Claudia D'Ambrosio, Andrea Lodi, Sven Wiese, Cristiana Bragalli, ***Mathematical programming techniques in water network optimization****, European Journal of Operational Research, Volume 243, Issue 3, 2015, Pages 774-788, ISSN 0377-2217*

The final published version is available online at:
https://doi.org/10.1016/j.ejor.2014.12.039

Mathematical Programming techniques in Water Network Optimization

C. D'Ambrosio[a], A. Lodi[b,*], S. Wiese[b], C. Bragalli[c]

[a]*CNRS & LIX, Ecole Polytechnique, 91128 Palaiseau CEDEX, France*
[b]*DEI, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy*
[c]*DICAM, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy*

Abstract

In this article we survey mathematical programming approaches to problems in the field of water network optimization. Predominant in the literature are two different, but related problem classes. One can be described by the notion of network design, while the other is more aptly termed by network operation. The basic underlying model in both cases is a nonlinear network flow model, and we give an overview on the more specific modeling aspects in each case. The overall mathematical model is a Mixed Integer Nonlinear Program having a common structure with respect to how water dynamics in pipes is described. Finally, we survey the algorithmic approaches to solve the proposed problems and we discuss computation on various types of water networks.

Keywords: Networks, Mixed Integer Nonlinear Programming, Combinatorial optimization, Global optimization

1. Introduction

In classic network flow problems the task is to route a flow through a network from a set of sources to a set of sinks. This point of view can become coarse when dealing with pressurized water networks, where the fluid is transported in pipes with no air contact and thus possibly varying pressure levels. To accurately model the physical aspects of such networks one can introduce pressure variables at nodes in addition to flow variables on arcs. In this modeling enhancement, what actually induces a flow between two nodes is explained by a pressure difference. To subsume a broader field of applications, the additional variables in such approaches are also referred to as node potentials, including as well the electric potential of a point in an electric circuit. Pressure again is an important quantity in gas networks. In water network optimization, such a modeling approach has experienced eminent interest in order to develop physically sound models for real-world applications. The drawback of the resulting accuracy gain is the fact that the relation between flow and potential difference usually leads to nonlinear equations. Together with discrete decisions that can be made regarding different network elements, this puts the optimization tasks faced here in the context of Mixed Integer Nonlinear Programming (MINLP).

In the following we focus on surveying topics related to the optimization of water networks. We use the term water networks to subsume anything that is named by water distribution networks, water supply systems, or combinations of the two. The term network underlines the fact that we deal with applications in which the underlying structure can be modeled by a graph in a mathematical sense. Out of the different stages in that water network optimization can be subdivided, we focus on the somewhat different tasks of *optimal design of water networks* on the one hand and *optimal operation of water networks* on the other. On both sides, one assumes to have an underlying network with a fixed topology, i.e., a fixed set of nodes and arcs representing sources, sinks, pipes, pumps, valves, and tanks. The former of the two optimization tasks, i.e., the design problem, usually disregards pumps, valves and tanks. One then seeks to choose for each pipe in the network a diameter among a discrete set of commercially available diameters in a cost-minimal way, while maintaining the satisfiability of all costumer demands located at sinks. The diameter has of course an important impact on the pipe's capacity, but also on the pressure distribution in the network. The operation problem instead typically assumes fixed pipe diameters but allows for the modeling of pumps, valves, and tanks. The task is then to operate pumps and valves, which again affect flow and pressure distribution, over a certain time horizon in order to satisfy the customer demands, while minimizing the operational costs mainly arising from power consumption of pumps. In its full-scale form, the operation problem hence incorporates the aspect of time into the model and is thus a *dynamic* problem, unlike the *static* design problem. In spite of these differences, there are still some obvious similarities from a mathematical point of view, due to the way water dynamics in a pipe is described. Typically, the majority of the arcs in a network is constituted by pipes, and the equation associated with a pipe will be at the heart of the present survey.

As mentioned earlier, the problems we consider here belong to the class of MINLPs, and the presented approaches exploit (different variants of) different algorithmic paradigms to solve MINLPs, including Mixed Integer Linear Programming (MILP) techniques. We try to provide an overview of how different techniques succeed in different situations. We are inter-

*Corresponding author

Email addresses: dambrosio@lix.polytechnique.fr (C. D'Ambrosio), andrea.lodi@unibo.it (A. Lodi), sven.wiese@unibo.it (S. Wiese), cristiana.bragalli@unibo.it (C. Bragalli)

ested in mathematical programming approaches, i.e., methods that explicitly use a mathematical programming model. We do not dwell on the variety of meta-heuristic approaches that exist in the field. For a broader discussion of topics related to water networks that covers also aspects that are not of algorithmic nature, we refer the reader to [1].

2. Modeling

To begin with, we present the main modeling aspects found in the literature concerning the design and the operation problem. We say the main aspects, because different variants at different levels of detail can be found. The most detailed modeling description of the relevant aspects is provided in [2]. A network is naturally represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where nodes stand for sources and sinks and arcs stand for pipes, pumps, and valves. Tanks are usually modeled as nodes, but this is not always true.

2.1. Flow & pressure

A main difference between the design and the operation problem is of course the contrast between the static and the dynamic setting. In the latter, in principle all variables and parameters can be made continuous-time dependent on $t \in [0, T]$, where $[0, T] \subset \mathbb{R}$ is the considered time horizon. However, to get a tractable optimization model, time is usually discretized and the quantities depend on the discrete time period $n \in \{1, \ldots, N\}$ of length $\tau \in \{\tau_1, \ldots, \tau_N\}$. A typical planning horizon is one day, divided in 24 hourly periods. In [2] it is pointed out that the discretization has another practical motivation. Namely, demand forecasts and electricity price tariffs are usually given for discrete and not continuous time. In the following we will highlight the time dependency of variables or parameters with a superscript n only when different periods are involved in an equation or constraint. Otherwise, the equation usually has to be imposed in every time period. In a static setting there is of course no time dependency to be highlighted. In any case, we introduce a flow variable q_a on each arc $a \in \mathcal{A}$. By allowing the flow to take negative values, a directed graph accounts for its both possible directions: a positive flow q_a on an arc $a = (i, j)$ means that it goes from i to j, while a negative value of q_a stands for a flow of amount $|q_a|$ from j to i. It is as well possible to allow only positive flow values and account for the directions with a binary variable.

From classical network flow problems one inherits the flow conservation constraints. For a node $i \in \mathcal{N}$ with demand d_i, which for the moment is assumed to be a constant, and the set of incoming and outgoing arcs, δ_i^- and δ_i^+, respectively, one has the linear constraint

$$\sum_{a \in \delta_i^-} q_a - \sum_{a \in \delta_i^+} q_a = d_i. \tag{1}$$

Sinks have a positive demand, which means that water actually leaves the network at those nodes. At sources, often called reservoirs in the water context, constraint (1) is usually not imposed. Otherwise, one can model d_i as a variable that can assume only non-positive values, possibly bounded from below. Of course, there can be nodes with zero demands. Sometimes all nodes with positive or zero demand are called junctions.

Next, one introduces the node potential variables h_i, $i \in \mathcal{N}$, representing the water pressure at a node. In water network optimization, pressure is conventionally measured as a height, or more precisely as the sum of the geographical elevation of a node above sea level or any other fixed reference point, called elevation head, and the so-called pressure head, that is an elevation difference according to the pressure of the water. The pressure head is proportional to the actual pressure when water is assumed to be incompressible, which is done in most applications. The node potential h_i is also called the hydraulic head at node i.

An important set of constraints arise from the fact that the two groups of variables introduced above are typically bounded. The absolute value of the flow is bounded by above due to the capacity of the arcs. For example, taking into account the flow's maximum velocity that is allowed in a pipe a, $v_a^{\max}$, the flow bound can be written [3] as

$$-\frac{\pi}{4} v_a^{\max} D_a^2 \leq q_a \leq \frac{\pi}{4} v_a^{\max} D_a^2, \tag{2}$$

where D_a is the diameter of pipe a. The node potentials have to stay between certain bounds in order to guarantee minimum and maximum pressure levels at the nodes. Usually, the node potentials are fixed at source nodes, reflecting the fact that at sources water is not pressurized, but it exploits a fixed geographical height.

2.2. Pipes

Typically, the majority of the arcs in a network represent pipes in which water is transported from one node to another, and this transportation is induced by different potentials at the nodes. The fundamental equation for a pipe $a = (i, j)$ is the head-loss equation, also nominated potential-flow coupling constraint in [4] (and we will use this term in the following), that is regularly of the form

$$h_i - h_j = \Phi_a(q_a), \tag{3}$$

where $\Phi_a : \mathbb{R} \to \mathbb{R}$ is a strictly increasing uneven function, concave on the negative half-axis of its domain and convex on the positive half-axis, see Figure 1. The induced flow is not linear in the potential difference because this kind of function accounts for the modeling of friction in the pipes. A positive flow as a function of the potential difference is strictly increasing but concave: higher flow values mean higher influence of friction. The other way round, for the same reason, a positive potential loss as a function of the flow is strictly increasing and convex. Equation (3) is also referred to as the potential-loss equation, because it describes the pressure, or equivalently the energy, loss along a pipe. It is easy to see that once constraints

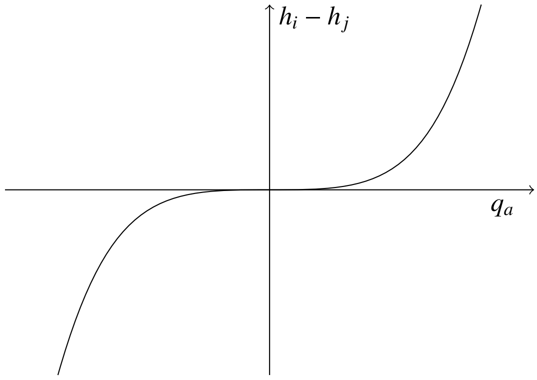


Figure 1: Potential-flow coupling function

(3) are embedded in a mathematical programming problem, the latter becomes non-convex. This is the case in most of the models we consider in this article.

