Pump scheduling in drinking water distribution networks with an LP/NLP-based branch and bound

Gratien Bonvin, Sophie Demassey, Andrea Lodi

June 18, 2019

Abstract

This paper offers a novel approach for computing globally optimal solutions to the pump scheduling problem in drinking water distribution networks. A tight integer linear relaxation of the original non-convex formulation is devised and solved by branch and bound where integer nodes are investigated through non-linear programming to check the satisfaction of the non-convex constraints and compute the actual cost. This generic method can tackle a large variety of networks, e.g. with variable-speed pumps. We also propose to specialize it for a common subclass of networks with several improving techniques, including a new primal heuristic to repair near-feasible integer relaxed solutions. Our approach is numerically assessed on various case studies of the literature and compared with recently reported results.

1 Introduction

To transition to a low-carbon energy system, EU countries have agreed a 40% cut in greenhouse gas emissions in 2030 compared to 1990 levels [9]. It would induce to shift the share of electricity generated from renewable energy sources, primarily by investing in wind and solar power generation capacities [28]. The incorporation of intermittent sources motivates a transition from "a power system in which controllable power stations follow electricity demand" to "an efficient power system overall where flexible producers, flexible consumers and storage systems respond increasingly to the intermittent supply of wind and solar power" [17].

The evolution of the power sector constitutes a significant issue but also an opportunity for drinking water distribution network (DWDN) operators. On the one hand, intermittency is likely to jeopardize electricity peak/off-peak tariffs, on which standard pump control strategies of DWDNs rely: they pump at night to take advantage of the lower cost [18]. Furthermore, a higher electricity average price would increase energy expenditure due to pump operation, which represents around 40% of the life cycle costs of a

pump [41]. On the other hand, the substitution of peak/off-peak tariffs for highly dynamic tariffs might be profitable to DWDNs given their flexible consumption and storage capacities: the pumps can be quickly and automatically operated, while the elevated water tanks act as (potential) energy storages and allow to partly dissociate pump operation and power consumption from water delivery to the end-consumers.

The introduction of dynamic tariffs motivates the use of optimization tools to schedule the pump operation on a daily horizon at a minimum operation cost, given water demand and electricity price forecasts [20, 35]. However, optimizing the day-ahead pump schedule of a DWDN remains a difficult task, because the pressure-related physical laws are non-convex and the pump operation decisions are discrete [12]. A direct resolution of the non-convex Mixed Integer Non Linear Programming (MINLP) formulation with a state-of-the-art solver, through spatial branch and bound and a systematic relaxation, is then not yet an option for most DWDNs of practical size [36, 4].

Following the traditional way to handle non-convex MINLP, a large share of the literature proposed to approximate the non-convex constraints with piecewise linear functions leading to solve a Mixed Integer Linear Programming (MILP) approximation of the problem, but the approach has two major drawbacks. First, the approximated solution is infeasible for the original problem if the approximation is not tight enough. Second, a tight approximation may require to introduce a large number of linearization points, and the associated binary variables make the MILP model hard to solve when the size of the DWDN grows.

Paper contribution. In this paper, we first introduce a tailored tractable relaxation of the non-convex constraints instead of systematic relaxations or piecewise-linear approximations: we compute a tight convex linear Outer Approximation (OA) of the non-convex equality constraints relaxed to inequalities. This two-step approximation leads to a MILP relaxation (vs. approximation) of reasonable size and that is consistent with the objective. Indeed, as observed in [4], minimizing the power consumption tends to satisfy the non-convex constraints at equality even if relaxed to inequalities.

Second, to solve the original non-convex MINLP, we present a variant of LP/NLP branch and bound for convex MINLPs [43], where no OA cuts are generated except the ones defining the MILP relaxation at the root node. The MILP relaxation is solved with a standard LP branch and bound. Each time an integer solution is found in the process, a non-convex NLP solver checks the satisfaction of the non-convex constraints and, if feasible, returns the actual cost of the solution. This results in an exact solution method: a spatial branch and bound – using two relaxations in a single tree – which is readily implemented by embedding an NLP solver, as lazy cuts, in a MILP

solver. (The relation of the proposed algorithm with respect to a few others in the literature is discussed in Section 4.)

The third originality of our approach, regarding the literature of pump scheduling, is to cover a broad variety of DWDNs with little or no restriction on the topology of the network (branched or with loops, directed or not), on the distribution of the elements (single or multiple sources, demand or tanks at intermediate nodes or at leaves), and on the nature of the elements (types of pumps and valves, models of the physical laws).

DWDNs considered in the literature often fall in the category of *DWDNs with binary settings* (BS), i.e. DWDNs where the operation mode of the active elements (valves and pumps) is binary (on/off), as opposed to *DWDNs with mixed settings* (MS), i.e. DWDNs that contain at least one variable-speed pump or pressure-reducing valve. We propose then to specialize our method by exploiting a feature of class BS: the fixed NLP subproblem reverts to a feasibility problem and we can generate effective combinatorial cuts for the MILP relaxation at infeasible nodes.

Furthermore, from the near-feasible solutions found at these nodes, we apply a primal heuristic which slightly adjusts the time step lengths to fix the tank level limit violations. Note that this new heuristic has a broader scope of application, as it could also be used to derive feasible schedules from approximated solutions of piecewise linear models.

Finally, we experimented our approach on various benchmark sets (*Simple FSD/VSD* [34], *AT(M)* [45, 10], *Poormond* [20, 38, 50] and *DWG* [56]), and drove an empirical comparison with recently reported results [10, 20, 38, 50] and with the reference global optimization solver BARON [47]. The computational results demonstrate the applicability of our generic solution method and also its efficiency regarding the results of the dedicated algorithms on given instances, although the resolution of the non-convex NLP subproblems remains a bottleneck for the largest instances of class MS.

Paper structure. The paper is structured as follows: Section 2 surveys the relevant literature on pump scheduling in DWDNs. A generic MINLP formulation of the problem is defined in Section 3. Section 4 describes our adapted LP/NLP branch and bound for non-convex MINLPs and specialization techniques to the class BS or MS of the network. Section 5 provides the MILP outer approximation. Section 6 describes the time-step adjustment heuristic. Section 7 presents the experimental results, comparisons and analysis.

2 Literature Review

An extensive literature has been devoted to the pump scheduling problem in DWDNs for almost half a century. In earliest contributions, the complex

hydraulic network model is often simplified by the use of mass-balance or regression models, where the pressure aspects are either fully neglected or approximated by calibrated curves [40]. For example, Ormsbee et al. [39] consider DWDNs with one single tank and multiple pump stations: they estimate, with regression curves, the minimum energy requirement associated with a specific tank water level transition and required pump flow, which they embed in a dynamic programming model of the tank water level trajectory.

As stated in [32], "deterministic methods started being supplemented by metaheuristics during the mid 1990s", in particular genetic algorithms [31] sometimes coupled with local search techniques [55], but also ant colony optimization [30] or simulated annealing [33]. Besides their non-exact nature, they appear to be not less expensive: for example in [21], a hybrid method based on linear programming shows a strong reduction in computing time with respect to a genetic algorithm.

Due to a large improvement of the dedicated methods and solvers [11], Mixed Integer Non-Linear Programming approaches have recently grown in popularity in the field of water network optimization [12], in particular to solve the static design problem of gravity-fed DWDNs. However, solving the non-convex MINLP formulation of the dynamic pump scheduling problem with any off-the-shelf global optimizer does still not scale up when the number of time steps or the network size increase [36]. In [23] for instance, the spatial branch and bound of SCIP is directly applied to two large case studies in the MS class but only the static variant of the problem, i.e. on one time step [12], is considered. Alternative methods based on mathematical programming relaxations or approximations have thus been investigated.

To tackle the large DWDN of Berlin, Burgschweiger et al. [7] rely on network reduction strategies, and on smoothing the valve and pump operation, in particular by aggregating the dozen of pumps installed in parallel at each station. The hierarchical approach solves the resulting continuous non-convex NLP, then determines the individual pump schedules to provide locally optimal solutions in less than 30 minutes. This continuous relaxation is however not suitable to a majority of rural DWDNs having only one or two pairs of pumps at each station.

In a significant share of the literature [19, 37, 56, 14, 36, 34], the nonlinear flow-coupling constraints are approximated by piecewise linear functions. While this technique outperforms a direct resolution through MINLP solvers [36], it is often limited to small DWDNs since computing a feasible schedule may require a fine-grained approximation resulting in a large MILP, especially in networks with multiple loops [35]. Furthermore, the MILP optimal solutions are not certified to correspond to feasible schedules (see also [5] in the context of the optimal design problem).

Ghaddar et al. [20] present a Lagrangian decomposition by dualizing the time-coupling constraints to separate the scheduling problem in independent

one-time step MINLPs. It provides valid lower bounds and the Lagrangian solutions are converted to feasible solutions by a simulation-based limited discrepancy search. Naoum-Sawaya et al. [38] obtained better solutions on the *Poormond* network (class BS) with another hybrid approach. They apply a Benders decomposition with combinatorial cuts to separate the decision on the binary operation variables, in the master problem, from the simulation of the hydraulic constraints. The solution space is only explored locally to speed up the search, thus the solutions have no performance guarantee. Costa et al. [10] also employ a simulation-optimization framework but which explicitly evaluates all the possible combinations of the binary operation variables. This full enumeration scheme was successfully applied to a small DWDN of class BS but it probably does not scale well. Shi and You [50] consider a similar decomposition but they develop an exact method and use a tight master MILP where the hydraulic constraints are only partially relaxed: non-convex outer approximations defined by piecewise-linear segments are automatically generated and refined during the search. Contrarily to the previous works, this approach directly applies to DWDNs of class MS, although only experiments on two small DWDNs of class BS with a limited number of time steps are presented. This exact method outperformed a direct resolution with SCIP on these cases, but the proposed piecewise-linear relaxation may require, like piecewise-linear approximations, a large number of auxiliary binary variables to model the linear segments. Bonvin et al. [4] exploited a specific property of a class of branched DWDNs to derive a tight convex relaxation with the same size of the original non-convex MINLP: they showed how to relax the head-flow coupling equalities into inequalities and convert the solutions to feasible near-optimal (even optimal if all pumps are identical) schedules. The same relaxation has been used in a heuristic to approximate the operation of fixed-speed pumps [36] and variable-speed pumps [34].

In this paper, we generalize the convex relaxation of [4] to DWDNs with loops, both in classes BS and MS, and devise an exact method based on a similar decomposition to [38] and [50]. The key differences with these approaches are that: (1) our MILP master relaxation is both tight and of limited size, and (2) the solutions are searched and evaluated within one single tree search. Our approach bears similarities with the LP/NLP branch-and-bound framework developed by Quesada and Grossmann [43] for general convex MINLPs but we progressively tighten the MILP relaxation with combinatorial cuts, as in [38], instead of OA cuts. Furthermore, we specialize the method for DWDNs of class BS by exploiting, like in [38, 10], the fact that the NLP subproblem can be turned into a feasibility check with a simple hydraulic simulation. A similar characteristic happens on the optimal DWDN design problem and was exploited by Raghunatan [44] to improve the LP/NLP branch and bound he applies to a convex formulation of this static problem.

A	arcs	J	nodes
$L \subset A$	pipes	$J_T \subset J$	tank nodes
$V \subset A$	valves	$J_S \subset J$	source nodes
$K \subset A$	pumps	$J_J \subset J$	internal nodes
$K_F \subset K$	fixed-speed pumps	$t \in [1, T]$	time periods
$K_V \subset K$	variable-speed pumps		
$x_{at} \in \{0, 1\}$	status of active element $a \in K \cup V$ in period t		
$w_{kt} \in [0, 1]$	speed of pump $k \in K_V$ in period t		
$q_{at} \in \mathbb{R}$	flow through $a \in A$ in period t		
$h_{jt} \geq 0$	hydraulic head at node $j \in J$ in period t		

Table 1: Summary of notation

Finally, as stated in [5], the variety of the modeling assumptions makes difficult to set up a formal comparison with alternative methods of the literature. In contrast with the problem of the optimal design of gravity-fed DWDNs where a benchmark set of 9 instances exists[1], the methods dedicated to the pump scheduling problem are often evaluated on only one or two instance sets which vary from study to study. Menke et al. [36] compared different implementations of mathematical programming approaches on small generated instances and concluded that piecewise-linear approximation was faster than the direct resolution of the non-convex model with SCIP solver. The two hybrid approaches by [20] and [38] were also rigorously compared on instances of the *Poormond* network [38]. For this paper, we built a benchmark set of 75 instances by applying the 5 electricity tariff profiles of [20] to a variety of networks with different characteristics coming from [34, 45, 20, 56]. While it was not an option to reimplement the complex methods of the literature, we propose to drive an empiric comparison with the computational results reported in four recent papers [20, 38, 10, 50].

3 Model formulation

This section describes the standard assumptions we used to model the different physical assets of DWDNs, and provides a non-convex MINLP formulation ($\mathcal{P}$) of the pump scheduling problem.

3.1 Notations and variables

A DWDN is described as a directed graph $\mathcal{G}$=(J,A), where nodes J are divided into tanks J_T, sources J_S and internal nodes J_J, and arcs A are divided into pumps K, pipes L and valves V [12]. The set of pumps K

[1] available at http://www.or.deis.unibo.it/research_pages/ORinstances/ORinstances.htm

is further split into fixed-speed pumps K_F and variable-speed pumps K_V. The scheduling horizon is discretized in T periods of equal length Δ_t (in h), during which we assume a steady state operation [6, 20][2]. As water demands and electricity tariffs often fluctuate on a daily basis, it is typically limited to one day: $T = 24$ and $\Delta_t = 1$.

