



HAL
open science

Strengthening mathematical models for pump scheduling in water distribution

Amirhossein Tavakoli, Valentina Sessa, Sophie Demassey

► **To cite this version:**

Amirhossein Tavakoli, Valentina Sessa, Sophie Demassey. Strengthening mathematical models for pump scheduling in water distribution. 4th International Conference on Applied Energy (ICAE2022), Aug 2022, Bochum, Germany. hal-03934746

HAL Id: hal-03934746

<https://hal.science/hal-03934746>

Submitted on 16 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Strengthening mathematical models for pump scheduling in water distribution^{1#}

Sophie Demasse¹, Valentina Sessa¹, Amirhossein Tavakoli^{1,2}

¹ CMA, Mines Paris-PSL University, Sophia-Antipolis, France

² Université Côte d'Azur, France (Corresponding Author)

ABSTRACT

Pump scheduling is a reputed demand-response tool for drinking water distribution. It is also a notorious difficult optimization problem. Mathematical programming is a privileged resolution approach, but efficiency highly depends on the strength of the considered mathematical models. This paper presents bound tightening and cut generation techniques to strengthen pump scheduling models. Numeric evaluations on both approximate and exact models show the benefit, if not the necessity, of such preprocessing.

Keywords: water networks, pump scheduling, mixed-integer nonlinear programming, preprocessing

1. INTRODUCTION

A drinking water distribution network is an energy-intensive utility with strong mitigation capacities: by intelligently managing the storage capacities of the water tanks and the nonlinear efficiency of the pumps, the energy consumption of pumping can be driven by tariff-based incentive [1,2]. Pump scheduling is the operational problem of deciding when and how to activate the pumps in a near future — the typical planning horizon is the day after — to meet the forecast demand profile at minimum cost given a dynamic electricity tariff. This optimization problem conjugates binary decisions (pump on/off status) and nonconvex constraints (hydraulic head/flow relations) and is thus notoriously hard. Besides metaheuristics, mathematical programming is a popular resolution approach, as it is able to assess the quality of the computed solutions [1,3]. When based on a monolithic — more or less accurate — formulation of the problem [2,4,5], it is also easily implemented on top of any off-the-shelf mixed-integer linear programming (MILP) solver. The underlying branch-and-bound algorithm relies on the iterative resolution of the linear programming (LP) relaxation, and its efficiency heavily depends on the strength and the size of this relaxation: too loose and the number of iterations increases, too large and the computational time for each iteration increases. As illustrated in Fig. 1 (left), a nonconvex constraint can be relaxed linearly: given the bounds on the implied

variables, the strength of the relaxation is directly linked to the tightness of these bounds. Preprocessing the mathematical model with bound tightening (BT) techniques [7,8] is thus of the utmost importance. Modern solvers run efficient systematic routines, but a deeper reduction is expected from problem-specific reasoning. The main issue is to figure out sophisticated, but fast approaches in order to balance the additional time for preprocessing with the expected time reduction of the optimization process.

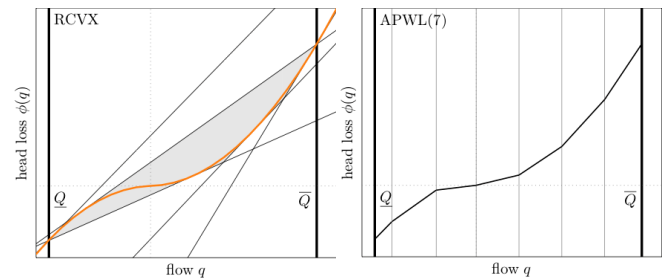


Fig. 1 Left: a nonconvex constraint (orange) and a convex relaxation RCVX (gray) defined by linear constraints within the variable bounds Q , \bar{Q} . Right: a piecewise linear approximation (APWL) defined with additional binary variables.

In this paper, we explore different techniques to be used in a preprocessing phase, both to tighten the variable bounds and to generate new valid constraints. We then present an empirical study to evaluate the impact of this preprocessing in a range of contexts: simple and complex networks, large and fine grain planning horizons, relaxed and approximate models, and exact and approximate approaches. Our experimental results highlight the importance of preprocessing in the different contexts, allowing the solvers either to compute feasible solutions in a reasonable computational time for the hardest instances or to significantly accelerate their convergence to near-optimal solutions, if not optimal. This work aims at emphasizing that the direct application of off-the-shelf MILP solvers leads to practical and competitive approaches for pump scheduling optimization when suitable BT techniques are enforced in a preprocessing step. To the best of our knowledge, the valid inequalities are novel in the context of water distribution and the iterative bound tightening technique by considering

^{1#} This is a paper for the 14th International Conference on Applied Energy - ICAE2022, Aug. 8-11, 2022, Bochum, Germany.

multi-period relaxation has not been appeared in the literature.

The next section presents a standard formal definition of the problem as a mixed-integer nonlinear program. Section 3 presents two preprocessing techniques, both for bound tightening and for constraint generation. Section 4 shows the experimental results. Concluding remarks and hints for future research are presented in the last section of this paper.