Explicit forms of the potential-flow coupling equation are the so-called Hazen-Willliams equation,

$$h_i - h_j = \frac{\text{sign}(q_a)|q_a|^{1.852} \cdot 10.7 \cdot L_a}{k_a^{1.852} D_a^{4.87}}, \tag{4}$$

or the Darcy-Weisbach equation,

$$h_i - h_j = \frac{\text{sign}(q_a) q_a^2 \cdot 8 \cdot L_a \cdot f_a}{\pi^2 g \cdot D_a^5}, \tag{5}$$

both of which include some constants like the gravitational acceleration g, or ones depending on the pipe, such as the length L_a or the roughness coefficient k_a depending on the material of the pipe. In the denominator appears the diameter D_a of a pipe, which in optimal water network design becomes a discrete variable but stays a constant in the operation case. The friction factor f_a actually depends on the flow in a nonlinear manner. There are several approximation formulas, for details see again [2]. The most simplifying approximation neglects the dependency on the flow and thus treats the friction factor as a constant. This is done in most optimization models, see, e.g., [5, 6]. It can however be kept in the model, see, e.g., [7]. As well in the Hazen-Williams equation, the quantity corresponding to the friction factor does not depend on the flow (it depends on the diameter though). Another downside of the above head-loss equations can arise when studying their differentiability for $q_a = 0$. The second derivative of the Darcy-Weisbach function is discontinuous at the origin, while the Hazen-Williams equation is not even second order differentiable there. This can create problems when relying on derivative-based optimization methods as in [2], which is why therein is used some kind of second order polynomial approximation of the Darcy-Weisbach equation with constant friction factor. The Darcy-Weisbach equation is often found with the substitution $\text{sign}(q_a)q_a^2 = q_a|q_a|$.

The pipe models seen so far assume a constant flow in a pipe, which is natural in a stationary setting. When dealing with a dynamic problem, this can be modeled even more accurately. A pipe model that allows for varying flow inside a pipe is based on the so-called water hammer equations. This set of partial differential equations describes the variation of the state vector (q, h) in function of time and space,

$$\frac{\partial h}{\partial t} + \frac{c^2}{gA}\frac{\partial q}{\partial x} = 0, \tag{6}$$

$$\frac{\partial q}{\partial t} + gA\frac{\partial h}{\partial x} = -\lambda \frac{q|q|}{2DA}, \tag{7}$$

again with some coefficients among which the diameter D and the cross-sectional area A of a pipe and a friction coefficient λ, like before possibly depending on the flow q. Note that when one assumes constant pressure and flow over time, i.e., turns to the static case, constraint (6) implies constant flow in the pipe, and constraint (7) becomes the Darcy-Weisbach equation with $\frac{\partial h}{\partial x}$ approximated by $\frac{h_j - h_i}{L_a}$ and $f_a = \lambda$.

Here, just like the time dimension, that, as already mentioned, is usually discretized in optimization models, the same can be done in space, for example by implicit box schemes, see, e.g., [8]. However, a slight modification of the underlying graph model is necessary, see, e.g., [7]. In fact, the flow in a pipe $a = (i, j)$ is not constant anymore, so a single variable q_a is not appropriate. Choosing the length of a pipe as spatial step size in the discretization scheme results in one flow value at the beginning of a pipe, as well as one at the end. This can be modeled by intermediate nodes. For node $i \in \mathcal{N}$ let δ_i^- denote the set of incoming pipes and δ_i^+ the set of outgoing ones. Then, consider the additional nodes i_a, $a \in (\delta_i^- \cup \delta_i^+)$, which can be imagined to be attached to the graph between the original node and the corresponding incident arc. We then have two flow variables per pipe per time step, $q_{i_a}^n$ and $q_{j_a}^n$. With this, the discretized versions of (6) and (7) for a pipe $a = (i, j)$ become

$$\frac{h_i^{n+1} + h_j^{n+1}}{2\tau_n} - \frac{h_i^n + h_j^n}{2\tau_n} + \frac{c^2}{gA} \cdot \frac{q_{j_a}^{n+1} - q_{i_a}^{n+1}}{L_a} = 0, \tag{8}$$

$$\frac{q_{i_a}^{n+1} + q_{j_a}^{n+1}}{2\tau_n} - \frac{q_{i_a}^n + q_{j_a}^n}{2\tau_n} + gA\frac{h_j^{n+1} - h_i^{n+1}}{L_a} = -\frac{1}{2DA}\left(\frac{\lambda(|q_{i_a}^{n+1}|)q_{i_a}^{n+1}|q_{i_a}^{n+1}|}{2} + \frac{\lambda(|q_{j_a}^{n+1}|)q_{j_a}^{n+1}|q_{j_a}^{n+1}|}{2}\right). \tag{9}$$

While equation (8) is linear, (9) is not, just like its static counterpart (3).

2.3. Pumps

On the design side, the problem that is usually attacked disregards pumps and thus assumes that the underlying network is gravity-fed. In other words, a source node has a higher elevation than the nodes to which it induces a flow. If a network is fed with groundwater, so that gravity does not suffice, or also if the flow has to be transported over very long distances and hence loses too much pressure along an arc in order to satisfy the required minimum pressure level at the end node, certain node potentials have to be raised. This can be done by pumps.

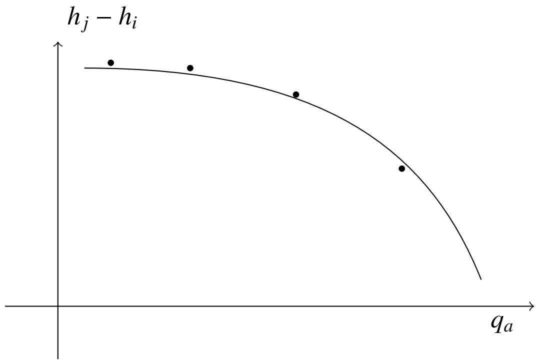


Figure 2: Fitted characteristic curve of a pump

Moreover, on the operational side pumps can serve to fill tanks for intermediate storage purposes (see below).
A pump is usually modeled as an arc $a = (i, j)$, so in particular there is a flow q_a passing through it. In comparison to pipes, pumps have a negligible length and thus the matter of constant or variable flow in a pump is not encountered. The flow through a pump is usually restricted in sign, that is, $q_a \geq 0$, allowing for flows from i to j only. Usually, the flow through a pump is considered semicontinuous, that is, either zero or in an interval $[\underline{q}_a, \bar{q}_a]$, with $\underline{q}_a > 0$. This already shows the need of a binary variable x_a for a pump, switching on the pump ($x_a = 1$) or switching it off ($x_a = 0$), i.e., acting like a closed valve ($q_a = 0$, see below).
In the context of gas networks, the authors in [4] present an elegant overall MINLP formulation for network flow problems exploiting a rather simple static model for potential raising elements, i.e., pumps in the context of water networks. The potential-flow coupling equation in [4] has the explicit form $h_i - h_j = \alpha_a q_a |q_a|^{\kappa_a}$, where α_a and κ_a summarize all the physical aspects of flow and pipe. The potential loss on the left-hand side is manipulated into a potential raise by multiplying the whole equation by minus one and adding a variable operating term,

$$h_j - h_i = -\alpha_a q_a |q_a|^{\kappa_a} + \beta_a y_a. \tag{10}$$

Here either the bounds of the variable y_a or the sign of the coefficient β_a have to be set in such a way that $\beta_a y_a \geq 0$. In other words, instead of losing some potential along the arc representing a pump, at its head node, the flow can have a potential value strictly higher than the one at its tail node. The same equation with $\beta_a y_a \leq 0$ can be used to model network elements that are able to reduce the pressure in some way.
A different pump model tries to replicate the characteristic curve of a pump. More precisely, for a pump $a = (i, j)$, the potential raise can be written as

$$h_j - h_i = \Delta H_a - \gamma_a^1 q_a^{\gamma_a^2}, \tag{11}$$

where the parameters ΔH_a, γ_a^1 and γ_a^2 are chosen to approximate empirical pump data. A typical characteristic curve is shown in Figure 2. There are also pumps that operate at variable speed $w_a \geq 0$. Basically, this quantity allows to shift the pump's characteristic curve in the plane and thus to manipulate the relation between potential raise and flow through the pump. The above equation becomes

$$h_j - h_i = w_a^2 \left(\Delta H_a - \gamma_a^1 \left(\frac{q_a}{w_a} \right)^{\gamma_a^2} \right). \tag{12}$$

Also for a variable-speed pump the parameters of their characteristic curve are estimated through empirical data of the pump working at its nominal speed ($w_a = 1$). Equation (11) describes fixed-speed pumps that work at their nominal speed only. Instead, second order polynomials are used to fit the characteristic curve of a pump in [6].
The obvious differences between the model approaches (10) and (11)/(12) can be explained by the discrepancy between network design and network operation. When designing a network, the pump to be installed might be chosen from a certain production series. A typical pressure raise that can be realized by the pumps of such a series is shown schematically in Figure 3. A very simple way to model this situation is actually equation (10). Things change when considering the task to operate an already existing pump in an existing network. This pump is already chosen and has its own specific characteristic curve, which can be modeled quite accurately, for example by equations (11) or (12), respectively.
Another important quantity in the context of pumps is their power consumption p_a, which often appears in the objective function. The power consumption can be modeled as proportional to the product of the pressure raise of a pump and the flow through it, which adds a nonlinear equality constraint as in [5]. Otherwise, an equation similar to (12) or again a polynomial of degree two is fitted to empirical pump data at nominal speed.
Pumps are part of the active elements in a network and thus they allow for some discrete decision, that is, turning them on or off. Being turned on or off has implications on two aspects, one of which is the already mentioned semi-continuity of the flow variables. The same is valid in an analogous fashion also for the speed of pumps w_a. The second aspect concerns the potential-flow coupling equations (10) - (12), that are to be imposed only when the pump is working ($x_a = 1$). When $x_a = 0$,

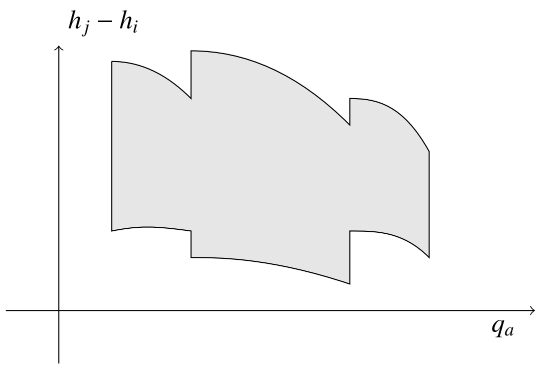


Figure 3: Operation range of a pump series

the pump is not working, and in this case it is usually regarded to act as a closed valve (see below). The potential values at the two ends of the arc are then uncoupled and do not follow any equation.

