Optimization and Engineering
https://doi.org/10.1007/s11081-018-9412-7

RESEARCH ARTICLE

Global optimality bounds for the placement of control valves in water supply networks

Filippo Pecci[1] · **Edo Abraham**[2] · **Ivan Stoianov**[1]

Received: 29 May 2018 / Revised: 19 November 2018 / Accepted: 19 November 2018
© The Author(s) 2018

Abstract

This manuscript investigates the problem of optimal placement of control valves in water supply networks, where the objective is to minimize average zone pressure. The problem formulation results in a nonconvex mixed integer nonlinear program (MINLP). Due to its complex mathematical structure, previous literature has solved this nonconvex MINLP using heuristics or local optimization methods, which do not provide guarantees on the global optimality of the computed valve configurations. In our approach, we implement a branch and bound method to obtain certified bounds on the optimality gap of the solutions. The algorithm relies on the solution of mixed integer linear programs, whose formulations include linear relaxations of the nonconvex hydraulic constraints. We investigate the implementation and performance of different linear relaxation schemes. In addition, a tailored domain reduction procedure is implemented to tighten the relaxations. The developed methods are evaluated using two benchmark water supply networks and an operational water supply network from the UK. The proposed approaches are shown to outperform state-of-the-art global optimization solvers for the considered benchmark water supply networks. The branch and bound algorithm converges to good quality feasible solutions in most instances, with bounds on the optimality gap that are comparable to the level of parameter uncertainty usually experienced in water supply network models.

Keywords Global optimization · Mixed-integer nonlinear programming · Valve placement · Pressure management · Water supply networks

✉ Filippo Pecci
f.pecci14@imperial.ac.uk

[1] Department of Civil and Environmental Engineering (InfraSense Labs), Imperial College London, London, UK

[2] Faculty of Civil Engineering and Geosciences, TU Delft, Stevinweg 1, 2628 CN Delft, The Netherlands

Published online: 28 November 2018

1 Introduction

The efficient management of hydraulic pressure in pipes results in reduction of leakage (Lambert 2000; Wright et al. 2015) and risk of pipe failure (Lambert and Thornton 2011), and it is therefore one of the main operational challenges in water supply networks (WSNs). Here we consider pressure management using pressure control valves, which regulate pressure at their outlet. We investigate the problem of simultaneously optimizing the placement and operational settings of control valves in WSNs, where the objective is to minimize average zone pressure (AZP). AZP is used as a surrogate measure for leakage. The problem formulation includes flows across network links and hydraulic heads at nodes as continuous decision variables. In addition, binary variables are introduced to model the placement of valves. Mass and energy conservation laws are enforced as optimization constraints, resulting in a nonconvex mixed integer nonlinear program (MINLP). The solution of process network optimization problems frequently relies on the solution of MINLPs. Some examples include synthesis of heat exchanger networks (Zamora and Grossmann 1998), multi-period blending (Kolodziej et al. 2013), optimal design and operation of gas networks (Pfetsch et al. 2015; Humpola and Fügenschuh 2015), and water supply networks (DAmbrosio et al. 2015). In the framework of WSNs, MINLP formulations are ubiquitous and employed in a variety of applications, ranging from optimal network design (Bragalli et al. 2012; Sherali et al. 1999) to pump scheduling (Menke et al. 2015; Gleixner et al. 2012).

Both heuristic and mathematical optimization methods were applied in previous work to solve the problem of optimal valve placement in water networks. Heuristic approaches based on genetic algorithms (GAs) have been widely used for solving the considered problem—see Reis et al. (1997), Araujo et al. (2006), Nicolini and Zovatto (2009), Liberatore and Sechi (2009), Ali (2015), De Paola et al. (2017). However, they present some limitations. Firstly, they can not guarantee optimality of the computed solutions, not even local optimality. Moreover, the number of objective function evaluations and hydraulic simulations required by these approaches grows rapidly with the size of the network, precluding the application of GAs when large operational water network are considered. Previous work has also investigated the application of mathematical optimization methods for the solution of the problem of optimal valve placement in WSNs—see Hindi and Hamam (1991), Eck and Mevissen (2012), Dai and Li (2014), Pecci et al. (2017a, b). Since the considered problem is nonconvex, approaches implemented in previous literature do not provide theoretical guarantees on the global optimality of the computed valve configuration.

This paper investigates mathematical optimization methods to generate a certified bound on the optimality gap of the computed solutions for the problem of optimal valve placement in WSNs, guaranteeing ϵ-sub-optimality. We formulate a branch and bound method, which is a common approach in global optimization. To the best of the authors knowledge, global optimization methods have not been previously applied to the problem of optimal valve placement in WSNs. Previous

literature has investigated global optimization techniques for optimal design of WSNs (Sherali et al. 1999; Raghunathan 2013). However, when pipe diameters are fixed, hydraulic heads and flow rates are uniquely determined and can be found by solving a strictly convex optimization problem (Raghunathan 2013). As a result, in the case of optimal WSN design, it is sufficient to focus the branch and bound on integer decision variables. On the contrary, when optimal operation of water supply networks is considered, spatial branch and bound is needed (Gleixner et al. 2012). In the present manuscript, we consider the problem of optimal valve placement in WSNs, where locations and operational settings of the control valves need to be simultaneously optimized. Therefore, branching is required on both continuous and integer variables.

The implemented branch and bound algorithm relies on a sequence of lower and upper bounds to the optimal value of the nonconvex MINLP in study—for a general review see Tawarmalani and Sahinidis (2002). Since all convex constraints within the problem formulation for optimal valve placement are linear, it is particularly convenient to generate lower bounds using linear relaxations of the nonconvex constraints (Tawarmalani and Sahinidis 2002, Chapter 4). The nonconvexity of MINLPs arising in the framework of water networks is due to the absolute power functions representing friction energy losses within the system's conservation laws—see Eq. (2a). Analogous nonconvex expressions have previously been studied in other engineering frameworks, where linear relaxations were formulated—see Humpola and Fügenschuh (2015), Gleixner et al. (2012), Liberti and Pantelides (2003), Tawarmalani and Sahinidis (2002), Udell and Boyd (2015), Vigerske (2012). We define linear relaxations of the nonconvex equality constraints considered here by extending the formulation proposed in Liberti and Pantelides (2003) for monomials of odd degree. Such linear relaxations define an outer approximation of the convex envelopes of the nonconvex equality constraints. We investigate the use of different number of linearizations for the outer approximation. The strength of the linear relaxations depends on the diameter of the decision variables' domain (Puranik and Sahinidis 2017). Therefore, we implement a domain reduction procedure, based on the solution of a series of linear programs (LPs). The proposed approach takes advantage of the underlying network structure to reduce the number of linear programming solves. Benefits and limitations of the developed methods are evaluated using two benchmark water networks, and a large-scale operational network from the UK. Moreover, the numerical results show that the proposed approach outperforms state-of-the-art global optimization solvers for the considered benchmark water networks. The branch and bound framework has enabled the convergence to ϵ-sub-optimal solutions for the problem of optimal valve placement, with bounds on the optimality gap comparable to the order of parameter and data uncertainties inherent in operational network models.

2 Problem formulation

A water supply network with n_0 water sources, n_n demand nodes and n_p pipes, is modelled as a directed graph with $n_n + n_0$ nodes and n_p edges. The operation of a network is considered under n_l different demand conditions during the diurnal cycle.

The nodal demands are denoted by $d^t \in \mathbb{R}^{n_n}$, while known hydraulic heads at water sources are indicated by $h_0^t \in \mathbb{R}^{n_0}$, for each $t = 1, \dots, n_l$. Furthermore, the vector of node elevations is represented by $\zeta \in \mathbb{R}^{n_n}$. Given $t \in \{1, \dots, n_l\}$, we consider hydraulic heads $h^t \in \mathbb{R}^{n_n}$ and flow rates $q^t \in \mathbb{R}^{n_p}$ as continuous decision variables. Moreover, vector $\eta^t \in \mathbb{R}^{n_p}$ is included to model the unknown head loss introduced by the action of pressure control valves. We introduce auxiliary variables $\theta^t \in \mathbb{R}^{n_p}$ to isolate the nonconvex terms representing the friction head losses occurring within the pipes of a network. These can be expressed by either the Hazen-Williams (HW) or Darcy-Weisbach (DW) formulae. Since both friction head loss formulae involve non-smooth nonconvex terms, it is convenient to use smooth quadratic approximations, computed over a range of flow (Pecci et al. 2017c). When a suitable quadratic approximation has been determined, it can be written as $\phi_j(q_j^t) = q_j^t(a_j|q_j^t| + b_j)$, with $a_j \geq 0$ and $b_j \geq 0$, for all $j = 1, \dots, n_p$ and $t = 1, \dots, n_l$—see also Eq. (28) in Appendix 1.

In this work, we investigate the optimal placement and operation of pressure control valves to minimize average zone pressure (AZP), defined as:

$$\frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \tag{1}$$

where $w_i := \sum_{j \in I_{(i)}} \frac{L_j}{2}$ and $I(i)$ is the set of indices for links incident at node i, L_j is the length of pipe j and $W = \sum_{i=1}^{n_n} w_i$ is a normalisation factor.

The optimization of valve placement and operation to minimize AZP is primarily subject to conservation energy (2b) and mass (2c) laws:

$$\theta^t = \boldsymbol{\Phi}(q^t), \quad t = 1, \dots n_l, \tag{2a}$$

$$A_{12} h^t + A_{10} h_0^t + \eta^t + \theta^t = 0, \quad t = 1, \dots, n_l \tag{2b}$$

$$A_{12}^T q^t - d^t = 0, \quad t = 1, \dots, n_l, \tag{2c}$$

where matrices $A_{12} \in \mathbb{R}^{n_p \times n_n}$ and $A_{10} \in \mathbb{R}^{n_p \times n_0}$ are edge-node incidence matrices for unknown and known head nodes, respectively. Moreover, we define the vector of friction head losses as $\boldsymbol{\Phi}(q^t) := [\phi_1(q_1^t) \dots \phi_{n_p}(q_{n_p}^t)]^T$.

We consider vectors of binary variables $z^+, z^- \in \{0, 1\}^{n_p}$ with the following properties:

- $z_j^+ = 1 \Leftrightarrow$ there is a valve on link j in the assigned positive flow direction
- $z_j^- = 1 \Leftrightarrow$ there is a valve on link j in the assigned negative flow direction
- $z_j^+ = z_j^- = 0 \Leftrightarrow$ there is no valve on link j
- $z_j^+ + z_j^- \leq 1$ prevents the placement of two valves on the same link.

The placement of control valves in a WSN is modelled by big-M constraints, ensuring that additional head losses have the same direction as flows and friction head

losses across the valves. The following constant vectors and matrices are used to formulate the big-M constraints. Given a vector of maximum allowed velocities across network pipes $v^{\max} \in \mathbb{R}^{n_p}$, let $q_L^t = -Sv^{\max}$, and $q_U^t = Sv^{\max}$, where $S := \mathbf{diag}(\pi D_1^2/4, \ldots, \pi D_{n_p}^2/4)$, with $D \in \mathbb{R}^{n_p}$ vector of pipe diameters. Let $Q_L^t := \mathbf{diag}(q_L^t)$, and $Q_U^t := \mathbf{diag}(q_U^t)$. Moreover, set $\theta_L^t := \boldsymbol{\Phi}(q_L^t)$, $\theta_U^t := \boldsymbol{\Phi}(q_U^t)$, $\Theta_L^t := \mathbf{diag}(\theta_L^t)$, and $\Theta_U^t := \mathbf{diag}(\theta_U^t)$. Minimum and maximum allowed hydraulic heads at nodes are specified by $h_{\min}^t, h_{\max}^t \in \mathbb{R}^{n_n}$, respectively. Define $\eta_L^t \in \mathbb{R}^{n_p}$ ad $\eta_U^t \in \mathbb{R}^{n_p}$ as follows:

$$\begin{aligned} (\eta_L^t)_k &:= (h_{\min}^t)_i - (h_{\max}^t)_j, \quad \forall i \xrightarrow{k} j \\ (\eta_U^t)_k &:= (h_{\max}^t)_i - (h_{\min}^t)_j, \quad \forall i \xrightarrow{k} j \end{aligned} \tag{3}$$

and set $N_L^t := \mathbf{diag}(\eta_L^t)$, $N_U^t := \mathbf{diag}(\eta_U^t)$. We formulate the following constraints:

$$\eta^t - N_U^t z^+ \leq 0, \quad t = 1, \ldots, n_l \tag{4a}$$

$$-\eta^t + N_L^t z^- \leq 0, \quad t = 1, \ldots, n_l \tag{4b}$$

$$-q^t - Q_L^t z^+ \leq -q_L^t, \quad t = 1, \ldots, n_l \tag{4c}$$

$$q^t + Q_U^t z^- \leq q_U^t, \quad t = 1, \ldots, n_l \tag{4d}$$

$$-\theta^t - \Theta_L^t z^+ \leq -\theta_L^t, \quad t = 1, \ldots, n_l \tag{4e}$$

$$\theta^t + \Theta_U^t z^- \leq \theta_U^t, \quad t = 1, \ldots, n_l. \tag{4f}$$

Let n_v be the number of valves to be installed. The desired properties of the binary variables are enforced by the linear constraints:

$$z^+ + z^- \leq \mathbf{1}_{n_p \times 1} \tag{5a}$$

$$\sum_{j=1}^{n_p} (z_j^+ + z_j^-) = n_v \tag{5b}$$

$$z^+, z^- \in \{0, 1\}^{n_p}. \tag{5c}$$

Finally, physical and operational bounds on all continuous variables are enforced by the following constraints:

$$q_L^t \leq q^t \leq q_U^t, \quad t = 1, \ldots, n_l \tag{6a}$$

$$\theta_L^t \leq \theta^t \leq \theta_U^t, \quad t = 1, \ldots, n_l \tag{6b}$$

$$\eta_L^t \leq \eta^t \leq \eta_U^t, \quad t = 1, \ldots, n_l \tag{6c}$$

$$h_{\min}^t \leq h^t \leq h_{\max}^t, \quad t = 1, \ldots, n_l. \tag{6d}$$

The valve placement optimization problem can be written in a compact form as:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \\ \text{subject to} \quad & \theta^t - \boldsymbol{\Phi}(q^t) = 0, \quad \forall t \in \{1, \dots, n_l\} \quad (2a) \\ & (q, h, \eta, \theta, z^+, z^-) \in \mathrm{F}(Q), \qquad \text{(MINLP}(Q)\text{)} \end{aligned}$$

where $q := (q^t)_{t=1,\dots,n_l}$, $h := (h^t)_{t=1,\dots,n_l}$, $\eta := (\eta^t)_{t=1,\dots,n_l}$, $\theta := (\theta^t)_{t=1,\dots,n_l}$. Furthermore, $\mathrm{F}(Q) \subset \mathbb{R}^{n_l n_p} \times \mathbb{R}^{n_l n_n} \times \mathbb{R}^{n_l n_p} \times \mathbb{R}^{n_l n_p} \times \{0,1\}^{n_p} \times \{0,1\}^{n_p}$ is the set defined by (2b)–(6d), which depends on $Q := \{q_L^t, q_U^t\}_{t=1,\dots,n_l}$. Problem MINLP($Q$) is a mixed integer nonlinear program (MINLP). Furthermore, the equality constraints (2a) are nonconvex, hence the problem formulation results in a nonconvex MINLP.

