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 tailored 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.

---

[*]Center for Applied Mathematics, Mines ParisTech, PSL Research University, Sophia-Antipolis, France, `sophie.demassey@mines-paristech.fr`

[†]CERC and École Polytechnique de Montreal, Montreal, Canada, `andrea.lodi@polymtl.ca`

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 [38]. 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, 33]. 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]. Solving 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 [34, 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. Polyhedral Outer Approximations (OA) of the non-convex equality constraints are devised in a way that optimal relaxed solutions are expected (in practice) to be close to satisfy the equality constraints. These relaxations do not require additional discrete variables as the OA regions are convex, and a minimal number of planes are generated w.r.t. some gap parameter ϵ. They result in a MILP relaxation (vs. approximation) of reasonable size and that is consistent with the objective.

Second, to solve the original non-convex MINLP, we present a variant of LP/NLP branch and bound for convex MINLPs [42], 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 tries to slightly adjust the time step lengths to fix the tank level limit violations. Indeed, when using a large discretization step, these violations could sometimes not be fixed with another (binary) assignment of pumps and valves, although switching pumps a few minutes earlier or later is technically possible and can solve these violations. Our heuristic is thus an alternative to optimizing on smaller time steps. Note also 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* [35], *AT(M)* [44, 10], *Poormond* [20, 37, 49] and *DWG* [56]), and drove an empirical comparison with recently reported results [10, 20, 37, 49] and with the reference global optimization solver BARON [46]. 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 solving 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 [39]. For example, Ormsbee et al. [40] 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 [31], "deterministic methods started being supplemented by metaheuristics during the mid 1990s", in particular genetic algorithms [50] sometimes coupled with local search techniques [55], but also ant colony optimization [30] or simulated annealing [32]. 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 [34]. 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, 36, 56, 14, 34, 35], the non-linear flow-coupling constraints are approximated by piecewise linear functions. While this technique outperforms a direct approach with a global optimization solver [34], 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 [33]. 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. [37] 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 [49] 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 solution 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 [34] and variable-speed pumps [35].

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 [37] and [49]. The key differences with these approaches are that: (1) our MILP master relaxation is tighter than [37] and

smaller than [49], 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 [42] for general convex MINLPs but we progressively tighten the MILP relaxation with combinatorial cuts, as in [37], instead of OA cuts. Furthermore, we specialize the method for DWDNs of class BS by exploiting, like in [37, 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 [43] to improve the LP/NLP branch and bound he applies to a convex formulation of this static problem.

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[1] exists, 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. [34] compared different implementations of mathematical programming approaches on small generated instances and concluded that piecewise-linear approximation was faster than solving the non-convex model with SCIP. The two hybrid approaches by [20] and [37] were also rigorously compared on instances of the *Poormond* network [37]. 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 [35, 44, 20, 56]. While it was not an option to reimplement the complex methods of the literature, we propose to perform an empiric comparison with the computational results reported in four recent papers [20, 37, 10, 49].

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 is further split into fixed-speed pumps K_F and variable-speed pumps K_V. The scheduling horizon is discretized in T periods, $t \in \mathbb{T} = \{0, \ldots, T-1\}$, that

---

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

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

Table 1: Summary of notation

we assume w.l.o.g. of equal length Δ (in hours). As water demands and electricity tariffs often fluctuate on a daily basis, the horizon is typically limited to one day: $T = 24$ and $\Delta = 1$. We also assume a steady-state operation of the network during each time period [6, 20].[2]

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 $a \in A$ during period $t \in \mathbb{T}$; hydraulic head $h_{jt} \geq 0$ is the sum of the geographical elevation and the water pressure head (in meters) at node $j \in J$ at the end of period $t \in \mathbb{T}$; binary variable $x_{at} \in \{0,1\}$ models the status of an active element $a \in K \cup V$ during period $t \in \mathbb{T}$, e.g. whether a pump is on or off, or a gate valve is open or close; finally, for variable-speed pumps $k \in K_V$, continuous variable $0 \leq w_{kt} \leq 1$ gives the normalized speed value during period $t \in \mathbb{T}$.

3.2 Nodes

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

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

with $D_{jt} \geq 0$ the forecasted water demand rate (in m^3/h) for period t. Note that pressure-dependent 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 meters), thus

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

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

---

[2]In [36], 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.

7

Sources. We assume that head level H_{jt} (in meters) at source node $j \in J_S$ varies in time $t \in \mathbb{T}$ 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 [45]. 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

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

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

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

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 $h_{j(t+1)} - h_{jt}$ 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 $H_{j0} \in [\underline{H_j}, \overline{H_j}]$, namely

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

$$h_{j0} = H_{j0} \leq h_{jT}. \tag{7}$$

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, 41]. Our approach is compatible with all these relations but, to facilitate the comparison with previously proposed methods, we adopt here, for each pipe $l = ij \in L$ and time $t \in \mathbb{T}$, 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. [49])

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

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

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

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

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

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

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

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 whose computation is discussed in Section 5.1. Note that $x_{kt} = w_{kt}$ for fixed-speed pump $k \in K_F$.

The maintenance cost of a pump can represent around 10% of its overall net present value lifecycle cost [38]. Sound practices can limit this cost, for example by restricting the number $N \in \mathbb{T}$ of daily pump switches or fixing the minimum number of periods $\tau_1 \in \mathbb{T}$ (resp. $\tau_0 \in \mathbb{T}$) 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 pump $k \in K$ is switched on (resp. off) at

9

time $t \in \mathbb{T}$ and by the following constraints:

$$\sum_{t=1}^{T-1} y_{kt} \leq N, \tag{12}$$

$$y_{kt} \geq x_{kt} - x_{k(t-1)}, \quad \forall t \in \{1, \ldots, T-1\} \tag{13}$$

$$y_{kt} \leq x_{kt'}, \quad \forall t, t' \in \mathbb{T} : 0 < t \leq t' < t + \tau_1 \tag{14}$$

$$z_{kt} \geq x_{k(t-1)} - x_{kt}, \quad \forall t \in \{1, \ldots, T-1\} \tag{15}$$

$$z_{kt} \leq 1 - x_{kt'}, \quad \forall t, t' \in \mathbb{T} : 0 < t \leq t' < t + \tau_0. \tag{16}$$

Valves. A large variety of valves with distinct functions exist for DWDNs [45]. 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 $t \in \mathbb{T}$, namely

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

$$\underline{Q_v}g(x_{vt}) \leq q_{vt} \leq \overline{Q_v}x_{vt}, \tag{18}$$

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 (see Section 5.1) 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 closed ($x_{vt} = 0$), 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. In our method, we use both formulations: the discrete linear one in the MILP relaxation model of the problem and the non-convex continuous one in the NLP model restrictions.

