Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

AMBROS M. GLEIXNER

HARALD HELD

WEI HUANG

STEFAN VIGERSKE

# Towards globally optimal operation of water supply networks

Supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin.

ZIB-Report 12-25 (July 2012)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

# Towards globally optimal operation of water supply networks*

Ambros M. Gleixner[†] Harald Held[‡] Wei Huang[§]

Stefan Vigerske[‖]

July 25, 2012

**Abstract**

This paper is concerned with optimal operation of pressurized water supply networks at a fixed point in time. We use a mixed-integer nonlinear programming (MINLP) model incorporating both the nonlinear physical laws and the discrete decisions such as switching pumps on and off. We demonstrate that for instances from our industry partner, these stationary models can be solved to $\varepsilon$-global optimality within small running times using problem-specific presolving and state-of-the-art MINLP algorithms.

In our modeling, we emphasize the importance of distinguishing between what we call real and imaginary flow, i.e., taking into account that the law of Darcy-Weisbach correlates pressure difference and flow along a pipe if and only if water is available at the high pressure end of a pipe. Our modeling solution extends to the dynamic operative planning problem.

## 1 Introduction

Water supply networks form a vital part of public, municipal infrastructure. Communal life and industrial activity base not only on the availability, but also the reliable distribution of, in particular, potable water. Installation, maintenance, and operation of a water supply network incur substantial costs. This article is concerned with mathematical optimization for cost- and energy-minimal network operation. For work on optimal network design using similar methodology, see, e.g., the recent article of Bragalli et al. [9] and references therein.

The article is organized as follows. In Section 2, we introduce the application background and put our research into the context of existing solution methodologies. Section 3 models the optimization problem as a *mixed-integer nonlinear program*

*This paper has been accepted for publication in the journal *Numerical Algebra, Control and Optimization*.

[†]Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, gleixner@zib.de.

[‡]Siemens AG, Corporate Technology (CT T DE TC 3), Otto-Hahn-Ring 6, 81739 Munich, Germany, harald.held@siemens.com.

[§]Corresponding author. Technical University of Munich, Department of Mathematics, M9, Boltzmannstr. 3, 85748 Garching b. Munich, Germany, huang@ma.tum.de.

[‖]Humboldt-Universität zu Berlin, Department of Mathematics, Unter den Linden 6, 10099 Berlin, Germany, stefan@math.hu-berlin.de. This author was supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin, see http://www.matheon.de/.

(MINLP) and Section 4 explains how this can be solved to globally proven optimality gaps. In Section 5 we describe a set of straightforward presolving steps reducing the size and difficulty of the model without cutting off optimal solutions. Section 6 presents results of computational experiments conducted on real-world instances provided by our industry partner Siemens AG, Corporate Technology, Modeling, Simulation & Optimization.[1] Finally, Section 7 contains concluding remarks.

# 2 Motivation

A pressurized water supply network is a system of pipes, valves, and pumps connecting the sources of water with the consumers. *Operative planning* must ensure that requested water is transported from the sources to the consumers, possibly using tanks as intermediate storage facilities. If gravitation does not suffice, pumps need to be activated thereby consuming energy at a certain cost. The second matter of expense is the resource itself: providing water at a source comes at different costs depending, e.g., on modes of extraction or purification at the facility.

The full-scale operative planning task typically covers the time span of one day with hourly demand forecasts for each consumer and comprises the decision when and which pumps are switched on and off, when and which tanks are filled for storage (typically at low demand), when they are deflated again (typically at peak demand), and where water is obtained. The benefit of *minimum cost operative planning* is both *economical and ecological*. Minimizing energy cost equals saving energy itself and preferring water from cheap sources usually means avoiding chemical treatment.

The resulting optimization problem includes discrete and continuous decisions and features nonconvex nonlinearities in constraints and objective function. The time dimension adds another difficulty. Due to this complexity, solution approaches found in the literature either simplify the physics involved, e.g., by dropping or linearizing the nonlinearities, or resort to locally optimal solution procedures such as solving NLP formulations to local optimality or applying (meta-)heuristic procedures without guaranteed optimality gaps. For an overview on work before 2004, see, e.g., Burgschweiger et al. [10]. For recent work, see, e.g., the theses of Huang [12], which has been the base for our paper, and Kolb [14], which adresses this problem heuristically from an optimal control perspective.

Although heuristic trade-offs may currently be necessary to obtain acceptable solution times, sacrificing global optimality remains unsatisfactory both from a theoretical and practical point of view. An interesting approach is investigated in [11, 13]. The authors describe how to approximate nonlinearities by piecewise linear functions in order to obtain a mixed-integer linear program (MIP), for which sophisticated solution algorithms are readily available. The approximation has to be determined *a priori*, but violation of the nonlinear constraints can be checked *a posteriori* and the approximation can be refined if necessary to obtain an $\varepsilon$-feasible solution. However, even then the approach does not guarantee a valid dual bound and optimal solutions may be cut off since precisely speaking a different problem is solved.

In our paper, we focus on solving a stationary version of the planning problem to $\varepsilon$-global optimality: Given fixed starting levels of tanks and constant demands of consumers, compute a pump configuration and a feasible flow through the network such as to minimize the variable operational cost incurred by purchase of energy and water. Although this alone does not address the full operative planning task, it arises naturally, e.g., as one-period subproblem in a time-discretized formulation.

[1] `http://www.ct.siemens.com/`

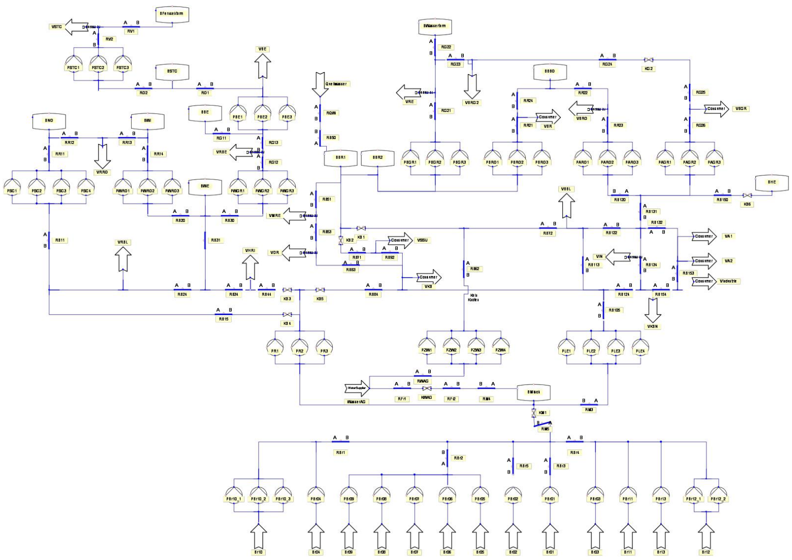

Figure 1: Schematic diagram of water supply network instance `n88p64a64` with 88 nodes (15 reservoirs, 11 tanks, 62 junctions), 22 consumers, 64 pipes, 55 pumps, and 9 valves.