The pump scheduling problem involves 4 sets of variables: $q_{at} \in \mathbb{R}$ denotes the water flow rate (in m^3/h) through arc a during period t; hydraulic head $h_{jt} \geq 0$ is the sum of the geographical elevation and the water pressure head (in m) at node j at the end of period t; binary variable $x_{at} \in \{0, 1\}$ models the status of an active element $a \in K \cup V$ during period t, e.g. whether a pump is turned on or off, or a gate valve is open or not; finally, for variable-speed pumps $k \in K_V$, continuous variable $0 \leq w_{kt} \leq 1$ gives the normalized speed value during period t.

3.2 Nodes

Internal nodes. Flow conservation at internal node $j \in J_J$ is enforced at any time t by

$$\sum_{ij \in A} q_{ijt} = \sum_{ji \in A} q_{jit} + D_{jt}, \tag{1}$$

with $D_{jt} \geq 0$ the forecasted water demand rate (in m^3/h) for period t. Note that pressure-dependant water leaks could be considered, by adding a term to (1) as in [51], but we neglect them here for simplicity. Water has also to be served with a minimal pressure level $\underline{P_j} \geq 0$ (in m), thus

$$h_{jt} \geq Z_j + \underline{P_j}, \quad \text{if } D_{jt} \neq 0, \tag{2}$$

where Z_j is the elevation (in m) of node j.

Sources. We assume that head level H_{jt} (in m) at source node $j \in J_S$ varies in time t but is exogenous as it is independent of the system operation, thus

$$h_{jt} = H_{jt}. \tag{3}$$

This is a common assumption as sources are often high capacity reservoirs such as lakes, rivers or groundwater aquifers [46]. In addition, one can enforce a daily maximal withdrawal limit $V_j \geq 0$ (in m^3) due to the capacity of raw water pumping stations or to a contractual agreement [7, 56], namely

$$\sum_{t \in T} \sum_{ji \in A} q_{jit} \Delta_t \leq V_j. \tag{4}$$

[2] In [37], transitional regimes are taken into account through the hammer equation but, as pointed out in [12], it is yet unclear whether the dynamic hydraulic behavior needs to be described this accurately in the context of pump scheduling.

Water tanks. Flow conservation at water tank $j \in J_T$ is enforced at any time t by

$$\sum_{ij \in A} q_{ijt} - \sum_{ji \in A} q_{jit} = \frac{S_j}{\Delta_t}(h_{jt} - h_{j(t-1)}). \tag{5}$$

The right-hand side represents the water tank net inflow during t, where $S_j > 0$ denotes the surface (in m^2) of the tank and $h_{jt} - h_{j(t-1)}$ models the variation of the water level. The water level is bounded by $[\underline{H_j}, \overline{H_j}]$ according to the geographic elevation, the capacity and the water volume reserved for emergency purposes. Finally, the water level at the end of the day is usually constrained to be greater than the initial level H_j^0, namely

$$\underline{H_j} \leq h_{jt} \leq \overline{H_j}, \tag{6}$$

$$h_{j0} = H_j^0 \leq h_{jT}. \tag{7}$$

3.3 Arcs

Pipes. Under the steady-state assumption, the Hazen-Williams or Darcy-Weisbach formulae are empirically-close approximations of the head losses due to friction through pipes [6]. Since they are sometimes difficult to handle in an optimisation framework, accurate quadratic approximations have been proposed [16, 42]. Our resolution scheme is compatible with these relations but, to facilitate the comparison with previously proposed methods, we adopt here, for each pipe $l = ij \in L$ and time t, the quadratic relation

$$h_{it} - h_{jt} = \Phi_l(q_{lt}) = A_l q_{lt} + B_l q_{lt}|q_{lt}|, \tag{8}$$

where A_l and B_l are real parameters that can be either extrapolated from experiments or approximated from the cited formulae.

Since Φ_l is not differentiable at 0, state-of-the-art global optimization solvers cannot handle this model. An alternative formulation (see, e.g. [50]) is to specify the flow direction with a binary variable $x_{lt} \in \{0, 1\}$ and to split the flow into its positive $q_{lt}^+ \geq 0$ and negative $q_{lt}^- \geq 0$ parts, as

$$q_{lt} = q_{lt}^+ - q_{lt}^-, \tag{8a}$$

$$q_{lt}^+ \leq \overline{Q_{lt}} x_{lt}, \tag{8b}$$

$$q_{lt}^- \leq |\underline{Q_{lt}}|(1 - x_{lt}), \tag{8c}$$

$$h_{it} - h_{jt} = A_l(q_{lt}^+ - q_{lt}^-) + B_l(q_{lt}^{+2} - q_{lt}^{-2}), \tag{8d}$$

where flow bounds $\underline{Q_{lt}} \leq 0 \leq \overline{Q_{lt}}$ can be computed as proposed in Section 5.1.

Pumps. When sources are elevated, gravity-fed water is supplied to the household connections with sufficient pressure. Otherwise, pumps are required to increase the hydraulic head within the network. Following [7, 37],

when a pump $k = ij \in K$ is active, the head increase between the inlet i and outlet j nodes at time t can be approximated by

$$\Psi_k(q_{kt}, w_{kt}) = w_{kt}^2 \left(\alpha_k - \beta_k \left(\frac{q_{kt}}{w_{kt}} \right)^{\gamma_k} \right), \tag{9^a}$$

where α_k, β_k, γ_k are real parameters derived from the pump manufacturer data, and the flow q_{kt} and speed w_{kt} variables are restricted to some positive intervals $[\underline{Q_k}, \overline{Q_k}]$ and $[\underline{W_k}, \overline{W_k}]$, with $\underline{W_k} = \overline{W_k} = 1$ for a fixed-speed pump. When the pump is inactive, flow and speed are null and the head at the inlet and outlet nodes remain uncoupled [12]. This behavior can be modeled by binary variable $x_{kt} \in \{0, 1\}$, with $x_{kt} = 1$ iff k is active at time t, and the following constraints:

$$\underline{M_k}(1 - x_{kt}) \leq h_{jt} - h_{it} - \Psi_k(q_{kt}, w_{kt}) \leq \overline{M_k}(1 - x_{kt}), \tag{9}$$
$$\underline{Q_k} x_{kt} \leq q_{kt} \leq \overline{Q_k} x_{kt}, \tag{10}$$
$$\underline{W_k} x_{kt} \leq w_{kt} \leq \overline{W_k} x_{kt}, \tag{11}$$

with $\underline{M_k}$ and $\overline{M_k}$ sufficiently large big-M values.

The maintenance cost of a pump can represent around 10% of its overall net present value lifecycle cost [41]. Sound practices can limit this cost, for example by restricting the number N of daily pump switches or fixing the minimum duration τ_1 (resp. τ_0) a pump has to remain on (resp. off) [29]. These constraints are modeled in [20] by using a binary variable y_{kt} (resp. z_{kt}) that is 1 if the pump k is switched on (resp. off) at time t and by

$$\sum_{t=2}^{T} y_{kt} \leq N, \tag{12}$$
$$y_{kt} \geq x_{kt} - x_{k(t-1)}, \tag{13}$$
$$y_{kt} \leq x_{kt'}, \quad \forall t' : t \leq t' < t + \tau_1 \tag{14}$$
$$z_{kt} \geq x_{k(t-1)} - x_{kt}, \tag{15}$$
$$z_{kt} \leq 1 - x_{kt'}, \quad \forall t' : t \leq t' < t + \tau_0. \tag{16}$$

Valves. A large variety of valves with distinct functions exist for DWDNs [46]. We focus on the three types appearing the most frequently in optimization studies: gate valves (GVs), check valves (CVs) and pressure-reducing valves (PRVs). Their purposes are to totally open or close a pipe, to avoid reversed flow, and to enforce a given head loss, respectively. Any of these valves $v = ij \in V$ can be modeled by two constraints at any time t, namely

$$\underline{M_v}(1 - x_{vt}) \leq h_{it} - h_{jt} \leq \overline{M_v}\,(1 - g(x_{vt}))\,, \tag{17}$$
$$\underline{Q_v} g(x_{vt}) \leq q_{vt} \leq \overline{Q_v} x_{vt}, \tag{18}$$

with x_{vt} a binary variable modeling the valve status, $\underline{Q_v}$, $\overline{Q_v}$, $\underline{M_v}$ and $\overline{M_v}$ sufficiently large big-M values and g a Boolean function defined by $g(x_{vt}) = x_{vt}$ if v is a GV or a CV and by $g(x_{vt}) = 1 - x_{vt}$ if v is a PRV.

For a GV, flow is null and head at inlet and outlet are decoupled if the valve is close ($x_{vt} = 0$), and heads and flow are untouched, otherwise. For a CV, by setting $\underline{Q_v} = \overline{M_v} = 0$, either the flow is positive and heads are untouched ($x_{vt} = 1$) or the quantity $h_{it} - h_{jt}$ is negative and the valve is closed ($x_{vt} = 0$) in order to prevent a negative flow. For a PRV, x_{vt} denotes the flow direction and the head drop $h_{it} - h_{jt}$ can be seen as a decision variable, then (17) forces the head to decrease in the flow direction. The alternative formulation of PRVs used in [6],

$$(h_{it} - h_{jt})q_{vt} \geq 0, \tag{17^a}$$

does not require binary variables but is non-convex.

3.4 Optimization task

The common objective of pump scheduling is to supply the forecast water demand with appropriate pressure requirements at minimal operating cost [6]. Following [56], we define a constant cost E_j per unit of raw water for its extraction and treatment at source $j \in J_S$, to which we add the energy costs induced by pumping. The power consumption of a variable-speed pump $k \in K_V$ can be approximated by [7, 37]

$$\Gamma_k^V(q_{kt}, w_{kt}) = w_{kt}^3 \left(\lambda_k + \mu_k \left(\frac{q_{kt}}{w_{kt}}\right)\right), \tag{19}$$

with λ_k and μ_k two parameters computed from experimental points provided by the pump manufacturer. For a fixed-speed pump, (19) is linear, i.e.

$$\Gamma_k^F(q_{kt}, x_{kt}) = \lambda_k x_{kt} - \mu_k q_{kt}, \tag{20}$$

since either $w_{kt} = x_{kt} = 1$ and $q_{kt} \geq \underline{Q_v}$, or $w_{kt} = x_{kt} = q_{kt} = 0$.

Given C_t the electricity unit cost at time t, the objective function to minimize is given by

$$\begin{aligned} \Gamma \;=\; & \sum_{t\in T}\sum_{j\in J_S}\sum_{ji\in A} E_j q_{ijt}\Delta_t + \sum_{t\in T}\sum_{k\in K_V} C_t\Gamma_k^V(q_{kt}, w_{kt})\Delta_t \\ + \; & \sum_{t\in T}\sum_{k\in K_F}\sum_{k\in K_F} C_t\Gamma_k^F(q_{kt}, x_{kt})\Delta_t. \end{aligned} \tag{21}$$

3.5 Summary of the mathematical model

With the help of notations given in Table 1, the Pump Scheduling Problem $(\mathcal{P})$ can be formulated as the non-convex MINLP

$$\min (21) \text{ subject to}$$

$$\text{Flow conservation} \begin{cases} \text{Equ. (1)} & \forall\, j \in J_J, \forall t \\ \text{Equ. (5)} & \forall\, j \in J_T, \forall t \end{cases}$$

$$\text{Head-flow relations} \begin{cases} \text{Equ. (8)} & \forall\, l \in L, \forall t \\ \text{Equ. (9)} & \forall\, k \in K, \forall t \\ \text{Equ. (17)} & \forall\, v \in V, \forall t \end{cases}$$

$$\text{Flow bounds} \begin{cases} \text{Equ. (10)} & \forall\, k \in K, \forall t \\ \text{Equ. (18)} & \forall\, v \in V, \forall t \end{cases}$$

$$\text{Speed bounds} \quad \text{Equ. (11)} \quad \forall k \in K_V, \forall t$$

$$\text{Head bounds} \begin{cases} \text{Equ. (2)} & \forall\, j \in J_J, \forall t \\ \text{Equ. (3)} & \forall\, j \in J_S, \forall t \\ \text{Equ. (6)} & \forall\, j \in J_T, \forall t \\ \text{Equ. (7)} & \forall\, j \in J_T \end{cases}$$

$$\text{Pump activation} \quad \text{Equ. (12)} - \text{(16)} \quad \forall k \in K$$

$$\text{Maximal withdrawal} \quad \text{Equ. (4)} \quad \forall j \in J_S$$

4 A LP/NLP-based branch and bound

Spatial branch and bound [52] is the best-known exact method for solving non-convex MINLP [2]. Its implementation in global optimization solvers is based on generic reformulation and linearization techniques to get valid relaxations for bounding, and thus is applicable to a broad variety of problems including the model $(\mathcal{P})$ defined in Section 3. However, the reformulation introduces an auxiliary variable for each elementary non-linear term and lifts then the model in a larger space: $O(T(|L| + |K_V|))$ new variables in our case. Each non-linear term being relaxed independently, the automatic reformulation may also be too weak to effectively prune the search tree [2]. Furthermore, global optimization solvers miss the advanced search and cut generation and management techniques that are part of modern MILP solvers. This section presents our implementation of a spatial branch and bound for $(\mathcal{P})$, based on the tailored MILP relaxation $(\mathcal{P}_\epsilon)$ described in Section 5, and built on top of the combination of a MILP solver with a non-convex NLP solver.

4.1 Two relaxations, one search tree

Theoretically, we solve $(\mathcal{P})$ within a single-tree branch and bound: we first branch on the binary variables x and use the LP relaxation of the MILP relaxation $(\mathcal{P}_\epsilon)$ for bounding. Once all the binary variables are fixed at a given node $(x = X)$, the search continues to solve the resulting restricted continuous non-convex NLP $(\mathcal{P}(X))$, now in the subtree, by evaluating the lower bounds with a systematic relaxation of $(\mathcal{P}(X))$, and by branching on fractional variables appearing in violated non-convex constraints.