In order to avoid these combinatorial aspects, the authors in [2] make some modeling modifications. A single pump $a = (i, j)$ in a graph often models an aggregation of real pumps in the network. The individual pumps in such a pumping station $a = (i, j)$ have a common pressure raise $h_j - h_i$, but the flow q_a through the pumping station is the sum of the individual flows. Approximately, the pressure raise of a pumping station becomes a degree of freedom independent of the aggregated flow value. To accurately measure the power consumption of such a pumping station, additional considerations about its efficiency are made. But all in all, the benefit is that there are no equations that have to be turned on or off.

2.4. Valves

Another set of active elements is given by valves. As for pumps, a valve is modeled by an arc $a = (i, j)$ with negligible length. There are models for different types of valves with no unified terminology in the literature. The authors in [7] for example consider a total of four different types of valves. Valves can be used to actively block the flow completely or to reduce the pressure or the flow in its direction by a controlled amount. They are usually modeled by a binary variable x_a. As an example consider the (rather simple) valve type that can be used to separate or join two pipes, that is, to avoid that fluid can pass between the two of them. If $\underline{q}_a, \bar{q}_a, \underline{h}_i, \bar{h}_i, \underline{h}_j$ and $\bar{h}_j$ denote the upper and lower bounds of the flow and potentials, an open ($x_a = 1$) or closed ($x_a = 0$) valve state can be modeled by the big-M constraints

$$\underline{q}_a x_a \leq q_a \leq \bar{q}_a x_a, \tag{13}$$

$$(\underline{h}_i - \bar{h}_j)(1 - x_a) \leq h_i - h_j \leq (\bar{h}_i - \underline{h}_j)(1 - x_a). \tag{14}$$

A closed valve forces the flow to be zero and decouples the two potentials, while an open valve forces the potential values to be equal.

Again in [2], the authors model one specific valve type without auxiliary binary variables. Namely, the valve type considered allows to reduce the pressure by a controlled amount, but only in the direction of the flow. This behavior is guaranteed by the inequality

$$(h_i - h_j)q_a \geq 0. \tag{15}$$

2.5. Tanks

Tanks can make the operation of the network more flexible. In a dynamic setting, where the demand at consumer nodes can vary in time, water can be stored in a tank during a period of low demand and be extracted from it to satisfy peak demands. In the static setting of [5] tanks are modeled as nodes in the graph that have a variable demand, which can also be negative so as to account for a positive initial tank filling. The water in a tank is usually not pressurized, which means the pressure head is zero, so that the potential value h_i represents the elevation head only. Accordingly, in [5] the potential value of the tanks is kept fixed as for reservoirs. In a dynamic setting, variable potential values can be used to describe the filling level of the tank at different points in time, which can then be linked to the tank's variable demand of a time period. A simplified version of the model for tank $i \in \mathcal{N}$ in [2] is given, for example, by

$$\sum_{a \in \delta_i^-} q_a^n - \sum_{a \in \delta_i^+} q_a^n = e_i^n, \tag{16}$$

$$e_i^n = \frac{1}{\tau_n}\left(h_i^n - h_i^{n-1}\right) A_i, \tag{17}$$

with e_i^n denoting the variable volumetric tank inflow and A_i being the cross-sectional area of the tank. A quite detailed tank model that includes binary variables is found in [6]. Unless the discretized water hammer equations (8)-(9) are used, the tank filling equations constitute the most important point where subsequent time periods are coupled. Note that it is also possible, see, e.g., [7], to model tanks as arcs in the graph, although this is far less common and essentially equivalent.

3. Algorithmic aspects of MINLP

In this section we outline some algorithmic paradigms widely applied to MINLPs, both convex and non-convex. We actually show those techniques that are useful and have been applied to water network optimization. For a very complete survey on MINLP techniques the reader is referred to [9].

3.1. Nonlinear branch and bound

The most straightforward extension of MILP techniques to the nonlinear case is nonlinear branch and bound. As in branch-and-bound or branch-and-cut schemes for MILP, initially all integer requirements on the variables are relaxed and subsequently forced to be fulfilled by branching. The resulting node problems in a branch-and-bound tree are Nonlinear Programming problems (NLPs), which can be tackeld by a variety of different NLP-solvers. In general, to solve such problems to global optimality is NP-hard, so that this is rarely an option in a subproblem of a search tree. However, there are algorithms for solving NLPs to local optimality in polynomial time (for example interior point methods), and one usually restricts to that in the nodes of a nonlinear branch-and-bound tree. In the case of convex MINLPs, the node problems are convex NLPs and their locally optimal solutions are automatically globally optimal and can be computed in polynomial time. Thus, the optimal solution values of node subproblems can be used for pruning rules in the tree. However, as already mentioned in Section 2.2, for the mathematical models including constraints (3), the nodes' NLPs are non-convex instead. In that case, pruning a node might be based on an only locally optimal solution, while the globally optimal one is disregarded. In consequence, the solution obtained by the algorithm cannot be guaranteed to

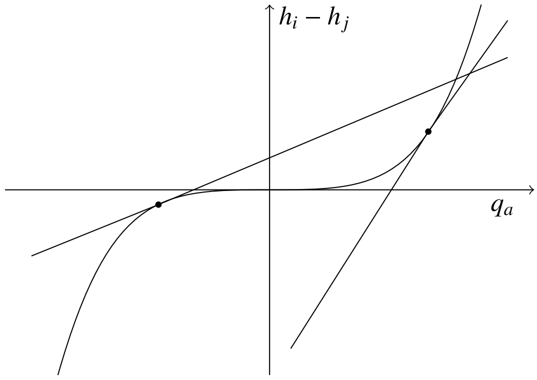


Figure 4: Outer approximation cuts of the potential-flow coupling function

be globally optimal either. Nevertheless, nonlinear branch and bound can be applied to non-convex MINLPs as an only heuristic solver. In the context of water network design this has been done in [3].

3.2. LP/NLP-based branch and bound

Another algorithmic framework for solving convex MINLPs to global optimality is the so-called LP/NLP-based branch and bound, which is a modification of the outer approximation algorithm. In outer approximation, the feasible region of an MINLP is approximated by a polyhedron that is formed by first order taylor approximations of the nonlinear functions at certain points, also called outer approximation cuts. Provided the nonlinear functions are convex, this results in a mixed integer linear relaxation of the original MINLP, which is solved to optimality. The first order taylor approximations of the nonlinear functions at the optimal solution are then used to refine the polyhedral relaxation. This process is repeated iteratively, which leads to a series of MILPs being solved. If the set of points of which the corresponding outer approximation cuts are added to the MILP is populated by the "correct" points, at some iteration the refined polyhedron will suffice to solve the MINLP to global optimality.

The above algorithmic scheme is able to provide globally optimal solutions to convex MINLPs, basically because the first order taylor approximations of convex functions do not cut off any part of the feasible region, while this is not true for non-convex functions. So also LP/NLP-based branch and bound leads to only heuristic solvers for non-convex MINLPs. The effect of outer approximation cuts on the potential-flow coupling equation (3) is depicted in Figure 4. (Actually, having a nonlinear equality constraint in this case, the failure of outer approximation cuts to approximate the feasible region can be seen even more drastically than for non-convex inequality constraints.)

Outer approximation is also referred to as a multi-tree method, because in each iteration an MILP is solved by exploring an entire search tree. At each iteration the (same) MILP is slightly modified and again solved by exploring another search tree from scratch. The LP/NLP-based branch and bound, which is implemented for example in `FILMINT` ([10]) and `BONMIN` [11], aims at reducing this approach into a single search tree. Instead of solving every single MILP to optimality and then adding the corresponding outer approximation cuts, in every integer feasible node of an initial MILP approximation, the corresponding NLP is solved to optimality and the outer approximation cuts corresponding to that optimal point are added. The Linear Programming (LP) relaxation with the newly added cuts is then resolved and the search continues.

Based on this algorithm and outer approximation cuts, a method to solve water network design problems to global optimality, despite containing the non-convex constraints (3), is developed in [12]. This is discussed in Section 4.