In heuristic or decomposition-based solution approaches these subproblems may have to be solved iteratively. Not least, the ability to compute proven optimal solutions to these stationary models can help in evaluating and improving heuristic solution techniques.

# 3 Model

The goal of this paper is to optimize the operation of a water supply network at a fixed point in time. Given filling levels of tanks and demands of consumers we wish to compute a feasible flow through the network such as to minimize the variable operational cost incurred by purchase of energy and water. In the following, we model this problem as a nonconvex *mixed-integer nonlinear program* (MINLP).

## 3.1 Network elements

Our model of a water supply network is based on a directed graph $G = (\mathcal{N}, \mathcal{A})$. The set of nodes $\mathcal{N} = \mathcal{J} \cup \mathcal{W}$ consists of *junctions* $j \in \mathcal{J}$ and *water sources* $w \in \mathcal{W}$ such as reservoirs or tanks. Consumers are located at junctions with nonzero demand $D_j > 0$. The arc set $\mathcal{A} = \mathcal{S} \cup \mathcal{P} \cup \mathcal{V}$, where $\mathcal{S}$ is the set of *pipe segments*, $\mathcal{P}$ is the set of *pumps*, and $\mathcal{V}$ is the set of *valves*.

In general, the direction of an arc does not prescribe the direction of flow through this element, but only defines the meaning of positive flow. Some arcs such as pumps, however, only permit one-directional flow. For a node $i$, we denote the set of in- and out-going arcs as $\delta^-(i) := \{ki \in \mathcal{A} : k \in \mathcal{N}\}$ and $\delta^+(i) := \{ik \in \mathcal{A} : k \in \mathcal{N}\}$, respectively.

Figure 1 shows an example of a real-world water supply network provided to us by Siemens AG. It consists of 88 nodes – 11 tanks, 15 reservoirs, 62 junctions, and

22 consumers – as well as 64 pipes, 55 pumps, and 9 valves.

## 3.2 Flow balance

Each arc $a = ij$ carries a signed flow $q_a \in [Q_a^{\min}, Q_a^{\max}]$; if positive then from $i$ to $j$, if negative then $|q_a|$ is transported backwards. At each junction $j$, the classical flow balance equation

$$\sum_{a\in\delta^-(j)} q_a - \sum_{a\in\delta^+(j)} q_a = D_j \tag{1}$$

must be satisfied. Junctions with positive demand $D_j > 0$ correspond to consumers, all others satisfy $D_j = 0$. The remaining nodes, the water sources $w \in \mathcal{W}$, provide the flow necessary to match the demand:

$$D_w^{\min} \leqslant \sum_{a\in\delta^+(w)} q_a - \sum_{a\in\delta^-(w)} q_a \leqslant D_w^{\max}. \tag{2}$$

While reservoirs allow for outflow only, i.e., $D_w^{\min} = 0$, tanks are used to store water and admit both in- and outflow.

## 3.3 Pressure

Water flow through the network is induced by different pressure levels at the nodes. Since water is approximately incompressible, static water pressure may be assumed proportional to the elevation $h$ (in meter) above a fixed point of measurement: pressure equals $\rho g h$, where $\rho$ is water density and $g$ is gravitational acceleration. According to convention, we measure pressure and pressure differences by the so-called head $h_i$ of a node $i$ and the head difference $\Delta h_a$ along an arc $a$.

While the pressure at water sources $w$ is fixed,

$$h_w = H_w^0,$$

the head at a junction $j$ may exceed the geodetic height,

$$h_j \geqslant H_j^0.$$

Their values are determined by pressure loss in pipes and valves and pressure increase in pumps according to the laws described in the following.

## 3.4 Pipe model

The flow of water through a pipe $s = ij$ is a function of the pressure levels $h_i$ and $h_j$ at its ends. The pressure loss along the pipe is described by the law of *Darcy-Weisbach*,

$$h_i - h_j = \lambda_s \operatorname{sgn}(q_s) q_s^2, \tag{3}$$

where $\operatorname{sgn}(q_s)$ denotes the sign of $q_s$. The loss coefficient $\lambda_s$ in this equation is computed as

$$\lambda_s = \frac{8L_s}{\pi^2 g d_s^5} f_s$$

involving properties of the pipe – length $L_s$ and inner diameter $d_s$ – and the Darcy friction factor $f_s$. The highly nonlinear dependency of $f_s$ on the flow rate $q_s$ is taken into account by simulation software, see, e.g., EPANET [22], but appears to be too detailed for an optimization model.

We use the law of Prandtl-Kármán,

$$f_s = \left(2\log_{10}\frac{\varepsilon_s}{3.71 d_s}\right)^2,$$

which eliminates the dependency on $q_s$ by assuming large Reynolds number and is a good approximation for hydraulically rough pipes. It tends to underestimate the induced flow for small pressure differences, hence yielding conservative solutions. The roughness parameter $\varepsilon_s$ only depends on the inner pipe surface. For more details on mathematical modeling of the physics of pressure loss, see, e.g., [10].

**Remark 1.** Because we handle the Darcy-Weisbach equation with bidirectional flow algorithmically, see Section 4.2, we do not need to include a forward and backward arc in our model with one nonlinear pressure loss constraint each as, e.g, in Sherali and Smith [17].

## 3.5 Valve model

A valve $v = ij$ can be used to block flow completely or decrease its pressure by a controlled amount in direction of the flow. We introduce a binary variable $y_v$ denoting the flow direction through the valve, $y_v = 1$ if positive, i.e., from $i$ to $j$. Then, we may model feasible valve states by

$$M(1 - y_v) \leqslant h_i - h_j \leqslant M y_v \tag{4}$$

and

$$M(1 - y_v) \leqslant q_v \leqslant M y_v, \tag{5}$$

where $M$ is chosen sufficiently large. If $q_v = 0$, the valve is closed and the pressure levels at $i$ and $j$ are uncoupled since $y_v$ can take either value. If $q_v \neq 0$, the pressure decreases in direction of the flow by $|\Delta h_v| = |h_i - h_j|$, possibly zero.