3 Solution method

The nonconvexity of MINLP(Q) is due to the presence of functions $(\phi_j(\cdot))_{j=1,\dots,n_p}$ within equality constraints (2a). Since the other constraints (2b)–(6d) are linear, it is convenient to consider linear relaxations of constraints (2a). Polyhedral relaxations for similar nonconvex expressions have been previously studied by Humpola and Fügenschuh (2015), Gleixner et al. (2012), Liberti and Pantelides (2003), Tawarmalani and Sahinidis (2002), Udell and Boyd (2015), Vigerske (2012). In this paper, linear relaxations developed in Liberti and Pantelides (2003) for monomials of odd degree are extended to the nonconvex equality constraints within the problem formulation for optimal valve placement in WSNs.

We implement a branch and bound method that relies on the generation of a sequence of lower and upper bounds to the optimal value. The present work takes a similar approach to Misener and Floudas (2013, 2014) and compute lower bounds solving Mixed Integer Linear Programming (MILP) relaxations of MINLP(Q). As discussed in Smith and Pantelides (1999), MILP relaxations result in tighter lower bounds then relaxed linear programs (LPs) and the algorithm is expected to converge in less iterations. However, they also require an higher computational effort for each branch and bound iteration. Nonetheless, as shown in the numerical results reported in Sect. 4, the implementation of state-of-the-art MILP solvers (e.g. Gurobi Optimization 2017) have enabled the application of the considered method to large problem instances.

3.1 Lower bounding MILP

Let $Q' = \{(q_L^t)', (q_U^t)'\}_{t=1,\dots,n_l}$ such that $q_L^t \le (q_L^t)' \le (q_U^t)' \le q_U^t$, for all $t \in \{1, \dots, n_l\}$. We consider the following restriction of MINLP(Q):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \\ \text{subject to} \quad & \theta^t - \Phi(q^t) = 0, \quad \forall t \in \{1, \dots, n_l\} \\ & (q, h, \eta, \theta, z^+, z^-) \in \mathrm{F}(Q'), \end{aligned} \tag{MINLP(Q')}$$

where $\mathrm{F}(Q') \subset \mathbb{R}^{n_l n_p} \times \mathbb{R}^{n_l n_n} \times \mathbb{R}^{n_l n_p} \times \mathbb{R}^{n_l n_p} \times \{0,1\}^{n_p} \times \{0,1\}^{n_p}$ is the set defined by (2b)–(6d) using the bounds in Q'. Let $y^*(Q')$ be the optimal value of MINLP(Q'). In the following, we formulate a MILP relaxation of MINLP(Q'), which yields a lower bound $\mathrm{L}(Q') \leq y^*(Q')$.

A detailed derivation of linear relaxations of (2a) is presented in Appendix 1, where the formulation for monomials of odd degree proposed by Liberti and Pantelides (2003) is extended to functions $(\phi_j(\cdot))_{j \in \{1,\dots,n_p\}}$. Such linear relaxations can be written as

$$R^t q^t + E^t \theta^t \leq r^t, \quad t = 1, \dots, n_l \tag{7}$$

for suitable matrices R^t and E^t, and vectors r^t, which depend on $(q_L^t)'$, $(q_U^t)'$, and a parameter $N_c \geq 0$ used to control the number of linearizations for the outer approximation of the convex envelopes of constraints (2a)—see Appendix 1. In Section 4, we investigate the effect of different number of linearizations on the convergence properties of the branch and bound algorithm. Using the above linear relaxations, it is possible to define the following MILP relaxation of MINLP(Q'):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \\ \text{subject to} \quad & R^t q^t + E^t \theta^t \leq r^t, \quad \forall t \in \{1, \dots, n_l\} \\ & (q, h, \eta, \theta, z^+, z^-) \in \mathrm{F}(Q'), \end{aligned} \tag{MILP(Q')}$$

Let $\mathrm{L}(Q')$ be the optimal value of MILP(Q'). If MILP(Q') is infeasible, then MINLP(Q') is infeasible and we have $\mathrm{L}(Q') = y^*(Q') = +\infty$. Otherwise, the solution of MILP(Q') yields a lower bound $\mathrm{L}(Q') \leq y^*(Q')$. Moreover, if the current best lower bound LB is available, set $\mathrm{L}(Q') \leftarrow \max(\mathrm{L}(Q'), \mathrm{LB})$.

3.2 Generation of upper bounds

Let $Q' = \{(q_L^t)', (q_U^t)'\}_{t=1,\dots,n_l}$ a set of bounds on flow variables as in Sect. 3.1. Recall that $y^*(Q')$ denotes the optimal value of MINLP(Q'). If MILP(Q') is infeasible, then set $U(Q') := +\infty$. Otherwise, let $\hat{z}^+, \hat{z}^- \in \{0,1\}^{n_p}$ be solution of MILP(Q'). We compute an upper bound to $y^*(Q')$ by fixing $z^+ = \hat{z}^+$ and $z^- = \hat{z}^-$ in MINLP(Q'). The resulting optimization problem is a nonconvex nonlinear program (NLP):

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \\ \text{subject to} \quad & \theta^t - \boldsymbol{\Phi}(q^t) = 0, \quad \forall t \in \{1, \dots, n_l\} \\ & (q, h, \eta, \theta, \hat{z}^+, \hat{z}^-) \in \mathrm{F}(Q'). \end{aligned} \tag{NLP(Q')}$$

If NLP(Q') is infeasible, set $\mathrm{U}(Q') := +\infty$. On the contrary, any (local) solution yields an upper bound $\mathrm{U}(Q') \geq y^*(Q')$.

The generation of a valid upper bound via the (local) solution of a nonlinear program is often computationally expensive, particularly when large problem instances are considered. However, NLP(Q') presents a high level of sparsity, that is retained by constraints (2b) and (2c) from the sparse structure of water supply networks. As a consequence, sparse NLP solvers offer scalable solution approaches for NLP(Q')—see the interior point method introduced in Waechter and Biegler (2006).

3.3 Domain reduction

The strength of the MILP relaxations is expected to have a significant impact on the convergence properties of branch and bound schemes (Belotti et al. 2009). This is confirmed by the numerical results reported in Sect. 4. As discussed in Appendix 1, in the case considered here, smaller ranges of flows lead to tighter relaxations. Therefore, we investigate pre-processing strategies to reduce the domains of the flow variables, focusing on Optimization Based Bound Tightening (OBBT), which relies on the solution of a series of optimization problems to tighten upper and lower bounds on selected variables—for a recent review on variable bound tightening approaches see Puranik and Sahinidis (2017).

In order to highlight the separable structure of the feasible set of MINLP(Q) with respect to the time indices, we introduce the time indexed sets $\{G^t(Q)\}_{t=1,\dots,n_l}$ such that $G^t(Q) \in \mathbb{R}^{n_p} \times \mathbb{R}^{n_n} \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_p} \times \{0,1\}^{n_p} \times \{0,1\}^{n_p}$ is defined by

$$(q^t, h^t, \eta^t, \theta^t, u_1^t, u_2^t) \in G^t(Q) \, \forall t \in \{1, \dots, n_l\} \Leftrightarrow (q, h, \eta, \theta, z^+, z^-) \in F(Q) \tag{8}$$

where $z^+ = u_1^t$ and $z^- = u_2^t$, for all $t \in \{1, \dots, n_l\}$, and $q = (q^t)_{t=1,\dots,n_l}$, $h = (h^t)_{t=1,\dots,n_l}$, $\eta = (\eta^t)_{t=1,\dots,n_l}$, $\theta = (\theta^t)_{t=1,\dots,n_l}$. By definition of $\{G^t(Q)\}_{t=1,\dots,n_l}$, the feasible set of MINLP(Q) is equivalent to the feasible set of the following problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n_l W} \sum_{t=1}^{n_l} w^T (h^t - \zeta) \\ \text{subject to} \quad & \theta^t - \boldsymbol{\Phi}(q^t) = 0, \quad \forall t \in \{1, \dots, n_l\} \\ & \left(q^t, h^t, \eta^t, \theta^t, u_1^t, u_2^t\right) \in G^t(Q), \quad \forall t \in \{1, \dots, n_l\} \\ & u_1^t - z^+ = \mathbf{0}_{n_p \times 1}, \quad \forall t \in \{1, \dots, n_l\} \\ & u_2^t - z^- = \mathbf{0}_{n_p \times 1}, \quad \forall t \in \{1, \dots, n_l\}. \end{aligned} \tag{9}$$

Let $\sigma \in \{-1, 1\}$, $t \in \{1, \ldots, n_l\}$, and $j \in \{1, \ldots, n_p\}$. Consider the mixed integer linear program:

$$\begin{aligned} \text{minimize} \quad & \sigma q_j^t \\ \text{subject to} \quad & R^l q^l + E^l \theta^l \leq r^l, \quad \forall l \in \{1, \ldots, n_l\} \\ & \left(q^l, h^l, \eta^l, \theta^l, u_1^l, u_2^l\right) \in G^l(Q), \quad \forall l \in \{1, \ldots, n_l\} \\ & u_1^l - z^+ = \mathbf{0}_{n_p \times 1}, \quad \forall l \in \{1, \ldots, n_l\} \\ & u_2^l - z^- = \mathbf{0}_{n_p \times 1}, \quad \forall l \in \{1, \ldots, n_l\}. \end{aligned} \tag{10}$$

where R^l, E^l and r^l represent the linear relaxations of (2a) computed using the flow bounds in Q as described in Appendix 1. The feasible set of Problem (10) is a relaxation of the set of feasible solutions of Problem (9). Tightening upper and lower bounds on the flow variables would require the solution of $2n_l n_P$ mixed integer linear programs analogous to (10). Mixed integer linear programs can be solved by state-of-the-art MILP solvers (Gurobi Optimization 2017). However, the dimension of Problem (10) and the number of problems to be solved can pose significant computational challenges when considering large-scale water networks. Therefore, the remainder of this section investigates tailored strategies to reduce the computational effort required by the considered domain reduction procedure.

Firstly, Problem (10) is converted into a linear program (LP) by replacing $G^t(Q)$ with a valid polyhedral relaxation $\widehat{G^t(Q)}$. Furthermore, the resulting LP can be decoupled with respect to the time indices $l \in \{1, \ldots, n_l\}$, omitting the consistency constraints in Problem (10). For each $t \in \{1, \ldots, n_l\}$ and $j \in \{1, \ldots, n_p\}$, we consider the following relaxation of Problem (10):

$$\begin{aligned} \text{minimize} \quad & \sigma q_j^t \\ \text{subject to} \quad & R^t q^t + E^t \theta^t \leq r^t \\ & (q^t, h^t, \eta^t, \theta^t, u_1^t, u_2^t) \in \widehat{G^t(Q)} \end{aligned} \tag{$\text{LP}_Q^{\sigma,t,j}$}$$

The OBBT method solves $\text{LP}_Q^{1,t,j}$ to compute a new lower bound $(q_L^t)_j' \geq q_{Lj}^t$, and $\text{LP}_Q^{-1,t,j}$ to obtain $(q_U^t)_j' \leq q_{Uj}^t$. If Q' is the resulting set of bounds, then MINLP(Q') and MINLP(Q) have the same optimal solution. Further domain reductions are achieved by iteratively applying the described process. In order to reduce the number of flow variables whose bounds are tightened through the solution of linear programs, we exploit the underlying network (graph) structure of $\text{LP}_Q^{\sigma,t,j}$. In what follows, few graph-theoretical definitions for WSNs are reviewed. The degree of an unknown head node in a WSN graph is defined as the number of links connected to the node. We define a *tree* in a WSN graph as an acyclic connected subgraph such that at least one of its unknown head nodes has degree one, and only one of its nodes is connected to either a looped part of the network or to a fixed head node (Deuerlein 2008). Such a unique node of a given tree is called its root node. The *forest* of a water network is

defined as the disjoint union of all trees in the network (Deuerlein 2008). The part of the network which is not contained in the forest but includes the roots of all the trees is called *core*. If a link belongs to the forest of a network graph, its flow is uniquely determined by the demand assigned to the adjacent forest nodes (Elhay et al. 2014). As a consequence, flows across forest links do not depend from the optimization process and their values are fixed. The forest-core decomposition of a WSN graph can be implemented using the algorithm presented in Elhay et al. (2014).