In practice, we embed a non-convex NLP solver in a modern MILP solver, using the so-called *lazy callback* functionality, to automatically drive the search and take advantage of its advanced implementations. Relaxation $(\mathcal{P}_\epsilon)$ is solved by the MILP solver but we specifically manage the incumbent update as follows: In a callback function, at every node where a new incumbent solution (X, W, Q, H) of $(\mathcal{P}_\epsilon)$ is found (either at a leaf node or by a heuristic), we check its feasibility against $(\mathcal{P})$. The function calls the global optimization solver on the restricted non-convex NLP $(\mathcal{P}(X, \bar{z}))$ obtained by fixing the integer variables $x = X$ in $(\mathcal{P})$ and by bounding the optimal solution value by the current MILP incumbent value, say $\bar{z}$. If the restricted problem is feasible, then it returns an actual feasible solution of $(\mathcal{P})$, and we update the incumbent with its actual value, which may differ from the cost of the relaxed solution. Hence, whether the relaxed solution is feasible or not, the node is discarded from the search.

This solution scheme is similar to the LP/NLP branch-and-bound algorithm originally developed by Quesada and Grossmann [43] for convex MINLP optimization problems, with the difference that the MILP relaxation is not refined with OA cuts during the search. Indeed, OA cuts are not necessarily valid in non-convex optimization and the method cannot directly be used in the non-convex case. Instead, we generate a set of alleged active OA cuts, once for all before the search, when constructing the MILP relaxation $(\mathcal{P}_\epsilon)$. Our solution scheme (with actually two different implementations for classes BS and MS) is also in the spirit of the branch-and-check algorithm [53] which, in its original framework, solves the restricted subproblems with constraint programming and generates no-good cuts by linearizing logical conditions. More recently, Dan et al. [13] propose to treat subtrees of a unique MILP relaxation of a class of MINLP with equilibrium constraints as separated search optimization problems and solve them by refining their associated formulation.

Relative to our problem, the static optimal design of gravity-fed DWDNs has been tackled by Raghunathan [44] through a similar scheme with two differences. First, in that application, the objective depends on the binary (pipe size choice) variables only, hence, a relaxed solution is discarded from the search only if infeasible; otherwise, its relaxed cost matches its actual cost and the incumbent is updated as usual. Second, an interesting feature

in [44] is that the restricted subproblem ($\mathcal{P}(X)$) gets reduced to a feasibility problem since, in gravity-fed DWDNs, flows and pressures (the continuous variables) are fully determined once the sizes of the pipes (the binary variables) are decided. In the pump scheduling problem, the same feature may appear depending on the nature of the active elements (pumps and valves) in the considered DWDN. Hereafter, we propose to characterize the subclasses of DWDNs having or not this feature and give implementation details to improve the algorithm in both cases.

4.2 DWDNs with binary settings

This class defines the set of DWDNs that contain as active elements only fixed-speed pumps, check valves (CVs) and gate valves (GVs). Elements of these types operate in one state on or off at each time, then can be modeled with only binary variables x. In this context, after fixing the active element states, flows and pressures through the DWDN are uniquely determined if the two following conditions – which are met for us – hold: (1) Φ_l is monotonically increasing with the flow value for each pipe $l \in L$, according to [54], and (2) $\Psi_k(.,1)$ is strictly decreasing with the flow value for each fixed speed pump $k \in K_F$, according to [48]. Notice also that the unicity of the head-flow configuration is not jeopardized by the presence of check or gate valves, according to [48]. Therefore, the restricted subproblem ($\mathcal{P}(X)$) allows at most one feasible solution when all the binary variables x are fixed to specific values X.

As a consequence, instead of solving $\mathcal{P}(X)$ with an NLP solver, the hydraulic feasibility of a new incumbent relaxed solution X is verified by running an extended period analysis: starting with the time period $t = 1$, we apply the Newton method[3] [54] to quickly compute the unique head-flow configuration consistent with the actual water demands, tank and source heads and active element status. Then we check that it satisfies internal node head and pump flow bounds (2) and (10) and compute the associated operation cost $z_t(X_t)$. Then, tank heads for the following time period $t = 2$ are updated according to flow conservation (5) and checked against their bounds (6). As long as no constraint is violated, the simulation goes successively over all time periods $t \in \{1, \ldots, T\}$. The solution is feasible if the final tank heads at $t = T$ satisfy their bounds (7). In this case, the subproblem returns the total operation cost $z(X) = \sum_{t=1}^{T} z_t(X_t)$. Otherwise, as soon as a constraint is violated, say at time period $t = \bar{t}$, the simulation is halted and the relaxed solution X is registered as infeasible. As in the Benders decomposition of [38], we can then reinforce our MILP relaxation ($\mathcal{P}_\epsilon$) with

[3]It is the core method of the EPANET simulator which is used in other optimization/simulation approaches like [38, 10] but we use our own implementation to match the quadratic model of the pipe head losses.

the combinatorial infeasibility cut

$$\sum_{t=1}^{\bar{t}} \left(\sum_{\substack{a \in K \cup V: \\ X_{at}=0}} x_{at} + \sum_{\substack{a \in K \cup V: \\ X_{at}=1}} (1 - x_{at}) \right) \geq 1. \tag{22}$$

The so-called [1] *canonical cut* (22), mostly known as a *no-good cut* [15], forces here to swap at least one binary decision X_{at} for some active element a and time period $t \in \{1, \ldots, \bar{t}\}$ to prevent the infeasibility at time period $\bar{t}$.

4.3 DWDNs with mixed settings

This class defines the set of DWDNs that contain at least one variable-speed pump (VSP) or one pressure-reducing valve (PRV). The operating modes of these active elements are not discrete and their models require a continuous variable which is not implied by the on/off status: pump speed w_{kt} for VSPs and head drop $h_{it} - h_{jt}$ for PRVs. Applying the branch and bound to this class of problems requires this time a non-convex NLP solver to optimize the restricted subproblem, where all binary variables are fixed.

In our implementation, we use Baron after bringing two adjustments to the NLP restricted model ($\mathcal{P}(X)$): we remove the non-differentiability related with the second derivative of function ϕ at $q = 0$ by replacing constraints (8) by (8^a) - (8^d), and we model PRVs with the non-convex constraints (17^a). The first condition introduces discrete variables in the NLP model, while the latter condition means that, at a given integer node $x = X$, we optimize a less restricted subproblem where the flow directions through the PRVs ($x_{vt}, v \in V_{PR}, t \in \{1, \ldots, T\}$) are unfixed. Once checked, whether a feasible solution is found or not, the following combinatorial infeasibility cut is added to the MILP ($\mathcal{P}$):

$$\sum_{t=1}^{T} \left(\sum_{\substack{a \in K \cup V_C \cup V_G: \\ X_{at}=0}} x_{at} + \sum_{\substack{a \in K \cup V_C \cup V_G: \\ X_{at}=1}} (1 - x_{at}) \right) \geq 1. \tag{23}$$

Solving the resulting restricted non-convex MINLP remains a hard task and the branch and bound can possibly not finish in reasonable time. In this case, we propose to turn the solution process into a heuristic providing also a lower bound and then a certificate of performance of the returned solution. First, the problem can be solved within a given ratio of the optimum $0 < G < 1$. Each restricted subproblem is then solved after setting $\bar{z}(1 - G)$ as an upper bound on the optimal value (where $\bar{z}$ is the current MILP incumbent). Second, to prevent the branch and bound to get stuck in a node X which is difficult to close, we fix a time limit T_s to the NLP subproblem resolution. If available, the best feasible solution found is used to update

the MILP incumbent $\bar{z}$ and we record the final lower bound $L(X)$. At the end of the branch and bound, the global lower bound L is then corrected by $L = \min(L, \min_{X \in S} L(X))$, where S denotes the set of unresolved integer nodes. Note that after correction, the final optimality gap $\frac{z^* - L}{L}$ may be greater than G.

Finally, a local optimization solver can be used instead of the global optimization solver to handle the restricted non-convex NLP subproblems. In our experiments, we tested Bonmin [3] alone which regularly computes good feasible solutions in short computing times. Note that in this case, our resolution scheme keeps providing a global lower bound even if Bonmin does not return value $L(X)$ for the unsolved integer nodes $X \in S$. We use instead the optimum $L_\epsilon(X \setminus X_V)$ (or a lower bound) of the restricted MILP $(\mathcal{P}_\epsilon(X \setminus X_V))$ obtained by fixing all the binary variables to X except for the status of the PRVs. When solving this MILP at optimality was too long, we fixed the optimality gap (we used $G_L = 1\%$ for DWG with $T = 48$) and used the final lower bound.

5 An ϵ-Outer Approximation

In the case of a DWDN with only one-way pipes and fixed-speed pumps, a convex MINLP relaxation of the pump scheduling problem $(\mathcal{P})$ is readily obtained by relaxing the equality in the head-flow coupling constraints (8) and (9^a) to inequality [4]. This relaxation is no longer convex in the general case when considering two-direction pipes or variable-speed pumps. To get a tractable MILP relaxation $(\mathcal{P}_\epsilon)$ of $(\mathcal{P})$ for bounding in the branch and bound described in Section 4, we now describe tight linear outer approximations of the non-convex constraints (8), (9^a), and, for variable-speed pumps, (19). As explained below, the synergy between these relaxations and the objective function motivates our approach.

First, minimizing power consumption tends to prevent extra head losses through the pipes. As depicted in Figure 1, for a given flow value through a pipe $l = ij \in L$, the minimum head loss (in absolute value, in red) admitted by the convex hull relaxation (in the hatched area) satisfies the coupling constraint (8) (on the orange curve Φ_l) except for flow values close to 0 (where the relaxation may under-estimate the head loss). We show in Section 5.1 how to tighten the flow bounds $\underline{Q_l}$ and $\overline{Q_l}$ to reduce this faulty interval $[\overline{Q_l}(1-\sqrt{2}), \underline{Q_l}(1-\sqrt{2})]$. We then provide in Section 5.2 an OA $\widetilde{\Phi_l}$ (the dashed lines) within a maximal distance $\epsilon > 0$ on the rest of the flow domain, i.e such that $|h_i - h_j| \geq |\Phi_l(q_l)| - \epsilon$ for all $(q_l, h_i - h_j) \in \widetilde{\Phi_l}$ and $q_l \in [\underline{Q_l}, \overline{Q_l}(1-\sqrt{2})] \cup [\underline{Q_l}(1-\sqrt{2}), \overline{Q_l}]$.

Second, for a given speed value, a pump $k = ij \in K_V$ is enforced to pump as much water as possible in periods when it is on because of the fixed operating cost $\alpha_k > 0$ in equation (19). Thus, minimizing power

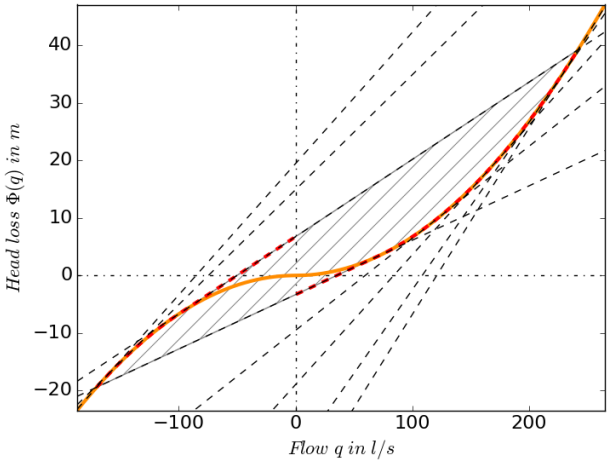


Figure 1: A convex relaxation (hatched area) of head loss in pipes (in orange) on the interval $[\underline{Q}, \overline{Q}]$.

consumption tends to maximize the head increase $h_j - h_i$ through the pumps. As depicted in Figure 2a, we exhibit in Section 5.3 a family of linear over-estimators of the characteristic curve $\Psi_k(q_k, w_k)$ in constraint (9) that is valid for a wide range of pumps. As for the pipes, given a precision value $\epsilon > 0$, a relaxation $\overline{\Psi}_k$ is built by generating a subset of these estimators in such a way that if $(h_j - h_i, q_k, q_k) \in \overline{\Psi}_k$ and $w_k = 1$, then $h_j - h_i \leq \Psi_k(q_k, w_k) + \epsilon$.

Finally, as the objective function may not be linear when variable-speed pumps are present, we exhibit in Section 5.4 a family of linear under-estimators of the power consumption Γ_k (19) depicted in Figure 2b.

5.1 Bound tightening

The quality of an approximation highly depends on the sharpness of the bounds on the variables from which it is built. To tighten the relaxations of the non-convex constraints, the variable domains and the big-M values in the indicator constraints, we estimate, as a preprocessing step, static bounds of the dynamic variables of the problem: flow bounds $\underline{Q_a}$ and $\overline{Q_a}$ for each pipe, pump or valve $a \in A$, speed bounds $\underline{W_k}$, $\overline{W_k}$ for each pump $k \in K_V$, and head increase bounds $\underline{M_a}$ and $\overline{M_a}$ (resp. $\underline{P_a}$ and $\overline{P_a}$) for each inactive (resp. active) pump or valve $a \in A$.