Feasible valve states could alternatively be modeled by the nonlinear constraint $\Delta h_v q_v \geqslant 0$, however, introducing an auxiliary binary variable $y_v$ improves the computational behavior of our branch-and-cut solution approach presented in Section 4.

**Remark 2.** For clarity of presentation, we use the same $M$ constant in all big-M constraints of our model. In our computations we choose $M$ for each constraint individually as small as possible, depending on the bounds of the variables involved.

## 3.6 Pump model

The geographically given head differences in a water supply network usually do not generate sufficient flow between water sources and consumers to satisfy the demand. Pumps are used to increase the pressure at water sources or within the network, thereby consuming energy.

The pressure increase generated by a pump depends on the speed at which it is operated and the flow through the pump. This relationship is measured empirically and recorded as characteristic curve of the pump. For a pump $p = ij$ operated at constant speed – all pumps in the instances available to us operate at a single fixed speed –, it may be approximated as

$$\Delta h_p = \Delta H_p^{\max} - \gamma_p^1 q_p^{\gamma_p^2}, \tag{6}$$

where parameters $\Delta H_p^{\max}$, $\gamma_p^1$, and $\gamma_p^2$ are chosen to fit the characteristic curve. The more water flows through the pump, the less the pressure increases.

If a pump is switched off it acts like a closed valve, i.e., $q_p = 0$ and $h_i$ and $h_j$ are uncoupled. If it is active then the flow must be within some interval $[Q_p^{\min}, Q_p^{\max}]$ and the head increase is enforced as in (6). Using a binary variable $x_p$ for the pump status, we can model this by

$$Q_p^{\min} x_p \leqslant q_p \leqslant Q_p^{\max} x_p \tag{7}$$

and

$$M(x_p - 1) \leqslant h_i - h_j - \Delta h_p \leqslant M(1 - x_p). \tag{8}$$

Note that pumps allow for positive flow direction only and $q_p$ is a semi-continuous variable, $Q_p^{\max} \geqslant Q_p^{\min} > 0$.

The energy consumed to generate a head difference of $\Delta h_p$ at flow rate $q_p$ can be approximated by $\rho g \Delta h_p q_p / \eta_p$, where $\eta_p$ is the pump efficiency. We assume that the range of feasible flow rates $[Q_p^{\min}, Q_p^{\max}]$ is chosen such that the pump operates close to its maximum efficiency and hence $\eta_p$ can be treated as constant. For more detailed mathematical models of water pumps, see, e.g., [10].

### 3.7 Real and imaginary flow

As explained above, different pressure levels at the ends of a pipe induce nonzero flow according to the law of Darcy-Weisbach as given by equation (3). However, this only holds if water is indeed present at the high-pressure node. With active elements like closed valves or inactive pumps, pipes may become empty. In this case, strict enforcement of (3) leads to a physically unsound model.

As an example, consider the subnetwork shown in Figure 2 taken from the real-world instance in Figure 1. An elevated tank $t_1$ is connected to the network via valve $k_1$. Pipe $s_3$ leads downwards, i.e., $H_{j_2}^0 > H_{j_1}^0$. Suppose now valve $k_1$ is closed. By flow balance, $q_{s_3} = 0$, and for (3) to hold we need $h_{j_1} = h_{j_2}$, i.e., the head at $j_1$ must lie strictly above its geodetic height. In reality, however, the subnetwork functions as if $s_3$, $j_2$, $k_1$, and $t_1$ were not present, hence $h_{j_1} = H_{j_1}^0$ might be a valid state.

We call head levels at nodes without water and the flow that would be induced by these head levels according to the law of Darcy-Weisbach *imaginary* as opposed to *real*. In the above example, the incorrect assumption was to enforce equation (3) although the head at $j_2$ is imaginary in solutions with closed valve $k_1$.

**Remark 3.** So far we have not seen this distinction being made in the literature. Although it may be that depending on the structure of the network all head levels can be validly assumed to be real, we believe this to be a potential source for harmful modeling gaps. Note that this distinction is equally necessary for the full-scale operative planning problem and can be made by the same constraints proposed here.

To distinguish between real and imaginary heads, we introduce a binary variable $z_j$ at each junction $j \in \mathcal{J}$ forced to 1 if the head is strictly greater than its geodetic height,

$$h_j \leqslant H_j^0 + M z_j, \tag{9}$$

or if flow passes through $j$, i.e.,

$$-M z_j \leqslant q_a \leqslant M z_j \tag{10}$$

for all $a \in \delta(j)$. Water supply networks are usually operated such that water sources are never completely empty and may be assumed as real, $z_w = 1$ for all $w \in \mathcal{W}$.

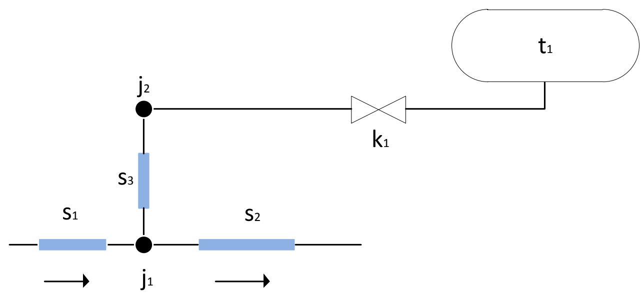


Figure 2: Subnetwork with imaginary flow for closed valve $k_1$.

Furthermore, we need to model how water is propagated along pipes: If a pipe $ij$ is level then water is present at $i$ if and only if it is present at $j$, i.e.,

$$z_i = z_j \tag{11}$$

for all $ij \in \mathcal{S}$ with $H_i^0 = H_j^0$. For pipes with nonzero slope two implications hold: First, if the geodetically higher node, node $i$, say, is real, so is the lower node $j$,

$$z_i \leqslant z_j. \tag{12}$$

Second, if the lower node $j$ is real and contains water with higher pressure than $H_i^0$, then also $i$ must be real,

$$h_j \leqslant H_i^0 + M z_i. \tag{13}$$

Finally, we enforce equation (3), the law of Darcy-Weisbach, between (and only between) real nodes:

$$\Delta h_s = \lambda_s \operatorname{sgn}(q_s) q_s^2 \tag{14}$$

and

$$M(z_i + z_j - 2) \leqslant h_i - h_j - \Delta h_s \leqslant M(2 - z_i - z_j) \tag{15}$$

for all pipes $s = ij \in \mathcal{S}$.