In the bound tightening approach proposed here, upper and lower bounds on flow variables corresponding to forest links are set a priori. Then, let core links j_1 and j_2 be connected in series through a node i. For each $t \in \{1, \dots, n_l\}$, flow variables in $\mathrm{LP}_Q^{\sigma,t,j}$ are subject to the following mass conservation laws:

$$A_{12}(j_1, i) q^t_{j_1} + A_{12}(j_2, i) q^t_{j_2} = d^t_i, \quad \forall t \in \{1, \dots, n_l\}. \tag{11}$$

Hence:

$$q^t_{j_2} = A_{12}(j_2, i) d^t_i - A_{12}(j_2, i) A_{12}(j_1, i) q^t_{j_1}, \quad \forall t \in \{1, \dots, n_l\}. \tag{12}$$

Given a sequence of core links connected in series, it is possible to select one link as representative and derive relations analogous to (12) for all other links in the sequence, proceeding by substitution. Let $\mathscr{C} \subset \{1, \dots, n_p\}$ be a subset of indices corresponding to network's links belonging to the core, where a unique representative for each sequence of links have been selected. An illustrative example of the definition of $\mathscr{C}$ for a simple network model is included in Appendix 2.

The proposed strategy is to perform OBBT only for those flow variables whose indices are in $\mathscr{C}$ and then propagate the bounds to the remaining core links using relations analogous to (12). Algorithm 1 is implemented for iterative domain reduction on the flow variables in MINLP(Q). Such iterative method can fail to converge to a fixed point in finite time (Puranik and Sahinidis 2017). Therefore, Algorithm 1 is terminated when the progress in the domain reduction is not significant or when the maximum number of iterations is reached. Given a possible choice of lower and upper bounds $Q' = \{(q^t_L)', (q^t_U)'\}$, define:

$$\mathbf{diam}(Q') := \max_{\substack{t = 1, \dots, n_l \\ j = 1, \dots, n_p}} ((q^t_U)'_j - (q^t_L)'_j). \tag{13}$$

The analytical study of the number of iterations required to generate tight variable bounds represents an open research problem (Puranik and Sahinidis 2017). Therefore, the maximum number of iterations in Algorithm 1 is set to 10 based on empirical experience. The number of linear programming solves at each iteration of Algorithm 1 is $2n_l|\mathscr{C}|$, where $|\mathscr{C}|$ is expected to be significantly smaller than n_p when operational water networks with a relatively low number of loops are considered. Furthermore, the *for-loops* within Algorithm 1 can be executed in parallel, as each iteration does not depend from the others.

Algorithm 1 Domain reduction on flow variables.

Initialisation:
  $k = 0$, $\rho = 0$, tol $= 0.95$, $K_{\max} = 10$;
  Define Q^0 as Q where known flow values across forest links have been fixed;
while $\rho \leq$ tol **and** $k \leq K_{\max}$ **do**
  for $j = 1 : |\mathscr{C}|$ **do**
    for $t = 1 : n_l$ **do**
      solve $\mathrm{LP}_{Q^k}^{1,t,\mathscr{C}(j)}$ and obtain $(q_L^t)'_{\mathscr{C}(j)}$;
      solve $\mathrm{LP}_{Q^k}^{-1,t,\mathscr{C}(j)}$ and obtain $(q_U^t)'_{\mathscr{C}(j)}$;
    end for
  end for
  propagate the new upper and lower bounds to links involved in pipe sequences;
  complete the vectors $(q_L^t)' \in \mathbb{R}^{n_p}$ and $(q_U^t)' \in \mathbb{R}^{n_p}$ with fixed flow rates across forest links;
  $Q^{k+1} := \{((q_L^t)', (q_U^t)')\}_{t=1,\ldots,n_l}$;
  $\rho \leftarrow \frac{\mathbf{diam}(Q^{k+1})}{\mathbf{diam}(Q^k)}$;
  $k \leftarrow k + 1$;
end while

3.4 Branch and bound algorithm

Let Q^{init} be the set of initial lower and upper bounds on flow variables. It can either be $Q^{\text{init}} = Q$ or $Q^{\text{init}} = Q^{\text{tight}}$, where Q^{tight} represents tightened lower and upper bounds on flow variables resulting from Algorithm 1. We denote by $y^* = y^*(Q)$ the optimal value of MINLP(Q). Then, by definition, $y^* = y^*(Q^{\text{init}})$. The branch and bound method iteratively generates a hierarchy of optimization problems, starting from MINLP(Q^{init}), represented by a binary tree named branch and bound tree. In particular, the algorithm creates a partition of $\mathrm{F}(Q^{\text{init}})$ given by $\{\mathrm{F}(Q') \mid Q' \in \mathscr{Q}\}$ such that

$$\min_{Q' \in \mathscr{Q}} \mathrm{L}(Q') \leq y^* \leq \min_{Q' \in \mathscr{Q}} \mathrm{U}(Q') \tag{14}$$

As standard in branch and bound methods, the proposed algorithm starts by computing lower and upper bounds to y^*—this is iteration 0. Then, the root problem MINLP(Q^{init}) is branched into two optimization problems—see Sect. 3.5. After branching, the algorithm computes lower and upper bounds on the two descendant problems as outlined in Sects. 3.1 and 3.2, respectively. The best upper and lower bounds are updated as necessary. Finally, a new problem from the branch and bound tree is selected for branching and the iteration is repeated. The method stops when a termination criterion is met and its implementation is described in Algorithm 2. The output of the algorithm is either an ε-sub-optimal solution to MINLP(Q) or a feasible solution with a certified bound on the optimality gap greater than ε.

The computed optimality gaps should be considered within the range of uncertainties that are inherent in modelling of operational water networks. As discussed in

Wright et al. (2015), pressure control in operational water networks is subject to multiple sources of data and modelling errors. These include the stochastic nature of customer demand, uncertainty in the hydraulic model parameters, network connectivity, measurements accuracy, and factors affecting the physical operation of control and isolation valves.

Algorithm 2 Branch and bound.

input: initial bounds on flow variables Q^{init};
Initialisation:
  $\mathcal{Q} := \{Q^{\text{init}}\}$, $j = 0$, MaxIter, $T_{\max}$
compute $\text{L}(Q^{\text{init}})$ and $\text{U}(Q^{\text{init}})$; // *Sections 3.1 and 3.2*
$\text{LB} \leftarrow \text{L}(Q^{\text{init}})$;
$\text{UB} \leftarrow \text{U}(Q^{\text{init}})$;
$j \leftarrow j + 1$;
while $\text{UB} - \text{LB} > \varepsilon$ **and** (j$\leq$ MaxIter **or** `Time` $< T_{\max}$) **do**
  select $Q^b \in \mathcal{Q}$ such that $\text{L}(Q^b) = \text{LB}$;
  split $\text{F}(Q^b)$ into $\text{F}(Q^{\text{left}})$ and $\text{F}(Q^{\text{right}})$; //*Sections 3.5*
  compute $\text{L}(Q^{\text{left}})$ and $\text{U}(Q^{\text{left}})$; // *Sections 3.1 and 3.2*
  compute $\text{L}(Q^{\text{right}})$ and $\text{U}(Q^{\text{right}})$; // *Sections 3.1 and 3.2*
  $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{Q^b\} \cup \{Q^{\text{left}}, Q^{\text{right}}\}$;
  $\text{LB} \leftarrow \min_{Q' \in \mathcal{Q}} \text{L}(Q')$;
  $\text{UB} \leftarrow \min(\text{UB}, \text{U}(Q^{\text{left}}), \text{U}(Q^{\text{right}}))$;
  $j \leftarrow j + 1$;
end while

Remark 1 If $Q' \in \mathcal{Q}$ is such that $\text{L}(Q') > \text{UB}$, then the global optimum cannot belong to $\text{F}(Q')$ and $\text{MINLP}(Q')$ can be pruned from the branch and bound tree. However, the selection strategy implemented in Algorithm 2 implies that the branch and bound iterations will terminate before Q' is selected for branching. Therefore, the method does not explicitly implement a pruning procedure.

3.5 Branching strategy

Let $\text{MINLP}(Q^b)$ be a problem in the branch and bound tree such that

$$\text{L}(Q^b) = \min_{Q' \in \mathcal{Q}} \text{L}(Q') \tag{15}$$

Branching $\text{MINLP}(Q^b)$ into two descendant problems $\text{MINLP}(Q^{\text{left}})$ and $\text{MINLP}(Q^{\text{right}})$ is equivalent to partitioning $\text{F}(Q^b)$ into two sets $\text{F}(Q^{\text{left}})$ and $\text{F}(Q^{\text{right}})$. A good branching strategy should aim to improve the lower bound (i.e. $\text{L}(Q^{\text{left}}) \geq \text{L}(Q^b)$ and $\text{L}(Q^{\text{right}}) \geq \text{L}(Q^b)$), and maintain a balanced branch and bound tree (Belotti 2013, Section 5.3.1). A branch and bound tree is said balanced if all problems in the tree are similarly difficult to solve. In this work, the following branching rule is implemented.

Branching rule Let $Q^b := \{(q_L^t)^b, (q_U^t)^b\}_{t=1,\dots,n_l}$ and $(\hat{q}, \hat{h}, \hat{\eta}, \hat{\theta}, \hat{z}^+, \hat{z}^-) \in \mathrm{F}(Q^b)$ be a solution of MILP(Q^b). Define indices $(l,k) \in \{1,\dots,n_l\} \times \{1,\dots,n_p\}$ such that

$$\hat{\theta}_k^l - \phi_k(\hat{q}_k^l) = \max_{\substack{t=1,\dots,n_l \\ j=1,\dots,n_p}} |\hat{\theta}_j^t - \phi_j(\hat{q}_j^t)|. \tag{16}$$

Let $Q^{\text{left}} := \{(q_L^t)^{\text{left}}, (q_U^t)^{\text{left}}\}_{t=1,\dots,n_l}$ and $Q^{\text{right}} := \{(q_L^t)^{\text{right}}, (q_U^t)^{\text{right}}\}_{t=1,\dots,n_l}$ be as follows:

$$\left(q_L^t\right)_j^{\text{left}} := (q_L^t)_j^b, \quad \forall (t,j) \in \{1,\dots,n_l\} \times \{1,\dots,n_p\} \tag{17}$$

$$\left(q_U^t\right)_j^{\text{left}} := (q_U^t)_j^b, \quad \forall (t,j) \in \{1,\dots,n_l\} \times \{1,\dots,n_p\} \setminus (l,k) \tag{18}$$

$$\left(q_L^t\right)_j^{\text{right}} := (q_L^t)_j^b, \quad \forall (t,j) \in \{1,\dots,n_l\} \times \{1,\dots,n_p\} \setminus (l,k) \tag{19}$$

$$\left(q_U^t\right)_j^{\text{right}} := (q_U^t)_j^b, \quad \forall (t,j) \in \{1,\dots,n_l\} \times \{1,\dots,n_p\} \tag{20}$$

$$\left(q_U^l\right)_k^{\text{left}} := \hat{q}_k^l \tag{21}$$

$$\left(q_L^l\right)_k^{\text{right}} := \hat{q}_k^l \tag{22}$$

Then, define of $\mathrm{F}(Q^{\text{left}})$ and $\mathrm{F}(Q^{\text{right}})$ using the bounds on flow variables contained in Q^{left} and Q^{right}.

The above branching strategy improves the lower bound, by definition of the linear relaxations given in Appendix 1. In fact, the formulation of MILP(Q^{left}) and MILP(Q^{right}) requires the inclusion of linear inequalities corresponding to the polyhedral relaxations computed with the new bounds on variable q_k^l. Such refined linear relaxations are exact at $\hat{q}_k^l$ resulting in tighter MILP relaxations. If $\hat{q}_k^l$ is too close to either $(q_L^l)_k^b$ or $(q_U^l)_k^b$, the implemented branching strategy can result in an unbalanced branch and bound tree. However, the relaxation error $|\hat{\theta}_j^t - \phi_j(\hat{q}_j^t)|$ is larger the more distant $\hat{q}_j^t$ is from $(q_L^t)_j^b$ and $(q_U^t)_j^b$, see Fig. 11. As a result, in most cases, $\hat{q}_k^l$ is expected to be closer to the middle point of the interval $[(q_L^t)_k^b, (q_U^t)_k^b]$ than to its extremes.

Table 1 Problem size of the 3 case studies

Name	# Cont. var.	# Bin. var.	# Lin. Constr.	# Non-convex terms
PescaraNet	365	198	1591	99
Net25	3192	74	9762	888
BWFLnet	28,251	2620	96,599	7107

The number of binary variables for BWFLnet reflects the restriction of candidate valve locations to those links with diameter greater or equal to 0.1 (m)

4 Results and discussion

In this section, the developed methods are evaluated using two benchmark network models, and a large operational water supply network from the UK. The corresponding problem sizes are reported in Table 1. All experiments presented below were conducted within MATLAB 2016b-64 bit for Windows 7, installed on a 2.50 GHz Intel Xeon(R) CPU E5-2640 0 with 12 Cores and 12 GB of RAM. The MILPs and LPs involved in Algorithms 1 and 2 are solved using GUROBI (v7.5) (Gurobi Optimization 2017), which is accessed via its MATLAB interface. In addition, all iterations in Algorithm 1 were executed in series, and GUROBI was forced to operate on a single-thread; all other parameters in GUROBI are set to their default values. Local solutions to the upper bounding NLPs considered in Sect. 3.2 are computed using IPOPT (v.3.12.6) (Waechter and Biegler 2006), accessed in MATLAB via an interface of the OPTI Toolbox (Currie and Wilson 2012). In the implementation of IPOPT, sparse gradients and Jacobians are directly supplied to the solver, in order to take advantage of the very sparse structure of our problem. The global optimization solvers BARON (v18.8.23) (Klnç and Sahinidis 2018) and SCIP (v3.2.1) (Gamrath et al. 2016) were implemented for the direct solution of the considered nonconvex MINLPs. The solver SCIP is accessed in MATLAB via an interface provided by OPTI Toolbox and relies on SoPLEX (v221) as linear solver and IPOPT (v3.12.6)

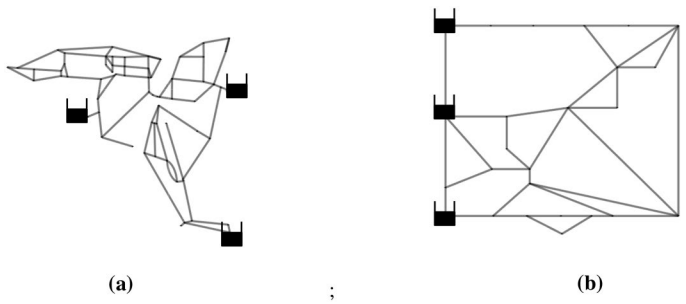


Fig. 1 Benchmark network models. **a** Layout of PescaraNet. **b** Layout of Net25

as nonlinear solver. In our implementation, we accessed BARON via its MATLAB interface, and used IBM ILOG CPLEX (v12.8) as linear solver, while BARON was allowed to select the nonlinear solver according to its dynamic strategy (Sahinidis 2018). Finally, we set $\varepsilon = 10^{-6}$, MaxIter $= +\infty$, and define $\texttt{Gap} := 100 \cdot \frac{UB-LB}{LB}$.

