the objective functional $J(p, q)$ in (2.17). Gradients of $J(p, q)$ are computed from the numerical solution of the adjoint equations derived from the discrete model equations. We note that the adjoint equations act backwards in time and need the complete forward solution obtained by the simulator. As optimizer we choose the well-known SQP-solver DONLP2. It is based on a sequential equality constrained quadratic programming method with an active set technique. Bounds on the variables are treated in a gradient-projection like fashion. For more details consult [19, 20].

5. Numerical illustrations. First numerical results are given for two example networks, which are used to compare our linear and nonlinear approach to technical transient gas network optimization. One compressor has to be optimized in the first network, whereas for the second network, two compressors have to be efficiently operated with respect to a temporally increasing customer’s demand. In Section 5.3, we present optimal controls for a real-life problem.

The network structures of the example problems are shown in Fig. 5.1 and Fig. 5.2.

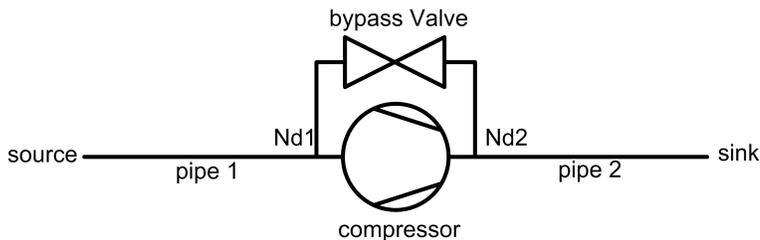


Fig. 5.1: First example network consisting of one source, one sink, two pipes, one compressor, and its bypass valve. There are two inner nodes $Nd1$ and $Nd2$ before and after the compressor.

All pipes are discretized by a two-point discretization. We perform four time steps with a constant $\tau = 1h$. The errors of the local interpolation are less than 0.5% for the scalar and bivariate functions P , I , and R , and less than 5% for the trivariate

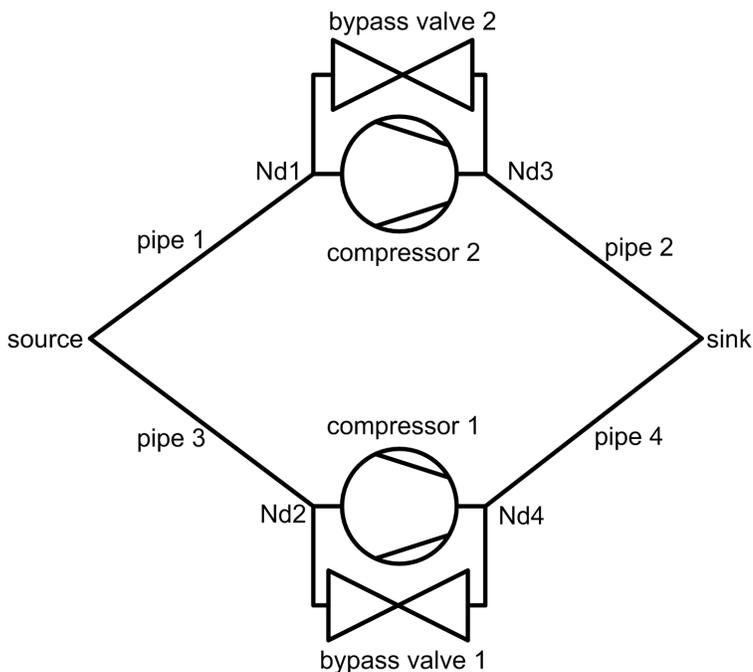


Fig. 5.2: Second example network consisting of one source, one sink, four pipes, two compressors, and their accompanying bypass valves. There are four inner nodes labelled from $Nd1$ to $Nd4$.