These static bounds can be obtained using optimization-based bound tightening [22] (OBBT): quantities q_a for $a \in A$, w_k for $k \in K$ and $(h_j - h_i)$ for $ij \in K \cup V$ are, successively, minimized and maximized under the

following set of constraints ($\mathcal{C}$):

$$\sum_{ij \in A} q_{ij} = \sum_{ji \in A} q_{ji} + d_j, \qquad \forall j \in J_J \qquad (24)$$

$$h_i - h_j = \Phi_{ij}(q_{ij}), \qquad \forall ij \in L \qquad (25)$$

$$\underline{M_k}(1 - x_k) \leq h_j - h_i - \Psi_k(q_k, w_k) \leq \overline{M_k}(1 - x_k), \quad \forall k \in K \qquad (26)$$

$$\underline{M_v}(1 - x_v) \leq h_i - h_j \leq \overline{M_v}\,(1 - g(x_v))\,, \qquad \forall v \in V \qquad (27)$$

$$\underline{Q_k} x_k \leq q_k \leq \overline{Q_k} x_k, x_k \in \{0, 1\} \qquad \forall k \in K \qquad (28)$$

$$\underline{W_k} x_k \leq w_k \leq \overline{W_k} x_k, \qquad \forall k \in K \qquad (29)$$

$$\underline{Q_v} g(x_v) \leq q_v \leq \overline{Q_v} x_v, \qquad \forall v \in V \qquad (30)$$

$$\min_{1 \leq t \leq T}(D_{jt}) \leq d_j \leq \max_{1 \leq t \leq T}(D_{jt}), \qquad \forall j \in J_J \qquad (31)$$

$$\underline{H_j} \leq h_j \leq \overline{H}_j, \qquad \forall j \in J_T \qquad (32)$$

$$\min_{1 \leq t \leq T}(H_{jt}) \leq h_j \leq \max_{1 \leq t \leq T}(H_{jt}), \qquad \forall j \in J_S. \qquad (33)$$

where the variable bounds $\underline{Q_a}$, $\overline{Q_a}$ for $a \in K \cup V$ and $\underline{W_a}$, $\overline{W_a}$ for $a \in K$ are initialized with the technical information provided by the component manufacturer and the big-M values $\underline{M_a}$, $\overline{M_a}$ for $a \in K \cup V$ are fixed to arbitrary large values. These bounds are valid since for any solution (q, h, x) of $(\mathcal{P})$, the static configuration (q_t, h_t, x_t, d_t) at any time t satisfies these constraints.

For DWDNs of limited size, we opted for solving these non-convex MINLPs directly with BARON after adding binary variables to model the flow direction in the non-smooth constraints (25) (see (8^a)-(8^d)). For larger DWDNs or for DWDNs where the flow direction is unknown in a large share of pipes (see network AT(M) in the experiments), solving the non-convex MINLP may become prohibitive and solving the continuous relaxation may be considered as an alternative. Another approach is to sample discrete values for the continuous variables (pressure decrease for PRVs, pump speed for VSPs, water tank heads), then to apply the Newton method [54] to compute the head-flow equilibrium within the network for each period t and each configuration of binary and continuous values. Finally, a safety margin is applied to the obtained extreme values.

5.2 Outer approximation of the head loss in a pipe

We now devise linear functions to under- and over-approximate the quadratic curve Φ representing the head loss through a pipe (8), given tight bounds $\underline{Q}$ and $\overline{Q}$ on the flow values, as depicted in Figure 1.

Proposition 1 *Given a real function ϕ defined on $\mathbb{R}$ by $\phi(q) = Aq + Bq|q|$ with $A, B \in \mathbb{R}$, $B > 0$, let $f_{q^*}(q) = \phi'(q^*)(q - q^*) + \phi(q^*)$ for $q^* \neq 0$ denote the tangent of ϕ at q^* with, by extension, $f_0 = 0$, and $g_{[q_1,q_2]}(q) =$*

$\frac{\phi(q_2)-\phi(q_1)}{q_2-q_1}(q-q_2)+\phi(q_2)$ *for* $q_1 \neq q_2$ *denote the straight line intersecting* ϕ *at* q_1 *and* q_2*. Then, on any interval* $[\underline{Q}, \overline{Q}]$*, with* $\underline{Q} < \overline{Q}$*:*

1. $\phi \leq f_{q^*}\ \forall q^* \leq \min(\overline{Q}, \overline{Q}(1-\sqrt{2}))$*, if* $\underline{Q} < \overline{Q}(1-\sqrt{2})$*,*
2. $\phi \leq g_{[\underline{Q},\overline{Q}]}$*, otherwise;*
3. $\phi \geq f_{q^*}\ \forall q^* \geq \max(\underline{Q}, \underline{Q}(1-\sqrt{2}))$*, if* $\overline{Q} > \underline{Q}(1-\sqrt{2})$*,*
4. *and* $\phi \geq g_{[\overline{Q},\underline{Q}]}$*, otherwise.*

Proof: We prove the validity of the upper bounds (1,2); the proof for the lower bounds (3,4) is similar given the symmetry $\phi(q) = -\phi(-q)$.

On $\mathbb{R}_-$, ϕ is concave (since $\phi'' = -2B \leq 0$), so its graph lies below its tangents: $\phi(q) \leq f_{q^*}(q)$ for all $q, q^* \leq 0$. On $\mathbb{R}_+$, ϕ is convex (since $\phi'' = 2B \geq 0$), so its graph lies below the line segment between any two points of the graph: $\phi(q) \leq g_{[q_1,q_2]}(q)$ for all $0 \leq q_1 \leq q \leq q_2$. It proves the proposition when $\overline{Q} \leq 0$ in case 1 and when $\underline{Q} \geq 0$ in case 2.

Suppose now that $\underline{Q} < 0 < \overline{Q}$ and note, by direct computation, that ϕ is continuous at 0, $Q^* = \overline{Q}(1-\sqrt{2}) < 0$ and $f'_{Q^*} = \phi'(Q^*) = g'_{[Q^*,\overline{Q}]}$, so $f_{Q^*} = g_{[Q^*,\overline{Q}]}$, i.e. the tangent at Q^* intersects ϕ at $\overline{Q}$.

In case 1, $\underline{Q} < Q^* < 0 < \overline{Q}$, consider f_{q^*} the tangent at any $q^* < Q^*$: by concavity, $f_{q^*} \geq f_{Q^*}$ on $[Q^*, +\infty)$, $f_{q^*} \geq \phi$ and $f_{Q^*} \geq \phi$ on $\mathbb{R}_-$. In particular, $f_{Q^*}(0) \geq \phi(0)$ then, since $f_{Q^*}(\overline{Q}) = \phi(\overline{Q})$, $f_{q^*} \geq f_{Q^*} \geq g_{[0,\overline{Q}]} \geq \phi$ on $[0, \overline{Q}]$.

In case 2, $Q^* \leq \underline{Q} < 0 < \overline{Q}$: $g'_{[\underline{Q},\overline{Q}]} \geq g'_{[Q^*,\overline{Q}]}$ (by direct computation), $g'_{[Q^*,\overline{Q}]} = f'_{Q^*} \geq f'_{\underline{Q}}$ (by concavity), and $g_{[\underline{Q},\overline{Q}]}(\underline{Q}) = \phi(\underline{Q}) = f_{\underline{Q}}(\underline{Q})$ (by definition), then $g_{[\underline{Q},\overline{Q}]} \geq f_{\underline{Q}} \geq \phi$ on $[\underline{Q}, 0]$. In particular, $g_{[\underline{Q},\overline{Q}]}(0) \geq \phi(0)$ and, since $g_{[\underline{Q},\overline{Q}]}(\overline{Q}) = \phi(\overline{Q})$, then $g_{[\underline{Q},\overline{Q}]} \geq g_{[0,\overline{Q}]} \geq \phi$ on $[0, \overline{Q}]$. ■

Building on Proposition 1, we relax the non-convex constraint (8) for each pipe $l = ij \in L$ and time t to

$$h_{it} - h_{jt} \leq \begin{cases} g^l_{[\underline{Q_l},\overline{Q_l}]}(q_{lt}) & \text{if } \underline{Q_l} \geq \overline{Q_l}(1-\sqrt{2}) \\ f^l_{q^*}(q_{lt})\ \forall q^* \in \overline{N^\epsilon_l} & \text{otherwise} \end{cases} \tag{$\overline{8_\epsilon}$}$$

$$h_{it} - h_{jt} \geq \begin{cases} g^l_{[\overline{Q_l},\underline{Q_l}]}(q_{lt}) & \text{if } \overline{Q_l} \leq \underline{Q_l}(1-\sqrt{2}) \\ f^l_{q^*}(q_{lt})\ \forall q^* \in \underline{N^\epsilon_l} & \text{otherwise.} \end{cases} \tag{$\underline{8_\epsilon}$}$$

In our implementation, the sets $\underline{N^\epsilon_l}$ and $\overline{N^\epsilon_l}$ of points, at which OA constraints are generated, are built progressively in such a way that the distance between $\Phi^l(q)$ and the closest approximation $f^l_{q^*}(q)$ never exceeds a fixed precision value $\epsilon > 0$.

5.3 Over approximation of the pump head increase

To approximate the head increase function Ψ_k through an active pump $k = ij \in K$, we consider the points that operate a minimal increase: $\Psi_k(q,w) \geq \underline{P_k}$, where $\underline{P_k} > 0$ is computed from the pump manufacturer information or, with more precision, in our preprocessing step by minimizing $h_j - h_i$ under constraints $(\mathcal{C})$ and the additional constraint $x_k = 1$ for the given pump k.

To simplify the two next propositions, we replace a bivariate function with its graph in the 3-dimensional space, e.g. $(q,w,p) \in \Psi \iff \Psi(q,w) = p$ and denote with subscripts its monovariate restrictions, e.g. $\Psi_w(q) = \Psi(q,w)$ or $\Psi_{q=a}(w) = \Psi(a,w)$. The two propositions are illustrated on Figures 2a and 2b.

Proposition 2 *Given a real bivariate function Ψ defined on $\mathbb{R}_+ \times (0,1]$ by $\Psi(q,w) = w^2(\alpha - \beta\frac{q^\gamma}{w^\gamma})$ with positive parameters α, β, γ and $1 \leq \gamma \leq 3$, and a positive lower bound $P \leq \alpha$, then:*

1. *$\Psi(q,w) \geq P$ only on the domain $\mathcal{D} = \{(q,w) \mid \sqrt{\frac{P}{\alpha}} \leq w \leq 1, 0 \leq q \leq s(w)\}$ where $(s(w), w, P)$ defines the non-empty intersection of Ψ with the plane $p = P$, i.e. $s(w) = \Psi_w^{-1}(P) = w(\frac{\alpha}{\beta} - \frac{P}{\beta w^2})^{\frac{1}{\gamma}}$, $\forall w \in [\sqrt{\frac{P}{\alpha}}, 1]$.*

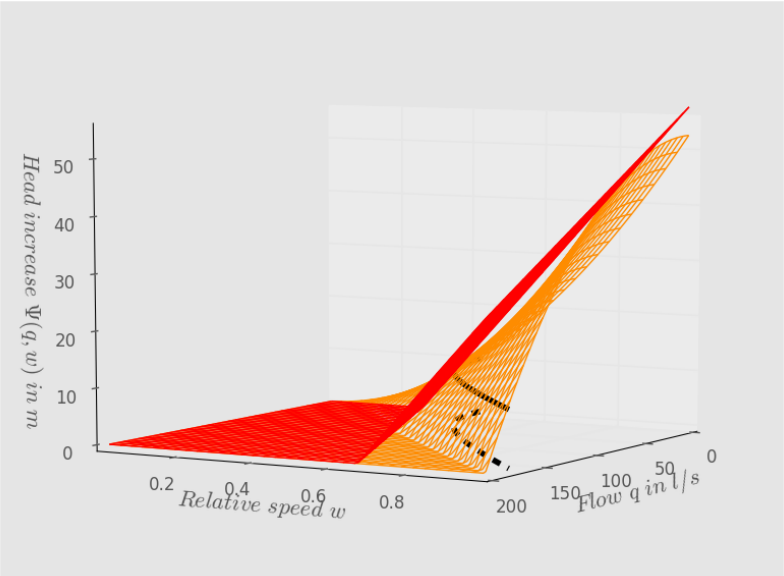
2. *For any $0 < q^* < s(1)$, if the tangent line of Ψ at $(q^*, 1)$ in the plane $w = 1$ intersects the tangent line of Ψ in the plane $p = P$ at some point $(s(w^*), w^*)$, defined by $s(w^*) + s'(w^*)(1 - w^*) = q^* + \frac{P - \Psi_{w=1}(q^*)}{\Psi'_{w=1}(q^*)}$ with $w^* \in [\sqrt{\frac{P}{\alpha}}, 1]$, then Ψ lies below the plane Π^* containing these two lines, i.e. $\Psi(q,w) \leq \Pi^*(q,w) = \Psi_{w=1}(q^*) + \Psi'_{w=1}(q^*)(q - q^*) - \Psi'_{w=1}(q^*)s'(w^*)(w-1)$ for any $(q,w) \in \mathcal{D}$.*

Proof: Since $0 < \frac{P}{\alpha} \leq 1$ and Ψ_w is strictly decreasing, a direct computation proves the first assertion. Assuming that P denotes a lower bound of Ψ, we now restrict our study to the domain of definition $\mathcal{D}$.