3.4 Optimization task

The common objective of pump scheduling is to supply the forecast water demand with appropriate pressure requirements at minimal operating

10

cost [6]. Following [56], we define a constant cost $E_s \in \mathbb{R}_+$ per unit of raw water for its extraction and treatment at source $s \in J_S$, to which we add the energy costs induced by pumping. The power consumption of a variable-speed pump $k \in K_V$ for a period $t \in \mathbb{T}$ can be approximated by [7, 36]

$$\Gamma_k(q_{kt}, w_{kt}) = w_{kt}^2 \left(\lambda_k w_{kt} + \mu_k q_{kt} \right), \tag{19}$$

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.

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

since either $w_{kt} = x_{kt} = 1$ or $w_{kt} = x_{kt} = q_{kt} = 0$.

Given $C_t \in \mathbb{R}_+$ the electricity unit cost at time $t \in \mathbb{T}$, the objective function to minimize is given by $\sum_{t \in \mathbb{T}} \Gamma^t(q_t, w_t) \Delta$ with:

$$\Gamma^t(q_t, w_t) = \sum_{s \in J_S} \sum_{sj \in A} E_s q_{sjt} + \sum_{k \in K} C_t \Gamma_k(q_{kt}, w_{kt}). \tag{21}$$

3.5 Summary of the mathematical model

In summary, a formulation of the Pump Scheduling Problem as a non-convex MINLP can be stated as follows:

$$(\mathcal{P}): \min_{x,w,q,h,h'} \sum_{t \in \mathbb{T}} \Gamma^t(q_{kt}, w_{kt}) \Delta \tag{21}$$

s.t.

$$h'_{jt} = H_{jt} \qquad\qquad \forall j \in J_S, t \in \mathbb{T} \tag{3}$$

$$\sum_{t \in \mathbb{T}} \sum_{ji \in A} q_{jit} \Delta \leq V_j \qquad\qquad \forall j \in J_S \tag{4}$$

$$\sum_{ij \in A} q_{ijt} - \sum_{ji \in A} q_{jit} = \frac{S_j}{\Delta}(h'_{j(t+1)} - h'_{jt}) \qquad \forall j \in J_T, t \in \mathbb{T} \tag{5}$$

$$h'_{jt} \in [\underline{H_j}, \overline{H_j}] \qquad\qquad \forall j \in J_T, t \in \mathbb{T} \cup \{T\} \tag{6}$$

$$h'_{j0} = H_{j0} \leq h'_{jT} \qquad\qquad \forall j \in J_T \tag{7}$$

$$\underline{W_k} x_{kt} \leq w_{kt} \leq \overline{W_k} x_{kt} \qquad\qquad \forall k \in K, t \in \mathbb{T} \tag{11}$$

$$(q_t, h_t) \in \mathcal{F}_t(x_t, w_t, h'_t) \qquad\qquad \forall t \in \mathbb{T} \tag{22}$$

$$x \in \mathcal{S} \tag{23}$$

where

$$\mathcal{S} = \{x \in \{0,1\}^{(K \cup V) \times \mathbb{T}}, \exists y \in \{0,1\}^{K \times \mathbb{T}}, z \in \{0,1\}^{K \times \mathbb{T}}, (12) - (16)\}$$

defines the profiles of pumps and valves activity that satisfy the lifetime-preserving operational constraints, and

$$\mathcal{F}_t(x_t, w_t, h'_t) = \{(q_t, h_t) \in \mathbb{R}^A \times \mathbb{R}^J, (1), (2), (8), (9^a), (10), (17), (18),$$
$$h_{jt} = h'_{jt} \ \forall j \in J_S \cup J_T\}$$
(24)

defines the solutions of the so-called *steady-state network analysis problem* during period $t \in \mathbb{T}$ associated to the given operation status of pumps and valves (x_t, w_t), initial heads at source and tanks h'_t and demand rates at junctions D_t.

A key characteristic of this problem is that, since all the head loss functions (Φ for pipes and $-\Psi(\cdot, w_t)$ for pumps) increase monotonically with the flow and since either demand or head is known at each node, the system of equations $\{(1), (8), (9^a)\}$ has an unique solution in $\mathbb{R}^A \times \mathbb{R}^J$, which can efficiently be computed with a gradient method – an application of the Newton-Raphson algorithm – known as the *Todini-Pilati method* [54, 47]. It means that, in our context, we can compute the (unique) solution of $\mathcal{F}_t(x_t, w_t, h'_t)$, or prove that none exists by: *i*) removing the arcs corresponding to inactive pumps and valves ($x_{at} = 0$), *ii*) fixing active pump speeds w_t, heads at sources and tanks h'_t and demands at junctions D_t, *iii*) applying the Todini-Pilati method to the resulting network, and *iv*) checking that the computed solution is not out of range according to (2), (10), and (18). Remark that, we found no such violation in all our experiments, although our approach makes an intensive use of this procedure (see Sections 4.2 and 6) when considering networks with binary settings (BS). It is likely because our test networks have realistic topologies (and (2) are naturally induced) and do not include strict operational restrictions on the flow values ((10) and (18) are always satisfiable). Finally, in the BS case thereafter, we simply denote the network analysis subproblem (24) $\mathcal{F}_t(x_t, h'_t)$, as BS networks only contain fixed-speed pumps for which $w = x$ holds.