Time [h]	0	1	2	3	4
H_{c1} [kW]	0	0	0	0	600.00
H_{c2} [kW]	0	0	705.89	1352.98	1479.70
$\sum_c \int_0^4 F_c dt$ [m ³]					1039.23

Table 5.1: Network 2. First feasible control found by CPLEX.

function F_c . The arising MILPs are solved with CPLEX [4], where the run time for the branch-and-cut algorithm is limited to 120s. In order to provide an admissible solution to start with, we first compute a steady state with all compressors switched off. For this, we set $p=70\text{bar}$ at the source for the first problem and $p=65\text{bar}$ for the second one. In both cases, $q=1.5 \times 10^6\text{m}^3/\text{h}$ is taken at the sink. We consider the optimal control as sufficiently accurate if all pressure values stay within their desired boxes or exceed their bounds by not more than 0.5bar.

Finally, all pipes are 50km long and have a diameter $D=1\text{m}$. Their roughness is $k=0.01\text{mm}$. All compressors are of the same type and their specific constants are $d_c=0.053286$ and $d_h=2.9818\text{kWh}/\text{m}^3$.

The computations were done on a PC with Intel processor x86.

5.1. Network with one compressor. The first test example serves to illustrate the performance of the MILP approach. There is one source, where gas is fed into the network, and one sink representing a customer. The customer's demand is $q=$

Time [h]	0	1	2	3	4
H_{c1} [kW]	0	0	0	600.00	619.43
H_{c2} [kW]	0	0	600.00	600.00	688.25
$\sum_c \int_0^4 F_c dt$ [m ³]					822.95

Table 5.2: Network 2. Best control found by CPLEX.

Time [h]	0	1	2	3	4
H_{c1} [kW]	0	0	600	600	943.65
H_{c2} [kW]	0	0	600	600	943.65
$\sum_c \int_0^4 F_c dt$ [m ³]					1121.37

Table 5.3: Network 2. Optimal control solution found by the SQP method.

$2 \times 10^6 \text{m}^3/\text{h}$, which is $5 \times 10^5 \text{m}^3/\text{h}$ higher than the value for the steady state solution. For all times, the pressure $p=70\text{bar}$ at the source is kept fixed. The specific bounds, set for this problem, are

$$q_{min} = 1.4 \times 10^6 \text{m}^3/\text{h}, q_{max} = 2.1 \times 10^6 \text{m}^3/\text{h}, p_{min} = 58\text{bar}, p_{max} = 71\text{bar}$$

and

$$H^{min} = 1.0 \times 10^5 \text{kW}, H^{max} = 6.0 \times 10^5 \text{kW}.$$

CPLEX delivers an optimal control for which the compressor is running at its technical minimum all the time. The correctness of the solution is confirmed by our simulator. All bounds are satisfied. The CPU time is negligible. It is remarkable that although the problem seems to be simple, the nonlinear global optimization software BARON [18] could not find a feasible solution.

5.2. Network with two compressors. Our second example network is slightly more complex. We shall compare optimal control solutions obtained with MILP and the optimizer DONLP2.

The customer's demand at the sink varies linearly in time from $1.625 \times 10^6 \text{m}^3/\text{h}$ to $2.0 \times 10^6 \text{m}^3/\text{h}$. Again, the pressure $p=65\text{bar}$ at the source is kept fixed. The flow rate at the source into the network is forced to be in the interval $[1.4 \times 10^6 \text{m}^3/\text{h}, 2.2 \times 10^6 \text{m}^3/\text{h}]$. For all other elements of the network, we have the specific bounds

$$q_{min} = 7.0 \times 10^5 \text{m}^3/\text{h}, q_{max} = 1.1 \times 10^6 \text{m}^3/\text{h}, p_{min} = 61\text{bar}, p_{max} = 65\text{bar}$$

and

$$H^{min} = 600\text{kW}, H^{max} = 1500\text{kW}.$$

CPLEX finds a feasible solution to the linearized problem within 2s. Passing this solution to our simulator and solving the nonlinear discrete equations, we find that the variations with respect to the pressure bounds sum up to 0.0167bar , which is negligible. The fuel gas consumption of the compressors is about 1039m^3 . The control variables, i.e., the compressor power H_c , are given in Tab. 5.1.