3.3. Spatial branch and bound

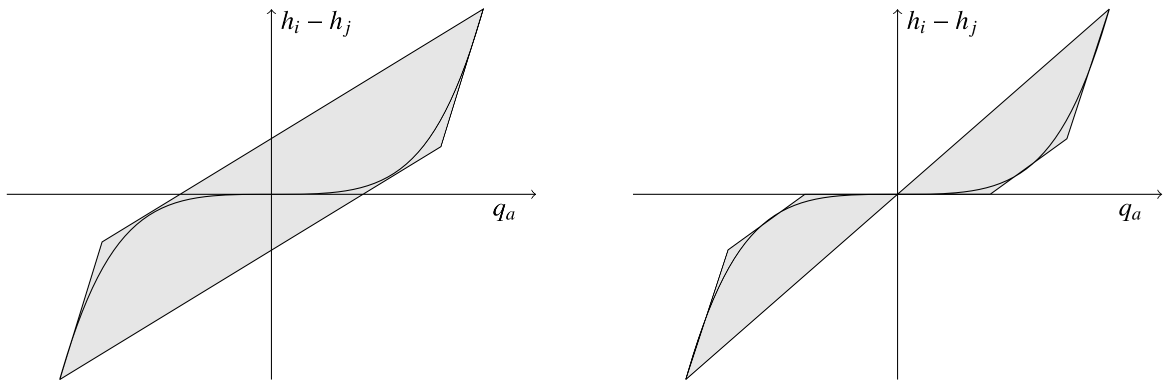
As mentioned, solving NLPs to global optimality is a NP-hard task in general. Anyway, one way to do so is to use spatial branching, a paradigm first investigated by the global optimization community. Integrating spatial branching with mixed integer branching techniques opens the possibility of developing general-purpose algorithms that can in principle solve non-convex MINLPs to global optimality. The basic idea of such an algorithm remains to iteratively divide the problem into subproblems and to solve (usually linear) relaxations of these. The division of subproblems though is done not only by branching on integer variables, but also by branching on continuous ones. In other words, in a subproblem of the branching tree the continuous domain of some nonlinear function is divided at some breakpoint into two smaller domains, thus creating two new subproblems. Provided the relaxation of a nonlinear constraint becomes tighter when the domain of the corresponding nonlinear function is reduced, spatial branching gradually refines the relaxations, see Figure 5. Branching is continued until finally the relaxations are tight enough to provide solutions that are ϵ-feasible for the original problem.

A way to obtain relaxations of subproblems is to use reformulation techniques aiming for a reformulation of all nonlinear functions into some "basic" functions. For these basic functions, linear relaxations are then readily available. The tightness of the relaxations and thus performance of the algorithm are highly dependent on the bounds on the domains of the nonlinearities. This fact makes an efficient domain propagation between subproblems essential.

Such an algorithm is implemented for example in the MINLP framework `COUENNE` [13]. Also the constraint integer programming framework `SCIP` [14] employs spatial branching if applied to MINLPs, see [15]. In water network optimization, `SCIP` has been applied in different contexts, as we will see further down.

3.4. Piecewise linear approximation

Another way to approach MINLPs is to approximate all nonlinearities by piecewise linear functions. The advantage is that a piecewise linear function can be modeled by linear constraints in mixed integer variables, which opens the possibility of applying MILP solvers to an approximated MINLP. It has to be clear though, that this is only an approximation of the original problem. Therefore, when the MINLP at hand is convex, there are



(a) (b)

Figure 5: Linear relaxation of the potential-flow coupling contraint and its refinement after spatial branching in the origin

more attractive methods (see above) towards which one usually orients. But when dealing with non-convex MINLPs, piecewise linear approximation becomes an intriguing alternative to expensive and sometimes ineffective MINLP methods. Especially in water network optimization, due to the presence of constraint (3) (among others), MILP approximations have experienced vast attention.

Piecewise linear approximations roughly work as follows. Imagine the domain of a univariate function to be partitioned into several intervals. That function can then be approximated by the line segments connecting the end points of these intervals (the break points), see Figure 6a. This simple point of view already makes clear that the accuracy of such an approximation depends on the number of break points.

There are several methods to model univariate piecewise linear functions, such as the so-called incremental method or the convex combination method. For a comparison of different piecewise linear modeling techniques, particularly applied to nonlinear network flow problems, see [16]. It is also in this context that topics related to special ordered sets of type 2 and therefore developed branching rules came up [17]. The piecewise linear modeling techniques have also been extended to multivariate functions, as becomes necessary, e.g., when attempting to approximate the function in (9) as done in [7] and [18]. Instead of intervals, the domain of such a function is usually partitioned into several simplices, and over each simplex the function is approximated by an affine function. Some kind of curse of dimension is encountered here, because the number of required simplices, that is directly connected to the number of auxilary binary variables in a mixed-integer formuation, is strongly growing with the dimension of the function's domain. Therefore, in this context we often find reformulation ideas that increase the number of variables and constraints, but decrease the dimension of the functions to be approximated.

Anyway, one has to keep in mind that all the above leads to an approximation of the original problem only. This means that the obtained globally optimal solution of the MILP is not necessarily globally or locally optimal for the MINLP. Even worse, it might not even be feasible to the nonlinear constraints. A way to estimate a priori the approximation error and to refine it until it stays inside some desired predefined error bound is proposed in [16]. The authors also show how to use piecewise linear functions to create relaxations of the MINLP instead of approximations only. Basically, the maximum error of an approximation over each simplex is overestimated as tightly as possible, and a continuous variable bounded by this maximum error is added to the function value. The resulting relaxation of this procedure is depicted in Figure 6b.

There has also been some effort to integrate MILP and NLP concepts. Very intuitively, by fixing the integer variables to the values of a globally optimal MILP solution, one could solve the remaining NLP to local optimality. This would guarantee at least the feasibility of the solution, but not necessary global optimality. For a more detailed idea of how to integrate MILP and NLP in the context of piecewise linear approximations in the water context see [19].

Also interesting is the extension of piecewise linear relaxations to an algorithm where the approximation is refined dynamically in order to compute a globally optimal solution for the originally non-convex MINLP. This approach has been investigated in the context of gas networks in [20].

4. Convex leaf problems

As mentioned earlier, the combinatorial optimization problems surveyed in this article are in the problem class of non-convex MINLPs. However, a recurring feature is the possibility of solving certain subproblems by convex optimization methods.

Consider a static network as in Section 2 with no active elements and with fixed pipe diameters, that is, a network with a flow variable q_a for each arc and a potential variable h_i for each node. Denote the set of source nodes by $\mathcal{N}^{src}$ and assume further that the potential values are fixed to some value h_i^{src} at source nodes $i \in \mathcal{N}^{src}$. Finally, assume that the flow conservation constraints (1) hold at each non-source node for a given set of fixed demands, and that each arc satisfies a potential-flow coupling equation like (3), where the only requirement about the function $\Phi_a : \mathbb{R} \to \mathbb{R}$ is strictly increasing monotonicity. Mathematically, this gives rise to the feasibility problem

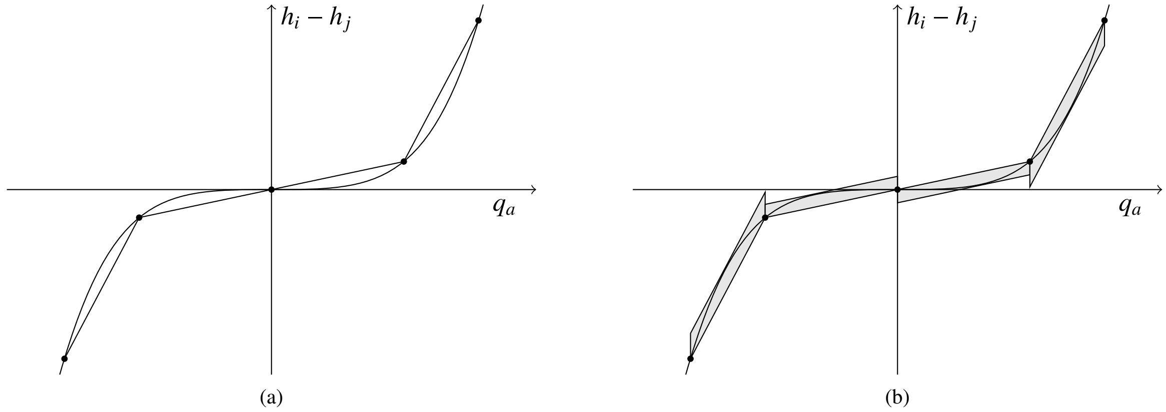


Figure 6: Piecewise linear approximation and piecewise linear relaxation of a potential-flow coupling constraint

$$\sum_{a\in\delta_i^-} q_a - \sum_{a\in\delta_i^+} q_a = d_i \qquad \forall\, i \in \mathcal{N} \setminus \mathcal{N}^{src}, \tag{18}$$

$$h_i - h_j = \Phi_a(q_a) \qquad \forall\, a \in \mathcal{A}, \tag{19}$$

$$h_i = h_i^{src} \qquad \forall\, i \in \mathcal{N}^{src}, \tag{20}$$

$$q_a \in \mathbb{R} \qquad \forall\, a \in \mathcal{A}, \tag{21}$$

$$h_i \in \mathbb{R} \qquad \forall\, i \in \mathcal{N} \setminus \mathcal{N}^{src}. \tag{22}$$

A reformulation of the above as a strictly convex optimization problem was originally studied in [21]. It is shown that the solution space of the above set of $|\mathcal{A}| + |\mathcal{N}|$ equations in the same number of variables has a unique solution. A slightly different version of the problem is studied in [4]; the flow conservation constraint is imposed at every node but the node potentials are not fixed at sources. The number of variables and equations stays the same, and the existence of a solution is given for balanced flows only, that is, $\sum_{i\in\mathcal{N}} d_i = 0$. In that case, one flow conservation constraint at a node becomes redundant and the solution is a one-dimensional subspace. More precisely, the vector of flow variables q is unique and the vector of potential values h is on a straight line in $\mathbb{R}^{|\mathcal{N}|+|\mathcal{A}|}$. It is easy to deduce that if one fixes the potential at any node, for example a source node, also the potential values become unique here. The most crucial assumption leading to the striking argument in a potential proof of the uniqueness in both versions is just the monotonicity of the function Φ_a. More precisely, a potential duplicity of flows can arise in the presence of circles only. By a circle in a graph we mean the existence of two different paths between two nodes. As an illustrating example consider a simple (sub-)graph as in Figure 7, see also [22]. (Actually, this is a multigraph, the two arcs a_1 and a_2 are to be thought of as two different paths from node i to node j.) Flow conservation implies that the relation between the flow values q_{a_1} and q_{a_2} is a line with negative slope, say $q_{a_2} = \Delta - q_{a_1}$ for some $\Delta \in \mathbb{R}$. From the potential-flow coupling constraints (3) we get

$$\Phi_{a_1}(q_{a_1}) = \Phi_{a_2}(\Delta - q_{a_1}). \tag{23}$$

The function on the right-hand side of (23) is now strictly decreasing in q_{a_1} and as such has exactly one intersection point with the strictly increasing function on the left-hand side. Such considerations and the resulting methods are implemented in widely used software packages like EPANET [23], designed for numerically calculating flow and potential values in pressurized water networks.