**Remark 4.** Note that both in reality and in our model a node may be real in spite of zero flow through the node: $z_j = 1$ and $q_a = 0$ for all $a \in \delta(j)$. As an example, imagine an additional, closed valve at node $j_1$ in Figure 2, while valve $k_1$ is open. Then pipe $s_3$ would be completely filled with water from the tank, hence nodes $j_1$ and $j_2$ would be real. At the same time, the water column in the pipe yields pressure $h_{j_1} = h_{j_2}$ and so the law of Darcy-Weisbach is satisfied by zero flow, $q_{s_3} = 0$.

## 3.8 Objective function

Our goal is to minimize the variable operational costs incurred by purchasing water fed into the network and the energy needed to operate pumps. The energy consumption of a pump $p$ equals

$$\frac{\rho g \Delta h_p q_p}{\eta_p} \overset{(6)}{=} \frac{\rho g}{\eta_p}\Big(\Delta H_p^{\max} q_p - \gamma_p^1 q_p^{1+\gamma_p^2}\Big).$$

| variable | interpretation |
|---|---|
| $h_i$ | pressure potential (head) at node $i \in \mathcal{N}$ $[m]$ |
| $\Delta h_a$ | pressure increase/decrease at pump or pipe $a \in \mathcal{P} \cup \mathcal{S}$ $[m]$ |
| $q_a$ | volumetric flow rate in arc $a \in \mathcal{A}$ $[m^3/s]$ |
| $x_p$ | binary indicator whether pump $p \in \mathcal{P}$ is switched on |
| $y_v$ | binary indicator for direction of valve $v \in \mathcal{V}$ |
| $z_i$ | binary indicator whether node $i \in \mathcal{N}$ is real |

Table 1: Variables of the optimization model.

The total objective function has the form

$$\begin{aligned}&\sum_{w\in\mathcal{W}} \alpha_w \Big( \sum_{a\in\delta^+(w)} q_a - \sum_{a\in\delta^-(w)} q_a \Big) + \sum_{p\in\mathcal{P}} \frac{\beta\rho g}{\eta_p} \Delta h_p q_p \\ &\quad = \sum_{w\in\mathcal{W}} \alpha_w \Big( \sum_{a\in\delta^+(w)} q_a - \sum_{a\in\delta^-(w)} q_a \Big) + \sum_{p\in\mathcal{P}} \frac{\beta\rho g}{\eta_p} \Big( \Delta H_p^{\max} q_p - \gamma_p^1 q_p^{1+\gamma_p^2} \Big),\end{aligned} \tag{16}$$

where $\beta > 0$ is the energy cost and $\alpha_w \geqslant 0$ the cost for purchasing water at source $w$, which may vary from source to source. Tanks function as storage facilities within the network – the only water sources allowing for inflow – and do not incur costs. Note that the objective function can be expressed in terms of flow variables only and each summand is a univariate function.

### 3.9 Summary

Table 1 summarizes the variables used in our optimization model. The complete nonconvex MINLP now reads

$$\left.\begin{aligned}
\min \quad & \sum_{w\in\mathcal{W}} \alpha_w \Big( \sum_{a\in\delta^+(w)} q_a - \sum_{a\in\delta^-(w)} q_a \Big) + \sum_{p\in\mathcal{P}} \frac{\beta\rho g}{\eta_p} \Big( \Delta H_p^{\max} q_p - \gamma_p^1 q_p^{1+\gamma_p^2} \Big) \\
\text{s.t.} \quad & (1-2), (4-5), (6-8), (9-15), \\
& x_p \in \{0,1\}, q_p \in [0, Q_p^{\max}], \Delta h_p \in [0, \Delta H_p^{\max}] && \text{for all } p \in \mathcal{P}, \\
& y_v \in \{0,1\}, q_v \in [Q_v^{\min}, Q_v^{\max}] && \text{for all } v \in \mathcal{V}, \\
& q_s \in [Q_s^{\min}, Q_s^{\max}] && \text{for all } s \in \mathcal{S}, \\
& z_j \in \{0,1\}, h_j \in [H_j^0, H_j^{\max}] && \text{for all } j \in \mathcal{J}, \\
& z_w = 1, h_w = H_w^0 && \text{for all } w \in \mathcal{W}.
\end{aligned}\right\} \tag{17}$$

It features two types of nonlinearities, the energy consumption of pumps in the objective function and the Darcy-Weisbach equation along each pipe, both of which are nonconvex. Together with the discrete states encoded in the binary variables this yields a highly nonconvex solution space.

## 4 Global solution approach

The problem formulation given in the previous section is a nonconvex MINLP. Its combination of discrete and continuous nonconvexities – binary decision variables for pump status, valve direction, and imaginary flow plus nonconvex nonlinear terms (6) and (14) – results in a challenging optimization problem. In the following we describe how well-known algorithmic techniques can be applied to solve them to $\varepsilon$-global optimality.

## 4.1 Branch-and-bound

A common methodology to handle nonconvex optimization problems is *branch-and-bound* [15], where the problem is successively divided into smaller subproblems until the individual subproblems are sufficiently easy to solve. Additionally, bounding is used to detect early whether improving solutions can be found in a subproblem and avoid enumerating suboptimal parts of the feasible region. Thereby, bounds on the optimal objective function value are computed from a computationally tractable relaxation of the current subproblem.

For nonconvex MINLPs, typically an efficiently solvable convex (linear or nonlinear) relaxation is used for *bounding*, obtained by dropping integrality conditions and replacing nonconvex nonlinear functions by convex estimators [18]. *Branching* (problem division) is done with respect to either discrete variables that take a fractional value in the relaxation's solution or variables that appear in violated nonconvex constraints. The purpose of the latter is, that a reduction of a variable's domain yields tighter convex estimators, which in turn may allow to cut off the infeasible solution from the relaxation.

Branch-and-bound algorithms for general MINLPs are implemented by the solvers BARON [18], Couenne [3], LINDO API [16], and SCIP [2, 19]. By default, all of them employ a linear relaxation.

We used the solver SCIP, a framework for solving constraint integer programs by a branch-and-bound algorithm. Arguably, from the solvers listed above, it provides the strongest support for solving mixed-integer programs (MIPs), which is necessary to address the combinatorial aspect of our optimization problem. Its state-of-the-art MIP features include cutting plane separators, primal heuristics, domain propagation algorithms, and support for conflict analysis [1, 2]. Recently, SCIP has been extended to handle also nonlinear constraints [8, 19].