Time [h]	0	1	2	3	4
H_{c1} [kW]	0	0	0	600	988.03
H_{c2} [kW]	0	0	600	600	988.04
$\sum_c \int_0^4 F_c dt$ [m ³]					935.03

Table 5.4: Network 2. Optimal control solution found by the SQP method, if the switching decisions from the best CPLEX-run are used.

Time [h]	0	1	2	3	4
H_{c1} [kW]	2000	0	1949.91	2077.30	4027.47
H_{c2} [kW]	2000	0	1972.06	1633.65	2716.44
H_{c3} [kW]	2000	1932.78	2297.38	2703.24	2800.39
$\sum_c \int_0^4 F_c dt$ [m ³]					7598.24

Table 5.5: Network 3. Optimal control solution found by the SQP method.

After a run time of 120s, we find a solution that is not worse than the optimal solution by a factor of 1.03. The fuel gas consumption of both compressors is about 823m³. In this case, the simulator delivers a total violation of the pressure bounds of 0.46bar, which is still in the acceptable range. The corresponding control is given in Tab. 5.2. We observe that the solution is improved at the price of larger pressure violations. This is possible due to the linearization error.

The problem is also solved with the SQP-solver DONLP2 coupled with our simulator. The result for the control is revealed in Tab. 5.3. An interesting observation is that the SQP-solver finds a symmetric solution that is optimal for the switching decisions made, but worse than the unsymmetric solutions found by the MILP approach. Using the switching decisions from the best CPLEX-run, the SQP-solver improves its solution by 16.6%. The results are shown in Tab. 5.4.

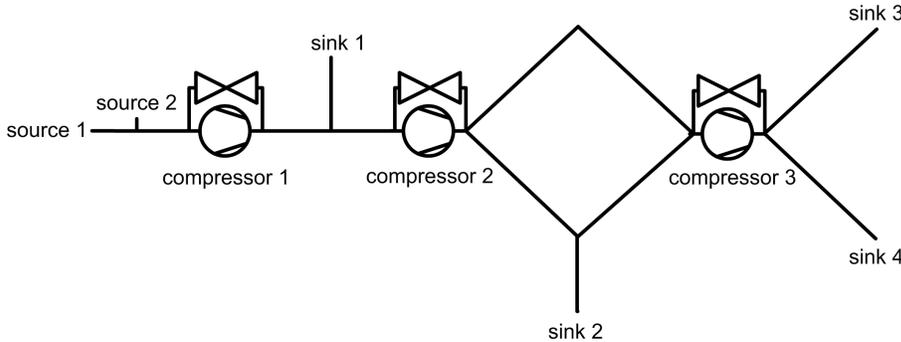


Fig. 5.3: Real-life network consisting of two sources, four sinks, a couple of pipes, three compressors, and their accompanying bypass valves.

5.3. Real-life network with three compressors. Our third example is taken from a real-life application. It consists of three compressors and four customers as

Time [h]	0	1	2	3	4
H_{c1} [kW]	2000	1000.00	1000.00	1000.00	1000.00
H_{c2} [kW]	2000	1000.00	1000.00	1000.00	1000.00
H_{c3} [kW]	2000	1624.78	1465.47	1808.16	2116.54
$\sum_c \int_0^4 F_c dt$ [m ³]					5430.36

Table 5.6: Network 3. Best solution found by CPLEX with maximum violation of 1.5bar for the pressure bounds.