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 $x = (x_{at})_{a \in K \cup V, t \in \mathbb{T}}$ 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 ($x = X \in \{0,1\}^{(K \cup V) \times \mathbb{T}}$), 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 global optimization 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 [42] 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 [43] 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 [43] 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, the restricted subproblem $\mathcal{P}(X)$ at an integer node $x = X \in \{0,1\}^{(K \cup V) \times \mathbb{T}}$ admits at most one feasible solution $q \in \mathbb{R}^{A \times \mathbb{T}}$, $h \in \mathbb{R}^{J \times \mathbb{T} \cup \{T\}}$, defined recursively by:

$$
\begin{cases}
h'_{j0} = H_{j0} & \forall j \in J_T \\
h'_{jt} = H_{jt} & \forall j \in J_S, t \in \mathbb{T} \\
h'_{j(t+1)} = h'_{jt} + \frac{\Delta}{S_j}(\sum_{ij \in A} q_{ijt} - \sum_{ji \in A} q_{jit}) & \forall j \in J_T, t \in \mathbb{T} \\
(q_t, h_t) \in \mathcal{F}_t(X_t, h'_t) & \forall t \in \mathbb{T}
\end{cases}
\tag{25}
$$

Indeed, as explained in Section 3.5, for given X_t and h'_t, the network analysis problem $\mathcal{F}_t(X_t, h'_t)$ has at most one solution.

As a consequence, we do not solve $\mathcal{P}(X)$ with an NLP solver but with the following iterative algorithm, called *extended period analysis* and also implemented in the notorious hydraulic simulator EPANET: At each iteration $t \in \mathbb{T}$, we compute the unique solution (q_t, h_t) of $\mathcal{F}_t(X_t, h'_t)$ with the Todini-Pilati method, check the bounds (2), (10) , and (18), and compute the associated operation cost $z_t(X_t) = \Delta\Gamma^t(q_t, X_t)$. We then compute the water level in the tanks at the end of period t, given by h'_{t+1} with (5), and check if they satisfy the tank capacities (6). If no bound is violated at any time $t \in \mathbb{T}$ and if the final tank levels at $t = T$ also satisfy (6) and (7), then the solution is returned with operation cost $z(X) = \sum_{t \in \mathbb{T}} z_t(X_t)$. Otherwise, as soon as a constraint is violated, say at time period $t = \bar{t}$, the simulation can be prematurely halted and the relaxed solution X said

infeasible. As in the Benders decomposition of [37], we can then reinforce our MILP relaxation $(\mathcal{P}_\epsilon)$ with the combinatorial infeasibility cut

$$\sum_{t=0}^{\bar{t}} \left(\sum_{\substack{a \in K \cup V: \\ X_{at}=0}} x_{at} + \sum_{\substack{a \in K \cup V: \\ X_{at}=1}} (1 - x_{at}) \right) \geq 1. \tag{26}$$

The so-called [1] *canonical cut* (26), 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 $t \in \{0, \ldots, \bar{t}\}$ to prevent the infeasibility at time period $\bar{t}$. Although not necessarily efficient in general, the combinatorial cuts were effective in accelerating the branch and bound in our experiments (see Section 7.2.1), at least when they stemmed from infeasibilities detected at early 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 solve the restricted subproblem $\mathcal{P}(X)$ with the global optimization solver Baron after bringing two adjustments to the model: (a) 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 (b) we model PRVs with the non-convex constraints (17^a). The first condition introduces one binary variable for each bidirectional pipe and each time, 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 ($x_{vt}, v \in V_{PR}, t \in \mathbb{T}$) are unfixed. Once checked, whether a feasible solution is found or not, the following combinatorial infeasibility cut is added to the MILP $(\mathcal{P})$:

$$\sum_{t \in \mathbb{T}} \left(\sum_{\substack{a \in K \cup V_C \cup V_G: \\ X_{at}=0}} x_{at} + \sum_{\substack{a \in K \cup V_C \cup V_G: \\ X_{at}=1}} (1 - x_{at}) \right) \geq 1. \tag{27}$$

Possibly, the resulting restricted non-convex MINLPs cannot be solved at global optimality in reasonable time. In this case, we propose to turn our whole solution process into a heuristic providing also a lower bound and then a certificate of performance of the returned solution. First, we fix a

stopping tolerance $0 < \texttt{tol} < 1$ and solve each restricted subproblem after setting $\bar{z}(1 - \texttt{tol})$ as an upper bound on the optimal value, where $\bar{z}$ is the current relaxed MILP incumbent. Second, to prevent our branch and bound to get stuck at integer nodes which are hard to close, we solve each restricted subproblem $\mathcal{P}(X)$ within a fixed time limit and record both the best feasible solution found, which is possibly used to update the MILP incumbent $\bar{z}$, and 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 unsolved integer nodes. Note that after this correction, the final optimality gap $\frac{z^* - L}{L}$ may be greater than $\texttt{tol}$.

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 algorithm 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 tolerance (we used $\texttt{tol} = 1\%$ 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, and thus cannot be used for bounding in the branch and bound described in Section 4.

Thus, we propose instead to build a tractable MILP relaxation $(\mathcal{P}_\epsilon)$ of $(\mathcal{P})$ where the non-convex functions are replaced by polyhedral outer approximations (OA) defined by a minimal number of planes so as to be tight in this sense: the projection of any optimal relaxed solution on any such polyhedron is expected to be at a distance lower that some given tolerance $\epsilon > 0$ from the non-convex curve. Indeed, as previously observed on a specific branched network [4], we expect that when relaxing the equality head-flow constraints, then pipe head losses (resp. pump head gains) tend to be underestimated (resp. overestimated) in optimal relaxed solutions, and variable-speed pumps tend to run with high speeds. Reasons for that are: (1) a minimum head is required at each node, but only water tanks have a maximum head limit related to their maximum capacity, and because of the optimization criterion, this maximum limit is reached on a limited number
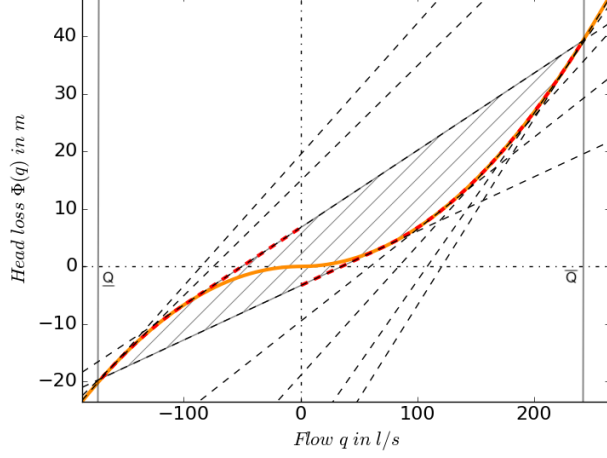
16

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

of time steps, (2) due in particular to their fixed operating costs, pumps are called heavily infrequently – rather than lightly frequently – thus inducing high speeds and high head gains at the periods they are active. Under this rationale, we propose to build the relaxation with tight underestimators for the pipe head losses, and tight overestimators for the pump head gains when running at speed $w = 1$. Furthermore, in order to keep the relaxation tractable, we consider polyhedral OA of the convex functions, which can then be modeled with only linear constraints and no additional discrete variables.