Firstly, we consider PescaraNet, a reduced model of the water supply network for the city of Pescara (Bragalli et al. 2012). The network's layout is presented in Fig. 1a. This network model has 68 demand nodes, 99 pipes and 3 water sources. Moreover, friction head losses are modelled using the HW formula. In this case, the formulation of MINLP(Q) includes a single demand condition and results in a relatively small size nonconvex MINLP whose characteristics are reported in Table 1. This water network has been already considered in Eck and Mevissen (2013) where a local optimization method was applied for computing optimal valve locations in PescaraNet. However, Eck and Mevissen (2013) did not include details on upper and lower bounds of flow variables, which precludes a direct comparison with the solutions obtained here. In the present implementation, maximum allowed velocity in each pipe is set to 2 (m/s) and minimum required pressure at each node to 19 (m). For every link $j \in \{1 \dots, n_p\}$, a quadratic head loss approximation $\phi_j(\cdot)$ is defined as in Pecci et al. (2017c).

The second case study is named Net25. This benchmark network model has been used to evaluate solution approaches for optimal valve placement problems in previous literature using heuristics (Reis et al. 1997; Araujo et al. 2006; Liberatore and Sechi 2009; Nicolini and Zovatto 2009; Ali 2015; De Paola et al. 2017) and mathematical optimization methods (Eck and Mevissen 2012; Dai and Li 2014; Pecci et al. 2017a, b). The results presented in this section allow the quantification of the level of sub-optimality of valve configurations for Net25 previously computed using heuristics and local optimization methods. The network has 22 nodes, 37 pipes and 3 reservoirs—see Fig. 1b. Details on pipes' characteristics, nodal demands and reservoirs' levels are presented in Jowitt and Xu (1990) and Dai and Li (2014). The hydraulic model of Net25 uses the HW formula to model friction head losses. We observe that Net25 has a smaller dimension with respect to PescaraNet; nonetheless, it results in a larger nonconvex MINLP as the problem formulation considers 24 demand conditions, one for each hour of the day—see Table 1. Moreover, we set the maximum allowed velocity in each pipe to 1 (m/s) and the minimum pressure at each node to 30 (m). Analogously to what done for PescaraNet, for each link j, a quadratic approximation of the friction losses $\phi_j(\cdot)$ is computed as proposed in Pecci et al. (2017c).

The developed methods are tested for optimizing the placement of 1 to 5 control valves in both PescaraNet and Net25, where the objective to be minimized is AZP. In these experiments, we set a time limit of 7200 (s). Given $\delta > 0$, let $\rho(\delta)$ be the percentage of problems with remaining Gap smaller or equal than δ, defined as

$$\rho(\delta) := 100 \frac{\left\{ \text{\# instances with } \texttt{Gap} \leq \delta \text{ at termination} \right\}}{\text{\# test problems}}. \tag{23}$$

For the purpose of this numerical study, we consider a uniform distribution of values of δ between 0 and 100, spaced by 10^{-4}. Let $N_c \geq 0$ be the parameter controlling the number of linearizations for the outer approximation of the convex envelopes of constraints (2a) - see Appendix 1. In the following, the symbol BB_{N_c} indicates the implementation of Algorithm 2 where the polyhedral relaxations (7) are computed as detailed in Appendix 1. When Algorithm 1 is implemented as pre-processing routine to reduce the domain of the flow variables, the combination of Algorithms 1 and 2 is denoted by dBB_{N_c}. Moreover, in order to ensure a total maximum computational time of 7200 (s), the value of $T_{\max}$ in Algorithm 2 is adjusted to account for the time spent in Algorithm 1.

Initially, we evaluate the performance of BB_{N_c} with $N_c \in \{0, 1, 3, 5\}$. The complete numerical results are reported in Tables 2, 3, 4, 5, 6, 7, 8, and 9 of Appendix 3. When no domain reduction is applied, the solutions computed after 7200 (s) by BB_{N_c} are within 10% of optimality, for all $N_c \in \{0, 1, 3, 5\}$. However, in the case of Net25, the obtained valve placements for $n_v \in \{3, 4, 5\}$ differ from the best-known solutions for the case study, with respect to those reported in Eck and Mevissen (2012), Dai and Li (2014), Pecci et al. (2017a), Pecci et al. (2017b). As shown in Fig. 2a, the inclusion of more linearizations for the outer approximation of the convex envelope of (2a) improves the convergence properties of the branch and bound algorithm with respect to BB_0. In addition, we observe that algorithms BB_1, BB_3, and BB_5 had similar performances. However, only BB_1 terminated in all instances with a `Gap` smaller than 7%.

We investigate the implementation of algorithms dBB_{N_c}, with $N_c \in \{0, 1, 3, 5\}$, for solving the considered optimal valve placement problems in PescaraNet and Net25. Firstly, Algorithm 1 was applied to tighten the flow variable bounds, for each $N_c \in \{0, 1, 3, 5\}$. When considering PescaraNet, the implementation of Algorithm 1 required the solution of 656 LPs in the case of $n_v = 1$ and $N_c = 0$. In comparison, the domain reduction procedure required the solution of 492 LPs in all the other cases. The average computational time required to run Algorithm 1 in PescaraNet is 18 (s)—see Tables 10, 12, 14, 16 for more details. The application of Algorithm 1

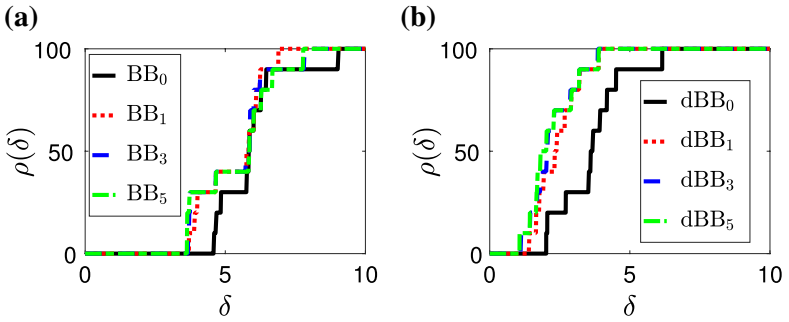


Fig. 2 Comparison of different number of linearizations for the outer approximation of the convex envelopes of (2a), with and without the application of domain reduction. **a** Performance of the branch and bound algorithm for different values of $N_c \in \{0, 1, 3, 5\}$. **b** Performance of the branch and bound algorithm with the application of domain reduction for different values of $N_c \in \{0, 1, 3, 5\}$

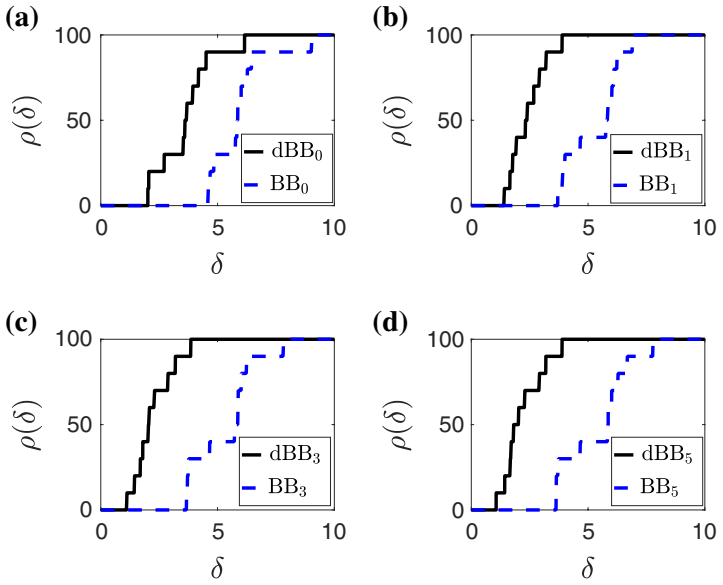


Fig. 3 Comparisons between implementations of the branch and bound algorithm with and without the application domain reduction as pre-solving routine

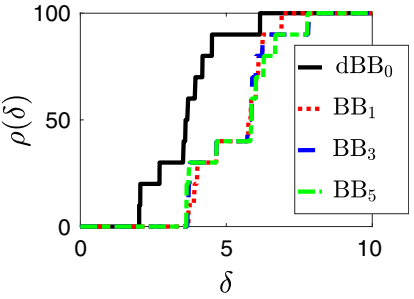


Fig. 4 Comparison between the combination of Algorithms 1 and 2 with $N_c = 0$, and the direct implementation of Algorithm 2 with $N_c \in \{1, 3, 5\}$

as domain reduction procedure for Net25 required the solution of 5568 LPs for each $n_v \in \{1, \ldots, 5\}$ and $N_c \in \{0, 1, 3, 5\}$, and an average computational time of 58 (s)—see Tables 11, 13, 15, 17. Algorithm 2 was then applied to the problem formulations with tightened bounds on the flow variables. The results are summarised in Tables 10, 11, 12, 13, 14, 15, 16, 17. As shown in Fig. 2b, in most instances, the solutions computed by dBB$_{N_c}$ are within 5% of optimality, for all $N_c \in \{0, 1, 3, 5\}$. Moreover, all the computed feasible solutions for Net25 equal the best-known valve locations obtained for the considered case study with respect to those reported in Eck and Mevissen (2012), Dai and Li (2014), Pecci et al. (2017a, b). According to these numerical results, the implementations of dBB$_{N_c}$ with $N_c > 0$ have better computational performance than dBB$_0$.

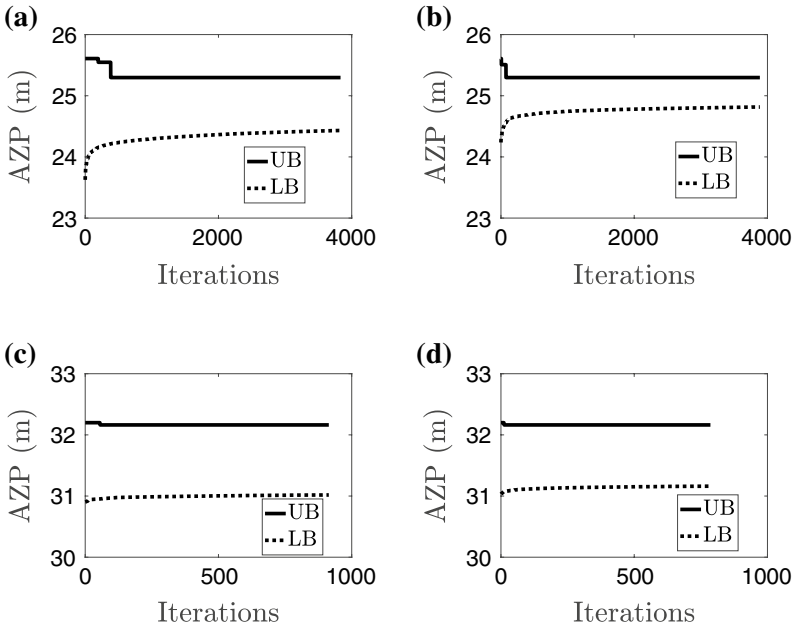


Fig. 5 Progress of dBB_0 and dBB_1 when applied to the problems of optimal placement of 3 valves in PescaraNet and Net25. **a** Progress of dBB_0 on PescaraNet for $n_v = 3$. **b** Progress of dBB_1 on PescaraNet for $n_v = 3$. **c** Progress of dBB_0 on Net25 for $n_v = 3$. **d** Progress of dBB_1 on Net25 for $n_v = 3$

In order to further investigate the effect of the domain reduction procedure on the convergence properties of Algorithm 2, we compare the implementation of the branch and bound algorithm with and without tightening the bounds on flow variables. As shown in Figs. 3, the domain reduction procedure significantly improved the performance of the branch and bound method for the considered problems of optimal valve placement in PescaraNet and Net25, irrespectively of the number of linearizations used. Furthermore, as shown in Fig. 4, the implementation of dBB_0 achieves better performance than the direct applications of branch and bound algorithm with $N_c \in \{1, 3, 5\}$. These results suggest that, for these benchmark water networks, increasing the number of linearizations improves the branch and bound ability to reduce the optimality gap, but not as much as the implementation of the developed domain reduction procedure. Furthermore, all the numerical experiments show rapid convergence of the algorithms to the final feasible solution and the associated optimality gap. As examples, Fig. 5 reports the progress of dBB_0 and dBB_1 when solving the problems of optimal placement of 3 valves on PescaraNet and Net25, respectively. In particular, it shows that good quality (if not optimal) solutions are computed early in the algorithm, while the remaining iterations are mostly used to improve the lower bounds.