Time [h]	0	1	2	3	4
H_{c1} [kW]	2000	1000.00	1104.97	1117.02	2552.42
H_{c2} [kW]	2000	1900.04	1559.86	1172.43	1794.92
H_{c3} [kW]	2000	1307.18	2036.70	2491.78	2807.41
$\sum_c \int_0^4 F_c dt$ [m ³]					6880.55

Table 5.7: Network 3. Optimal control solution found by the SQP method, if the switching decisions from the best CPLEX-run are used.

shown in Fig. 5.3. This problem is significantly more challenging than the previous ones. Again we consider optimal control solutions obtained by CPLEX and the optimizer DONLP2 for a four hour time period. The customers' demands at all sinks increases linearly from $3.6 \times 10^5 \text{m}^3/\text{h}$ to $4.2235 \times 10^5 \text{m}^3/\text{h}$, except the second sink, where the outflow is a constant, $1.8 \times 10^5 \text{m}^3/\text{h}$. There is a fixed pressure $p=70\text{bar}$ at the first source while a constant inflow rate $q=7.2 \times 10^5 \text{m}^3/\text{h}$ is given at the second source.

For all elements of the network, the corresponding flow and pressure values are bounded from below and above. Typically, the flow is allowed to vary by some $10^5 \text{m}^3/\text{h}$ and the pressure values by about 3bar. For the compressors, we have the specific bounds

$$H^{min} = 1000\text{kW}, H^{max} = 10000\text{kW}$$

and the individual constants $d_c = 0.053287, 0.051955, 0.055951$, in (2.16) and $d_h = 2.981750\text{kWh}/\text{m}^3, 3.058204\text{kWh}/\text{m}^3, 2.839763\text{kWh}/\text{m}^3$. Recall $H_c = d_h F_c$.

After a few seconds, the SQP-solver DONLP2 coupled with our simulator delivers a solution resulting in a fuel gas consumption of about 7598m^3 . Details are given in Tab. 5.5. CPLEX finds a first feasible solution within 180s, but this does not give a better solution when passed to the nonlinear optimizer. After about 7 hour runtime, optimality is proven for the linearized model. The corresponding optimal control is given in Tab. 5.6. The fuel gas consumption of all compressors is about 5430m^3 , however the pressure bounds are violated by 1.6bar – a value, which is three times higher than allowed. Using the binary decisions made and the control as initial state for the SQP-solver, we obtain an improved feasible control solution with an overall fuel gas consumption of 6881m^3 , see Tab. 5.7.

6. Summary and Conclusions. We have considered the problem of technical transient gas network optimization. The gas flow through pipelines is modelled by the instationary nonlinear isothermal Euler equations. The objective function, which

is the overall sum of the fuel gas consumption of all compressors, is neither convex nor concave. To attack the minimization problem, we have applied mixed integer linear and sequential quadratic nonlinear programming-based algorithms. In a first step, the Euler equations were discretized by a symmetric implicit box scheme. To set up the mixed integer linear program (MILP), the incremental method was then used to approximate the nonlinearities through piecewise linear functions.

On the basis of numerical experiments made for two academic problems and one real-life application, we have come to three main conclusions. (i) Our algorithms are able to compute feasible and physically meaningful optimal control solutions to transient gas network problems. Although less known in nonlinear network optimization, MILPs are remarkably useful to handle combinatorial constraints and to find the best binary decisions since their solution strategies guarantee to derive globally optimal solutions. (ii) The classical sequential quadratic programming (SQP) extended with a continuous treatment of binary control variables is able to deliver feasible optimal control solutions in short running times, which seems to be also true for more complex gas network problems. However, these solutions are in general only locally optimal. (iii) Better results could be achieved by combining both the MILP and the SQP approach. In a first step, the MILP solver equipped with an a priori control of the linearization errors provides a globally optimal set of combinatorial decisions for the switching network components, that is for compressors and valves. Taking the corresponding discrete states as input, the SQP method is then used to solve the remaining nonlinear optimization problem. The solution computed in such a way is optimal with respect to the underlying linear model and satisfies all side constraints, especially the often crucial bounds for the pressure variables.

Inspired by our first promising results, we will develop feedback strategies to pass information from the SQP solver back to the MILP solver in order to speed up the branch-and-cut algorithm in CPLEX. We will also study sensitivities of the binary decisions in terms of the linearization errors.