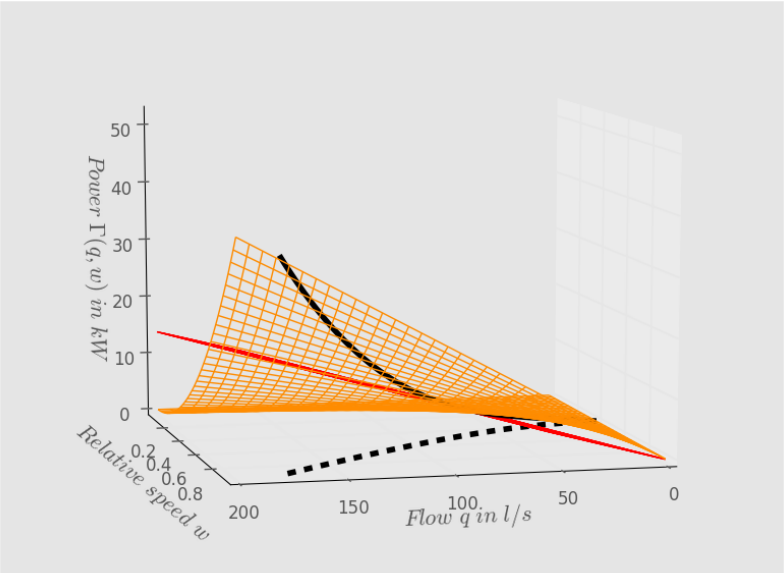
The restriction $\Psi_q(w) = \Psi(q,w)$ to any fixed plane q is convex since, for $(q,w) \in \mathcal{D}$, $\Psi''_q = 2\alpha - \beta\frac{q^\gamma}{w^\gamma}(2-\gamma)(1-\gamma)$, hence $\Psi''_q \geq 0$ if $1 \leq \gamma \leq 2$, and $\Psi''_q \geq 2\alpha - \beta\frac{s(w)^\gamma}{w^\gamma}(2-\gamma)(1-\gamma) \geq 2\alpha - 2(\alpha - \frac{P}{w^2}) \geq 0$ if $2 < \gamma \leq 3$. Since the restriction $\Pi^*_q(w) = \Pi^*(q,w)$ to the plane q is a line, we just need to show that $\Psi_q(w) \leq \Pi^*(q,w)$ at $w = 1$ and at $w = s^{-1}(q)$ in order to show that $\Psi \leq \Pi^*$ on $\mathcal{D}$.

Case $w = 1$: In the plane $w = 1$, $\Psi_{w=1}$ is clearly concave, then it lies below its tangent $\Pi^*_{w=1}$ at $q = q^*$. Thus $\Psi_q(1) = \Psi_{w=1}(q) \leq \Pi^*_{w=1}(q) = \Pi^*(q,1)$.

Case $w = s^{-1}(q)$: Note that $\Pi^*_{w=1}$ is strictly decreasing and $\Pi^*_{w=1}(q^*) = \Psi(q^*,1) > P$, then line $\Pi^*_{w=1}$ intersects the plane $p = P$ at some point $(q', 1, P)$ with $q' > q^*$. Considering their restrictions to the plane $p = P$, Π^* is by definition the tangent line to Ψ (then to the curve defined by



(a) Head increase



(b) Power consumption (nonconvex part)

Figure 2: Illustrations of (a) a linear over-estimator Π^* (in red) of the head increase Ψ (in orange) and (b) a linear under-estimator Π_* (in red) of the non-convex addend Γ (in orange) of the power consumption. The black curve lines depict function $s(w)$, the maximum flow value for a given speed value w, projected on Ψ and Γ: (a) Π^* is tangent to the black curve $\Psi(s(w), w)$ and also to Ψ at some point $(q^*, 1)$ in the plane $w = 1$. (b) Π_* is tangent to the black curve $\Gamma(s(w), w)$ at some w_* and meets Γ at $(0, 1)$.

s) at $(s(w^*), w^*)$ going through $(q', 1)$. Observe by computation that s is non-decreasing and concave (since $s' \geq 0$ and $s'' \leq 0$ as $\gamma \geq 1$), and that the restriction Π^*_q of Π^* to a fixed plane q is non-decreasing (since $\Pi^{*\prime}_q \geq 0$). Hence, for any $(q, w) \in \mathcal{D}$ such that $\Pi^*(q, w) = P$, we have: $q \geq s(w)$ (since s concave), then $w \leq s^{-1}(q)$ (since s non-decreasing), then $\Psi_q(s^{-1}(q)) = P = \Pi^*_q(w) \leq \Pi^*_q(s^{-1}(q)) = \Pi^*(q, s^{-1}(q))$ (since Π^*_q non-decreasing). ■

Note that Proposition 2 only applies when the characteristic pump function Ψ_k is between linear and cubic in the flow ($1 \leq \gamma \leq 3$). This range does not restrict the practicability of the method as the pump head increase can be reasonably represented by a quadratic curve of the flow [8] and that reported values do not depart significantly from $\gamma = 2$ [6]. Proposition 2 also provides a linear relaxation for fixed-speed pumps, although the tighter relaxation $h_{jt} - h_{it} \leq \Psi_k(q_{kt}, 1)$ may sometimes be directly handled by efficient solvers, such as second-order cone solvers when the function is quadratic ($\gamma = 2$) [4].

As for relaxing the head loss in pipes, in our implementation, we progressively generate approximations Π^*_k for a subset Q^ϵ_k of points $0 < q^* < s(1)$ such that the distance between $\Psi_k(q, 1)$ and the closest approximation $\Pi^*_k(q, 1)$ never exceeds a fixed precision value $\epsilon > 0$.

For each pump $k = ij \in K$ and for each time t, constraint (9) is then relaxed to

$$h_{jt} - h_{it} \leq \Psi_{w=1}(q^*)x_{kt} + \Psi'_{w=1}(q^*)(q_{kt} - q^* x_{kt}) \\ - \Psi'_{w=1}(q^*)s'(w^*)(w_{kt} - x_{kt}) + \overline{M_k}(1 - x_{kt}), \forall\ q^* \in Q^\epsilon_k, \quad (9_\epsilon)$$

where $\overline{M_k}$ is computed in the preprocessing step as the maximum head difference $h_j - h_i$ under constraints ($\mathcal{C}$) and the additional constraint $x_{kt} = 0$. Constraints (9_ϵ) are then reduced to $h_{jt} - h_{it} \leq \Pi^*(q_{kt}, w_{kt})$ when $x_{kt} = 1$ and $h_{jt} - h_{it} \leq \overline{M_k}$ when $x_{kt} = 0$.

5.4 Under approximation of the power consumption

The power consumption of a fixed-speed pump is linear in the flow through the pump, but it becomes polynomial in the speed value for a variable-speed pump. Next proposition describes a family of linear under-estimators of the non-convex addend of the power consumption function in this latter case. As in Proposition 2, the study of the function is limited to domain $\mathcal{D}$ on which the pump operates with a minimum pressure increase P. Furthermore, we restrict the proof to the case where $\sqrt{\frac{P}{\alpha}} \geq \frac{1}{3}$, i.e. $\frac{\alpha}{9} \leq P \leq \alpha$, and $\mu > 0$. This reasonable assumption is satisfied by all the instances in our benchmarks although μ may sometimes be negative as in [56].

Proposition 3 *Given a real bivariate function Γ defined on $\mathcal{D}$ (see Proposition 2) by $\Gamma(q, w) = \mu q w^2$ with $\mu > 0$, let γ denote its restriction to the*

surface $q = s(w)$ *(i.e.* $\Psi(q, w) = P$*):* $\gamma(w) = \Gamma(s(w), w)$ *for* $w \in [\sqrt{\frac{P}{\alpha}}, 1]$*. Then,* $\Gamma \geq \Pi_*$ *on* $\mathcal{D}$*, for any* $w_* \in [\sqrt{\frac{P}{\alpha}}, 1]$*, where* Π_* *denotes the plane passing through* $M_0 = (0, 1, 0)$ *and tangent to* γ *at* $M_* = (s(w_*), w_*, \gamma(w_*))$ *and which is formally defined by*

$$\Pi_*(q, w) = \frac{b_*}{c_*}(1 - w) - \frac{a_*}{c_*}q,$$

with $(a_*, b_*, c_*) = u_* \times v_*$ *the cross product of* $u_* = (s(w_*), w_* - 1, \gamma(w_*))$ *the vector pointing from* M_0 *to* M_* *and* $v_* = (s'(w_*), 1, \gamma'(w_*))$ *the tangent vector of* γ *at* M_**.*

Proof: Let π_* denote the intersection of Π_* with the surface $q = s(w)$, i.e. $\pi_*(w) = \Pi_*(s(w), w)$, we first show that $\gamma \geq \pi_*$ on this surface. By construction, π_* is tangent to γ at M_* and, by direct computation, we show that γ is convex ($\gamma'' \geq 0$) and that π_* is concave ($\pi_*'' \leq 0$), then $\gamma(w) \geq \gamma(w_*) + \gamma'(w_*)(w - w_*) = \pi(w_*) + \pi'(w_*)(w - w_*) \geq \pi_*(w)$ for all $w \in [\sqrt{\frac{P}{\alpha}}, 1]$.

Now consider, for any $w_+ \in [\sqrt{\frac{P}{\alpha}}, 1]$, the plane Π_+ containing M_0, $M_+ = (s(w_+), w_+, \gamma(w_+))$ and vector $(0, 0, 1)$. Let D_+ be the straight line of Π_+ passing through M_0 and M_+ and let D_*^+ be the line at the intersection of Π_* and Π_+. By definition, D_*^+ pass through M_0 and $M = (s(w_+), w_+, \pi_*(w_+))$ and, because $\gamma(w_+) \geq \pi_*(w_+)$, then D_*^+ lies below D_+ in plane Π_+, i.e. if $(q, w, p_1) \in D$ and $(q, w, p_2) \in D_+$, then $p_1 \leq p_2$.

Let C denote the intersection of Γ with Π_+, then C intersects D_+ in M_0 and M_+. If $w_+ = 1$, then Π_+ is the plane $w = 1$ and C is the straight line defined by $w = 1$ and $p = \mu q$, so it coincides with D_+. Otherwise, if $w_+ < 1$, then an equation for Π_+ is given by $q = \pi_+(w) = \frac{s(w_+)}{1 - w_+}(1 - w)$. Because Γ is restricted to domain D and $\pi_+(w) \leq s(w)$ implies $w \geq w_+$ and vice-versa, then C is defined by the parametric equation $q = \pi_+(w)$ and $p = \Gamma(\pi_+(w), w)$ for $w \in [w_+, 1]$. Consider, for example, the first-order condition

$$\begin{aligned} &< \nabla\Gamma(\pi_+(w_2), w_2) - \nabla\Gamma(\pi_+(w_1), w_1), (\pi_+(w_2), w_2) - (\pi_+(w_1), w_1) > \\ &= \mu \frac{s(w_+)}{1 - w_+}(w_2 - w_1)^2(2 - 3(w_1 + w_2)) \leq 0, \quad \text{if } w_1 + w_2 \geq \frac{2}{3}. \end{aligned}$$

It shows that C is concave on the segment $(w_+, 1)$ then it lies above D_+, then above D_*^+. This being true for any $w_+ \in [\sqrt{\frac{P}{\alpha}}, 1]$, it proves that Γ lies above Π_*. ■

The power consumption of a variable-speed pump $k \in K_V$ is given by (19) as the sum of a convex function λw^3, which can be approximated from below by its tangent lines ($\lambda w_*^2(3w - 2w_*)$, for any $w^* > 0$), and

of function $\mu q w^2$ studied in Proposition 3. In our implementation, for each variable-speed pump $k \in K_V$, we generate a fixed number n_k of linear underestimators by setting W_k the set of n_k values of w_* evenly distributed in the interval $[\sqrt{\frac{P_k}{\alpha_k}}, 1]$. Then we introduce, for each time t, two new decision variables $y^1_{kt} \geq 0$ and $y^2_{kt} \geq 0$ with the constraints

$$y^1_{kt} \geq \lambda w_*^2(3w_{kt} - 2w_*), \forall w_* \in W_k, \tag{34}$$

$$y^2_{kt} \geq \frac{b_*}{c_*}(1 - w_{kt}) - \frac{a_*}{c_*}q_{kt}, \forall w_* \in W_k. \tag{35}$$

Finally, we relax the objective function (21) to

$$\sum_{t \in T} \Delta_t \Big(\sum_{ji \in A | j \in J_S} E_j q_{ijt} + \sum_{k \in K_V} C_t(y^1_{kt} + y^2_{kt}) + \sum_{k \in K_F} C_t \Gamma^F_k(q_{kt}, x_{kt}) \Big). \tag{36}$$

6 A time-step duration adjustment-based heuristic for class BS

As explained in Section 5, we expect that integer feasible nodes X of the MILP relaxation $(\mathcal{P}_\epsilon)$ are either feasible or close to feasible for $(\mathcal{P})$. In this context, close to feasible means that the restricted NLP subproblem $(\mathcal{P}(X))$ has no solution but there exists a pump schedule that is physically feasible – but possibly not a solution of our model $(\mathcal{P})$ – at a slight distance of X. This is particulary true for instances of class BS where the binary variables x are the unique decision variables so that, contrarily to class MS, small flow imbalances cannot be compensated by a different setting of the continuous (implied) variables. Small imbalances result in low violations of the water tank level limits (6) and (7). These violations could be prevented by allowing to switch the active elements not just at fixed times. This motivates the following primal heuristic to turn close-to-feasible solutions of $(\mathcal{P})$ to feasible pump schedules, by adjusting the duration of the time periods. We describe it in the context of instances of class BS, i.e. without variable-speed pumps or pressure-reducing valves.