Precisely, Figure 1 depicts, for a given pipe $l = ij \in L$, the head loss function $q \mapsto \Phi_l(q)$ (the orange curve) on $[\underline{Q_l}, \overline{Q_l}]$, and the polyhedral outer approximation P_l^ϵ (the hatched area) that we consider in our relaxed model, i.e., we relax the nonconvex equality constraint $h_{jt} - h_{it} = \Phi_l(q_{ijt})$ to the linear system $(q_{ijt}, h_{jt} - h_{it}) \in P_l^\epsilon$ for each $t \in \mathbb{T}$. P_l^ϵ is built so that head loss underestimations (on the dashed red piece-wise line) are tight as follows:

$$|\delta_h| \geq |\Phi_l(|q|)| - \epsilon, \forall (q, \delta_h) \in P_l^\epsilon, q \in [\underline{Q_l}, \overline{Q_l}(1 - \sqrt{2})] \cup [\underline{Q_l}(1 - \sqrt{2}), \overline{Q_l}].$$

Note that the underestimation may exceed ϵ when flow is close to 0, i.e., in the interval $[\overline{Q_l}(1 - \sqrt{2}), \underline{Q_l}(1 - \sqrt{2})]$. Both the size of this interval and the number of planes required to satisfy our definition of tightness above are inversely related to the sharpness of the bounds $\underline{Q_l}$ and $\overline{Q_l}$.

In this section, we first show how to tighten these bounds (as well as the big-M values in our model), then we exhibit an outer approximation P_l^ϵ of the head loss function Φ_l that meets our tightness condition. We then present

17
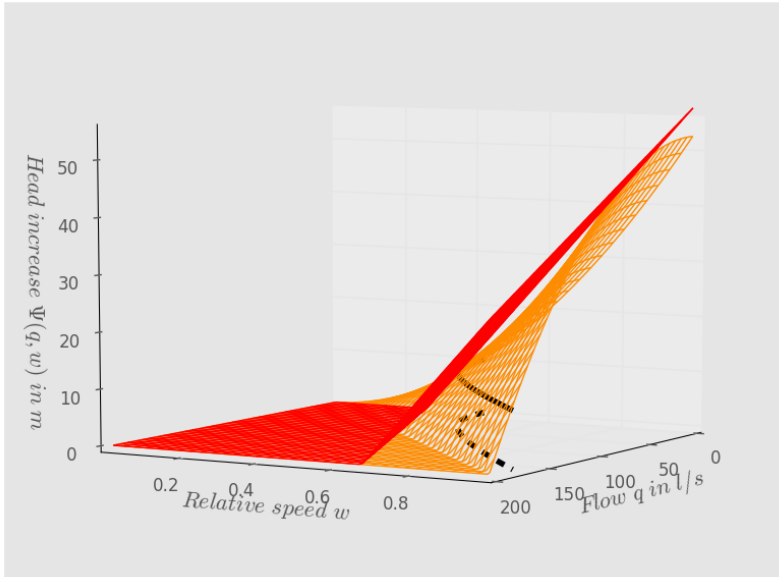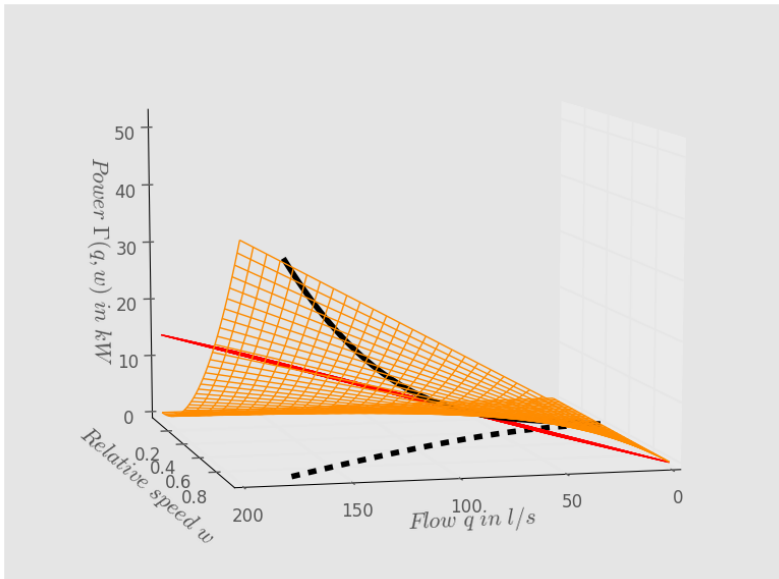
(a) Head increase

(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 $(0, 1)$.

a similar approach to relax the head-flow coupling constraints (9) through a pump $k \in K$ while limiting the head gain overestimation. As depicted in Figure 2a, we exhibit an outer approximation P_k^ϵ of the characteristic curve $\Psi_k(q, w)$ that satisfies

$$|\delta_h| \le |\Psi_k(q)| + \epsilon, \forall (q, \delta_h) \in P_k^\epsilon, w = 1.$$