## 4.2 Outer approximation

For the nonlinear functions $q_s \mapsto \lambda_s \mathrm{sgn}(q_s) q_s^2$ from constraint (14) and $q_p \mapsto -\gamma_p^1 q_p^{1+\gamma_p^2}$ in the objective function (16), SCIP generates a linear outer approximation along their convex and concave envelopes. If the relaxation's solution violates nonlinear constraints, the outer approximation is tightened by branching on the flow variables $q_s$ and $q_p$. For $q_s \mapsto \lambda_s \mathrm{sgn}(q_s) q_s^2$, this is illustrated in Figure 3. For further details, we refer to [19].

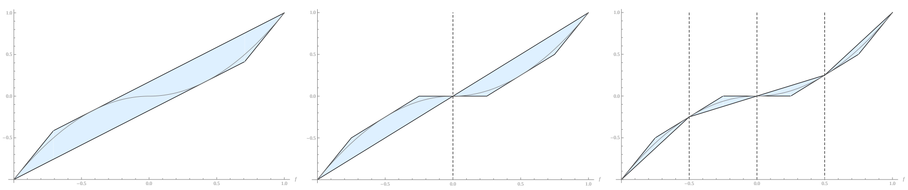

Figure 3: Linear outer approximation of the nonlinear function $q_s \mapsto \lambda_s \mathrm{sgn}(q_s) q_s^2$ and effect of branching on $q_s$.

To improve performance, SCIP uses the constraints to propagate a reduction in one variable's domain to other variables. For example, if the bounds on variable $\Delta h_s$ in constraint $\Delta h_s = \lambda_s \mathrm{sgn}(q_s) q_s^2$ are reduced to $[\underline{\Delta h_s}, \overline{\Delta h_s}]$, the bounds of $q_s$ can be tightened to

$$\left[\mathrm{sgn}(\underline{\Delta h_s})\sqrt{|\underline{\Delta h_s}|/\lambda_s}, \mathrm{sgn}(\overline{\Delta h_s})\sqrt{|\overline{\Delta h_s}|/\lambda_s}\right],$$

which allows for a tighter linear outer approximation. Similarly, tighter bounds for $\Delta h_s$ may be deduced from domain reductions for $q_s$.

### 4.3 Primal solutions

Although in theory, it suffices to collect feasible solutions of the relaxation at leaves of the branch-and-bound tree, in practice, it is highly beneficial to apply heuristic procedures interleaved with the global search. Finding good solutions early in the search allows the user to stop the solution early if he is already satisfied with the achieved solution quality. Algorithmically, better primal bounds allow the branch-and-bound tree to be pruned earlier and can hence improve solver performance.

SCIP uses several primal heuristics to find feasible solutions early in the search. First, SCIPs default MIP primal heuristics [4] are applied to find a point that is feasible for the linear relaxation plus the integrality requirements, but may violate some of the nonlinear constraints. Subsequently, the binary variables ($x$, $y$, $z$) are fixed to their value in this solution and the resulting nonlinear program (NLP) is solved to local optimality using Ipopt [20]. If the NLP is feasible, any solution is also feasible for the original MINLP.

Second, SCIP employs various large neighbourhood search heuristics extended from MIP to MINLP [4, 7] or specifically designed for MINLP [5, 6]. These heuristics use the relaxation solution or previously found feasible solutions to construct a hopefully easier sub-MINLP by restricting the search space, e.g., via variable fixings. The reduced problem is then partially solved by a separate SCIP instance.

## 5 Reformulation and presolving

This section outlines a set of straightforward problem-specific presolving steps that help to reduce both size and difficulty of given instances of type (17). The reductions explained in the following are exact in the sense that a feasible solution is cut off only if another essentially identical solution remains.

### 5.1 Fixing and propagating $z$ variables

At junctions with nonzero demand, flow balance requires nonzero flow on at least one incident arc. Trivially, (10) implies that the head is real:

$$j \in \mathcal{J}, D_j > 0 \Longrightarrow z_j = 1.$$

Using these fixings and the water sources known to be real, some of the constraints (11−13) may then become redundant or can be used to fix further $z$ variables to one.

### 5.2 Breaking symmetry in pump stations

A design commonly found in water supply networks is a collection of identical pumps $p_1, \dots, p_N \in \mathcal{P}$ that are connected in parallel within a so-called *pump station* as depicted in Figure 4.

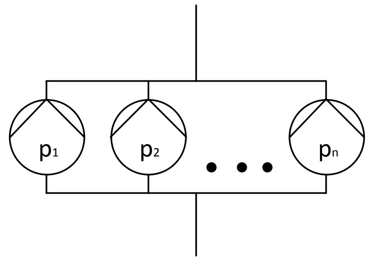


Figure 4: Pump station with pumps connected in parallel.

All active pumps increase the pressure by an equal amount and their flow rates add up. Since only the number of active pumps is relevant and not which pumps are active, the standard symmetry breaking constraints

$$x_{p_1} \leqslant \ldots \leqslant x_{p_N}$$

are valid. This reduces the search space for feasible choices of active pumps significantly from $2^N$ to $N+1$.

## 5.3 Contracting subsequent pipes

Suppose a zero demand junction $j$ is incident with two pipes, one entering, $ij$, and one leaving, $jk$. Flow balance enforces $q_{ij} = q_{jk} =: \tilde{q}$ and if nonzero flow passes through $j$ the Darcy-Weisbach equations read $h_i - h_j = \lambda_{ij}\mathrm{sgn}(\tilde{q})\tilde{q}^2$ and $h_j - h_k = \lambda_{jk}\mathrm{sgn}(\tilde{q})\tilde{q}^2$. These two constraints are equivalent to

$$h_i - h_k = (\lambda_{ij} + \lambda_{jk})\mathrm{sgn}(\tilde{q})\tilde{q}^2$$

and

$$h_j = \frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}}.$$

We want to exploit this to replace pipes $ij$ and $jk$ by a new, aggregated pipe $ik$ with loss coefficient $\lambda_{ij} + \lambda_{jk}$ and consequently remove junction $j$ from the network.