In conclusion, we compare the performance of dBB_5 with state-of-the-art global optimization solvers SCIP and BARON for the solution of the considered nonconvex MINLPs. A time limit of 7200 (s) was set for all solvers. All other options in BARON and SCIP are set to their default values. As reported in

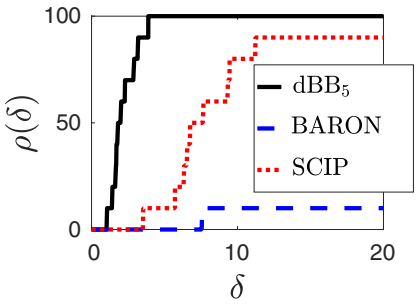


Fig. 6 Comparison between the combination of Algorithms 1, and 2 with $N_c = 5$ and two state-of-the-art global optimization solvers

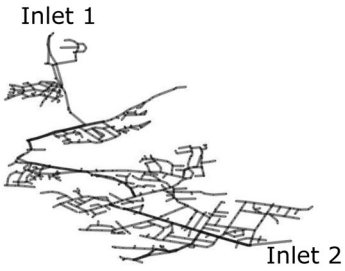


Fig. 7 Layout of BWFLnet

Tables 18 and 19, BARON failed to produce a feasible solution in all instances except one. These results are in line with the numerical experiments reported in Misener and Floudas (2014), where both SCIP (v3.0) and BARON (v12.7.2) failed to generate feasible solutions for the majority of test problems related to optimal design of WSNs—see Tables 40 and 41 in the supplementary material to Misener and Floudas (2014). Tables 20 and 21 show that SCIP has failed to find a feasible solution only for the problem of optimal placement of 5 valves in Net25. However, in most instances, SCIP resulted in a `Gap` larger than 5%. Furthermore, the majority of the feasible solutions computed by SCIP differ from the best-known solutions for Net25, which are reported in Pecci et al. (2017a) and were computed by the convex MINLP solver BONMIN (Bonami et al. 2008). Results reported in Tables 16, 17, 18, 19, 20, 21 show that the lower bounds computed by dBB$_5$ are tighter than those obtained by BARON and SCIP, in all instances. As illustrated in Fig. 6, after two hours of computations, the developed branch and bound method consistently results in smaller optimality gaps than the two state-of-the-art solvers.

4.1 Case study 3: BWFLnet

Finally, the proposed global optimization method for optimal valve placement has been applied to BWFLnet, the network model of the Smart Water Network Demonstrator (Field Lab) operated by Bristol Water, InfraSense Labs at Imperial College London and Cla-Val (Wright et al. 2014). This water supply network consists of 2310 nodes, 2369

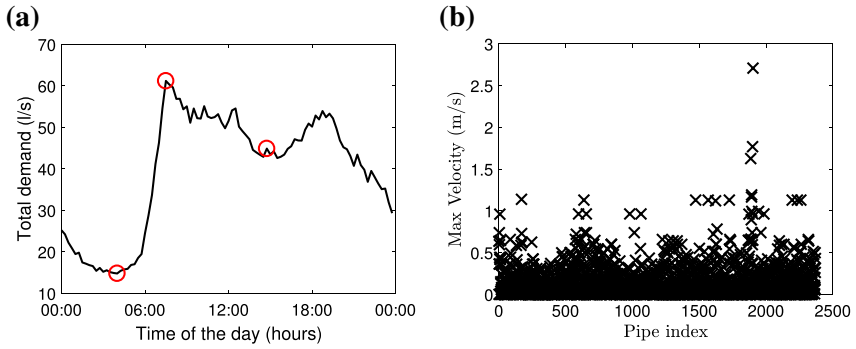


Fig. 8 Total network demand pattern and maximum velocities achieved in simulation in BWFLnet. **a** Total demand profile for BWFLnet. The circles correspond to the three time steps selected in this study. **b** Maximum velocity achieved in simulation across each link

pipes and 2 inlets (with fixed known hydraulic heads); its graph is presented in Fig. 7. The HW formula is used to model friction losses within BWFLnet.

The size of BWFLnet is two orders of magnitude larger than the other considered benchmark WSNs. In BWFLnet, the network's operator has already installed three automatic pressure control valves and two boundary control valves, which are optimally controlled in order to minimize AZP (Wright et al. 2015). In the problem formulation for optimal valve placement in BWFLnet, we model the three pressure control valves as open smooth pipes. Moreover, details on the operation of the two boundary valves have been provided by the valves' manufacturer. The daily demand profiles in BWFLnet include 96 different demand conditions, one every 15 minutes. However, the problem of optimal valve placement and operation in BWFLnet under 96 different demand conditions would result in a large MINLP with 227, 424 nonconvex constraints. In order to limit the problem size, MINLP(Q) is formulated on BWFLnet using only three most relevant demand conditions, which were selected to capture network's typical daily operational conditions at low night time demand, morning peak demand, and afternoon dip—see Fig. 8a. The AZP values computed using a problem formulation restricted to these 3 demand conditions are expected to be close to those obtained for a full set of 96 different demand conditions—this is confirmed to be the case for BWFLnet, as shown in Tables 24 and 26.

Note that the highest impact in terms of AZP reduction will be achieved by controlling pressure through pipes carrying large quantity of water, hence links with small diameters are not likely to be good candidates for control valve placement. As a consequence, links in BWFLnet whose diameter is smaller than 100 (mm) are discarded from the set of candidate valve locations. The minimum allowed pressure at all demand nodes is set to 18 (m), while this value is relaxed to zero for nodes with no demand. A quadratic approximation of friction losses is computed as discussed in Pecci et al. (2017c). The size of the resulting MINLP is shown in Table 1.

Multiple simulations of BWFLnet, with no installed valves, under different demand conditions, show that the maximum velocities achieved across network pipes do not exceed 3 (m/s)—see Fig. 8b. In view of Fig. 8b, for most pipes, feasible velocities are

expected to be significantly smaller than 3 (m/s). As a consequence, when upper and lower bounds on the flow variables are based on a maximum allowed velocity of 3 (m/s) for all pipes, the polyhedral relaxations (7) are not expected to be tight, leading to loose MILP relaxations—see also the discussion in Appendix 1. In order to tighten the polyhedral relaxations (7) and avoid unnecessarily large flow bounds, we define a tailored maximum allowed velocity for each link. It is known that the placement of control valves can result in velocity changes across network links (Abraham et al. 2018). In order to limit the possibility of discarding optimal solutions from the feasible set, we implement the following heuristic. Given a link $j \in \{1, \ldots, n_p\}$, let v_j^{sim} be the maximum velocity achieved in simulation under multiple demand conditions. If $v_j^{\text{sim}} \leq 1$ ($\frac{m}{s}$), then the optimized velocities across link j are allowed to exceed v_j^{sim} by at least a factor of two. In comparison, for links where the simulated velocities achieved higher rates, the allowed increment is more limited. The implemented strategy is detailed in the following:

$$v_j^{\max} := \begin{cases} 1 & \text{if } 0 \leq v_j^{\text{sim}} < 0.5 \\ 2 & \text{if } 0.5 \leq v_j^{\text{sim}} < 1.0 \\ 2.5 & \text{if } 1.0 \leq v_j^{\text{sim}} < 2.0 \\ 3 & \text{if } 2.0 \leq v_j^{\text{sim}} < 3.0, \end{cases} \tag{24}$$

Then, MINLP(Q) was formulated on BWFLnet using tailored maximum velocities to define upper and lower bounds on flow variables Q. We consider the optimal placement of 1 to 5 control valves in BWFLnet. In the numerical experiments reported in this section, we set a time limit $T_{\max} = 86{,}400$ (s) (1 day). We investigate the direct application of Algorithm 2 with $N_c \in \{0, 3\}$, without implementing the domain reduction procedure. The results are summarised in Tables 22 and 23. In the case of $N_c = 0$, the relative optimality gap obtained for $n_v \in \{1, 2, 3\}$ are larger than 30%. No feasible solution was found after a day of computations for $n_v \in \{4, 5\}$. Including more linearizations for the outer approximation of the convex envelopes of (2a) results in small improvements for the cases of $n_v \in \{1, 2\}$, while no feasible solution was obtained when $n_v \in \{3, 4, 5\}$—see Table 23.

Then, the domain reduction procedure described in Algorithm 1 was applied as pre-processing reoutine to tighten the flow variable bounds within the formulation of MINLP(Q) on BWFLnet. In this case, the fraction of links involved in LP solves corresponds to less than the 10% of the whole set of network links. Such reduction is due to the structure of BWFLnet, a typical water network from the United Kingdom, where a considerable portion of links belongs to the forest. Moreover, hydraulic models of operational networks often present sequences of links connected in series, used to model the existence of different customer connections along network pipes. As a result, the proposed graph-theory based decomposition has significantly reduced the computational cost associated with Algorithm 1, which required the solution of 2532 LPs for each case of $n_v = 1, \ldots, 5$. In the numerical experiments reported here, such LPs were solved sequentially. As a result, Algorithm 1 required

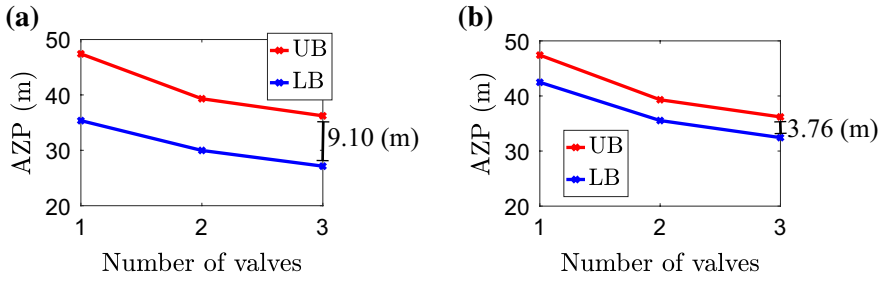


Fig. 9 Upper and lower bounds for the solution of MINLP(Q) on BWFLnet. **a** Upper and lower bounds computed by Algorithm 2, with $N_c = 0$. **b** Upper and lower bounds computed by the combination of Algorithm 1 and 2, with $N_c = 0$

roughly one hour of CPU time, in all instance—see Tables 24 and 25. However, as previously observed, the LPs solved at each iteration of Algorithm 1 do not depend on each other. Therefore, in a practical implementation, they can be solved in parallel, exploiting the existence of multiple computational cores.

Algorithm 1 was then applied to MINLP(Q^{tight}), with $N_c \in \{0, 3\}$—see Tables 24 and 25. When $N_c = 0$, the bounds on the absolute optimality gaps obtained for $n_v \in \{1, 2, 3\}$ are between 3 and 5 (m)—see also Fig. 9. Optimality bounds of such magnitude are comparable to the order of uncertainty affecting pressure control in BWFLnet (Wright et al. 2015). Hence, the quality of the computed feasible solutions is considered to be acceptable. Again, in the cases of $n_v \in \{4, 5\}$, no feasible solution was found before reaching the time limit. Similar optimality gaps were obtained when $N_c = 3$, but the algorithm failed to compute a feasible solution for $n_v \in \{3, 4, 5\}$. These results suggest that the inclusion of different number of linearizations for the outer approximation of the convex envelopes of (2a) did not improve the performance of the branch and bound algorithm for solving the problem of optimal valve placement in BWFLnet.

We observe that the feasible solutions computed with and without domain reduction are the same. However, as shown in Fig. 9, the application of Algorithm 1 has significantly improved the quality of the lower bounds computed by the branch and bound algorithm. Nonetheless, the optimality gaps reported in Table 24 are too wide to be reduced within the prescribed time limit, even if the domain reduction routine is applied. The numerical results show that, in all problem instances, most pipes in BWFLnet experience low velocities, which are significantly smaller than their expected maximum velocities, despite the implemented heuristic to tailor maximum allowed velocities. As a consequence, the polyhedral relaxations of Eq. (2a) corresponding to these pipes are not sufficiently tight—see also the discussion in Appendix 1.

Next, given optimal valve locations for $n_v \in \{1, 2, 3\}$, computed with Algorithm 2, we have formulated a nonconvex nonlinear program to optimize valve operation under the complete set of 96 demand conditions. Such NLP is obtained from the formulation of MINLP(Q) by fixing the values of the binary variables corresponding to optimized valve locations, and it is equivalent to the problem of optimizing the actuators operation. Algorithm 1 is applied to tighten the bounds on flow variables, and

Algorithm 2 is subsequently implemented to solve the nonconvex NLP with tightened variable bounds. Lower and upper bounds computed at the root of the branch and bound tree are reported in Table 26, for $n_v \in \{1, 2, 3\}$. As expected, the upper bounds to the AZP values computed considering the full set of multiple demand conditions are close to those obtained for the restriction to three demand conditions. Moreover, in this case, the domain reduction procedure has resulted in considerably tighter lower bounds, without performing any branching operation. In contrast, the results reported in Table 24 indicate that that inclusion of valve locations as unknowns significantly reduces the ability of Algorithm 1 to compute tight estimates of flow variable domains. This situation has an interpretation from the hydraulic application perspective. In fact, appropriate changes in network topology induced by valves closures can result in increased flow velocities across some pipes (Abraham et al. 2018).

In conclusion, we observe that the average computational cost associated with the solution of a single MILP relaxation for BWFLnet is considerably higher than for Net25 and PescaraNet. In fact, solving the MILP relaxation at the root of the branch and bound tree for BWFLnet required 2–6 orders of magnitude more computational time than what experienced for Net25 and PescaraNet. This behaviour is predictable, as the problem of optimal valve placement and operation in BWFLnet results in a MINLP whose size is one to two orders of magnitude larger than the size of the problem formulation on Net25 and PescaraNet—see Table 1. It is known that computational effort required to solve a mixed-integer program grows combinatorially with the size of the problem. These numerical experiments were conducted on a single computational thread of a desktop machine, using a standard implementation of GUROBI for the solution of the MILP relaxations. The availability of additional computational capability and the use of a tailored MILP solver could speed-up the optimization process.