2. THE PUMP SCHEDULING MODEL

We briefly present a mathematical model for the pump scheduling problem. More details can be found, e.g., in [4, 5]. A water distribution network can be formalized as a directed graph $G = (J, A)$: arcs $a \in A$ are either pipes $a \in A_L$ or pumps $a \in A_K$, and nodes $j \in J$ represent either sources $j \in J_S$, or service nodes (also called junctions) $j \in J_J$, or water tanks $j \in J_T$. The scheduling horizon is discretized in typically $T = 24$ hourly time steps: $t \in \mathbb{T}$, with $\mathbb{T} = \{0, \dots, T - 1\}$. Some arcs are controllable, namely fixed-speed pumps and pipes equipped with gate valves, and have two possible states (i.e., on or off) at each time instant t . This is formulated as a boolean variable: $x_{at} = 0$ if arc $a \in A$ is inactive (i.e., flow cannot pass) and $x_{at} = 1$, otherwise. Each arc $a \in A$ between the nodes i and j is also associated with a function ϕ_a , which relates the flow $q_{at} \in \mathfrak{R}$ with the head (potential) loss $v_{at} = h_{it} - h_{jt} \in \mathfrak{R}$ at time $t \in \mathbb{T}$. The head h_{jt} at a water tank j is proportional to the filling level, which is limited by the tank capacity. The pump scheduling problem can be stated as:

$$(P): \min \sum_{\mathbb{T}} \sum_{A_K} c_{at}^0 x_{at} + c_{at}^1 q_{at} \\ \text{s.t.}:$$

$$q_{jt} = D_{jt}, \forall j \in J_L, t \in \mathbb{T} \quad (1)$$

$$h_{jt+1} = h_{jt} + \sigma_j q_{jt}, \forall j \in J_T, t \in \mathbb{T} \quad (2)$$

$$q_{at} \geq \underline{Q}_{at} x_{at}, \forall a \in A, t \in \mathbb{T} \quad (3)$$

$$q_{at} \leq \overline{Q}_{at} x_{at}, \forall a \in A, t \in \mathbb{T} \quad (4)$$

$$v_{at} \geq \phi(q_{at}) + \underline{V}_{at}(1 - x_{at}), \forall a \in A, t \in \mathbb{T} \quad (5)$$

$$v_{at} \leq \phi(q_{at}) + \overline{V}_{at}(1 - x_{at}), \forall a \in A, t \in \mathbb{T} \quad (6)$$

$$\underline{H}_{jt} \leq h_{jt} \leq \overline{H}_{jt}, \forall j \in J_T, t \in \mathbb{T} \quad (7)$$

$$x \in X, x \in \{0, 1\}^{A \times \mathbb{T}} \quad (8)$$

The objective function is to minimize the electricity cost of pumping and the constraints describe, at each time step: (1) demand satisfaction at junctions j where $q_{jt} = \sum_{a \in A} q_{at}$ is the residual flow, (2) flow conservation at tanks, (3)-(4) flow bounds w.r.t. arc status, (5)-(6)

potential-flow relation w.r.t arc status, (7) tank capacities, (8) any additional linear condition on the controllable arcs (e.g., dependencies between parallel pumps or usage limits). Model (P) is a mixed-integer nonlinear program where disjunctions on the arc activity are modeled in constraints (2)-(5) with the so-called ‘big-M’ values: $\underline{Q} \leq \overline{Q}$ denote bounds on the flows q in active arcs, and $\underline{V} \leq \overline{V}$ denote bounds on the head loss v in inactive arcs.

In this paper, we consider solving (P) with an approximate method, as proposed e.g., in [2], and an exact method, proposed in [4]. Both are based on solving a MILP and using a hydraulic simulator to attempt to recover the feasibility of the MILP solutions w.r.t. (P). The two MILPs are built from (P) and differ in the reformulation of the nonconvex functions ϕ in constraints (5)-(6) as depicted in Fig. 1: the exact method uses the convex relaxation RCVX (left) and the approximate method uses the piecewise-linear approximation APWL (right). Each delimiting plane is defined by a linear constraint, as well as, in APWL, an additional binary variable. The speed of resolution of a MILP mostly depends on the strength of its LP relaxation (obtained by relaxing the integrality constraints (8)). We denote with R the feasible set of the LP relaxation, either of APWL or RCVX. The strength of R is, in turn, closely tied to the tightness of the variable bounds figuring in constraints (3)-(7), as well as to the number of binary variables which can be fixed a priori to 0 or 1.

3. OPTIMIZATION-BASED BOUND TIGHTENING (OBBT)

In this section, we present different bound tightening techniques for preprocessing the LP relaxation R. These are feasibility-oriented and optimization-based, i.e., they rely on minimizing/maximizing a variable on a suited (possibly nonconvex) relaxation of (P) to derive lower/upper bounds of the variable. For any such OBBT auxiliary problem, we denote with O its relaxed feasible set, as opposed to the relaxed set R. The choice of relaxation O is very sensitive: too tight and solving one auxiliary problem may be as hard as solving (P), too loose and it provides no new information to R that the optimization process cannot infer alone.