REFERENCES

- [1] J.J. Bartholdi III, P. Goldsman, *Continuous Spatial Indexing of Surfaces - Part 1: Standard Triangulations*, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332-0205 USA, 2001.
- [2] L.T. Biegler, A. Wächter, *On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming*, Math. Progr. 100, pp. 25–57, 2006.
- [3] R.G. Carter, *Pipeline optimization: dynamic programming after 30 years*, In: 30th Annual Meeting of Pipeline Simulation Research Group, 1998.
- [4] ILOG CPLEX 10.0, *ILOG CPLEX 10.0 User Manual*, ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA, 2006.
- [5] O. Kolb, J. Lang, P. Bales, *Adaptive Linearization for the Optimal Control Problem of Gas Flow in Pipeline Networks*, Preprint 2553, Department of Mathematics, Technische Universität Darmstadt, 2008.
- [6] O. Kolb, P. Bales, J. Lang *Moving Penalty Functions for Optimal Control with PDEs on Networks*, Preprint 2562, Department of Mathematics, Technische Universität Darmstadt, 2008.
- [7] R. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press 2002.
- [8] D. Mahlke, A. Martin, S. Moritz, *A Simulated Annealing Algorithm for Transient Optimization in Gas Networks*, Math. Meth. Oper. Res. 66, pp. 99–116, 2007.
- [9] A.S. Manne, H.M. Markowitz, *On the Solution of Discrete Programming Problems*, Econometrica 110, pp. 25–84, 1957.
- [10] A. Martin, M. Möller, S. Moritz, *Mixed Integer Models for the Stationary Case of Gas Network Optimization*, Math. Progr. 105, pp. 563–582, 2006.

- [11] M. Möller, *Mixed Integer Models for the Optimisation of Gas Networks in the Stationary Case*, PhD thesis, Technische Universität Darmstadt, Department of Mathematics, 2004.
- [12] S. Moritz, *A Mixed Integer Approach for the Transient Case of Gas Network Optimization*, PhD Thesis, Technische Universität Darmstadt, Department of Mathematics, 2007.
- [13] G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1999.
- [14] A.J. Osiadacz, S. Swierczewski, *Optimal control of gas transportation systems*, In: Proceedings of IEEE Conference on Control Applications, Vol. 2, pp. 795–796, 1994.
- [15] M. Padberg, *Approximating Separable Nonlinear Functions via Mixed Zero-One Programs* Operation Research Letters 27, pp. 1–5, 2000.
- [16] T.P.A. Perdicoulis, L.R. Fletcher, *Decentralised dynamic optimisation of gas*, In: 31st Annual Meeting of Pipeline Simulation Interest Group, 1999.
- [17] N.L. Ramchandani, *Optimisation of gas networks using Nash equilibria derived from dynamic non-cooperative game theory*, PhD thesis, University of Stanford, 1993.
- [18] N. V. Sahinidis, M. Tawarmalani, *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs*, User's Manual, 2005.
Available at <http://www.gams.com/dd/docs/solvers/baron.pdf>.
- [19] P. Spellucci, *A New Technique for Inconsistent Problems in the SQP Method*, Math. Meth. Oper. Res. 47, pp. 355–400, 1998.
- [20] P. Spellucci, *An SQP Method for General Nonlinear Programs Using Only Equality Constrained Subproblems*, Math. Prog. 82, pp. 413–448, 1998.
- [21] M.C. Steinbach, *On PDE solution in transient optimization of gas networks*, J. Comp. Appl. Math. 203, pp. 345–361, 2007.
- [22] B. Wendroff, *On centered difference equations for hyperbolic systems*, J. Soc. Ind. Appl. Math. 8, pp. 549–555, 1960.
- [23] D. Wilson, *Polyhedral Methods for Piecewise Linear Functions*, PhD thesis, University of Kentucky, 1998.