As already mentioned, problem (18)-(22) represents a network with only pipes with fixed diameters and relaxed bounds on the variables. However, when we allow for discrete variable diameters, the problem occurs also as leaf problem with relaxed bounds in a branch-and-bound tree, i.e., when all integer variables are fixed. In such a situation, a subproblem of this type can in principle be solved by a NLP-solver that solves to local optimality only. The fact that the bounds are relaxed does not create a problem in the exploitation of this property, because their satisfaction can easily be checked after having obtained a unique solution. Also when the potential values are not unique but located on a line, it is easy to check whether there is one solution inside the bounds. Also in other situations uniqueness of solutions in subproblems can occur. Consider, for example, a network with working pumps that follow equation (11). Here, the pressure gain $h_j - h_i$ on arc $a = (i, j)$ is strictly decreasing in the flow q_a, meaning that the pressure loss is strictly increasing. At least this is true in the domain of q_a, so when relaxing the bounds maybe the function has to be extended conveniently, in such a way that it stays strictly increasing on $\mathbb{R}$. However, also a network with such pumps has a unique solution in its leafs, i.e., when the discrete decisions on which pumps are actually working and which are switched off have been taken.

The situation becomes different when pumps follow models (10) or (12). What actually happens here is that a degree of

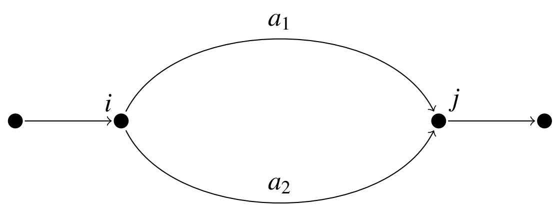


Figure 7: Simple example of a network containing a circle

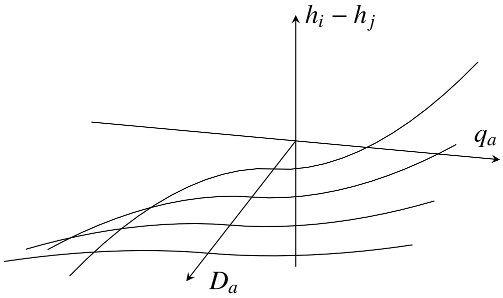


Figure 8: The Hazen-Williams equation for some discrete diameters

freedom is added. In the equation for a fixed-speed pump (11) the pressure raise is uniquely determined by the unique flow through the pump. This is not true for variable-speed pumps, where the pressure raise can be influenced by variation of the speed w_a. The same happens for model (10), where the pressure raise can be influenced by variation of the variable y_a independently of the flow q_a. Uniqueness or just convexity gets lost here. Staying with the previous example, imagine the arc a_2 in Figure 7 to be a pump obeying equation (10). Equation (23) then becomes

$$\Phi_{a_1}(q_{a_1}) = \Phi_{a_2}(\Delta - q_{a_1}) + \beta_{a_2} y_{a_2} \tag{24}$$

for some $\beta_{a_2} \in \mathbb{R}$, which is clearly representing a non-convex subspace of $\mathbb{R}^2$. A remedy, just like in [4], is to discretize the degree of freedom, i.e., to allow the variables y_a or w_a to take values in a discrete set only. Then, when they are fixed to one of these values in a leaf of a branch-and-bound tree, the functions describing the pressure loss are strictly increasing in the flow q_a and uniqueness or convexity can be exploited.

We believe that there is a lot of potential in the property of uniqueness or convexity, which has partially been exploited in water network optimization as we will see later. It might also be a key feature when developing techniques for discretized dynamic problems in water network optimization. For example, one can prove uniqueness of the solution when tanks are coupled by an equation of type (17) in a passive network with relaxed bounds considered over several discrete time periods.

5. Solving water network optimization problems

In this section we show in more detail what kind of approaches were presented in the literature for solving water network design problems, on the one hand, and water network operation problems, on the other hand.

5.1. Optimal design of water networks

On the design side, one specific formulation together with a set of literature and real-world instances have been studied by several authors. As anticipated, active elements as pumps and valves and also tanks are disregarded. In return, we can choose the diameter for each pipe from a discrete set. The formulation basically consists of the flow conservation constraint (1) for each non-source node and the Hazen-Willliams equation (4) for each pipe. The diameter D_a on a single pipe a is a variable constrained to belong to a discrete set $\{D_{a,1}, \dots, D_{a,r_a}\}$. The Hazen-Williams equation for some discrete diameters is depicted in Figure 8.

To each diameter $D_{a,i}$ is associated a positive unit length cost $C_{a,i}$ in such a way that costs increase with the diameter. Finally, there are lower and upper bounds on the flows, cf. (2), as well as lower and upper bounds on the node potentials, and potentials at source nodes are fixed. In an MINLP formulation, the membership of the diameter to a discrete set can be represented by introducing additional binary variables $X_{a,i}$, $i = 1, \dots, r_a$ and using SOS-1 type equations,

$$\sum_{i=1,\dots,r_a} X_{a,i} = 1, \tag{25}$$

$$\sum_{i=1,\dots,r_a} D_{a,i} X_{a,i} = D_a, \tag{26}$$

in which case the objective function becomes $\sum_{a\in\mathcal{A}} L_a \left(\sum_{i=1,\dots,r_a} C_{a,i} X_{a,i}\right)$. As seen in Section 4, if the diameters were fixed, this would result in a convex (feasibility) problem. This does not hold for a variable diameter, especially not when it is discrete. As mentioned, there is a set of 9 instances for the design problem, out of which 4 are smaller literature instances and 5 are larger real-world ones representing water networks of three Italian cities, one of which is counted three times due to three different diameter sets. The characteristics of the instances are subsumed in Table 1. The set of available diameters is actually the same for each pipe in each instance, i.e., r_a is constant for each $a \in \mathcal{A}$.

The nonlinear branch-and-bound algorithm implemented in `BONMIN` [11] was applied to these instances in [3]. Remember that this acts as a heuristic solver for non-convex MINLPs. Several amendments to the model, on the one hand, and to the algorithm itself, on the other hand, were made. In [3] for example a fitted polynomial $C_a(D_a)$ is used as objective function instead of the sum $\sum_{i=1,\dots,r_a} X_{a,i} C_{a,i}$ in order to get a smooth function. Moreover, this function usually produces tighter bounds because the optimal cost value for pipe a of some NLP relaxation is a point on the graph of $C_a(D_a)$ instead of just a point in the convex hull of the points $(D_{a,1}, C_{a,1}), \dots, (D_{a,r_a}, C_{a,r_a})$, see Figure 9. However, the polynomial is not chosen to be an exact fit in the arguments $\{D_{a,1}, \dots, D_{a,r_a}\}$, which is why the fitted objective is only used

instance name	#nodes	of which sources	#pipes	#diameters
`shamir`	7	1	8	14
`hanoi`	32	1	34	6
`new york`	20	1	21	12
`blacksburg`	31	1	35	11
`foss_poly_0`	37	1	58	7
`foss_iron`	37	1	58	13
`foss_poly_1`	37	1	58	22
`pescara`	71	3	99	13
`modena`	272	4	317	13

Table 1: Characteristics of water network design instances

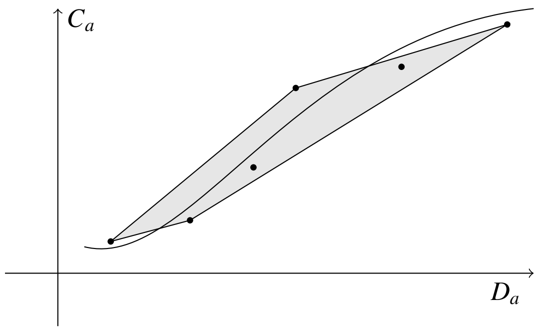


Figure 9: Fitted polynomial and convex hull of diameter costs

to guide the search in the tree, while the real objective is used to calculate the cost of integer feasible solutions. This is a nice example of how problem specific solution paradigms can give rise to enhancements of general-purpose solvers. Indeed, the option of working with two objective functions was added to the implementation of `BONMIN` afterwards. Also the implementation of proper SOS-1 branching in `BONMIN`, which can be used instead of the binary requirement of the variables $X_{a,i}$, was stimulated by the water network design application.

A modified LP/NLP-based branch-and-bound framework that exploits the convexity structure presented in Section 4 is proposed in [12]. This algorithm is then exact for the non-convex design problem. There are three crucial points in the approach. First, each arc in the network is cloned as many times as there are diameters available on it. One therefore gets much more potential-flow coupling constraints, but each of them is given by a univariate function (because the diameter on a cloned arc is fixed). Second, for each arc, it is then kept record of the flow direction by explicitely introducing an additional binary variable indicating the direction. The gain of both reformulations above is that the resulting non-convex constraints can easily be relaxed to two convex constraints, because only an either concave or convex part of the function Φ has to be regarded. For example, for a positive flow q_a with upper bound $\bar{q}_a$, these two constraints are precisely

$$h_i - h_j \geq \Phi_a(q_a) \quad \text{and} \quad h_i - h_j \leq \frac{\Phi_a(\bar{q}_a)}{\bar{q}_a} \cdot q_a. \qquad (27)$$

This situation is depicted in Figure 10.