In case nonzero flow $\tilde{q} \neq 0$ is guaranteed to pass through the pipe, we only need to ensure satisfiability of $h_j \geqslant H_j^0$ by

$$\frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}} \geqslant H_j^0. \tag{18}$$

To account for $\tilde{q} = 0$, however, we need to keep variable $z_j$ in the model, since it may be zero even if $z_i = z_k = 1$. (As an example consider the case that junction $j$ is located much higher than $i$ and $k$ and can hence block flow even if water is available at $i$ and $k$.)

Darcy-Weisbach holds if and only if all three nodes $i$, $j$, and $k$ have real head, i.e., constraint (15) becomes

$$M(z_i + z_j + z_k - 3) \leqslant h_i - h_k - \Delta h_{ik} \leqslant M(3 - z_i - z_j - z_k). \tag{15a}$$

Constraints (9–13) involving $z_j$ remain unchanged. To ensure (18) if $j$ is real, we add constraint

$$\frac{\lambda_{jk}h_i + \lambda_{ij}h_k}{\lambda_{jk} + \lambda_{ij}} \geqslant H_j^0 - M(1 - z_j). \tag{19}$$

The cases of two pipes entering or leaving a zero demand junction work analogously. Pipe sequences with several inner nodes $ij_1, j_1j_2, \ldots, j_Nk$ can be treated identically – for each inner node we only need to add its $z$ variable to (15a) and include constraint (19).

Note that these presolving steps do not just yield a smaller problem, but most importantly a more linear one because we remove nonlinear equations of type (14).

## 5.4 Contracting pipe-valve-sequences

Suppose a pipe $ij \in \mathcal{S}$ and a valve $jk \in \mathcal{V}$ are connected by a zero demand junction $j$. Flow balance enforces $q_{ij} = q_{jk} =: \tilde{q}$. Figure 5 shows the feasible values of pressure loss $h_k - h_i$ versus $\tilde{q}$. While the Darcy-Weisbach equation forces the pressure loss along the pipe onto the dashed line, the valve allows for larger pressure loss in absolute value. The feasible region is hence a union of two convex sets, the dotted area for backward flow and the shaded area for forward flow.

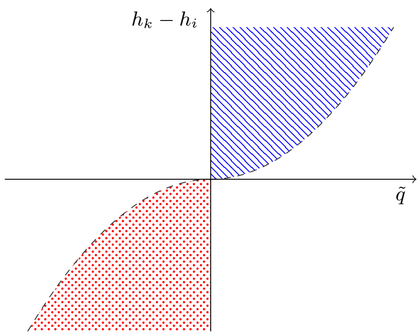


Figure 5: Feasible values of pressure loss versus flow through a pipe-valve-sequence $ij \in \mathcal{S}$, $jk \in \mathcal{V}$.

This can be exploited replacing pipe $ij$ and valve $jk$ by a new arc $a = ik$ and relaxing valve constraints (4) and (5) and pipe constraints (14) and (15) to

$$M(y_a - 1) \leqslant q_a \leqslant My_a \tag{5}$$

for flow direction as before,

$$\Delta h_a \geqslant \lambda_{ij} q_a^2 \tag{14a}$$

for the minimum pressure loss, and

$$M(z_i + y_a - 2) \leqslant h_i - h_k - \Delta h_a \tag{15b}$$

and

$$h_i - h_k + \Delta h_a \leqslant M(1 - z_k + y_a) \tag{15c}$$

for the relaxed Darcy-Weisbach equation.

This reduction replaces the nonconvex, nonconcave constraint (14) by a convex quadratic constraint. Again, other combinations of arc directions work analogously.

**Remark 5.** The above presolving steps simplify the model in two ways. First, they reduce the problem size by eliminating variables and constraints. Second, and even more importantly, they remove some of the nonconvex Darcy-Weisbach equations either completely – when contracting pipe sequences – or replace them by convex constraints – when contracting a pipe-valve-sequence. This significantly reduces the amount of spatial branching needed in the branch-and-bound solution procedure.

# 6 Computational experiments

This section presents the results of our computational experiments on two networks provided by our industry partner Siemens AG. Figure 6 shows a small water supply network `n25p22a18` on 25 nodes (1 reservoir, 4 tanks, 20 junctions), 4 consumers, 22 pipes, 12 pumps, and 6 valves. The second network `n88p64a64` on 88 nodes (15 reservoirs, 11 tanks, 62 junctions), 22 consumers, 64 pipes, 55 pumps, and 9 valves is depicted in Figure 1. Each network comes with hourly demand forecast for one day.

## 6.1 Experimental setup

The goal of our experiments was to investigate whether and how fast the stationary version of the operative planning problem in form of the MINLP model (17) can be solved to $\varepsilon$-global optimality and to evaluate the computational impact of the presolving reductions described in Section 5.

Exemplarily, we selected the demand forecasts for 0-1 am (low demand), 6-7 am (first peak demand), 12-1 pm (medium demand), and 6-7 pm (second peak demand). The results for these scenarios were representative for the other hours.

For the tank levels, we considered two scenarios. In the *medium tank level scenario*, we assume all tanks to be half-full; in this case, a large portion of the demand may be satisfied by emptying the tanks only, without significant pump activity. However, such a solution will be very greedy and also the difficulty of the MINLPs may be reduced. Therefore, for a second test, we select the tanks that—if the first solution was implemented—would run empty first and set them to their minimum filling level, hence only allowing for inflow into these tanks; for network `n25p22a18` we reset the first, for `n88p64a64` we reset the first four tanks that would run empty to their minimum filling levels. We refer to this as *low tank level scenario*.

For our experiments we solely used academic software that is available in source code. We ran SCIP 2.1.1 [24] with SoPlex 1.6.0 [25] as LP solver, Ipopt 3.10.1 [23] as NLP solver, CppAD 20110101.5 [21] as expression interpreter for evaluating nonlinear functions, and Zimpl 3.2 [26] as modeling language. SCIP was run with default settings and a time limit of one hour. We conducted the experiments on an AMD Opteron 6174 with 2.2 GHz and 128 GB RAM.

## 6.2 Computational results

First, we evaluate the impact of the problem-specific presolving steps described in Section 5. Table 2 shows how these help to reduce the size of the problems in number of variables "vars", binary variables "bin", number of constraints "cons" and number of nonlinear constraints "nlin". Note that the problem reductions apply to the structure of the network and are indepent of demand forecast or tank levels. The numbers given are computed before applying SCIP's presolving. Fixed variables and bound constraints are not counted. The largest reduction occurs in the number of binary variables, which are reduced by 14% and 18%, respectively. The number of nonlinear constraints is only slightly reduced.