Finally, as the objective function may not be linear, we exhibit a family of linear under-estimators of the power consumption Γ_k (19) of a variable-speed pump $k \in K^V$, 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 $(\mathcal{C})$:

$$\sum_{ij \in A} q_{ij} = \sum_{ji \in A} q_{ji} + d_j, \qquad \forall j \in J_J \qquad (28)$$

$$h_i - h_j = \Phi_{ij}(q_{ij}), \qquad \forall ij \in L \qquad (29)$$

$$\underline{M_k}(1 - x_k) \le h_j - h_i - \Psi_k(q_k, w_k) \le \overline{M_k}(1 - x_k), \forall k \in K \qquad (30)$$

$$\underline{M_v}(1 - x_v) \le h_i - h_j \le \overline{M_v}(1 - g(x_v)), \qquad \forall v \in V \qquad (31)$$

$$\underline{Q_k}x_k \le q_k \le \overline{Q_k}x_k, x_k \in \{0, 1\} \qquad \forall k \in K \qquad (32)$$

$$\underline{W_k}x_k \le w_k \le \overline{W_k}x_k, \qquad \forall k \in K \qquad (33)$$

$$\underline{Q_v}g(x_v) \le q_v \le \overline{Q_v}x_v, \qquad \forall v \in V \qquad (34)$$

$$\min_{1 \le t \le T}(D_{jt}) \le d_j \le \max_{1 \le t \le T}(D_{jt}), \qquad \forall j \in J_J \qquad (35)$$

$$\underline{H_j} \le h_j \le \overline{H_j}, \qquad \forall j \in J_T \qquad (36)$$

$$\min_{1 \le t \le T}(H_{jt}) \le h_j \le \max_{1 \le t \le T}(H_{jt}), \qquad \forall j \in J_S. \qquad (37)$$

where the 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 sufficiently large values. A bound computed at one iteration can be exploited to tighten the model at

the next iterations. The final result then depends on the order for evaluating the bounds. The computed bounds are valid since for any solution (q, h, x) of $(\mathcal{P})$, the steady-state configuration (q_t, h_t, x_t, d_t) at any time $t \in \mathbb{T}$ satisfies all these constraints.

In our experiments, we opted for solving these non-convex MINLPs directly with Baron, without updating the bounds between two iterations, and after adding binary variables to model the flow direction in the non-smooth constraints (29) (see (8^a)-(8^d)). However, for DWDNs of large size or with many bidirectional pipes, solving the non-convex MINLP may quickly become prohibitive. On the AT(M) network, where all the 41 pipes are bidirectional, we thus relaxed the integrality constraints and solved the NLP relaxation instead, again with Baron. Finally, a safety margin is applied to the obtained extreme values to take into account the inaccuracies of the optimal solution returned by the solver: a fixed tolerance value greater than the expected error is added to the upper bounds and subtracted from the lower bounds.

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 $\underline{Q} < \overline{Q}$:*

$$\phi \leq \begin{cases} f_{q^*}, \forall q^* \leq \min(\overline{Q}, \overline{Q}(1 - \sqrt{2})) & \text{if } \underline{Q} < \overline{Q}(1 - \sqrt{2}) \\ g_{[\underline{Q}, \overline{Q}]} & \text{otherwise} \end{cases} \tag{38}$$

$$\phi \geq \begin{cases} f_{q^*}, \forall q^* \geq \max(\underline{Q}, \underline{Q}(1 - \sqrt{2})) & \text{if } \overline{Q} > \underline{Q}(1 - \sqrt{2}) \\ g_{[\overline{Q}, \underline{Q}]} & \text{otherwise.} \end{cases} \tag{39}$$

Proof: We prove the validity of the upper bounds (38) in case 1 ($\underline{Q} < \overline{Q}(1 - \sqrt{2})$) and in case 2 ($\underline{Q} \geq \overline{Q}(1 - \sqrt{2})$); the proof for the lower bounds (39) 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, $\overline{f_{q^*}} \geq f_{Q^*}$ 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 $t \in \mathbb{T}$ to

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

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

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 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 $\Pi^{*'}_q \geq 0$). 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). $\blacksquare$

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_k^ϵ 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 $t \in \mathbb{T}$, constraint (9) is then relaxed to

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

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 depicted in Figure 2b. 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

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

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 under-estimators 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 $t \in \mathbb{T}$, two new decision variables $y_{kt}^1 \geq 0$ and $y_{kt}^2 \geq 0$ with the constraints

$$y_{kt}^1 \geq \lambda w_*^2(3w_{kt} - 2w_*), \forall w_* \in W_k, \tag{40}$$

$$y_{kt}^2 \geq \frac{b_*}{c_*}(1 - w_{kt}) - \frac{a_*}{c_*}q_{kt}, \forall w_* \in W_k. \tag{41}$$

Finally, we relax the objective function (21) to

$$\sum_{t \in \mathbb{T}} \Delta \Big(\sum_{ji \in A | j \in J_S} E_j q_{ijt} + \sum_{k \in K_V} C_t(y_{kt}^1 + y_{kt}^2) + \sum_{k \in K_F} C_t \Gamma_k(q_{kt}, x_{kt}) \Big). \tag{42}$$

24

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

As explained in Section 5, we expect that a pump schedule that is feasible for the MILP relaxation $(\mathcal{P})$ is either feasible for $(\mathcal{P})$ or there exists a pump schedule in a neighborhood, that is feasible in practice, but possibly not for our model $(\mathcal{P})$ due to the time discretization. This is particulary true for instances of class BS where pump speeds or valve pressure reductions cannot be adjusted to compensate for the small flow imbalances resulting from the violations of the tank level limits (6) and (7). These violations could be prevented by allowing to switch the active elements not only at the precise times fixed by the discretization. This motivates the following primal heuristic to turn near-feasible solutions of $(\mathcal{P})$ to feasible pump schedules, by adjusting the time step lengths. We describe it in the context of instances of class BS, i.e. without variable-speed pumps or pressure-reducing valves. Note that this heuristic is not specific to our branch and bound as it could be applied to any near-feasible pump schedule like, e.g., the solution of a piecewise-linear approximated model of $(\mathcal{P})$.