All three case studies show that good quality solutions are often computed at the root node of the branch and bound tree (i.e. iteration 0). This is in accordance with the work by Diamond et al. (2018), presenting a number of examples where good quality solutions to nonconvex optimization problems are recovered from the solution of suitable convex relaxations. The results suggest that Algorithm 2 can be early terminated to generate good quality solutions of MINLP(Q), by opportunely setting time limit or maximum number of iterations. Moreover, observe that Algorithm 2 provides more information than local optimization methods for optimal valve placement in WSNs studied in previous work. In fact, the algorithm always generates a certified bound on the optimality gap of the computed solution. Local optimization methods can be implemented before starting Algorithm 2, in order to rapidly generate good quality feasible solutions.

5 Conclusions and future work

In this manuscript, we have investigated the application of branch and bound strategies to compute ε-sub-optimal solutions for the problem of optimal valve placement and operation in water supply networks. The implemented algorithm relies on the solution of MILP relaxations of the original nonconvex MINLP. Furthermore, a tailored domain reduction procedure was implemented to tighten the MILP relaxations. In contrast to previously published solution methods for optimal valve placement in

water networks, the presented algorithm terminates with a certified bound on the optimality gap of the computed solution, thus providing additional information to support the design and operation of water supply networks.

The proposed branch and bound method has successfully generated good quality feasible solutions in two benchmark water networks and a large water supply network, after few iterations, with bounds on the optimality gap comparable to the order of uncertainty usually experienced in pressure control of water supply networks. Furthermore, the results suggest that the proposed domain reduction strategy is more effective in improving the convergence properties of the algorithm, than simply increasing the number of linearizations used to define the polyhedral relaxations. Moreover, the results reported in this manuscript show that, for the considered benchmark water networks, the proposed branch and bound algorithms outperform state-of-the-art global optimization solvers SCIP and BARON. These results highlight the challenges of applying *off-the-shelf* global solvers to the problem in study.

However, the lower bounds generated by the algorithm experience slow progress, as shown in Fig. 5. They can be improved by performing Algorithm 1 on Q^{left} and Q^{right}, so that the new bounds on the branching variable can be propagated to the remaining flow variables. Recall that, at each iteration of Algorithm 1, the required solutions of the $2n_l|\mathscr{C}|$ linear programs can be computed in parallel. As a consequence, if enough computational cores are available, the outlined bound propagation strategy can be applied at each branch and bound iteration, without dramatically increasing the overall computational time. Furthermore, future work should investigate the inclusion of additional valid linear inequalities in the formulation of MILP(Q'). A possible strategy is to use the locally optimal solutions generated at each stage of the branch and bound method, following an approach similar to Humpola et al. (2016). In addition, when large operational water networks are considered, tailored solvers for the relaxed MILPs can be designed using suitable decomposition strategies, following ideas discussed in Vigerske (2012, Chapters 3 and 4).

Although the present work has focused on the minimization of AZP, other convex objective functions can be minimized within the same framework, with little modifications to the discussed formulation. In particular, if the objective function is nonlinear, the generation of lower bounds requires the solution convex MINLPs. Furthermore, optimal design (Bragalli et al. 2012), optimal valve control (Wright et al. 2015), and optimal pump scheduling (Menke et al. 2015) problems result in optimization problems involving nonconvex constraints like (2a). As a consequence, using suitable linear relaxations and NLP sub-problems, Algorithms 1 and 2 can be applied to guarantee the optimality of the solution found, or a bound on the optimality gap, for such nonconvex optimization problems.

Acknowledgements The authors thank the anonymous reviewers for their helpful comments and suggestions, and for their valuable advice on the formulation of the polyhedral relaxations described in Appendix 1. This work was supported by the NEC-Imperial Smart Water Systems project, and EPSRC (EP/P004229/1, Dynamically Adaptive and Resilient Water Supply Networks for a Sustainable Future).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix 1: Polyhedral relaxations of nonconvex head loss constraints

We look for polyhedral relaxations of constraints

$$\theta_j^t - \phi_j(q_j^t) = 0, \quad t = 1, \dots, n_l,\ j = 1, \dots n_p$$

These will be written as

$$R^t q^t + E^t \theta^t \le r^t, \quad t = 1, \dots, n_l \tag{25}$$

for suitable matrices R^t and E^t and constant vectors r^t that depend on q_L^t and q_U^t. Moreover, the present appendix demonstrates that the strength of the relaxations depends on the tightness of the bounds on the flow variables.

In the following, the indices t and j are omitted as the mathematical derivation does not depend on them. Consider the set

$$\left\{ (q, \theta) \;\middle|\; \theta = \phi(q), \quad \forall q \in [q_L, q_U] \right\} \tag{26}$$

A convex relaxation is given by

$$\left\{ (q, \theta) \;\middle|\; \underline{\phi}(q) \le \theta \le \bar{\phi}(q), \quad \forall q \in [q_L, q_U] \right\} \tag{27}$$

where $\underline{\phi}(\cdot)$ and $\bar{\phi}(\cdot)$ are suitable convex and concave functions, respectively. Analytical expressions for $\underline{\phi}(\cdot)$ and $\bar{\phi}(\cdot)$ can be derived following a similar procedure to the one illustrated in Liberti and Pantelides (2003) for monomials of odd degree. Recall that $\phi(\cdot)$ is defined as

$$\phi(q) := q(a|q| + b) \quad q \in [q_L, q_U], \quad a \ge 0, \quad b \ge 0. \tag{28}$$

We can assume that $a > 0$, otherwise $\phi(\cdot)$ is a linear function and relaxations are not needed. Let $q_L \le 0 \le q_U$ be upper and lower bounds on flow q. We look for $\bar{q} \le 0$ such that the line from $(\bar{q}, \phi(\bar{q}))$ to $(q_U, \phi(q_U))$ is tangent to the curve $\phi(\cdot)$ at $(\bar{q}, \phi(\bar{q}))$. This is represented by the equation:

$$\frac{\phi(\bar{q}) - \phi(q_U)}{\bar{q} - q_U} = \phi'(\bar{q}). \tag{29}$$

Multiplying by $(\bar{q} - q_U)$ on both sides of the previous equation yields:

$$\bar{q}(a|\bar{q}| + b) - q_U(a|q_U| + b) - (2a|\bar{q}| + b)(\bar{q} - q_U) = 0. \tag{30}$$

Since $a > 0$, $\bar{q} \le 0$ and $q_U \ge 0$, we finally obtain

$$-\bar{q}^2 + q_U^2 + 2\bar{q}q_U = 0. \tag{31}$$

Thus $\bar{q} := (1 - \sqrt{2})q_U$. Analogously, the unique point $\underline{q} \ge 0$ such that the line from $(\underline{q}, \phi(\underline{q}))$ to $(q_L, \phi(q_L))$ is tangent to $\phi(\cdot)$ in $(\underline{q}, \phi(\underline{q}))$ is defined as $\underline{q} := (1 - \sqrt{2})q_L$.

If $\bar{q}_L < \bar{q} < 0 < \underline{q} < q_U$, we have the following convex over and under estimators (see Fig. 10a):

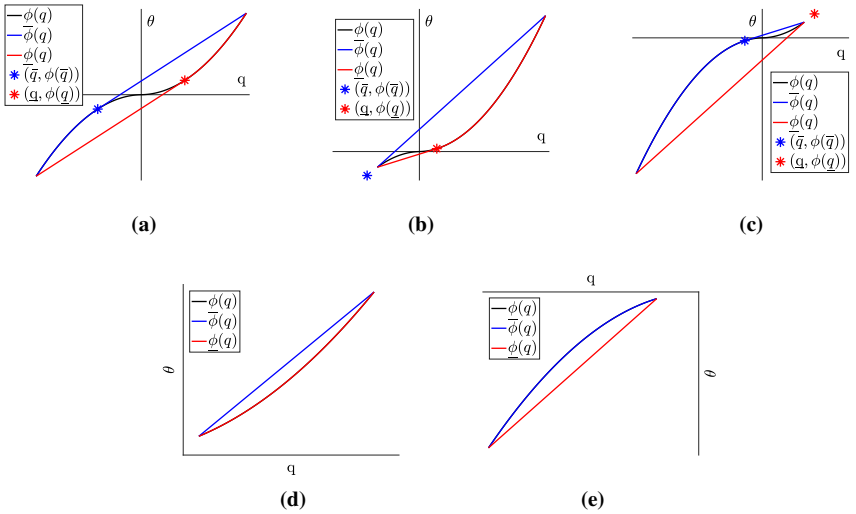


Fig. 10 Convex envelopes of nonconvex constraints $\theta = \phi(q)$. **a** $q_L < \bar{q} < 0 < \underline{q} < q_U$. **b** $\bar{q} \leq q_L < 0 < \underline{q} < q_U$. **c** $q_L < \bar{q} < 0 < q_U \leq \underline{q}$. **d** $0 \leq q_L < q_U$. **e** $q_L < q_U \leq 0$.

$$\overline{\phi}(q) = \begin{cases} \phi(q), & q_L \leq q \leq \bar{q} \\ \phi(q_U) + \frac{\phi(\bar{q}) - \phi(q_U)}{\bar{q} - q_U}(q - q_U), & \bar{q} \leq q \leq q_U \end{cases} \tag{32}$$

$$\underline{\phi}(q) = \begin{cases} \phi(q_L) + \frac{\phi(\underline{q}) - \phi(q_L)}{\underline{q} - q_L}(q - q_L) & q_L \leq q \leq \underline{q} \\ \phi(q) & \underline{q} \leq q \leq q_U \end{cases} \tag{33}$$

When $\bar{q} \leq q_L$, it is necessary to modify the definition of $\overline{\phi}(\cdot)$ as follows (see Fig. 10b):

$$\overline{\phi}(q) = \phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L), \forall q \in [q_L, q_U]. \tag{34}$$

Analogously, if $\underline{q} \geq q_U$ set (see Fig. 10c)

$$\underline{\phi}(q) = \phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L), \forall q \in [q_L, q_U]. \tag{35}$$

Now, assume that $0 \leq q_L < q_U$. In this case, the convex envelope (see Fig. 10d) is given by

$$\underline{\phi}(q) = \phi(q) \tag{36}$$

$$\overline{\phi}(q) = \phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \tag{37}$$

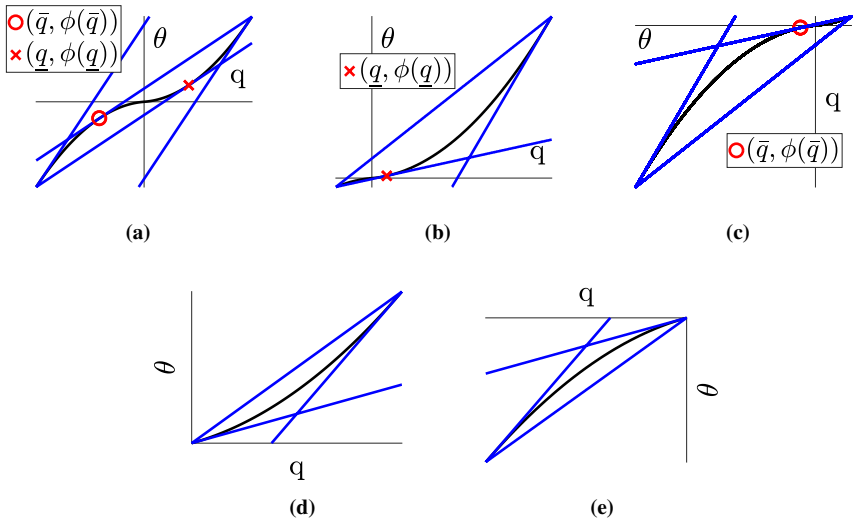


Fig. 11 Polyhedral relaxations of nonconvex constraints $\theta = \phi(q)$, with $N_c = 0$. **a** $q_L < \bar{q} < 0 < \underline{q} < q_U$. **b** $\bar{q} \le q_L < 0 < \underline{q} < q_U$. **c** $q_L < \bar{q} < 0 < q_U \le \underline{q}$. **d** $0 \le q_L < q_U$. **e** $q_L < q_U \le 0$

Analogously, if $q_L < q_U \le 0$, set (see Fig. 10e)

$$\underline{\phi}(q) = \phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \tag{38}$$

$$\overline{\phi}(q) = \phi(q) \tag{39}$$

In this manuscript, our aim is to apply a branch and bound method to the nonconvex problem MINLP(Q). At each step of the algorithm, a solution of a convex MINLP relaxation of MINLP(Q) will yield a lower bound on the optimal objective function value. Observe that, with the exception of (2a), all the other constraints in MINLP(Q) are linear. Therefore, it is convenient to study polyhedral relaxations of the nonconvex set in (26), which will result in a Mixed Integer Linear Programming (MILP) relaxation of MINLP(Q)—for a general review, see Tawarmalani and Sahinidis (2002, Chapter 4). Let $N_c \ge 0$ be a parameter used to control the number of linearizations for the outer approximation of the convex envelopes in Fig. 10.