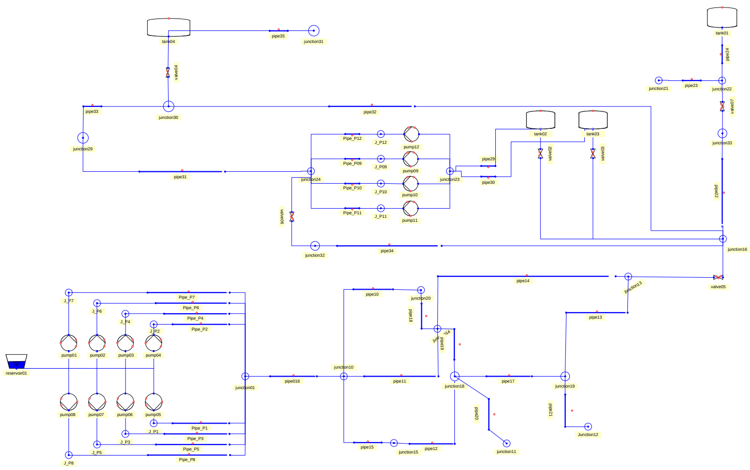

Figure 6: Schematic diagram of water supply network `n25p22a18` with 25 nodes (1 reservoir, 4 tanks, 20 junctions), 4 consumers, 22 pipes, 12 pumps, and 6 valves.

Table 3 compares running times and number of branch-and-bound nodes explored by SCIP when solving to optimality with a tolerance of $10^{-6}$. It can be seen that the scenarios for the smaller instance `n25p22a18` can all be solved within one second and can only improve minimally when using presolving. The most difficult instances are the low tank level scenarios for the larger network `n88p64a64`. Here, both solution time and number of branch-and-bound nodes decrease drastically when applying presolving. Due to smaller branch-and-bound trees, the instances are solved faster by a factor between 3.8 and 89.5. The only slowdown occurs on “0-1 am med” and “6-7 am med” because SCIP's primal heuristics do not find the optimal solution at the root node anymore. Nevertheless, these are solved within less than two seconds. All in all, the presolving steps presented in Section 5 proved highly beneficial in our experiments.

Finally, Table 4 presents our computational results for the presolved instances in more detail. From column “objval” listing the objective value of the optimal solution found, we can confirm the expectation that the low tank level scenarios always require more pumps being active, except for demand “6-7 pm” in `n25p22a18`, where the objective value remains at the same level. In all cases, the “low” scenarios take at least as long as the “med” scenarios. In particular for `n88p64a64`, this seems to explain why the “med” scenarios are computationally much easier: a solution with no active pumps is feasible and can be found and proven to be optimal very fast.

The last three columns analyze the solution progress in more detail, giving the time to find a first feasible solution, the time to achieve a proven primal-dual gap of

| network | without presolving | | | | with presolving | | | |
|---|---|---|---|---|---|---|---|---|
| | vars | bin | cons | nlin | vars | bin | cons | nlin |
| `n25p22a18` | 145 | 28 | 332 | 42 | 139 | 24 | 322 | 40 |
| `n88p64a64` | 561 | 99 | 1098 | 171 | 542 | 81 | 982 | 170 |

Table 2: Problem sizes without and with problem-specific presolving as described in Section 5.

| | scenario | | without presolving | | with presolving | |
|---|---|---|---|---|---|---|
| | demands | tanks | time | nodes | time | nodes |
| n25p22a18 | 0-1 am | med | 0.7s | 247 | 0.4s | 67 |
| | | low | 0.9s | 663 | 0.8s | 85 |
| | 6-7 am | med | 0.6s | 219 | 0.4s | 60 |
| | | low | 1.0s | 478 | 0.8s | 77 |
| | 12-1 pm | med | 0.5s | 76 | 0.6s | 76 |
| | | low | 1.0s | 239 | 0.9s | 172 |
| | 6-7 pm | med | 0.5s | 54 | 0.5s | 80 |
| | | low | 0.4s | 54 | 0.5s | 80 |
| n88p64a64 | 0-1 am | med | 0.4s | 1 | 1.1s | 75 |
| | | low | 11.2s | 3518 | 1.1s | 16 |
| | 6-7 am | med | 0.6s | 1 | 1.6s | 181 |
| | | low | 595.4s | 334128 | 12.8s | 5495 |
| | 12-1 pm | med | 3.6s | 1044 | 2.4s | 430 |
| | | low | 1941.4s | 1195329 | 21.7s | 6738 |
| | 6-7 pm | med | 4.2s | 1413 | 1.0s | 85 |
| | | low | 399.8s | 236966 | 104.0s | 64940 |

Table 3: Running times and number of branch-and-bound nodes to optimal solution without and with presolving as described in Section 5.

5%, and the time until an optimal solution is found. A gap of 5% is always reached within 2.4 seconds except for `n88p64a64` "12-1 pm low", where it takes 16.7 seconds. In almost all cases, the optimal solution is found at the very end of the solution process. For the instance `n88p64a64` "6-7 pm low" with longest running time of 104 seconds, however, the situation is reversed: the optimal solution is found already after 1.5 seconds and SCIP spends the remaining time to prove its optimality.

# 7 Concluding remarks

This paper has presented a small contribution to the task of optimal, i.e., energy- and cost-minimal, operative planning of water supply networks. Our research has focused on a stationary version of this challenging optimization problem and aimed at $\varepsilon$-globally optimal solution techniques. The MINLP model used is detailed in the sense that it incorporates the nonlinear physical laws as well as the discrete decisions involved.

In our modeling, we have emphasized the importance of distinguishing between what we call real and imaginary flow. The Darcy-Weisbach equation relating flow and pressure loss along a pipe must only be enforced if water is actually available at the high pressure end of the pipe. Our model to handle this distinction extends to the full dynamic operative planning problem.