6.1 Mathematical model of the neighborhood

Given a complete assignment $X \in \{0,1\}^{(K \cup V) \times \mathbb{T}}$ of the binary variables x of $(\mathcal{P}_\epsilon)$, let $\mathtt{act}(X,t) = \{a \in K \cup V | X_{at} = 1\}$ denote the configuration of active pumps and valves during time step $t \in \mathbb{T}$. We allow configuration $\mathtt{act}(X,t)$ to be active earlier or latter (up to 1 time step, i.e. during the times steps $t-1$ or $t+1$), and for a shorter or longer duration (between the start of time step $t-1$ and the end of time step $t+1$), without preemption. We formulate the following mathematical program to compute a solution of minimum power cost in the neighborhood of X thus created:

$$(\mathcal{N}_\rho(X)) : \min_{\delta,q,h} \sum_{(t,\sigma) \in \mathbb{T} \times \Sigma} \delta_t^\sigma \Gamma^t(q_t^\sigma, X_t^\sigma) \tag{43}$$

$$s.t. \ H_{jt}^\sigma = H_{jt}, \qquad\qquad \forall j \in J_S, (t,\sigma) \in \mathbb{T} \times \Sigma \tag{3'}$$

$$\sum_{(t,\sigma) \in \mathbb{T} \times \Sigma} -Q_{jt}^\sigma \delta_t^\sigma \leq V_j, \qquad\qquad \forall j \in J_S \tag{4'}$$

$$Q_{jt}^\sigma \delta_t^\sigma = S_j(H_{jt}^{\sigma+1} - H_{jt}^\sigma), \qquad\qquad \forall j \in J_T, (t,\sigma) \in \mathbb{T} \times \{1,2\} \tag{5'}$$

$$Q_{jt}^1 \delta_t^1 = S_j(H_{j(t+1)}^1 - H_{jt}^3), \qquad\qquad \forall j \in J_T, t \in \mathbb{T} \tag{5''}$$

$$\underline{H_j} + \rho\overline{H_j} \leq H_{jt}^\sigma \leq (1-\rho)\overline{H_j}, \qquad\qquad \forall j \in J_T, (t,\sigma) \in \mathbb{T} \times \Sigma \tag{6'ρ}$$

$$H_{j0}^1 = H_{j0} \leq H_{jT}^1 - \rho\overline{H_j}, \qquad\qquad \forall j \in J_T \tag{7'ρ}$$

$$\delta_t^1 + \delta_t^2 + \delta_t^3 = \Delta, \qquad\qquad \forall t \in \mathbb{T} \tag{44}$$

$$u_{t-1}^3 + u_t^1 \leq L_t \qquad\qquad \forall t \in \mathbb{T} \setminus \{0\} \tag{45}$$

$$\delta_{t-1}^3 - \delta_{t'-1}^3 - \delta_t^1 + \delta_{t'}^1 \geq (\tau_s + t - t')\Delta, \forall s \in \{0,1\}, (t,t') \in I^s \tag{46}$$

$$0 \leq \delta_t^2 \leq \Delta, \qquad\qquad \forall t \in \mathbb{T} \qquad\qquad (47)$$

$$0 \leq \delta_t^\sigma \leq u_t^\sigma \Delta, u_t^\sigma \in \{0,1\}, \qquad \forall t \in \mathbb{T}, \sigma \in \{1,3\} \qquad (48)$$

$$u_t^\sigma \in \{0,1\}, u_0^1 = 0, u_{T-1}^3 = 0, \qquad \forall t \in \mathbb{T}, \sigma \in \{1,3\} \qquad (49)$$

$$Q_{jt}^\sigma = \sum_{ij \in A} q_{ijt}^\sigma - \sum_{ji \in A} q_{jit}^\sigma, \qquad \forall j \in J_T \cup J_S, (t,\sigma) \in \mathbb{T} \times \Sigma$$

$$(50)$$

$$(q_t^\sigma, h_t^\sigma) \in \mathcal{F}_t(X_t^\sigma, H_t^\sigma) \qquad \forall (t,\sigma) \in \mathbb{T} \times \Sigma \qquad (22')$$

In this model, each time step $t \in \mathbb{T}$ is divided into three successive substeps $(t,1)$, $(t,2)$, and $(t,3)$ where the active configurations of pumps and valves are, respectively, $\texttt{act}(X, t-1)$, $\texttt{act}(X,t)$ and $\texttt{act}(X, t+1)$. The problem is to decide the length $\delta_t^\sigma \in [0,\Delta]$ of each substep (t,σ) with $t \in \mathbb{T}$ and $\sigma \in \Sigma = \{1,2,3\}$. Binary variables u are required to prevent preemption and extra pump switches: $u_t^1 = 1$ if configuration $\texttt{act}(X, t-1)$ is extended on period t ($\delta_t^1 > 0$ and $\delta_{t-1}^3 = 0$) and, symmetrically, $u_t^3 = 1$ if configuration $\texttt{act}(X, t+1)$ is advanced on period t ($\delta_t^3 > 0$ and $\delta_{t+1}^1 = 0$). Constraints (45) - (49) enforce nonpreemption, where $X_{at}^\sigma \in \{0,1\}$ denotes the active status of $a \in K \cup V$ during substep $(t,\sigma) \in \mathbb{T} \times \Sigma$ (i.e. $X_{at}^1 = X_{a(t-1)}$, $X_{at}^2 = X_{at}$, and $X_{at}^3 = X_{a(t+1)}$), and $L_t \in \{0,1\}$ denotes a configuration switch (i.e. $L_1 = 1$ iff $\texttt{act}(X, t-1) \neq \texttt{act}(X,t)$). Constraints (46) ensure that the maintenance constraints (13)-(16) are still satisfied after adjusting the time step length, where $(t,t') \in I^1 \subseteq \mathbb{T}^2$ (resp. I^0) if $t < t'$ and at least one element $a \in K \cup V$ is on (resp. off) at t and off (resp. on) at t'. Constraint set (22') ensures a feasible flow-head configuration during each time substep.

Parameter ρ in (6'ρ) and (7'ρ) has default value 0, but we can set it to a positive value (e.g. $\rho = 10^{-3}$) to strengthen the tank head bounds, in order to improve the convergence of the iterative scheme described thereafter. Indeed, we do not solve directly the non-convex MINLP ($\mathcal{N}_0(X)$) that is almost as hard as the original problem ($\mathcal{P}$). Instead, we heuristically solve, for a limited time, a sequence of MILP approximations where the stationary hydraulic components (22') are fixed and progressively refined.