6.1 Heuristic formulation

Let X be a feasible instantiation of the binary variables x of $(\mathcal{P}_\epsilon)$ and $A^X_t = \{a \in K \cup V | X_{at} = 1\}$ denote the combination of pumps and valves that are active at time step t. We allow A^X_t to be active earlier or latter (up to 1 time step), and for a shorter or longer duration (from 0 to 3 time steps). We formulate as a mathematical program denoted as $(\mathcal{H}_\alpha(X))$ the problem of finding a solution of minimum power cost in the neighborhood of X thus created. For this, we divide each time step t into three parts of variable

lengths δ_t^1, δ_t^2 and δ_t^3 in $[0, \Delta_t]$, during which, respectively, A_{t-1}^X, A_t^X and A_{t+1}^X are active.

$$z' = \min \sum_{t=1}^{T} \sum_{p=1}^{3} C_t \Gamma_k^F(q_{kt}^p, X_{kt}^p) \delta_t^p \tag{37}$$

$$s.t.\ \delta_t^1 + \delta_t^2 + \delta_t^3 = \Delta_t, \qquad \forall t \geq 1 \tag{38}$$

$$u_{t-1}^3 + u_t^1 \leq L_t \qquad \forall t \geq 2 \tag{39}$$

$$\delta_{t-1}^3 - \delta_{t'-1}^3 - \delta_t^1 + \delta_{t'}^1 \geq (\tau_s + t - t')\Delta_t, \forall s \in \{0,1\}, (t,t') \in I^s \tag{40}$$

$$0 \leq \delta_t^2 \leq \Delta_t, \qquad \forall t \geq 1 \tag{41}$$

$$0 \leq \delta_t^p \leq u_t^p \Delta_t, u_t^p \in \{0,1\}, \qquad \forall t \geq 1, p \in \{1,3\} \tag{42}$$

$$u_t^p \in \{0,1\}, u_1^1 = 0, u_T^3 = 0, \qquad \forall t \geq 1, p \in \{1,3\} \tag{43}$$

$$\sum_{ij \in A} q_{ijt}^p = \sum_{ji \in A} q_{jit}^p + D_{jt}, \qquad \forall j \in J_j, t \geq 1, p \in \{1,2,3\} \tag{1'}$$

$$h_{jt}^p \geq Z_j + \underline{P_j}, \qquad \forall j \in J_J, t \geq 1 | D_{jt} \neq 0, p \in \{1,2,3\} \tag{2'}$$

$$h_{jt}^p = H_{jt}, \qquad \forall j \in J_R, t \geq 1, p \in \{1,2,3\} \tag{3'}$$

$$\sum_{t \in T} \sum_{p=1}^{3} \sum_{ji \in A} q_{jit}^p \delta_t^p \leq V_j, \qquad \forall j \in J_S \tag{4'}$$

$$\sum_{ij \in A} q_{ijt}^p \delta_t^p = S_j (h_{jt}^p - h_{jt}^{p-1}), \qquad \forall j \in J_T, t \geq 1, p \in \{2,3\} \tag{5'}$$

$$\sum_{ij \in A} q_{ijt}^1 \delta_t^1 = S_j (h_{jt}^1 - h_{j(t-1)}^3), \qquad \forall j \in J_T, t \geq 1 \tag{5"}$$

$$\underline{H_j} + \alpha \overline{H_j} \leq h_{jt}^p \leq (1-\alpha)\overline{H_j}, \qquad \forall j \in J_T, t \geq 1, p \in \{1,2,3\} \tag{6'α}$$

$$h_{j0}^3 = H_j^0 \leq h_{jT}^3 - \alpha \overline{H_j}, \qquad \forall j \in J_T \tag{7'α}$$

$$h_{it}^p - h_{jt}^p = \Phi_{ij}(q_{ijt}^p), \qquad \forall ij \in L, t \geq 1, p \in \{1,2,3\} \tag{8'}$$

$$(h_{jt}^p - h_{it}^p - \Psi_{ij}(q_{kt}^p)) X_{kt}^p = 0, \qquad \forall k \in K, t \geq 1, p \in \{1,2,3\} \tag{9'}$$

$$\underline{Q_k} X_{kt}^p \leq q_{kt}^p \leq \overline{Q_k} X_{kt}^p, \qquad \forall k \in K, t \geq 1, p \in \{1,2,3\} \tag{10'}$$

$$h_{it}^p - h_{jt}^p \geq \underline{M_v}(1 - X_{vt}^p), \qquad \forall v \in V, t \geq 1, p \in \{1,2,3\} \tag{17'}$$

$$h_{it}^p - h_{jt}^p \leq \overline{M_v}(1 - X_{vt}^p), \qquad \forall v \in V, t \geq 1, p \in \{1,2,3\} \tag{17"}$$

$$\underline{Q_v} X_{vt}^p \leq q_{vt}^p \leq \overline{Q_v} X_{vt}^p, \qquad \forall v \in V, t \geq 1, p \in \{1,2,3\} \tag{18'}$$

where X_{at}^p denotes the status of active element a during part p of step t, i.e. $X_{at}^1 = X_{a(t-1)}$, $X_{at}^2 = X_{at}$, and $X_{at}^3 = X_{a(t+1)}$. Constraints (39) - (43) prevent to separate A_t^X in two parts, then the number of pump switches to increase, where $L_t = 0$ if $A_{t-1}^X = A_t^X$ and $L_t = 1$ otherwise. Constraints (40) ensure that the maintenance constraints (13)-(16) are still verified after adjusting the time step length, with $(t,t') \in I^1$ (resp. I^0) if at least one

active element $k \in K$ is on (resp. off) at t and off (resp. on) at t'. Constraints (1')-(18') ensure a feasible flow-head configuration during each time step part p. Parameter α in (6'α) and (7'α) has default value 0, but we set it to a positive value (e.g. $\alpha = 1‰$) to strengthen the water tank head bounds, in order to improve the convergence of the iterative resolution scheme described thereafter. Indeed, we do not solve the non-convex MINLP ($\mathcal{H}_\alpha(X)$) directly but propose to solve a sequence of MILP restrictions ($\mathcal{H}_\alpha(X, Q)$) obtained by fixing the flow variables at different values $q = Q$.

6.2 Solution scheme

Consider the part $p \in \{1, 2, 3\}$ of a time period $t \in \{1, \ldots, T\}$. For a DWDN of class BS, once the status X^p_{at} of each active elements $a \in K \cup V$ is fixed, the only quantities which can affect the head-flow equilibrium are the head levels h^p_{jt} at every water tank $j \in J_T$. Interestingly, if the proposed heuristic can recover a low cost feasible solution, it is unlikely that the tank head trajectories are significantly modified because they provide a lower bound on the objective for the node X under attention. Furthermore, a slight alteration of the water tank heads will cause only a limited modification of the head-flow configuration. These observations motivate the iterative Algorithm 1 where the lengths of the time periods and the flows in arcs are successively refined and possibly converge to a fix point.

Algorithm 1 Time-step duration adjustment-based heuristic for DWDNs of class BS

Input: $(H_{jt})_{j \in J_T, t \geq 1}$, $(X_t)_{t \geq 1}$, $\alpha \in \mathbb{R}_+$
Initialize: $(h^1_{jt}, h^2_{jt}, h^3_{jt}) = (H_{j(t-1)}, H_{jt}, H_{jt})$ $\forall j \in J_T, t \geq 1$
while time limit is not reached **do**
  Step 1 (Input: h / Output: $\bar{q}$): for all $t \geq 1, p \in \{1, 2, 3\}$, fix $\bar{h}^p_{jt} = h^p_{jt}$ $\forall j \in J_T$ and compute the head-flow configuration ($\bar{q}^p_t$, $\bar{h}^p_t$) with the Newton method [54].
  Step 2 (Input: $\bar{q}$ / Output: δ): solve ($\mathcal{H}_\alpha(X, \bar{q})$),
  if infeasible **then**
    break
  else
    Step 3 (Input: δ / Output: (q,h)): compute the head-flow configurations (q^p_t,h^p_t) for all $t \geq 1, p \in \{1, 2, 3\}$ with the extended period analysis (see Section 4.2) given time step durations δ^p_t.
    if (q,h,δ) is feasible for ($\mathcal{H}_0(X)$) **then**
      (q,h,δ) is a feasible solution. Compute z as in (37)
      Output: (X, q, h, δ, z)
    end if
  end if
end while

Instance	Class	$\|L\|$	$\|V\|$	$\|K\|$	$\|J_T\|$	$\|J_S\|$	$\|J_J\|$
Simple FSD	$(\mathcal{A})$	2	0	3	1	1	2
AT(M)	$(\mathcal{A})$	41	0	3	3	1	19
Poormond	$(\mathcal{A})$	44	4	7	5	1	46
Simple VSD	$(\mathcal{B})$	2	0	3	1	1	2
DWG	$(\mathcal{B})$	22	6	5	6	3	24

Table 2: Characteristics of DWDNs instances.

First, water tank heads h_{jt}^p are initialized to their values in the investigated solution (X, Q, H) of the MILP relaxation $(\mathcal{P}_\epsilon)$. Then, an iteration consists of three tasks. In the first step, we compute from the tank head values the head-flow configurations providing temporary values $\bar{q}_t^p$ for flows for all time step parts (t, p). In the second step, we fix the flow values q_t^p in the MINLP $(\mathcal{H}_0(X))$ and solve the restricted MILP $(\mathcal{H}_\alpha(X, \bar{q}))$ to obtain new values δ_t^p. In the third step, we simulate through an extended period analysis (see Section 4.2) the pump schedule corresponding to the new time step lengths. If the schedule is feasible for $(\mathcal{H}_0(X))$, then we compute its cost and possibly update the incumbent. Otherwise, when the schedule violates at least one constraint, next iteration starts with the last computed tank head values.

7 Experimental results

In this section, we report on the computational evaluation of our algorithm. In Section 7.1, we describe the benchmark set, while in Section 7.2 we discuss the computational results. Finally, in Section 7.3 we compare our results with those in the literature.

7.1 Experimental data

Experimental data consist of 5 case studies which cover different aspects that can be encountered in real-world DWDNs. Their characteristics are summarized in Table 2. *Simple FSD* (resp. *Simple VSD*) is a test network drawn from [34] with 1 source, 1 water tank, 2 pipes and 3 identical fixed-speed (resp. variable-speed) pumps operating in parallel. *AT(M)* is a modified version proposed in [45] and further investigated in [10] of the extensively studied hypothetical network Anytown [57][4]. It consists of 1

[4]With respect to [10], we connect the water tanks 165 and 265 with a pipe of zero length to prevent non-physical behaviors induced by the discretization, especially when Δ_t is large. This change is justified by the fact that the head levels in the two tanks are always very close, as shown in Figure 8 of [10]. The alternative used in [10] is to run the extended period analysis with a time step smaller than Δ_t.

source, 3 water tanks, 41 pipes and 3 identical fixed-speed pumps working in parallel. *Poormond* is adapted by [20] from the schematic representation of the Richmond water distribution system owned by Yorkshire Water in the UK [21]. It comprises 1 source, 5 water tanks, 44 pipes, 7 fixed-speed pumps and 4 check valves. Finally, *DWG* is a slightly modified version of a Belgium network operated by the water company De Watergroep [56][5]. It consists of 3 sources, 6 water tanks, 22 pipes, 5 fixed-speed pumps and 6 PRVs. *Simple VSD* and *DWG* are the only DWDNs of class MS because of the presence of 3 variable-speed pumps for the former and 6 PRVs for the latter.

To appreciate how the computational scheme responds to an increase number of time periods, we considered the cases T=12, 24 and 48 by smoothing the electrical tariff and water demand profiles, if need be. We considered five different daily electrical tariffs profiles that correspond to the wholesale prices occurring on the Single Electricity Market (SEM) on the five-day period starting from May 21, 2013 at 7am [49]. In summary, we built a benchmark of 75 instances that we are pleased to share under request.

For each instance, we applied the following solution process. First, the variable bounds are estimated with the procedure described in Section 5.1. Then, MILP relaxation ($\mathcal{P}_\epsilon$) is built – with parameters $\epsilon = 0.01\ m$ and $n_k = 10$ for all $k \in K_V$ – using the Gurobi Python API, and solved with Gurobi v.5.6.3 [24] on one thread of a 2× Xeon E5-2650V4 2.2.GHz, 256 GB RAM. The restricted NLPs ($\mathcal{P}(X)$) are investigated through a Gurobi callback function, which differs whether the DWDN is of class BS or MS. For BS, the extended period analysis is implemented in Python, as well as the primal heuristic which is launched at most one time each 30 seconds with a time limit of 10 seconds. For MS, the non-convex NLP is modeled with Pyomo [25] and solved successively by Bonmin (v.1.8.4) and Baron (v.18.5.8) with a time limit of 300 seconds each. Finally, the overall resolution scheme is stopped once reaching either the optimality gap G or the overall time limit, fixed respectively to 0% and 1 hour for BS, and to 1% and 2 hours for MS.

7.2 Computational results

Table 3 presents the results for DWDNs of class BS (*Simple FSD, AT(M), Poormond*) and MS (*Simple VSD, DWG*). For each instance, defined by a day and a time step number T, *Best* gives the cost of the best solution found within the given optimality gap and time limit, if available, otherwise the lower bound computed as in Section 4.3 (in parenthesis); *Gap* is either the

[5]To use our MINLP formulation, we made three modifications: 1) the minimal pressure level $\overline{P}$ is only required for internal nodes with positive demands, 2) we modeled the complex operation of the water tanks (see Eq.(5)-(9) in [56]) by preceding each water tank with a PRV, 3) we dropped the operating constraints related to raw water pump.