In this way, in [12] a convex MINLP is obtained that is a relaxation of the original MINLP. A branch-and-cut scheme as in an LP/NLP-based branch-and-bound algorithm is applied to such a convex MINLP relaxation. Instead of solving the associated NLP in integer feasible nodes, and this is the third crucial point, the unique solution to the feasibility problem stated in Section 4 is found. If this unique solution additionally satisfies flow and potential bounds, the node is feasible for the original non-convex MINLP. In this way a stronger condition, directly related to the original MINLP, is tested instead of just solving the sub-NLP of its convex relaxation. If the bounds are violated, the node is eliminated from the search tree by some cut. Note that a globally optimal solution to the non-convex MINLP is found since the objective function depends on the integer variables only.

5.1.1. Computation in the design case

Here we want to present the computational results that were reported in conclusion with the two algorithms presented in the previous section applied to the described water network design instances. We also include the results of applying plain `SCIP`, i.e., spatial branch and bound, to these instances, without any exploitation of problem structure, reported in [25, Sec. 8.2], and updated in [24]. The results of these three different approaches are shown in Table 2.

All reported results were obtained on different computers and with respect to some time limit and in one case with respect to an additional memory limit in terms of branch-and-bound nodes. The machine characteristics are given in the respective column. The best lower and upper bounds found are reported. For the `BONMIN`-based approach in the second column we obviously have no lower bounds due to the heuristic nature of the approach. The algorithm in [3] reached the time limit in all except the first instance. The results of the exact approach of [12], for brevity called `LP/CVXNLP`, are given in columns 3 and 4. Here, an optimal solution is found in all of the four smaller literature instances, as a side effect certifying the optimality of the heuristic solutions in column 2. For the larger instances, the algorithm `LP/CVXNLP` is not able to terminate within the specified limits, but in any case finds feasible solutions, that, except for the instance `pescara`, are worse than the solution found by the `BONMIN`-based algorithm. Taking into account the spatial branch-and-bound algorithm of `SCIP` in columns 5 and 6, there appear some inconsistencies. The reported optimal solution of the small instance `blacksburg` is smaller than the optimal solution found by `LP/CVXNLP`. In addition, the optimal solutions of `foss_poly_0` and `foss_iron` are below the lower bounds given by `LP/CVXNLP`. We can only explain these inconsistencies with numerical issues in either of the two algorithms. One source of these issues might be the reformulation techniques used in `SCIP`, which sometimes produce slight infeasibilities [26].

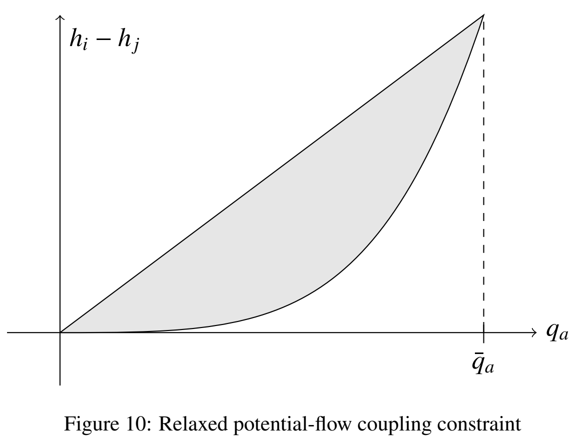


Figure 10: Relaxed potential-flow coupling constraint

instance name	BONMIN-based [3] *time limit 7,200 sec.* *single core 2.4 GHz* *1.94 GB RAM*	LP/CVXNLP [12] *time limit 14,400 sec.* *mem. limit 165,000 nodes* *dual core 2.63 GHz* *3 GB RAM*		SCIP [24] *time limit 10,000 sec.* *dual core 3.2 GHz* *48 GB RAM*	
	ub	lb	ub	lb	ub
shamir	419,000.00	419,000.00	419,000.00	419,000.00	419,000.00
hanoi	6,109,620.90	6,109,620.90	6,109,620.90	6,109,620.90	6,109,620.90
new york	39,307,799.72	39,307,799.72	39,307,799.72	23,363,470.00	39,307,800.00
blacksburg	118,251.09	118,251.09	118,251.09	116,945.00	116,945.00
foss_poly_0	70,680,507.90	70,062,509.72	71,741,922.90	67,559,218.00	67,559.218
foss_iron	178,494.14	177,515.24	178,494.14	175,992.00	175,992.00
foss_poly_1	29,117.04	26,236.17	31,352.87	28,043.86	28,043.86
pescara	1,820,263.72	1,700,108.17	1,814,271.91	1,639,746.00	1,938,885.00
modena	2,576,589.00	2,206,137,43	4,191,445.38	2,115,898.00	N/A

Table 2: Reported computational results for the water network design instances

5.2. Optimal operation of water distribution networks

The full-scale problem of optimal water network operation appears to be a rather hard task. By full-scale we mean being based on a time (and space) discretized formulation. To the best of our knowledge, there is no successful solution for this complete form in the literature. Three simplifications are proposed, which we will present in the following. Unfortunately, there does not seem to be an unified test set, which makes a concise comparison difficult.

The first simplification is obtained in [5] by dropping the time dimension and regarding a static operation problem. Such a problem may arise as subproblem in the full-scale task, possibly useful in heuristic approaches to the latter. At the basis is a network model with fixed-speed pumps following equation (11), valves that can be closed or reduce the pressure in the direction of the flow, and tanks with a fixed initial filling level. The pipes have a fixed diameter and the potential-flow coupling equation is given by (5). The objective function is the sum of the cost of water purchased at source nodes and the cost of the power consumption of pumps. An innovation of the study is the introduction of the concept of real and imaginary flows. It is observed that due to valves, it is possible that actually no water is present at certain nodes. Enforcing the potential-flow coupling constraint on an arc that is incident to such a node, which means inducing a flow on that arc, is wrong. If no water is present at a node, no flow emerging from it can be induced. The concept of real and imaginary flows is modeled with the help of additional binary variables. Some interesting preprocessing steps for that model are also presented, one of which can be applied to sequences of pipes and valves. Due to the presence of valves that can reduce the pressure in the direction of the pipe, the pressure-loss of an entire pipe-valve-sequence is not described by the Darcy-Weisbach equation itself but by some relaxation of it, see Figure 11. With some extra effort, this union of two convex sets can be modeled by convex constraints. Again, as in [12], the potential-flow coupling function's property of being "half-concave" and "half-convex" is exploited.
The resulting MINLP is solved by using SCIP, testing the approach on two real-world networks, of which the smaller one is schematically represented in Figure 12. The approach is tested on different scenarios in the two networks, given by different initial tank fillings and the forecast demands of different time windows of a day. The larger of the two networks consists of 88 nodes and 128 arcs, resulting in a program with about a hundred binary variables (without presolve). The most difficult scenario requires about half an hour of computing time. When applying the presolving steps, the computation times reduce drastically. All of the tested scenarios are solved within less than two minutes of computation time, on average much faster. A natural question to ask is how the solution method in [5], that is, the spatial branch-and-bound algorithm of SCIP, works on the full-scale operation problem. The authors reported that already going up to the full-scale problem with only two or three time periods is troublesome with the - at that time current - version of SCIP [27].

Another simplification of the full-scale operational problem is obtained by piecewise linearly approximating the

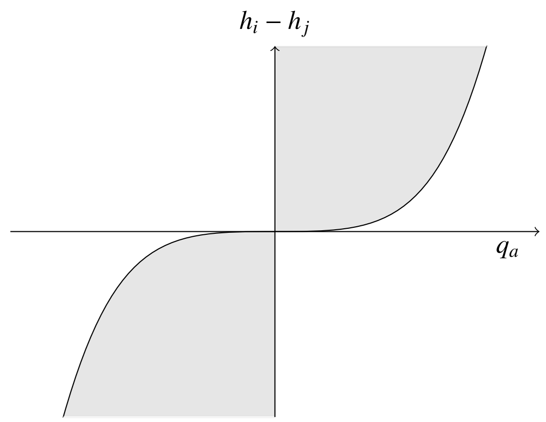


Figure 11: Pressure loss of a pipe-valve sequence

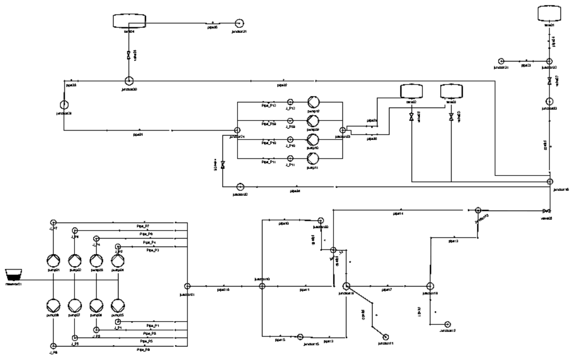


Figure 12: Schematic representation of a typical network

nonlinear functions in the time discretized model, as shown in [7] and [18], or in more detail in [28]. The model therein mainly consists of flow conservation constraints (1) adapted to time-dependent variables, potential-flow coupling constraints (8) and (9) and pump equations (12). As mentioned, four different types of valves are modeled and two types of constraints on each tank, that are modeled as arcs, are imposed. It is also accounted for modeling a terminal filling level of each tank and for the technical requirement of "breathing", that is filling and emptying the tank completely for a given number of times over the considered time horizon. Finally, there are additional linear constraints that account for minimum runtimes and downtimes of pumps.