6.2   An iterative heuristic solution approach

Indeed, model ($\mathcal{N}_0(X)$) has an interesting decomposable structure: i) the arc flows q are fully determined at every substep by only one steady-state hydraulic component (22'), ii) these components are independent and only linked to the remaining scheduling subproblem through the tank heads H, iii) each hydraulic component can efficiently be solved with the Todini-Pilati method, and iv) the remaining scheduling subproblem has only linear constraints. As we are interested in a primal (not dual) solution, we do not dualize the linking constraints in a Lagrangian decomposition fashion.

Instead, we progressively solve the subproblems by fixing the linking variables to their values in a previous subproblem solution. Precisely, at one iteration of our algorithm, we successively solve the hydraulic components $(q_t^\sigma, h_t^\sigma) \in \mathcal{F}_t(X_t^\sigma, H_t^\sigma)$ at every substep $(t, \sigma) \in \mathbb{T} \times \Sigma$ for given values of tank heads H_t^σ, and the scheduling subproblem

$$(\mathcal{N}_\rho(X, q)) : \min_{\delta, H}(43) \ s.t. \ (3') - (7'\rho), (44) - (50)$$

with the arc flows $q \in \mathbb{R}^{A \times \mathbb{T} \times \Sigma}$ set at their values in the hydraulic component solutions. If a scheduling solution δ is returned, we test its feasibility regarding the neighborhood model $(\mathcal{N}_0(X))$ by running an extended period analysis (as used to solve $(\mathcal{P}(X))$ in Section 4.2): when all variables δ are fixed, $(\mathcal{N}_0(X))$ admits at most one solution that can be computed by iterating over the time substeps $(t, \sigma) \in \mathbb{T} \times \Sigma$: starting with initial tank levels $H_0^1 = H_0$, we solve (22') at one substep and compute the tank levels for the next substep according to (5') or (5"). If one bound is violated at any substep (either in (22') or among (4'), (6'ρ), (7'ρ) with $\rho = 0$), then δ is not a feasible solution of $(\mathcal{N}_0(X))$ and we start another iteration of our heuristic, with the newly computed head tank profiles. Our iterative heuristic is summarized in Algorithm 1.

As stated before, the constraints in $(\mathcal{N}_0(X))$ that are more likely to be violated at Step 3 of the algorithm by a solution δ of $(\mathcal{N}_\rho(X, q))$ are the tank capacities (6'ρ) and (7'ρ) with $\rho = 0$, and these violations will tend to vanish from an iteration of the algorithm to the other. Indeed, the flows computed at Step 3 result from a slight alteration of the tank heads computed in Step 2. So, flows and tank heads do not differ significantly between Step 2 and Step 3. Because the tank heads at Step 2 all satisfy the strongest tank limits (6'ρ) and (7'ρ) with $\rho > 0$, we expect that the tank heads at Step 3 almost satisfy the tank limits with $\rho = 0$.

Finally, we propose to run this primal heuristic at some integer nodes X of our branch and bound by initializing the tank head profiles H to their values in the optimal relaxed solution of $(\mathcal{P}_\epsilon(X))$. Hence, if the heuristic does not abort in Step 2, then it returns a feasible solution of $(\mathcal{N}_0(X))$ with a cost close to the optimum of $(\mathcal{N}_\rho(X, q))$ for some flow profile q, which is itself close to an optimal flow of $(\mathcal{P}_\epsilon(X))$. Since $(\mathcal{N}_0(X))$ is a relaxation of $(\mathcal{P}(X))$ and that all these models share the same objective functions, we may expect that the heuristic solution has a low cost.

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.

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

    Input: $(h_{jt})_{j \in J_T, t \in \mathbb{T}}$, $(X_t)_{t \in \mathbb{T}}$, $\rho \in \mathbb{R}_+$
    Initialize: $(H^1_{jt}, H^2_{jt}, H^3_{jt}) = (h_{j(t-1)}, h_{jt}, h_{jt})$ $\forall j \in J_T, t \in \mathbb{T}$
    while time limit is not reached **do**
        Step 1 (Input: H / Output: q): solve (22') independently for all $(t, \sigma) \in \mathbb{T} \times \Sigma$ with the Todini-Pilati method [54]
        Step 2 (Input: q / Output: δ): solve the approximated MILP model $(\mathcal{N}_\rho(X, q))$
        if $(\mathcal{N}_\rho(X, q))$ infeasible **then**
            break
        else
            Step 3 (Input: δ / Output: (q,h)): run the extended period analysis by solving (22') on every substep $(t, \sigma) \in \mathbb{T} \times \Sigma$, starting with the initial tank levels $H^1_0 = H_0$, and updating the tank levels for the next substep according to (5') or (5") and δ^σ_t
            if all bounds holds in (22'), (4'), (6'ρ) and (7'ρ) with $\rho = 0$ **then**
                Output: (δ, q, h) a feasible solution of $(\mathcal{N}_0(X))$
            else
                set $H^\sigma_{jt} = h^\sigma_{jt}$ $\forall j \in J_T, (t, \sigma) \in \mathbb{T} \times \Sigma$
            end if
        end if
    end while

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 [35] 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 [44] and further investigated in [10] of the extensively studied hypothetical network Anytown [57].[3] It consists of 1 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 the Belgium network operated by the water company De Watergroep considered in [56], without the extra operational

---

[3]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 with long time steps. 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 on a smaller time step.

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

Table 2: Characteristics of DWDNs instances.

constraints.[4] 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.

It is standard in pump scheduling to consider an horizon of 1 day divided in 24 1-hour steps (see, e.g., references in [31]) as it likely matches the precision for demand forecasts. To appreciate how the computational scheme responds to the time precision, 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 [48]. 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 [27] on one thread of a $2\times$ 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 [24] 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 solution scheme is stopped once reaching either the optimality gap tolerance `tol` or the overall time limit, fixed respectively to 0% and 1 hour for BS, and to 1% and 2 hours for MS.

---

[4]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.

	Day	T = 12				T = 24				T = 48			
		Best	Gap	%CB	1st	Best	Gap	%CB	1st	Best	Gap	%CB	1st
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.