		T=12				T=24				T=48			
	Day	Best	Gap	%CB	1^{st}	Best	Gap	%CB	1^{st}	Best	Gap	%CB	1^{st}
Simple FSD	21	inf	<1s	51%	<1s	155.1	3s	33%	1s	150.9	1285s	1%	2s
	22	inf	<1s	34%	<1s	159.1	2s	29%	<1s	155.7	0.9%	3%	2s
	23	inf	<1s	34%	<1s	172.4	3s	39%	<1s	168.5	0.9%	<1%	4s
	24	inf	<1s	34%	<1s	181.7	6s	55%	1s	176.0	0.2%	<1%	<1s
	25	inf	<1s	34%	<1s	147.8	2s	42%	<1s	145.5	0.6%	<1%	<1s
AT (M)	21	766.3	17s	6%	9s	733.2	1.2%	26%	48s	731.8	1.5%	18%	195s
	22	796.4	7s	14%	5s	732.1	1.1%	26%	32s	730.6	2.7%	15%	514s
	23	825.5	23s	5%	12s	761.5	0.8%	28%	51s	765.0	2.9%	16%	367s
	24	884.2	16s	6%	10s	822.9	2.0%	26%	69s	824.0	2.6%	22%	99s
	25	845.8	4s	27%	3s	690.6	0.1%	16%	7s	685.6	3.7%	18%	143s
Poormond	21	111.6	404s	11%	61s	109.0	2.2%	<1%	52s	110.1	4.9%	<1%	561s
	22	113.6	342s	8%	31s	113.0	3.8%	<1%	87s	112.4	4.8%	<1%	556s
	23	126.6	230s	6%	31s	125.2	3.8%	<1%	54s	124.5	4.9%	<1%	262s
	24	138.9	465s	3%	31s	136.3	2.6%	<1%	51s	136.0	4.1%	<1%	174s
	25	113.4	359s	19%	32s	94.2	1.4%	<1%	52s	92.4	3.9%	<1%	212s
Simple VSD	21	148.2	<1s	79%	<1s	146.8	7s	14%	<1s	146.9	1.3%	<1%	<1s
	22	154.0	<1s	82%	<1s	152.4	6s	12%	<1s	151.5	1.2%	<1%	<1s
	23	167.5	<1s	76%	<1s	165.1	6s	11%	<1s	164.0	817s	<1%	<1s
	24	173.5	<1s	78%	<1s	172.2	6s	12%	<1s	171.2	3368s	<1%	<1s
	25	145.0	<1s	81%	<1s	139.8	3s	30%	<1s	140.9	742s	<1%	<1s
DWG	21	3379.3	1.6%	>99%	322s	(3266.5)	-	99%	-	(3266.9)	-	92%	-
	22	3469.1	4.2%	>99%	268s	(3292.3)	-	99%	-	(3284.8)	-	87%	-
	23	3635.4	4.5%	>99%	36s	(3428.9)	-	99%	-	(3417.9)	-	88%	-
	24	3689.4	1.5%	>99%	47s	(3549.8)	-	99%	-	(3549.1)	-	93%	-
	25	3602.3	12.2%	>99%	25s	(3128.1)	-	99%	-	(3122.9)	-	93%	-

Table 3: Results on the different networks of class BS and MS.

time (in s.) to find an optimal solution or its optimality gap (in %); *%CB* is the share of time spent in the callback function; *First* is the time to compute a first feasible solution.

7.2.1 Results for class BS

For all 45 instances of class BS, we computed solutions with an optimality gap of 5% and obtained the first solutions in less than 10 minutes. Except for the simplest instances, a small share of the overall computing time is spent in the callback function to evaluate the feasibility and possibly repair the integer relaxed solutions X. Indeed for class BS, subproblem ($\mathcal{P}(X)$) is a feasibility problem fast checked with the procedure described in Section 4.2, and the primal heuristic is launched only on 4% of the nodes.

We evaluated the impact of the combinatorial cuts (22) by dropping them, i.e. by generating, at each integer node X, the constraint (22) with $\bar{t} = T$ which cuts no other solution than X. For *Simple FSD* with $T = 24$ and *Poormond* with $T = 12$, the computational duration increased of 1.1 and 3.0 times in average. For *AT(M)* with $T = 12$, no feasible solutions are obtained in 10 minutes for 2 instances and an optimality gap above 7% is still present for the 3 others. Cuts (22) are then effective and even necessary to solve the problem in some cases.

The heuristic has contrasted performances over the three DWDNs. For

		T=12				T=24				T=48			
	Day	Best	Gap	%CB	1^{st}	Best	Gap	%CB	1^{st}	Best	Gap	%CB	1^{st}
Poormond	21	118.1	7.5%	21%	2531s	118.5	10.5%	12%	1458s	111.3	6.3%	4%	1229s
	22	(112.2)	-	27%	-	(108.5)	-	14%	-	115.9	7.7%	2%	1714s
	23	(124.4)	-	29%	-	132.6	9.6%	13%	2706s	134.1	11.8%	8%	308s
	24	(136.7)	-	24%	-	141.8	6.8%	13%	307s	140.0	6.8%	1%	501s
	25	(110.8)	-	34%	-	(91.4)	-	33%	-	98.6	10.1%	1%	312s

Table 4: Results on *Poormond* (BS class) without the primal heuristic.

Simple FSD, all instances are solved in less than 30 seconds for $T = 12$ and $T = 24$ and we do not call the heuristic in this case; for $T = 48$, most of the integer relaxed solutions (87%) were feasible. For *AT(M)* with $T = 24$ and $T = 48$, 12% of the feasible solutions as well as 9 out of the 10 best feasible solutions are computed by the heuristic. For *Poormond*, all computed feasible solutions are provided by the heuristic. The strength of the heuristic can be enlighted by considering the electricity tariffs of day 25. Indeed, it is about four times higher between 9.30am and 11.30am and significant savings can be obtained by turning the pump on as little as possible during this time window. On the *Poormond* instance, pumping was required during this time window. While the discretization imposes to switch the pumps on during a multiple of Δ_t, the heuristic allows to adjust this duration at its minimum.

We further investigated on *Poormond* the impact of the primal heuristic and, in turn, the strength of the MILP relaxation, by disabling the heuristic (see results in Table 4). For 6 out of the 15 instances, no solutions are obtained in the time limit of 1 hour. For the remaining 9 instances, the cost of the best solutions obtained was 5% higher on average, and the time needed to compute a first feasible solution is 15 times higher. These results show first that many integer relaxed solutions of $(\mathcal{P}_\epsilon)$ are infeasible for $(\mathcal{P})$. Indeed, with the heuristic disabled, only 26 out of the 105,159 potential candidates over the 15 instances were in fact feasible. However, many relaxed solutions are near feasible and the heuristic is able to quickly repair them and recover feasible solutions of good quality, which helps a lot in cutting the search tree. Indeed, with the heuristic on, only 36 relaxed solutions were to investigate and the heuristic was able to repair 33 out of them, even leading to solutions of lower cost for 30. Note finally, as expected, the higher deterioration when turning off the heuristic for instances with T=12 and T=24. Indeed, the heuristic helps to adjust the time step duration, then it allows to schedule with fewer and longer time steps without degrading the optimum.

These experiments show that the MILP relaxation is often tight for instances of class BS, providing low-cost and close-to-feasible solutions, and that it is well complemented by the heuristic, which appears to be a key factor of the overall solution scheme in some cases.

		T=12				T=24				T=48			
	Day	Best	Gap	% CB	1^{st}	Best	Gap	% CB	1^{st}	Best	Gap	% CB	1^{st}
DWG	21	3382.8	1.6%	94%	982s	(3266.7)	-	98%	-	(3267.0)	-	88%	-
	22	3398.2	1.8%	>99%	1814s	3420.6	3.7%	98%	7057s	(3284.9)	-	82%	-
	23	3555.6	1.8%	98%	668s	(3429.2)	-	98%	-	(3418.0)	-	82%	-
	24	3692.3	1.4%	88%	510s	3737.5	5.0%	98%	4568s	(3549.4)	-	86%	-
	25	3477.2	8.4%	>99%	509s	3312.7	5.3%	98%	1971s	3360.4	7.0%	91%	6958s

Table 5: Results on DWG (MS class) solving NLPs with Bonmin only

7.2.2 Results for class MS

High quality solutions are quickly computed for all 15 instances of *Simple VSD*: the first feasible solutions are found in less than 1 second and the best feasible solutions (with optimality gap $G = 1\%$) in less than 20 seconds. Incidentally, for each of the 85 integer nodes investigated, Baron was never called. Indeed, if $L_\epsilon(X)$ denotes the optimum of $(\mathcal{P}_\epsilon)$ at node X, Bonmin was always able to return feasible solutions of $(\mathcal{P}(X))$ with a cost smaller than $(1 + G)L_\epsilon(X)$.

Solving the instances of *DWG* was more difficult: for $T = 12$, feasible solutions are obtained in less than 2 minutes and best solutions found in 2 hours have an average optimality gap of 4.8%, but no feasible solutions are found for the largest instances with $T = 24$ and $T = 48$. Apart from the problem size, 3 characteristics of network *DWG* make these instances difficult: (a) the flow direction is unknown for 12 out of the 22 pipes. Each requires to introduce binary variables and linear and non-convex quadratic constraints (8^a)-(8^d) to the NLP subproblems $(\mathcal{P}(X))$; (b) the internal node pressure constraints (2) and the daily maximal withdrawal limits (4) are tight and make harder to recover feasibility from integer relaxed solutions; (c) *DWG* has PRVs but no variable-speed pumps, which offers less flexibility to readjust flows once the status of the pumps and valves is fixed since PRVs can only dissipate an excess of pressure, while variable-speed pumps can balance the pressure upward or downward. Hence, even for the smallest instances ($T = 12$), Bonmin and Baron were able to close (i.e. either to find a feasible solution in the gap limit G or to prove infeasibility) only 72 out of the 181 nodes investigated within the 5 minutes time limit, and they provided new incumbent solutions at 14 nodes. Only one improving solution was provided by Bonmin directly, but the mean computing time per node required by Baron was 192 seconds against only 6 seconds for Bonmin.

Table 5 presents the individual results when disabling Baron to solve the NLP subproblems heuristically with Bonmin only. Note that Bonmin almost always finished long before the 300s time limit. First, we observe in Table 5 (compared to Table 3) for the 5 instances with $T = 12$ that the best solutions found are improved, the mean cost being 1.5% lower and the mean optimality gap reduced from 4.8% to 3.0%. This improvement results from the increase of the number of nodes explored, from 36 to 741

on average. The search tree was even completely explored in the time limit for two instances: Day 21 in 6482s and Day 24 in 6596s. For $T = 24$ and $T = 48$, the heuristic approach allowed to compute feasible solutions within the time limit on 4 out of 10 instance.

7.3 Comparison with published results

This section provides an empiric comparison of our results with previous published results of alternative methods applied to one of the DWDNs presented so far. We evaluated our method on a different machine, but on the same data, including the electricity tariffs.

7.3.1 Enumeration and simulation [10] on AT(M)

In [10], Costa et al. apply their enumeration scheme on the AT(M) network (class BS) with 3 identical pumps and a rather complex pipe layout. The daily scheduling horizon is divided into 24 hours and the case study is investigated for different values of N the maximum number of pump activations. The search tree is built by deciding progressively for every hour the number of pumps to activate. Partial schedules at nodes are pruned if the hydraulic simulation with EPANET proves them to be infeasible. Performed on a PC (i7-4771 CPU, 3.5 GHz, 32 GB), the method computed optimal solutions in 425 seconds ($N = 1$), 10 hours ($N = 2$) and 81 hours ($N = 3$). Our LP/NLP branch and bound was not able to outperform these results: our computing time was longer for $N = 1$ (440 seconds) and a positive gap of 1.6% for $N = 2$ and 1.9% for $N = 3$ remained after the same amount of computing time for the two other cases. However, near-optimal solutions were quickly obtained with our algorithm with an optimality gap of less than 6% after 5 minutes, while no information on the quality of intermediate solutions are provided with the enumeration scheme if prematurely stopped because the search tree is explored by Breadth First Search.

7.3.2 Lagrangian relaxation [20] or Benders decomposition [38] on Poormond

Ghaddar et al. [20] and Naoum-Saway et al. [38] reported results on network *Poormond* (class BS) with $T = 48$ and 96 respectively. In both papers, modeling assumptions are identical and the only difference with ours is the shape of the pump power consumption function.[6] We then solved our model with $T = 48$ after computing the linear coefficients in (20) to approximate their cubic formulation, then recalculated the costs from the best solutions found. Bearing in mind that our method was performed on a different

[6]The authors of [20] and [38] do not mention check valves in their mathematical formulation, but likely include them in the hydraulic model of EPANET. That could explain the inconsistency between the lower bounds reported in [20] and our results.

	Day	LR [20]	BD [38]	LP/NLP	Δ
Poormond	21	130.7	129.74	116.05	10.6%
	22	139.4	131.39	119.35	9.2%
	23	140.2	140.59	124.12	13.0%
	24	151.8	147.73	137.68	6.8%
	25	130.3	125.57	93.32	25.7%

Table 6: Best costs obtained with a Lagrangian relaxation (LR), a Benders decomposition (BD), and our method (LP/NLP) with a time limit of 1 hour on different machines (LR and BD were performed on a RedHat Linux blade server 3.5 GHz). Δ gives the cost improvement of LP/NLP with respect to the best of LR and BD.

environment, we reported in Table 6 substantially better solutions for the five instances investigated.

Our approach bears similarities with the combinatorial Benders decomposition of [38] but it seems to benefit a lot of our tigher MILP relaxation. In [38], the non-convex constraints and the objective are almost totally relaxed in the master MILP. Furthermore, we showed in Section 7.2.1 the importance of the primal heuristic on these instances. Finally, note that the best solutions were found with the Benders decomposition of [38] when run in the manner of a local search, loosing thus the faculty to provide performance guarantee.

7.3.3 Outer-approximation algorithm [50] on simplified versions of Poormond

The outer-approximation algorithm proposed by Shi and You [50] is applied to two simplified versions of the instance Poormond with a scheduling horizon of 8 one-hour time periods. Despite some differences with our model (Hazen-William formula for ϕ, cubic function of the flow for Γ_k^F), the optimal solutions found are similar (the state of one pump differs during one time period for the *small-scale* case study while the two solutions are identical for the *large-scale* case study). Thus, we propose to compare the computing times to obtain them. The reported results were obtained, on an Intel Core i5-2400 CPU @ 3.10 GHz with 8GB RAM, using CPLEX 12 and CONOPT 3 for solving the MILP relaxations and NLP restrictions, respectively. The reported computing times are 60.74s and 321.69s, while we obtain 1.44s and 9.15s, respectively. This difference can be explained by analysing Figure 9 in [50] which decomposes the time needed to solve the *large-scale* instance into the different steps. We observe that the OA algorithm takes only two iterations, but most of the time is spent to solve two MILP relaxations: at the first iteration (190.35s) to obtain a relaxed integer solution, and at the

Instance	21	22	23	24	25
Simple VSD	148.2 (22s)	154.0 (25s)	167.5 (21s)	173.7 (26s)	145.0 (24s)
DWG	-	-	-	-	-

Table 7: Optimum values (with $G = 1\%$) and computation times for solving $(\mathcal{P})$ with Baron on the MS case studies with $T = 12$ in a time limit of 2 hours.

second iteration (129.51s) to prove that no cheaper solution exists. With our branch and bound, the optimal solution was obtained in less than 1s, then it took less than 9s to explore the search tree and evaluate 512 other integer nodes.