Preprocessing techniques, see Section 3.4 and [16], are used to approximate all nonlinear functions within a controlled error bound of 10^{-2}. Some results on the comparison between the different models for piecewise linear functions especially in the water network context are reported in [16]. The winning method therein is the incremental method, which is also the method of choice in [7] and [18]. The tests are conducted on three networks of varying size, optimized over a time horizon of either four hours divided into 12 time steps or one day divided into 24 hourly steps. It is interesting to note that there is some flexibility of what is represented by the objective function, according to the underlying application. So, in some tests, the objective function is chosen to be the minimization of the number of tanks with a filling level below a certain limit, so as to maximize the supply guarantee of the network. In other cases, the overall power consumption is minimized. As we have seen earlier, also the purchase cost of water at source nodes can be taken into account in the objective function. The largest considered network consists of 25 arcs, resulting in an approximating MILP with almost 11,000 binary variables, for which an MILP solver finds an optimal solution within 694 CPU seconds. Further computational results can be found in [28, Sec. 8.1].

A MILP approach through piecewise linear approximations is also presented in [6]. Besides the Darcy-Weisbach equation there are second order polynomials that represent the characteristic curve of pumps with fixed-speed only, while the empirical power consumption of a pump is fitted by (linear) polynomials of degree one. The objective function is again a combination of the cost of purchased water and the power consumption of pumps.

The nonlinear univariate functions are piecewise linearly approximated by a modification of the convex combination method. This modification was proposed in [29] and observes that in SOS-2 situations as for piecewise linear approximations, the required number of auxiliary binary variables is actually only logarithmic in the number of approximating line segments. The formulation is further strengthened by adding valid inequalities.

In addition, the authors actually use a piecewise linear relaxation as described in [16] rather than an approximation, see again Figure 6b. The test network consists of 30 arcs, and the time horizon of one day is divided into 5 time steps. The reported results consider linear relaxations based on a number of approximation intervals ranging from 2 to 8. An MILP solver terminates at an optimal solution in the range of less than 30 seconds.

Computational results obtained with the very detailed model developed in [2] are reported in the follow-up paper [30]. As stated at the end of Sections 2.3 and 2.4, there are ways to approximate the MINLP model without introducing binary variables. For their large network with 1,481 nodes and 1,935 arcs arising from the drinking water distribution network in the city of Berlin, the authors apply purely nonlinear programming techniques without any kind of branching, thus no search trees are explored. Due to the non-convexities that are still present, this leads again to only locally optimal solutions of the problem with discrete aspects neglected.

5.3. *Unified modeling of design and operation*

As mentioned before, the inherent difference of the design and the operation problem is the contrast between static and dynamic modeling. Dynamic modeling can make sense under two conditions. First of all, considering a dynamic model is useful only if exogenous parameters that have impact on the decision variables, like the demand patterns of consumers, naturally change over time. In such a situation, neglecting time completely and taking for example mean values to express these parameters results in an accuracy loss. Of course, also a discretization leads to mean values, but nevertheless these mean values are based on single time windows. The second condition is that it has to make physical sense to take different decisions at different points in time. A pump, for example, can be switched on and off over time. For design problems in the form that we discuss them here, this is hardly the case. The variables in our model represent the decision to build a pipe with some diameter, which is not something that is easily reversed within a reasonable time horizon.

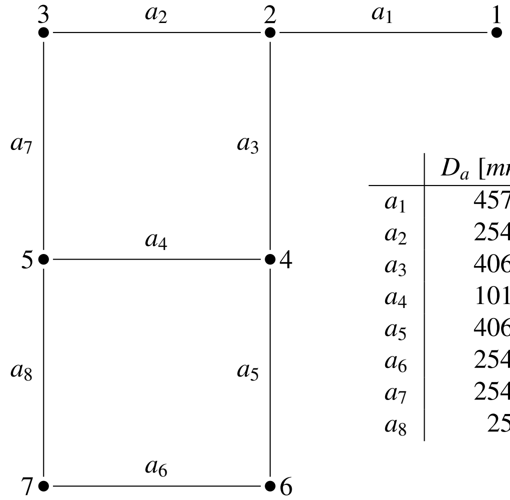
Nevertheless, even when neglecting the time dependency, it is quite natural to seek for an approach that from the modeling point of view handles both design and operation problems. This is done in [4] and [31] in the context of gas networks. The basic structure of that static model can in principle be applied to

water networks as well. Gas transmission networks are considered, with flow conservation constraints at the nodes and arc equations of type (10), where the operational component $\beta_a y_a$ is forced to be zero for pipes, positive for compressors (the equivalent of pumps in gas networks) or negative for pressure regulators. The striking point is that variable y_a is modeled as a discrete variable. This leads to the fact that the leaf problems in a search tree that branches on the integer variables are convex optimization problems, as shown in Section 4.
The first paper [4] is of operational type. Fixed entry and exit flows into and out of the network at certain nodes are given, and the task is to determine if the network is able to satisfy this scenario, also called "nomination" in this context. Thus, a feasibility problem without the minimization of power consumption or anything else is solved. The convex leaf problems are therefore solved by an NLP solver via different relaxation strategies. For example, one possibility is to relax the bounds on the variables q and h and minimize the bound violation. A leaf is then feasible if the optimal value of the relaxed problem is equal to zero. Networks with up to almost five hundred pipes are handled, although in the computations variables y_a's are relaxed as continuous. Hence the procedure is not able to proof infeasibility, but only the feasibility of a scenario. However, in the considered tests infeasibility did not occur.
The follow-up paper [31] treats the so-called topology optimization problems for the same type of gas networks. Based on the same equations, the model is now allowed to extend a network (that is infeasible for some scenario) by choosing on each arc exactly one of a discrete number of parallel network elements with different characteristics in a cost-minimal way. In principle, these elements can contain also pumps, but as a special case is contained the problem of choosing on each arc exactly one pipe out of a discrete set of pipes with varying parameters, i.e., the classic water network design problem. Again the convex leaf problems are solved through a relaxation. If such a relaxed leaf problem is infeasible for the original MINLP, a cut is derived that is based on information from the nicely interpreted dual problem of the relaxed leaf problem. Some tests were conducted on networks with only pipes, and it is shown that the cuts derived from the relaxed leaf problems can significantly reduce the computing times. This again underlines the potential that lies in the exploitation of the leaf problems.

6. Discussion

We have surveyed a class of challenging optimization problems and the attempts to solve them. The problems exhibit some differences among them, but all have in common the underlying nonlinear network flow model. The potential-flow coupling constraint is present in almost all formulations (and if not, its dynamic extension (8)-(9) is), and this constitutes one of the main difficulties.

We would like to discuss a very simple computational example. Consider the water network design instance `shamir` from Section 5.1. The network is depicted in Figure 13. Node 1 is the only source node and water is transported from there



	D_a [mm]
a_1	457.2
a_2	254.0
a_3	406.4
a_4	101.6
a_5	406.4
a_6	254.0
a_7	254.0
a_8	25.4

	$i = 1$	2	3	4	5	6	7
$\underline{h}_i$ [m]	210	180	190	185	180	195	190

Figure 13: The `shamir` network

to all the other (consumer) nodes. In that figure are also listed the diameters on each pipe, that we consider as fixed for the moment. This diameter set actually represents the optimal diameter set of the water network design problem, which on this small instance can be determined by any of the three methods presented in Section 5.1.1. Since this diameter set is optimal, it is in particular feasible, i.e., leads to a NLP-feasible (and unique) solution (q, h) that satisfies the flow conservation constraints, the potential-flow coupling constraints and the bounds on flow and pressure, and can be determined by any (also local) NLP solver.

The example helps to illustrate the physical bottleneck in many subproblems. This bottleneck is actually constituted by the lower pressure bounds. For example, decreasing the diameter on arc a_6 to the next available discrete one from the water network design instance would result in a network flow that does not violate the capacity on that arc, but the lower bound on the pressure of node 7. This again can be certified by any NLP solver. The same effect can be obtained by reducing the diameter on several other arcs.

Another characteristic of the underlying nonlinear network flow model is given by the property described in Section 4. We have seen that in some situations a nonlinear network flow in a subproblem is unique, provided it is feasible at all. It seems sensible to compute this solution by a convex NLP solver in polynomial time, instead of exploiting other methods. Take for example the classic formulation for the water network design problem described in Section 5.1. It turns out, that an MILP approach via piecewise linear approximations performs terribly bad. Indeed, even when linearizing the Hazen-Williams equation with only a few linearization points, no feasible solution is found by an MILP-solver within a reasonable time limit on

medium size instances. This is because the subproblems with fixed integer variables corresponding to the diameter choices, i.e., subproblems, that are encountered during a tree search, are either infeasible or have exactly one solution (note that as well the linearized version of the potential-flow coupling constraint is strictly increasing). Certifying infeasibility or searching this unique solution with a MILP formulation cannot be a good idea.
The same conceptual problem occurs if the subproblem with an at most unique solution is solved by spatial branching. A feasible solution will be found in at most one of the many branches. An additional difficulty in design problems often arises from the fact that the objective function is constant in such a subproblem, since it only depends on the diameter chioces. This results in exploring a sub-tree without the guidance of the objective function.
We conclude that in this situation, the most promising thing to do is to exploit the uniqueness of the leaf problems by some combination of integer branching and nonlinear programming techniques.

We now turn to the above example in order to shed some light on a side effect of MILP approximations of the potential-flow coupling equation. Imagine to have such an approximation for the `shamir` network with diameters fixed to the optimal diameter set, as above. Since the potential-flow coupling function is convex in $|q_a|$, the energy loss will be overestimated by this approximation, cf. Figure 6a. The worse the approximation is, the higher will be this overestimation. In fact, when fixing the diameters to the optimal diameter set and using five linearization points per pipe in a MILP model for our example, any MILP solver certifies the infeasibility of the optimal (and thus NLP-feasible) diameter set. The bottleneck are again the lower pressure bounds, this time at nodes $3, 5, 6$ and 7. So, MILP approximations lead to conservative solutions in general, because they overestimate the energy loss in a network. The worse the approximation is, the more conservative gets the solution. If one keeps the approximation coarse as to keep the number of auxilary binary variables low, fewer diameter configurations become feasible, which is why it becomes harder to find feasible solutions in the branch-and-bound tree. Refining the approximation augments the number of binaries and thus slows down the MILP solver.