Assume $q_L < \bar{q} < 0 < \underline{q} < q_U$. Let $N_c > 0$, and $q_L < q_1^M < \cdots < q_{N_c}^M < \bar{q}$ and $\underline{q} < q_1^S < \cdots < q_{N_c}^S < q_U$ be sequences of equidistant points, where $\bar{q}$ and $\underline{q}$ are the tangential points defined above. A polyhedral relaxation is given by the following linear inequalities (see Figs. 11a, 12a) :

$$\phi(q_L) + \frac{\phi(\underline{q}) - \phi(q_L)}{\underline{q} - q_L}(q - q_L) \le \theta \tag{40}$$

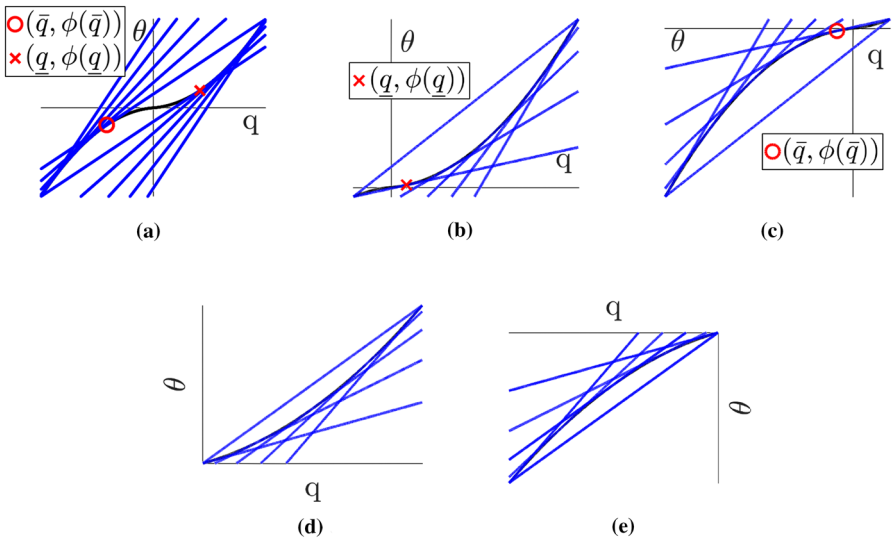


Fig. 12 Polyhedral relaxations of nonconvex constraints $\theta = \phi(q)$, with $N_c = 3$. **a** $q_L < \bar{q} < 0 < \underline{q} < q_U$. **b** $\bar{q} \le q_L < 0 < \underline{q} < q_U$. **c** $q_L < \bar{q} < 0 < q_U \le \underline{q}$. **d** $0 \le q_L < q_U$. **e** $q_L < q_U \le 0$.

$$\phi(q_U) + \frac{\phi(\bar{q}) - \phi(q_U)}{\bar{q} - q_U}(q - q_U) \ge \theta \tag{41}$$

$$\phi(q_U) + \phi'(q_U)(q - q_U) \le \theta \tag{42}$$

$$\phi(q_L) + \phi'(q_L)(q - q_L) \ge \theta \tag{43}$$

$$\phi(q_i^S) + \phi'(q_i^S)(q - q_i^S) \le \theta, \quad \forall i \in \{1, \dots N_c\} \tag{44}$$

$$\phi(q_i^M) + \phi'(q_i^M)(q - q_i^M) \ge \theta, \quad \forall i \in \{1, \dots N_c\} \tag{45}$$

If $N_c = 0$, the polyhedral relaxation is defined only by constraints (40)–(43).

Now, let $\bar{q} \le q_L < 0 < \underline{q} < q_U$. Assume that $N_c > 0$, and let $\underline{q} < q_1^S < \cdots < q_{N_c}^S < q_U$ be sequence of equidistant points. In this case, we consider the linear relaxation given by (see Figs. 11b, 12b):

$$\phi(q_L) + \frac{\phi(\underline{q}) - \phi(q_L)}{\underline{q} - q_L}(q - q_L) \le \theta \tag{46}$$

$$\phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \ge \theta \tag{47}$$

$$\phi(q_U) + \phi'(q_U)(q - q_U) \le \theta \tag{48}$$

$$\phi(q_i^S) + \phi'(q_i^S)(q - q_i^S) \le \theta, \quad \forall i \in \{1, \dots, N_c\} \tag{49}$$

If $N_c = 0$, the corresponding polyhedral relaxation is defined only by constraints (46)–(48).

Analogously, if $q_L < \bar{q} < 0 < q_U \le \underline{q}$ and $N_c > 0$, let $q_L < q_1^M < \cdots < q_{N_c}^M < \bar{q}$ be a sequence of equidistant points. We have (see Figs. 11c, 12c):

$$\phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \le \theta \tag{50}$$

$$\phi(q_U) + \frac{\phi(\bar{q}) - \phi(q_U)}{\bar{q} - q_U}(q - q_U) \ge \theta \tag{51}$$

$$\phi(q_L) + \phi'(q_L)(q - q_L) \ge \theta \tag{52}$$

$$\phi(q_i^M) + \phi'(q_i^M)(q - q_i^M) \ge \theta, \quad \forall i \in \{1, \dots, N_c\} \tag{53}$$

When $N_c = 0$, the polyhedral relaxation is defined by constraints (50)–(52).

Assume $0 \le q_L < q_U$. Let $N_c > 0$ and consider a sequence of equidistant points $q_L < q_1^S < q_{N_c}^S < q_U$. We define (see Fig. 11d, 12d):

$$\phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \ge \theta \tag{54}$$

$$\phi(q_L) + \phi'(q_L)(q - q_L) \le \theta\theta \tag{55}$$

$$\phi(q_U) + \phi'(q_U)(q - q_U) \le \theta\theta \tag{56}$$

$$\phi(q_i^S) + \phi'(q_i^S)(q - q_i^S) \le \theta, \quad \forall i \in \{1, \dots, N_c\}\theta \tag{57}$$

If $N_c = 0$, the polyhedral relaxation is defined only by constraints (54)–(56).

Finally, if $q_L < q_U \le 0$ and $N_c > 0$, let $q_L < q_1^M < \cdots < q_{N_c}^M < q_U$ be a sequence of equidistant points. A linear relaxation is given by (see Figs. 11e, 12e):

$$\phi(q_L) + \frac{\phi(q_U) - \phi(q_L)}{q_U - q_L}(q - q_L) \le \theta \tag{58}$$

$$\phi(q_U) + \phi'(q_U)(q - q_U) \ge \theta \tag{59}$$

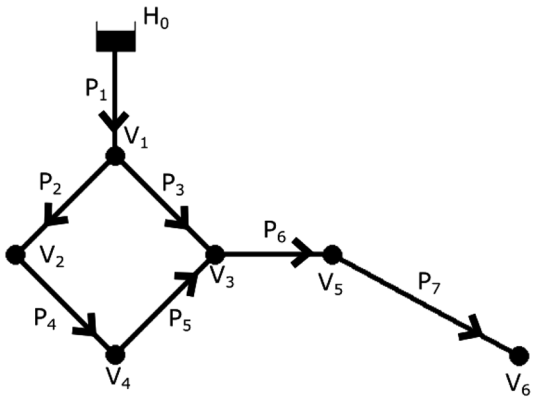
$$\phi(q_L) + \phi'(q_L)(q - q_L) \ge \theta \tag{60}$$

$$\phi(q_i^M) + \phi'(q_i^M)(q - q_i^M) \ge \theta, \quad \forall i \in \{1, \dots, N_c\} \tag{61}$$

When $N_c = 0$, the polyhedral relaxation is defined only by constraints (58)–(60).

The polyhedral relaxations of (26) defined above are illustrated in Figs. 11 and 12. Observe that their strength depends on q_L and q_U. Smaller ranges for $q_U - q_L$ lead to tighter linear relaxations. In conclusion, it is possible to write polyhedral relaxations of constraints (2a) as

Fig. 13 ToyNet



$$R^t q^t + E^t \theta^t \leq r^t, \quad \forall t = 1, \ldots, n_l \tag{62}$$

for suitable matrices R^t and E^t and vectors r^t that depend on q^t_L and q^t_U.

Appendix 2: An illustrative example

In this appendix, we illustrate the graph decomposition routine outlined in Sect. 3.3 to compute the index set $\mathscr{C}$ corresponding to links considered by the domain reduction procedure in Algorithm 1.

Consider the network ToyNet, whose layout is presented in Fig. 13. This network includes one fixed-head node H_0, and six unknown-head nodes. Moreover, node V_6 is the only node in the network with degree one. In this case, there is a unique tree, which is composed by links P_6 and P_7, and nodes V_3, V_5, and V_6. Node V_3 is the root of the tree. Following the notation introduced in Sect. 2, the operation of ToyNet is considered under n_l different demand conditions.

Let $t \in \{1, \ldots, n_l\}$. Equation (2c) at V_6 implies that $q^t_{P_7} = d^t_{V_6}$, where $d^t_{V_6}$ denotes the demand at node V_6 and time t. Therefore, flow across link P_7 is known a priori, and upper and lower bounds on flow variables corresponding to link P_7 are equal. Furthermore, Eq. (2c) imply that $q^t_{P_6} = d_{V_5} + d_{V_6}$. Hence, also the flow across link P_6 is known *a priori*. Furthermore, we observe that links P_2, P_4 and P_5 are connected in series. Equation (2c) yield that

$$q^t_{P_2} = q^t_{P_4} + d^t_{V_2} \tag{63}$$

$$q^t_{P_5} = q^t_{P_4} - d^t_{V_4} \tag{64}$$

As a result, it is possible to select link P_4 as representative and derive bounds on variables q_{P_2} and q_{P_5} using (63). In conclusion, the optimization based bound tightening

Table 2 Results obtained by Algorithm 2 with $N_c = 0$ for PescaraNet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	7684	25.24	26.87	6.46
2	7200	5720	24.44	26.65	9.04
3	7200	4938	24.14	25.55	5.84
4	7200	4200	23.96	25.07	4.60
5	7200	3756	23.81	24.97	4.85

Table 3 Results obtained by Algorithm 2 with $N_c = 0$ for Net25

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	1933	31.64	33.63	6.28
2	7200	1515	30.874	32.68	5.87
3	7200	1194	30.76	32.20	4.69
4	7200	1060	30.64	32.40	5.77
5	7200	1165	30.57	32.41	6.01

Table 4 Results obtained by Algorithm 2 with $N_c = 1$ for PescaraNet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	7232	25.32	26.87	6.11
2	7200	4368	24.62	26.32	6.90
3	7200	3117	24.32	25.30	4.01
4	7200	1824	24.12	25.07	3.92
5	7200	1461	23.96	24.85	3.70

Table 5 Results obtained by Algorithm 2 with $N_c = 1$ for Net25

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	2032	31.65	33.63	6.25
2	7200	1525	30.87	32.67	5.86
3	7200	1189	30.76	32.20	4.67
4	7200	1007	30.64	32.40	5.76
5	7200	1356	30.57	32.41	6.02

Table 6 Results obtained by Algorithm 2 with $N_c = 3$ for PescaraNet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	7894	25.38	26.87	5.89
2	7200	4746	24.72	26.65	7.83
3	7200	3098	24.39	25.30	3.71
4	7200	1726	24.16	25.07	3.76
5	7200	1309	24.00	24.89	3.68

Table 7 Results obtained by Algorithm 2 with $N_c = 3$ for Net25

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	1583	31.65	33.63	6.25
2	7200	1098	30.87	32.67	5.86
3	7200	918	30.76	32.20	4.66
4	7200	796	30.65	32.40	5.75
5	7200	1165	30.57	32.41	6.01

Table 8 Results obtained by Algorithm 2 with $N_c = 5$ for PescaraNet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	7055	25.38	26.87	5.86
2	7200	4274	24.72	26.64	7.78
3	7200	2762	24.41	25.30	3.64
4	7200	1551	24.17	25.07	3.73
5	7200	1194	24.01	24.89	3.64

Table 9 Results obtained by Algorithm 2 with $N_c = 5$ for Net25

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	1449	31.64	33.63	6.23
2	7200	1024	30.87	32.67	5.86
3	7200	852	30.76	32.20	4.67
4	7200	752	30.63	32.68	6.68
5	7200	905	30.57	32.41	6.01

Table 10 Results obtained by Algorithms 1 and 2 with $N_c = 0$ for PescaraNet

n_v	Time (s)	Time in Alg. 1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	16	6923	26.34	26.87	2.02
2	7200	13	4417	25.10	26.65	6.16
3	7200	13	3832	24.43	25.30	3.50
4	7200	13	2119	24.08	25.09	4.19
5	7200	12	1424	23.86	24.81	3.94

Table 11 Results obtained by Algorithms 1 and 2 with $N_c = 0$ for Net25

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	47	3088	32.46	33.63	3.61
2	7200	47	1750	31.27	32.67	4.51
3	7200	44	914	31.02	32.16	3.69
4	7200	44	662	30.92	31.75	2.72
5	7200	43	374	30.84	31.47	2.05

Table 12 Results obtained by Algorithms 1 and 2 with $N_c = 1$ for PescaraNet

n_v	Time (s)	Time in Alg. 1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	14	8486	26.49	26.87	1.42
2	7200	15	4650	25.61	26.06	1.78
3	7200	16	3886	24.82	25.30	1.94
4	7200	15	1531	24.47	25.06	2.41
5	7200	15	1405	23.24	24.89	2.67

Table 13 Results obtained by Algorithms 1 and 2 with $N_c = 1$ for Net25

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	54	2591	32.68	33.63	2.92
2	7200	52	1570	31.45	32.67	3.89
3	7200	52	786	31.16	32.16	3.21
4	7200	51	534	31.03	31.75	2.32
5	7200	50	275	30.96	31.47	1.66

Table 14 Results obtained by Algorithms 1 and 2 with $N_c = 3$ for PescaraNet

n_v	Time (s)	Time in Alg. 1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	18	8218	26.57	26.87	1.12
2	7200	20	4650	25.69	26.06	1.45
3	7200	20	3886	24.85	25.30	1.80
4	7200	19	1531	24.55	25.06	2.04
5	7200	19	1405	23.34	24.85	2.08

Table 15 Results obtained by Algorithms 1 and 2 with $N_c = 3$ for Net25

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	66	2218	32.69	33.63	2.88
2	7200	65	846	31.47	32.67	3.86
3	7200	63	547	31.17	32.16	3.20
4	7200	61	386	31.04	31.75	2.30
5	7200	59	200	30.95	31.47	1.70

Table 16 Results obtained by Algorithms 1 and 2 with $N_c = 5$ for PescaraNet

n_v	Time (s)	Time in Alg. 1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	22	8295	26.59	26.87	1.07
2	7200	24	4329	25.70	26.06	1.43
3	7200	24	1541	24.87	25.30	1.71
4	7200	24	2700	24.61	25.06	1.82
5	7200	23	1201	24.35	24.85	2.03