Through computational experiments on instances from industry, we demonstrated that the stationary models presented can be solved to global optimality within small running times using problem-specific presolving and a state-of-the-art MINLP solution algorithm.

| | scenario | | to optimality | | | time to | | |
|---|---|---|---|---|---|---|---|---|
| | demands | tanks | objval | time | nodes | first sol | 5% gap | best sol |
| n25p22a18 | 0-1 am | med | 42.63 | 0.4s | 67 | 0.2s | 0.4s | 0.4s |
| | | low | 64.55 | 0.8s | 85 | 0.5s | 0.6s | 0.8s |
| | 6-7 am | med | 42.51 | 0.4s | 60 | 0.2s | 0.2s | 0.4s |
| | | low | 62.82 | 0.8s | 77 | 0.5s | 0.7s | 0.8s |
| | 12-1 pm | med | 60.54 | 0.6s | 76 | 0.3s | 0.6s | 0.6s |
| | | low | 72.78 | 0.9s | 172 | 0.8s | 0.8s | 0.9s |
| | 6-7 pm | med | 60.54 | 0.5s | 80 | 0.1s | 0.2s | 0.5s |
| | | low | 60.54 | 0.5s | 80 | 0.1s | 0.2s | 0.5s |
| n88p64a64 | 0-1 am | med | 0 | 1.1s | 75 | 1.1s | 1.1s | 1.1s |
| | | low | 4.45 | 1.1s | 16 | 0.7s | 1.1s | 0.7s |
| | 6-7 am | med | 0 | 1.6s | 181 | 1.6s | 1.6s | 1.6s |
| | | low | 118.76 | 12.8s | 5495 | 0.6s | 0.9s | 12.8s |
| | 12-1 pm | med | 0 | 2.4s | 430 | 2.4s | 2.4s | 2.4s |
| | | low | 86.58 | 21.7s | 6738 | 12.2s | 16.7s | 21.7s |
| | 6-7 pm | med | 0 | 1.0s | 85 | 1.0s | 1.0s | 1.0s |
| | | low | 51.24 | 104.0s | 64940 | 0.8s | 1.0s | 1.5s |

Table 4: Detailed computational results for water supply networks `n25p22a18` and `n88p64a64` after presolving as described in Section 5.

**Acknowledgments.** Many thanks to Robert Schwarz and Jonas Schweiger for their valuable comments. Stefan Vigerske was supported by the DFG Research Center MATHEON *Mathematics for key technologies* in Berlin, see `http://www.matheon.de/`.

# References


[1] Tobias Achterberg. *Constraint Integer Programming.* PhD thesis, Technische Universität Berlin, 2007.

[2] Tobias Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009. `doi:10.1007/s12532-008-0001-1`.

[3] Pietro Belotti, Jon Lee, Leo Liberti, Francois Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4–5):597–634, 2009. `doi:10.1080/10556780903087124`.

[4] Timo Berthold. Primal Heuristics for Mixed Integer Programs. Master's thesis, Technische Universität Berlin, 2006.

[5] Timo Berthold and Ambros M. Gleixner. Undercover – a primal heuristic for MINLP based on sub-MIPs generated by set covering. In Pierre Bonami, Leo Liberti, Andrew J. Miller, and Annick Sartenaer, editors, *Proceedings of the EWMINLP*, pages 103–112, April 2010.

[6] Timo Berthold and Ambros M. Gleixner. Undercover – a primal MINLP heuristic exploring a largest sub-MIP. ZIB-Report 12-07, Zuse Institute Berlin, 2012. `http://vs24.kobv.de/opus4-zib/frontdoor/index/index/docId/1463/`.

[7] Timo Berthold, Stefan Heinz, Marc E. Pfetsch, and Stefan Vigerske. Large neighborhood search beyond MIP. In Luca Di Gaspero, Andrea Schaerf, and Thomas Stützle, editors, *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pages 51–60, 2011.

[8] Timo Berthold, Stefan Heinz, and Stefan Vigerske. Extending a CIP framework to solve MIQCPs. In Jon Lee and Sven Leyffer, editors, *Mixed-integer nonlinear optimization*, volume 154 of *The IMA volumes in Mathematics and its Applications*, pages 427–444. Springer, 2012. `doi:10.1007/978-1-4614-1927-3_15`.

[9] Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth. On the optimal design of water distribution networks: a practical MINLP approach. *Optimization and Engineering*, 13:219–246, 2012. `doi:10.1007/s11081-011-9141-7`.

[10] Jens Burgschweiger, Bernd Gnädig, and Marc C. Steinbach. Optimization models for operative planning in drinking water networks. ZIB-Report 04-48, Zuse Institute Berlin, 2004. `http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/823`.

[11] Björn Geißler, Oliver Kolb, Jens Lang, Günter Leugering, Alexander Martin, and Antonio Morsi. Mixed integer linear models for the optimization of dynamical transport networks. *Mathematical Methods of Operations Research*, 73:339–362, 2011. `doi:10.1007/s00186-011-0354-5`.

[12] Wei Huang. Operative Planning of Water Supply Networks by Mixed Integer Nonlinear Programming. Master's thesis, Freie Universität Berlin, 2011.

[13] Katrin Klamroth, Jens Lang, Günter Leugering, Alexander Martin, Antonio Morsi, Martin Oberlack, Manfred Ostrowski, and Roland Rosen. *Mathematical Optimization of Water Networks*, volume 162 of *International Series of Numerical Mathematics*. Birkhäuser-Science, Basel, 2012.

[14] Oliver Kolb. *Simulation and Optimization of Gas and Water Supply Networks*. PhD thesis, Technische Universität Darmstadt, 2011.

[15] Ailsa H. Land and Alison G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[16] Youdong Lin and Linus Schrage. The global solver in the LINDO API. *Optimization Methods & Software*, 24(4–5):657–668, 2009. `doi:10.1080/10556780902753221`.

[17] Hanif D. Sherali and Ernest P. Smith. A global optimization approach to a water distribution network design problem. *Journal of Global Optimization*, 11:107–132, 1997. `doi:10.1023/A:1008207817095`.

[18] Mohit Tawarmalani and Nikolaos V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming, Ser. A*, 99:563–591, 2004. `doi:10.1007/s10107-003-0467-6`.

[19] Stefan Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt Universität zu Berlin, 2012. Submitted.

[20] Andreas Wächter and Lorenz T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. `doi:10.1007/s10107-004-0559-y`.

[21] CppAD. A Package for Differentiation of C++ algorithms. `http://www.coin-or.org/CppAD/`.

[22] EPANET. A software that models water distribution piping systems. `http://www.epa.gov/nrmrl/wswrd/dw/epanet.html` (accessed March 2012).

[23] Ipopt. Interior Point Optimizer. `http://www.coin-or.org/Ipopt/`.

[24] SCIP. Solving Constraint Integer Programs. `http://scip.zib.de/`.

[25] SoPlex. Sequential object-oriented simPlex. `http://soplex.zib.de/`.

[26] Zimpl. Zuse Institute Mathematical Programming Language. `http://zimpl.zib.de/`.