At this point, it is possible to illustrate the role of pumps. Abstracting from the original physical setting of the `shamir` instance, imagine that there is a pump associated with arc a_3, modeled by the equation

$$h_4 - h_2 = -\frac{\operatorname{sign}(q_{a_3})|q_{a_3}|^{1.852} \cdot 10.7 \cdot L_{a_3}}{k_{a_3}^{1.852} D_{a_3}^{4.87}} + y, \qquad (28)$$

where $y \in \mathbb{R}_+$, cf. equation (10). Obviously, the network with fixed optimal diameters still has a NLP-feasible solution. But also decreasing the diameter on arc a_6 leads to a NLP-feasible solution now, namely with $y > 0$. Adding another such virtual pump on arc a_2 turns feasible also the above mentioned MILP approximation of the optimal diameter set with five linearization points. In this case, pumps could also be placed on arcs a_2 and a_5 or just on arc a_1.
In other words, pumps are somehow able to compensate for the approximation error made by MILP approximations. This could lead to computational advantages by producing more quickly diameter configurations that, tested a posteriori by solving convex leaf problems, show no need of pump usage. Of course, this depends on the cost of the pumps that is, however, somehow difficult to evaluate in a static framework like the design one. Namely, taking decisions on pump usage in the design context might imply operation problems that are feasible only if the pump is always used. Thus, the cost saved for installing pipes might be spent for the continuos use of pumps.
An interesting paradox is encountered in this context. Imagine to have some water network optimization problem for which all subproblems with fixed integer variables possess the convexity or uniqueness structure described in Section 4. Again, take for example the classic water network design problem and imagine to have pumps modeled by equation (28) in the network. In order to be able to use the above mentioned algorithmic combination of integer branching and explotation of uniqueness, one would have to discretize the variable y as in [4]. Otherwise, the introduced pump destroys the convexity of the subproblems. This is not true for an MILP approximation of the problem. Because everything is linear in such a model, the additive y does not destroy any convexity. On the contrary, it acts as a slack variable and can make things easier. Discretizing the variable y instead would again lead to the same situation as in a MILP approximation of the water network design problem without pumps: a branch-and-bound tree where subproblems solveable in polynomial time are instead attacked by integer branching.

Another aspect related to the role of pumps is more of algorithmic nature. As seen above, they actually have the ability to augment the number of feasible integer configurations in a tree search. It is in general easier to find feasible solutions. This algorithmic advantage can be seen directly. In an equation like (28), y acts as a slack variable. It becomes easier to find a solution somewhere above the graph of the Hazen-Williams function instead of exactly on that graph. Another piece of evidence of this can be found in [5]. One of the several effects of presolving therein is that some Darcy-Weisbach constraints are replaced by their relaxations depicted in Figure 11. Instead of being constrained to find a feasible point on the graph, we can find it somewhere above (or below) the graph, and the performance of the algorithm is drastically improved by presolving. Also pumps with variable speed in an operational setting somehow allow to augment the set of feasible solutions. All in all, it depends of course on the cost of the above "slacks", whether an overall procedure is improved or not.

Another way of approximating (sub-)problems in a manner that the space of feasible solutions is augmented is to use piecewise linear relaxations instead of piecewise linear approximations. In this way, it should become easier to find feasible solutions. In [16] is reported that the switch from piecewise linear approximation to piecewise linear relaxation does not result in an overall speed-up for the problem considered in that paper. Also for the water network design instances we did not experience a significant speed-up in the solution time when using a piecewise linear relaxation.

Acknowledgments

We would like to thank Jesco Humpola for sharing insights on nonlinear network flows and Stefan Vigerske and Ambros Gleixner for giving technical advice regarding SCIP. We also thank Wei Huang, Lars Schewe and Antonio Morsi for their helpful comments in the context of water network operation.

References

[1] B. Coelho, A. Andrade-Campos, Efficiency achievement in water supply systems - a review, Renewable and Sustainable Energy Reviews 30 (2014) 59–84.

[2] J. Burgschweiger, B. Gnädig, M. C. Steinbach, Optimization models for operative planning in drinking water networks, Optim. Eng. 10 (2009) 343–373.

[3] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, P. Toth, On the optimal design of water distribution networks: A practical minlp approach, Optim. Eng. 13 (2012) 219–246.

[4] A. Fügenschuh, J. Humpola, A unified view on relaxations for a nonlinear network flow problem, Tech. Rep. 13-31, Zuse-Institut Berlin (2013).

[5] A. M. Gleixner, H. Held, W. Huang, S. Vigerske, Towards globally optimal operation of water supply networks, Numerical Algebra, Control and Optimization 2 (2012) 695–711.

[6] D. Verleye, E.-H. Aghezzaf, Optimising production and distribution operations in large water supply networks: A piecewise linear optimisation approach, International Journal of Production Research 51 (23-24) (2013) 7170–7189.

[7] A. Morsi, B. Geißler, A. Martin, Mixed integer optimization of water supply networks, in: A. Martin, K. Klamroth, J. Lang, G. Leugering, A. Morsi, M. Oberlack, M. Ostrowski (Eds.), Mathematical Optimization of Water Networks, Vol. 162 of International Series of Numerical Mathematics, Springer Basel, 2012, Ch. 3, pp. 35–54.

[8] O. Kolb, J. Lang, Simulation and continuous optimization, in: A. Martin, K. Klamroth, J. Lang, G. Leugering, A. Morsi, M. Oberlack, M. Ostrowski (Eds.), Mathematical Optimization of Water Networks, Vol. 162 of International Series of Numerical Mathematics, Springer Basel, 2012, Ch. 2, pp. 17–33.

[9] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, Mixed-integer nonlinear optimization, Acta Numerica 22 (2013) 1–131.

[10] K. Abishek, S. Leyffer, J. Linderoth, Filmint: An outer approximation-based solver for convex mixed-integer nonlinear programs, INFORMS Journal on Computing 22 (4) (2010) 555–567.

[11] https://projects.coin-or.org/bonmin.

[12] A. U. Raghunathan, Global optimization of nonlinear network design, SIAM J. Optim. 23 (1) (2013) 268–295.

[13] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wächter, Branching and bounds tightening techniques for non-convex minlp, Optimization Methods and Software 24 (4-5) (2009) 597–634.

[14] T. Achterberg, Scip: solving constraint integer programs, Math. Prog. Comp. 1 (2009) 1–41.

[15] T. Berthold, S. Heinz, S. Vigerske, Extending a CIP framework to solve MIQCPs, in: J. Lee, S. Leyffer (Eds.), Mixed Integer Nonlinear Programming, Vol. 154 of The IMA Volumes in Mathematics and its Applications, Springer New York, 2012, pp. 427–444.

[16] B. Geißler, A. Martin, A. Morsi, L. Schewe, Using piecewise linear functions for solving MINLPs, in: J. Lee, S. Leyffer (Eds.), Mixed Integer Nonlinear Programming, Vol. 154 of The IMA Volumes in Mathematics and its Applications, Springer New York, 2012, pp. 287–314.

[17] E. M. L. Beale, J. A. Tomlin, Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables, in: J. Lawrence (Ed.), Proceedings of the fifth international conference on operational research, Tavistock Publications, London, 1970, pp. 447–454.

[18] B. Geißler, O. Kolb, J. Lang, G. Leugering, A. Martin, A. Morsi, Mixed integer linear models for the optimization of dynamical transport networks, Mathematical Methods of Operations Research 73 (3) (2011) 339–362.

[19] O. Kolb, A. Morsi, J. Lang, A. Martin, Nonlinear and mixed integer linear programming, in: A. Martin, K. Klamroth, J. Lang, G. Leugering, A. Morsi, M. Oberlack, M. Ostrowski (Eds.), Mathematical Optimization of Water Networks, Vol. 162 of International Series of Numerical Mathematics, Springer Basel, 2012, Ch. 4, pp. 55–65.

[20] B. Geißler, A. Morsi, L. Schewe, A new algorithm for MINLP applied to gas transport energy cost minimization, in: M. Jünger, G. Reinelt (Eds.), Facets of Combinatorial Optimization, Springer Berlin Heidelberg, 2013, pp. 321–353.

[21] M. Collins, L. Cooper, R. Helgason, J. Kennington, L. LeBlanc, Solving the pipe network analysis problem using optimization techniques, Management Science 24 (7) (1978) 747–760.

[22] S. Pilati, E. Todini, La verifica delle reti idrauliche in pressione, in italian.

[23] http://www.epa.gov/nrmrl/wswrd/dw/epanet.html.

[24] S. Vigerske, Solving MINLPs with SCIP, Conference talk ISMP, Berlin (August 2012).

[25] S. Vigerske, Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming, Ph.D. thesis, Humboldt-Universität zu Berlin (2012).

[26] S. Vigerske, private communication (2013).

[27] A. M. Gleixner, private communication (2013).

[28] A. Morsi, Solving minlps on loosely-coupled networks with applications in water and gas network optimization, Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2013).

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

[30] J. Burgschweiger, B. Gnädig, M. C. Steinbach, Nonlinear programming techniques for operative planning in large drinking water networks, The Open Applied Mathematics Journal 3 (2009) 14–28.

[31] A. Fügenschuh, J. Humpola, A new class of valid inequalities for nonlinear network design problems, Tech. Rep. 13-06, Zuse-Institut Berlin (2013).