The comparison highlights two key features of the proposed method. First, we explore one single search tree and evaluate all integer solutions, in the manner of the LP/NLP branch and bound, while the OA algorithm waits to evaluate the optimal solution of the MILP relaxation. This is justified as the evaluation step is cheap for instances of class BS. Second, our convex OA relaxation may perhaps be not as tight as the non-convex OA relaxation of [50], but it results in a much smaller MILP as we do not introduce additional binary variables to model piecewise linear segments. Finally, note that the MILP relaxation of [50] is automatically generated while we propose a tailored generic MILP relaxation, but both approaches have a wide range of application.

7.3.4 Direct resolution of Simple VSD and DWG through MINLP

Finally, we directly solved the non-convex MINLP formulations of the two DWDNs of class MS with Baron using the default parameters.[7] Results are given in Table 7 for $T = 12$ and can be compared to the results of the proposed approach summarized in Table 3. For *Simple VSD*, the direct resolution required more than 20 seconds to find and prove the optimum while our method required less than 1 second. For *DWG*, the direct resolution found no feasible solutions at all in 2 hours, while our method found at least one feasible solution for all instances in less than one hour and with an optimality gap lower than 15%.

8 Conclusion

In this paper, we presented a tailored LP/NLP-based branch-and-bound algorithm to solve at optimality a non-convex formulation of the pump

[7]Note that [34] and [56] report approximated solutions that cannot be directly compared with ours.

scheduling problem in DWDNs. This framework can readily be implemented by embedding a non-convex NLP solver as a lazy cut separator within a MILP solver. To our knowledge, this framework has never been applied in the context of water or gas networks. The full resolution scheme includes also several contributions such as a generic tight MILP relaxation that has provided the opportunity to deal with networks of class MS, i.e. with *multi-settings* active elements such as variable-speed pumps. For the other networks (of class BS: *binary-settings*), we added several improving techniques including a fast evaluation, cuts, and a new primal heuristic to turn slightly infeasible solutions into feasible solutions. A computational study on several benchmark instances, including a comparison with competing methods, highlights the strengths and weaknesses of the proposed approach. For DWDNs of class BS, it quickly computes near-optimal solutions but the lower bound evolves slowly afterwards. While a systematic piecewise linear relaxation as in spatial branch and bound does not seem worthwhile, the MILP relaxation could be refined during the search by branching on the non-convex constraints which are consistently violated. For DWDNs of class MS, the proposed method outperforms a direct resolution with a global optimization solver but solving the non-convex NLP restrictions at integer nodes remains a bottleneck as the problem size grows. One option could be to derive sufficient conditions to reject the infeasible integer nodes as done in [26, 27] in the context of the optimal design of gas networks. Finally, this work suggests that this two-step LP/NLP-based branch and bound could be considered in a broader context to solve non-convex MINLPs given efficient bound tightening techniques and a tight MILP relaxation.

Acknowledgments

We would like to thank Bradley Eck, Joe Naoum-Sawaya, Bruno de Athayde Prata, Hanyu Shi and Derek Verleye for sharing with us some complementary elements concerning their respective data and for providing us technical details concerning their model. The work of Gratien Bonvin was partially supported by a Doc.Mobility Grant (P1SKP2_174858) from the Swiss National Science Foundation.

References

[1] E. Balas and R. Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal on Applied Mathematics*, 23:61–69, 1972.

[2] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.

[3] Pierre Bonami, Lorenz T Biegler, Andrew R Conn, Gérard Cornuéjols, Ignacio E Grossmann, Carl D Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, et al. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

[4] Gratien Bonvin, Sophie Demassey, Claude Le Pape, Nadia Maïzi, Vincent Mazauric, and Alfredo Samperio. A convex mathematical program for pump scheduling in a class of branched water networks. *Applied Energy*, 185:1702 – 1711, 2017.

[5] Cristiana Bragalli, Claudia D'Ambrosio, Jon Lee, Andrea Lodi, and Paolo Toth. On the optimal design of water distribution networks: a practical minlp approach. *Optimization and Engineering*, 13(2):219–246, 2012.

[6] Jens Burgschweiger, Bernd Gnädig, and Marc C Steinbach. Nonlinear programming techniques for operative planning in large drinking water networks. *Open Applied Mathematics Journal*, 3:14–28, 2009.

[7] Jens Burgschweiger, Bernd Gnädig, and Marc C Steinbach. Optimization models for operative planning in drinking water networks. *Optimization and Engineering*, 10(1):43–73, 2009.

[8] Ron Carlson. The correct method of calculating energy savings to justify adjustable-frequency drives on pumps. *IEEE Transactions on Industry Applications*, 36(6):1725–1733, 2000.

[9] European Commission. 2030 energy strategy, 2014. [accessed: 18-Apr-2017].

[10] Luis Henrique Magalhães Costa, Bruno de Athayde Prata, Helena M Ramos, and Marco Aurélio Holanda de Castro. A branch-and-bound algorithm for optimal pump scheduling in water distribution networks. *Water resources management*, 30(3):1037–1052, 2016.

[11] Claudia D'Ambrosio and Andrea Lodi. Mixed integer nonlinear programming tools: an updated practical overview. *Annals of Operations Research*, 204(1):301–320, 2013.

[12] Claudia D'Ambrosio, Andrea Lodi, Sven Wiese, and Cristiana Bragalli. Mathematical programming techniques in water network optimization. *European Journal of Operational Research*, 243(3):774 – 788, 2015.

[13] Teodora Dan, Andrea Lodi, and Patrice Marcotte. An Exact Algorithm for a Class of Mixed-Integer Programs with Equilibrium Constraints. Technical Report DS4DM-2018-010, École Polytechnique de Montréal, 2018.

[14] Louise de La Perrière, Antoine Jouglet, Alexandre Nace, and Dritan Nace. Water planning and management: An extended model for the real-time pump scheduling problem. In *Advances in hydroinformatics*, pages 153–170. Springer, 2014.

[15] C. D'Ambrosio, A. Frangioni, L. Liberti, and A. Lodi. On interval-subgradient and no-good cuts. *Operations Research Letters*, 38(5):341–345, 2010.

[16] Bradley J. Eck and Martin Mevissen. Valve placement in water networks: Mixed-integer non-linear optimization with quadratic pipe friction. Technical report, IBM Research Report, September 2012.

[17] Federal Ministry for Economic Affairs and Energy (BMWi). *An Electricity Market for Germany's Energy Transition (Green Paper)*, October 2014.

[18] M. Feldman. Aspects of energy efficiency in water supply systems. In *The 5th IWA water loss reduction Specialist Conference*, pages 85–89, Capetown, South Africa, 2009.

[19] Björn Geißler, Oliver Kolb, Jens Lang, Günter Leugering, Alexander Martin, and Antonio Morsi. Mixed integer linear models for the optimization of dynamical transport networks. *Mathematical Methods of Operations Research*, 73(3):339–362, 2011.

[20] Bissan Ghaddar, Joe Naoum-Sawaya, Akihiro Kishimoto, Nicole Taheri, and Bradley Eck. A lagrangian decomposition approach for the pump scheduling problem in water networks. *European Journal of Operational Research*, 241(2):490 – 501, 2015.

[21] C. Giacomello, Z. Kapelan, and M. Nicolini. Fast hybrid optimization method for effective pump scheduling. *J. Water Resour. Plann. Manage.*, page 175, 2013.

[22] Ambros Gleixner, Timo Berthold, Benjamin Müller, and Stefan Weltge. Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, 67(4):731–757, 2017.

[23] Ambros M. Gleixner, Harald Held, Wei Huang, and Stefan Vigerske. Towards globally optimal operation of water supply networks. *Numerical Algebra, Control and Optimization*, 2(4):695–711, 2012.

[24] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016.

[25] William E. Hart, Carl D. Laird, Jean-Paul Watson, David L. Woodruff, Gabriel A. Hackebeil, Bethany L. Nicholson, and John D. Siirola. *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, second edition, 2017.

[26] Jesco Humpola and Armin Fügenschuh. A new class of valid inequalities for nonlinear network design problems. Technical report, Zuse-Institut, Berlin, 2013.

[27] Jesco Humpola and Felipe Serrano. Sufficient pruning conditions for minlp in gas network design. *EURO Journal on Computational Optimization*, 5(1-2):239–261, 2017.

[28] Vincent Krakowski, Edi Assoumou, Vincent Mazauric, and Nadia Maïzi. Reprint of feasible path toward 40–100power supply in france by 2050: A prospective analysis. *Applied Energy*, 184:1529 – 1550, 2016.

[29] Kevin E Lansey and Kofi Awumah. Optimal pump operations considering pump switches. *Journal of Water Resources Planning and Management*, 120(1):17–35, 1994.

[30] M. López-Ibáñez, T. D. Prasad, and B. Paechter. Ant colony optimization for the optimal control of pumps in water distribution networks. *J. Water Resour. Plann. Manage.*, page 337, 2008.

[31] G. Mackle, G. A. Savic, and G. A. Walters. Application of genetic algorithms to pump scheduling for water supply. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 400–405, 1995.

[32] Helena Mala-Jetmarova, Nargiz Sultanova, and Dragan Savic. Lost in optimisation of water distribution systems? a literature review of system operation. *Environmental Modelling & Software*, 93:209 – 254, 2017.

[33] G. Mccormick and R S Powell. Derivation of near-optimal pump schedules for water distribution by simulated annealing. *Journal of the Operational Research Society*, 55(7):728–736, 2004.

[34] RMM Menke, EA Abraham, and IS Stoianov. Modeling variable speed pumps for optimal pumpscheduling. In *World Environmental and Water Resources Congress*, pages 199–209, 2016.

[35] Ruben Menke, Edo Abraham, Panos Parpas, and Ivan Stoianov. Demonstrating demand response from water distribution system through pump scheduling. *Applied Energy*, 170:377 – 387, 2016.

[36] Ruben Menke, Edo Abraham, Panos Parpas, and Ivan Stoianov. Exploring optimal pump scheduling in water distribution networks with branch and bound methods. *Water Resources Management*, pages 1–17, 2016.

[37] Antonio Morsi, Björn Geißler, and Alexander Martin. Mixed integer optimization of water supply networks. In *Mathematical Optimization of Water Networks*, pages 35–54. Springer, 2012.

[38] Joe Naoum-Sawaya, Bissan Ghaddar, Ernesto Arandia, and Bradley Eck. Simulation-optimization approaches for water pump scheduling and pipe replacement problems. *European Journal of Operational Research*, 246(1):293–306, 2015.

[39] Lindell Ormsbee, Thomas Walski, Donald Chase, and Wayne Sharp. Methodology for improving pump operation efficiency. *Journal of Water Resources Planning and Management*, 115(2):148–164, 1989.

[40] Lindell E. Ormsbee and Kevin E. Lansey. Optimal control of water supply pumping systems. *Journal of Water Resources Planning and Management*, 120(2):237–252, 1994.

[41] J. Nault F. Papa. Lifecycle assessment of a water distribution system pump. *Journal of Water Resources Planning and Management*, 141(12), 2015.

[42] Filippo Pecci, Edo Abraham, and Ivan Stoianov. Quadratic head loss approximations for optimisation problems in water supply networks. *Journal of Hydroinformatics*, 19(4):493–506, 2017.

[43] Ignacio Quesada and Ignacio E Grossmann. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, 16(10-11):937–947, 1992.

[44] A. U. Raghunathan. Global optimization of of nonlinear network design. *SIAM Journal on Optimization*, 23(1):268–295, 2013.

[45] Zhengfu Rao and Fernando Alvarruiz. Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model. *Journal of Hydroinformatics*, 9(1):15–24, 2007.

[46] L Rossman. *EPANET*, 2000.

[47] N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.

[48] Rubén Orlando Salgado-Castro. *Computer modelling of water supply distribution networks using the gradient method.* PhD thesis, Newcastle University, 1988.

[49] SEMO. Single electricity market operator, 2016.

[50] Hanyu Shi and Fengqi You. Energy optimization of water supply system scheduling: Novel minlp model and efficient global optimization algorithm. *AIChE Journal*, 62(12):4277–4296, 2016.

[51] P. Skworcow, D. Paluszczyszyn, and B. Ulanicki. Pump schedules optimisation with pressure aspects in complex large-scale water distribution systems. *Drinking Water Engineering and Science*, 7(1):53–62, 2014.

[52] Edward Smith and Constantinos Pantelides. Global optimisation of nonconvex minlps. *Computers & Chemical Engineering*, 21:S791–S796, 1997.

[53] E. S. Thorsteinsson. Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In *International Conference on Principles and Practice of Constraint Programming (CP'01)*, volume 2239 of *Lecture Notes in Computer Science*, pages 16–30, 2001.

[54] E. Todini and S. Pilati. A gradient algorithm for the analysis of pipe networks. In Bryan Coulbeck and Chun-Hou Orr, editors, *Computer Applications in Water Supply: Vol. 1—systems Analysis and Simulation*, pages 1–20. Research Studies Press Ltd., Taunton, UK, UK, 1988.

[55] J. E. van Zyl, D. A. Savic, and G. A. Walters. Operational optimization of water distribution systems using a hybrid genetic algorithm. *J. Water Resour. Plann. Manage.*, page 160, 2004.

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

[57] Thomas M Walski, E Downey Brill Jr, Johannes Gessler, Ian C Goulter, Roland M Jeppson, Kevin Lansey, Han-Lin Lee, Jon C Liebman, Larry Mays, David R Morgan, et al. Battle of the network models: Epilogue. *Journal of Water Resources Planning and Management*, 113(2):191–203, 1987.