Table 17 Results obtained by Algorithms 1 and 2 with $N_c = 5$ for Net25

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	7200	80	937	32.68	33.63	2.92
2	7200	77	709	31.46	32.67	3.89
3	7200	74	465	31.17	32.16	3.19
4	7200	73	337	31.04	31.75	2.30
5	7200	71	188	30.95	31.47	1.67

Table 18 Results obtained by BARON (v18.8.23) for PescaraNet

n_v	Time (s)	LB (m)	UB (m)	Gap (%)
1	7200	23.92	–	–
2	7200	23.32	–	–
3	7200	22.84	–	–
4	7200	22.47	–	–
5	7200	22.21	–	–

Table 19 Results obtained by BARON (v18.8.23) for Net25

n_v	Time (s)	LB (m)	UB (m)	Gap (%)
1	7200	31.22	–	–
2	7200	30.37	32.67	7.58
3	7200	30.28	–	–
4	7200	30.25	–	–
5	7200	30.15	–	–

Table 20 Results obtained by SCIP (v3.2.1) for PescaraNet

n_v	Time (s)	LB (m)	UB (m)	Gap (%)
1	7200	25.95	26.87	3.54
2	7200	24.50	26.38	7.67
3	7200	23.39	25.61	9.46
4	7200	23.14	25.30	9.34
5	7200	22.62	25.16	11.23

Table 21 Results obtained by SCIP (v3.2.1) for Net25

n_v	Time (s)	LB (m)	UB (m)	Gap (%)
1	7200	31.50	33.63	6.75
2	7200	31.37	33.36	6.34
3	7200	30.39	32.39	6.56
4	7200	30.47	32.22	5.72
5	33	–	–	–

Table 22 Results obtained by Algorithm 2 with $N_c = 0$ for BWFLnet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	86,400	580	35.36	47.41	34.09
2	86,400	36	29.98	39.31	31.14
3	86,400	1	27.14	36.23	33.49
4	> 86,400	–	–	–	–
5	> 86,400	–	–	–	–

Table 23 Results obtained by Algorithm 2 with $N_c = 3$ for BWFLnet

n_v	Time (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	86,400	517	35.42	47.41	33.87
2	86,400	25	30.03	39.31	30.89
3	> 86,400	–	–	–	–
4	> 86,400	–	–	–	–
5	> 86,400	–	–	–	–

Table 24 Results obtained by Algorithms 1 and 2 with $N_c = 0$ for BWFLnet

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	86400	3707	783	42.48	47.41	11.61
2	86400	3293	29	35.54	39.31	10.62
3	86400	2885	1	32.44	36.20	11.58
4	> 86400	–	–	–	–	–
5	> 86400	–	–	–	–	–

Table 25 Results obtained by Algorithms 1 and 2 with $N_c = 3$ for BWFLnet

n_v	Time (s)	Time in Alg.1 (s)	No. Iter.	LB (m)	UB (m)	Gap (%)
1	86400	4304	596	42.52	47.41	11.50
2	86400	3720	20	35.61	39.31	10.41
3	> 86400	3360	–	–	–	–
4	> 86400	–	–	–	–	–
5	> 86400	–	–	–	–	–

Table 26 Lower and upper bounds at the root of the branch and bound tree obtained by Algorithm 2 with $N_c = 0$ for BWFLnet, with fixed valve locations, considering 96 different demand conditions. Algorithm 1 was applied as pre-solving routine

n_v	No.Iter	LB (m)	UB (m)	Gap (%)
1	1	47.05	47.79	1.60
2	1	39.21	39.36	0.40
3	1	36.06	36.21	0.40

described in Algorithm 1 is performed only for flow variables corresponding to links in $\mathscr{C} := \{P_1, P_4, P_3\}$.

Appendix 3: Supplementary material: tables with numerical results

See Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 and 26.

References

Abraham E, Blokker M, Stoianov I (2018) Decreasing the discoloration risk of drinking water distribution systems through optimized topological changes and optimal flow velocity control. J Water Resour Plan Manag 144(2):04017093. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000878

Ali ME (2015) Knowledge-based optimization model for control valve locations in water distribution networks. J Water Resour Plan Manag 141(1):04014048. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000438

Araujo LS, Ramos H, Coelho ST (2006) Pressure control for leakage minimisation in water distribution systems management. Water Resour Manag 20(1):133–149. https://doi.org/10.1007/s11269-006-4635-3

Belotti P (2013) Bound reduction using pairs of linear inequalities. J Glob Optim 56(3):787–819. https://doi.org/10.1007/s10898-012-9848-9

Belotti P, Lee J, Liberti L, Margot F, Wächter A (2009) Branching and bounds tightening techniques for non-convex MINLP. Optim Methods Softw 24(4–5):597–634. https://doi.org/10.1080/10556780903087124

Bonami P, Biegler LT, Conn AR, Cornuéjols G, Grossmann IE, Laird CD, Lee J, Lodi A, Margot F, Sawaya N, Wächter A (2008) An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optim 5(2):186–204. https://doi.org/10.1016/j.disopt.2006.10.011

Bragalli C, D'Ambrosio C, Lee J, Lodi A, Toth P (2012) On the optimal design of water distribution networks: a practical MINLP approach. Optim Eng 13(2):219–246. https://doi.org/10.1007/s11081-011-9141-7

Currie J, Wilson DI (2012) OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. In: Foundations of computer-aided process operations. https://www.inverseproblem.co.nz/OPTI/. Accessed 27 Nov 2018

Dai PD, Li P (2014) Optimal localization of pressure reducing valves in water distribution systems by a reformulation approach. Water Resour Manag 28(10):3057–3074. https://doi.org/10.1007/s11269-014-0655-6

De Paola F, Galdiero E, Giugni M (2017) Location and setting of valves in water distribution networks using a harmony search approach. J Water Resour Plan Manag 143(6):04017015. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000760

Deuerlein J (2008) Decomposition model of a general water supply network graph. J Hydraul Eng 134(6):822–832. https://doi.org/10.1061/(ASCE)0733-9429(2008)134:6(822)

Diamond S, Takapoui R, Boyd S (2018) A general system for heuristic minimization of convex functions over non-convex sets. Optim Methods Softw 33(1):165–193. https://doi.org/10.1080/10556788.2017.1304548

DAmbrosio C, Lodi A, Wiese S, Bragalli C, (2015) Mathematical programming techniques in water network optimization. Eur J Oper Res 243(3):774–788. https://doi.org/10.1016/j.ejor.2014.12.039

Eck BJ, Mevissen M (2012) Non-linear optimization with quadratic pipe friction. Technical report RC25307. IBM Research Division

Eck BJ, Mevissen M (2013) Fast non-linear optimization for design problems on water networks. In: World environmental and water resources congress 2013. American Society of Civil Engineers, Reston, VA, pp 696–705

Elhay S, Simpson AR, Deuerlein J, Alexander B, Schilders W (2014) A reformulated co-tree flows method competitive with the global Gradient Algorithm for solving the water distribution system equations. J Water Resour Plan Manag 140(12):04014040. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000431

Gamrath G, Fischer T, Gally T, Gleixner A, Hendel G, Koch T, Maher SJ, Miltenberger M, Muller B, Pfetsch ME, Puchert C, Rehfeldt D, Schenker S, Schwarz R, Serrano F, Shinano Y, Vigerske S, Weninger D, Winkler M, Witt JT, Witzig J (2016) The SCIP optimization suite 3.2. Technical report 15-60, Zuse Institute, Berlin

Gleixner A, Held H, Huang W, Vigerske S (2012) Towards globally optimal operation of water supply networks. Numer Algebra Control Optim 2(4):695–711. https://doi.org/10.3934/naco.2012.2.695

Gurobi Optimization (2017) Gurobi optimizer reference manual. https://www.gurobi.com/documentation/7.5/refman.pdf. Accessed 27 Nov 2018

Hindi KS, Hamam YM (1991) Locating pressure control elements for leakage minimization in water supply networks: an optimization model. Eng Optim 17(4):281–291. https://doi.org/10.1080/03052159108941076

Humpola J, Fügenschuh A (2015) Convex reformulations for solving a nonlinear network design problem. Comput Optim Appl 62(3):717–759. https://doi.org/10.1007/s10589-015-9756-2

Humpola J, Fügenschuh A, Koch T (2016) Valid inequalities for the topology optimization problem in gas network design. OR Spectr 38(3):597–631. https://doi.org/10.1007/s00291-015-0390-2

Jowitt PW, Xu C (1990) Optimal valve control in water distribution networks. J Water Resour Plan Manag 116(4):455–472. https://doi.org/10.1061/(ASCE)0733-9496(1990)116:4(455)

Kolodziej S, Castro PM, Grossmann IE (2013) Global optimization of bilinear programs with a multiparametric disaggregation technique. J Glob Optim 57(4):1039–1063. https://doi.org/10.1007/s10898-012-0022-1

Klnç MR, Sahinidis NV (2018) Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. Optim Methods Softw 33(3):540–562. https://doi.org/10.1080/10556788.2017.1350178

Lambert A (2000) What do we know about pressure: leakage relationships in distribution. systems? In: IWA conference system approach to leakage control and water distribution systems management. IWA

Lambert A, Thornton J (2011) The relationships between pressure and bursts a state-of-the-art update. Water21 21:37–38. IWA

Liberatore S, Sechi GM (2009) Location and calibration of valves in water distribution networks using a scatter-search meta-heuristic approach. Water Resour Manag 23(8):1479–1495. https://doi.org/10.1007/s11269-008-9337-6

Liberti L, Pantelides CC (2003) Convex envelopes of monomials of odd degree. J Glob Optim 25(2):157–168. https://doi.org/10.1023/A:1021924706467

Menke R, Abraham E, Parpas P, Stoianov I (2015) Exploring optimal pump scheduling in water distribution networks with branch and bound. Water Resour Manag 30(14):5333–5349. https://doi.org/10.1007/s11269-016-1490-8

Misener R, Floudas CA (2013) GloMIQO: global mixed-integer quadratic optimizer. J Glob Optim 57(1):3–50. https://doi.org/10.1007/s10898-012-9874-7

Misener R, Floudas CA (2014) ANTIGONE: Algorithms for coNTinuous/Integer Global Optimization of Nonlinear Equations. J Glob Optim 59(2–3):503–526. https://doi.org/10.1007/s10898-014-0166-2

Nicolini M, Zovatto L (2009) Optimal location and control of pressure reducing valves in water networks. J Water Resour Plan Manag 135(3):178–187. https://doi.org/10.1061/(ASCE)0733-9496(2009)135:3(178)

Pecci F, Abraham E, Stoianov I (2017a) Outer approximation methods for the solution of co-design optimisation problems in water distribution networks. IFAC-PapersOnLine 50(1):5373–5379. https://doi.org/10.1016/j.ifacol.2017.08.1069

Pecci F, Abraham E, Stoianov I (2017b) Penalty and relaxation methods for the optimal placement and operation of control valves in water supply networks. Comput Optim Appl 67(1):201–223. https://doi.org/10.1007/s10589-016-9888-z

Pecci F, Abraham E, Stoianov I (2017c) Quadratic head loss approximations for optimisation of problems in water supply networks. J Hydroinform 19(4):493–506. https://doi.org/10.2166/hydro.2017.080

Pfetsch ME, Fügenschuh A, Geißler B, Geißler N, Gollmer R, Hiller B, Humpola J, Koch T, Lehmann T, Martin A, Morsi A, Rövekamp J, Schewe L, Schmidt M, Schultz R, Schwarz R, Schweiger J, Stangl C, Steinbach MC, Vigerske S, Willert BM (2015) Validation of nominations in gas network optimization: models, methods, and solutions. Optim Methods Softw 30(1):15–53. https://doi.org/10.1080/10556788.2014.888426

Puranik Y, Sahinidis NV (2017) Domain reduction techniques for global NLP and MINLP optimization. Constraints 22(3):338–376. https://doi.org/10.1007/s10601-016-9267-5

Raghunathan AU (2013) Global optimization of nonlinear network design. SIAM J Optim 23(1):268–295. https://doi.org/10.1137/110827387

Reis LFR, Porto RM, Chaudhry FH (1997) Optimal location of control valves in pipe networks by genetic algorithm. J Water Resour Plan Manag 123(6):317–326. https://doi.org/10.1061/(ASCE)0733-9496(1997)123:6(317)

Sahinidis N (2018) BARON user manual v:18.8.23

Sherali HD, Subramanian S, Loganathan G (1999) Effective relaxation and partitioning schemes for solving water distribution network design problems to global optimality. J Glob Optim 19(1):1–26. https://doi.org/10.1023/A:1008368330827

Smith EMB, Pantelides CC (1999) A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Comput Chem Eng 23(4–5):457–478. https://doi.org/10.1016/S0098-1354(98)00286-5

Tawarmalani M, Sahinidis NV (2002) Convexification and global optimization in continuous and mixed-integer nonlinear programming, 1st, edn. edn. Springer, New York. https://doi.org/10.1007/978-1-4757-3532-1

Udell M, Boyd S (2015) Maximizing a sum of sigmoids. http://web.stanford.edu/~boyd/papers/pdf/max_sum_sigmoids.pdf. Accessed 27 Nov 2018

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

Waechter A, Biegler LT (2006) On the implementation of a primal–dual interior point filter line search algorithm for large-scale nonlinear programming. Math Program 106(1):25–57. https://doi.org/10.1007/s10107-004-0559-y

Wright R, Stoianov I, Parpas P, Henderson K, King J (2014) Adaptive water distribution networks with dynamically reconfigurable topology. J Hydroinform 16(6):1280–1301. https://doi.org/10.2166/hydro.2014.086

Wright R, Abraham E, Parpas P, Stoianov I (2015) Control of water distribution networks with dynamic DMA topology using strictly feasible sequential convex programming. Water Resour Res 51(12):99259941. https://doi.org/10.1002/2015WR017466

Zamora JM, Grossmann IE (1998) A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. Comput Chem Eng 22(3):367–384. https://doi.org/10.1016/S0098-1354(96)00346-8