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 time (in s.) to find an optimal solution or its optimality gap (in %); *%CB* is the share of time spent in the callback function; 1st 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 (26) by dropping them, i.e. by generating, at each integer node X, the constraint (26) with

			$T = 12$				$T = 24$				$T = 48$		
	Day	Best	Gap	%CB	1st	Best	Gap	%CB	1st	Best	Gap	%CB	1st
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.

$\bar{t} = T$ which cuts no other solution than X. For *Simple FSD* with $T = 24$ and *Poormond* with $T = 12$, the computational duration increased by 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 (26) are then effective and even necessary to solve the problem in some cases.

The heuristic has contrasted performances over the three DWDNs. For *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 highlighted 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 for a multiple of Δ, the heuristic allows to adjust this duration at its minimum.

We further investigated on *Poormond* our core branch and bound and, in particular, the strength of the MILP relaxation) by disabling the primal 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.

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 tolerance `tol` $= 1\%$) 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 $(1 + \mathtt{tol})L_\epsilon(X)$.

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

	Day	T = 12				T = 24				T = 48			
		Best	Gap	% CB	1st	Best	Gap	% CB	1st	Best	Gap	% CB	1st
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

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

In this section, we perform an approximate empirically comparison of our branch and bound with alternative methods proposed in the literature. Note that the codes of these alternatives are not available and that we have not reimplemented them (except in 7.3.4). Hence, our results on the test cases presented so far are put side-by-side with some results reported in previous publications and we draw some approximate empirical conclusions. For the comparison being as fair as possible, we recall, for each publication, the available details on the experimental set-up and the known differences with our models.

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.

	Day	LR [20]	BD [37]	LP/NLP	*savings*
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). Column *savings* gives the cost improvement of LP/NLP with respect to the best of LR and BD.

7.3.2 Lagrangian relaxation [20] or Benders decomposition [37] on Poormond

Ghaddar et al. [20] and Naoum-Saway et al. [37] reported results on network *Poormond* (class BS) with $T = 48$ and 96 respectively. In both papers, modeling assumptions are almost identical[5] and the only difference with ours is the shape of the pump power consumption function: we thus solved our model with $T = 48$ after computing the linear coefficients in (20) to approximate their cubic formulation, then recomputed the costs from the best solutions found. This explains the (small) difference between the cubic costs reported in Table 6 and the linear costs reported in Table 3. Bearing in mind that our method was performed on a different environment, we reported in Table 6 substantially better solutions for the five instances investigated.

Our approach bears similarities with the combinatorial Benders decomposition of [37] but it seems to benefit a lot from our tigher MILP relaxation. In [37], 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 [37] when run in the manner of a local search, loosing thus the faculty to provide performance guarantee.

7.3.3 Outer-approximation algorithm [49] on simplified versions of Poormond

The outer-approximation algorithm proposed by Shi and You [49] 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

---

[5]The authors of [20] and [37] 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.

(Hazen-William formula for ϕ_l, cubic function of the flow for Γ_k), the optimal solutions found on the 8-step horizon 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 [49] 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 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 [49], 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 [49] is automatically generated while we propose a tailored generic MILP relaxation, but both approaches have a wide range of application.

7.3.4 Solving the MINLP formulation of Simple VSD and DWG

Finally, we directly solved the non-convex MINLP formulations of the two DWDNs of class MS with Baron using the default parameters.[6] 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*, solving and proving optimality required more than 20 seconds while our method required less than 1 second. For *DWG*, no feasible solution has been found in 2 hours, while our method found at least one feasible solution for all instances in less than one hour and stopped with an optimality gap lower than 15%. Note that for $T \geq 24$ our approach was not able to compute a feasible solution for *DWG*.

---

[6]Note that [35] and [56] report approximated solutions that cannot be directly compared with ours.

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

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

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 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 solution scheme includes also several contributions such as a tailored 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. On DWDNs of class MS, the proposed method outperformed a direct application of 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 [25, 26] 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

36

Availability

The material related to the reproduction of the experiments is available under request to the second author. Parts of the code are also available through the *GOPS* (Global Optimisation for Pump Scheduling) project under the terms of the GPL license and can be downloaded at `https://github.com/sofdem/gopslpnlpbb`.

References

[1] Egon Balas and Robert 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, and Andreas Wächter. 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. `ec.europa.eu/energy/en/topics/energy-strategy/2030-energy-strategy`[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] Mordecai 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] Carlo Giacomello, Zoran Kapelan, and Matteo Nicolini. Fast hybrid optimization method for effective pump scheduling. *Journal of Water Resources Planning and Management*, pages 175–183, 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 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] 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.

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

[26] 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.

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

[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] Manuel López-Ibáñez, T.Devi Prasad, and Ben Paechter. Ant colony optimization for the optimal control of pumps in water distribution networks. *Journal of Water Resources Planning and Management*, page 337, 2008.

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

[32] Gregory McCormick and Roger Powell. Derivation of near-optimal pump schedules for water distribution by simulated annealing. *Journal of the Operational Research Society*, 55(7):728–736, 2004.

[33] 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.

[34] 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.

[35] Ruben Menke, Edo Abraham, and Ivan Stoianov. Modeling variable speed pumps for optimal pumpscheduling. In *World Environmental and Water Resources Congress*, pages 199–209, 2016.

[36] 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.

[37] 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.

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

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

[40] 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.

[41] 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.

[42] 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.

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

[44] 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.

[45] Lewis Rossman. *EPANET*, 2000.

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

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

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

[49] 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.

[50] Angus Simpson, Graeme Dandy, and Laurence Murphy. Genetic algorithms compared to other techniques for pipe optimization. *Journal of Water Resources Planning and Management*, 120(4):423–443, 1994.

[51] Piotr Skworcow, Daniel Paluszczyszyn, and Bogumil 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] Erlendur 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] Ezio Todini and Stefano 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] Jakobus van Zyl, Dragan Savić, and Godfrey Walters. Operational optimization of water distribution systems using a hybrid genetic algorithm. *Journal of Water Resources Planning and Management*, pages 160–170, 2004.

[56] Derek Verleye and El-Houssaine 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.