3.1. Steady-state relaxation with floating demand

The tightness of big-M values – here the bounds on variables q and v in constraints (3)-(6) – is known to have a major impact on the strength of R. Suited to these steady-state flow/potential variables, a small-size and accurate relaxation for O is obtained by ignoring in (P) all the time steps except one. Such a steady-state relaxation is considered in [4, 8] where demand D_{jt} in (2) is allowed to vary between some known minimum and maximum values.

Since the arc flow bounds $\underline{Q}_{at}, \overline{Q}_{at}$ are active in constraints (3)-(4) only if arc a is active at time t , we also enforce condition $x_{at} = 1$ in O when computing these bounds: if O is infeasible under this condition, then arc a cannot be active and variables x_{at} can be turned into constant 0 in R. Similarly, the head-loss bounds $\underline{V}_{at}, \overline{V}_{at}$ in constraints (5)-(6) are computed by enforcing $x_{at} = 0$ in O, and x_{at} can be fixed to 1 in R in case of infeasibility. This principle, known as probing [11], allows reducing both the big-M values and the number of binary variables in R. This preprocessing is very fast, as it consists in solving at most 4 small nonconvex MINLPs (2 bounds on 2 variables) for each arc. In practice, the minimum and maximum demand values in O can be obtained from the historical data of the given water network, making this preprocessing instance-independent.

3.2. Steady-state relaxation with fixed demand profiles

Still, probing and strong bound reduction are unlikely to occur without considering the exact demand profile D for a given instance. Given D , we can actually apply procedure of Section 3.1 to each time step t , independently, by enforcing in O the original constraints (2) with fixed demand values. The nature and size of the OBTT problems do not change. Still, their number is multiplied by T . One trivial improvement is to aggregate the time steps with the same demand, making this preprocessing still fast for the considered network sizes.

3.3. Multi-period relaxation for state variables

One major feasibility issue in (P) comes from the tank capacities in constraints (7), as tight bounds $\underline{H}, \overline{H}$ on the water level in the tanks may have a great impact on the strength of both R and the steady-state relaxations. OBTT should then also be applied to these variables h and the whole procedure – including the one described in Section 3.2 – be iterated so that any bound improvement is propagated by updating O in the subsequent OBTT problems. However, the steady-state relaxation is not relevant for applying OBTT to variables h since the level at time t directly depends on the decisions made at $t - 1$ and before. We thus consider a multi-period relaxation O instead. The period length (from 2 to T) and the strength of O (with or without nonconvex or integrality constraints) must be chosen carefully w.r.t. the complexity of the water network and the criticality of these bounds. Employing a loose relaxation may still benefit the complete OBTT procedure, by propagation.

3.4. Probing on related network elements

We can apply the values obtained by probing techniques to related network elements. For instance, if a branch is served by one pump, we can strengthen constraints (7) by

$$h_{jt} \geq \underline{H}_{jt}^1 x_{at} + \underline{H}_{jt}^0 (1 - x_{at}) \quad (9)$$

where $\underline{H}_{jt}^1, \underline{H}_{jt}^0$ are the lower bounds on the level of a tank j computed conditionally to the status of the pump a serving it.

Similarly, we can consider another usual scheme: a pipe outgoing from a pumping station made of interdependent pumps. By examining the effect on the pipe flow of activating each possible combination of pumps separately, we may observe disjoint operation intervals. We can then convexify the disjoint regions for flow and head loss following techniques from disjunctive programming [12] in order to strengthen constraints (3)-(6). Note that computing the conditional bounds on the pipe appearing in these disjunctive constraints does not ask for running additional OBTT in the preprocessing, when it can directly be inferred from the conditional bounds on the pumps, from the pump interdependencies, and from flow conservation.

3.5. Pushing the objective with Mixed-Integer Rounding

OBTT applies not only to decision variables y but also to any (linear) variable composition $f(y)$. The resulting bounds are then enforced in (P) with additional linear constraints, e.g., $f(y) \geq \underline{F}$. This procedure must be reserved for some promising compositions. We applied it to the minimum total of active pumps $\sum_{a \in A, t \in \overline{\mathbb{T}}} x_{at}$ with $\overline{\mathbb{T}} \subset \mathbb{T}$. The intuition behind this choice is: (i) pushing up this lower bound mechanically increases the objective cost on R, (ii) limiting the period length in O (to $[0, \tau]$ or $[\tau, T]$) ensures its tractability, (iii) less flexibility is given to O when restricted to those intervals as the level of the tanks is fixed at times 0 and T , (iv) since the number of active pumps (corresponding to $f(y)$ here) is an integer, we can apply a Mixed-Integer Rounding technique [11] to lift the constraint above as follows:

$$f(y) \geq \frac{1}{\underline{F} - \lfloor \underline{F} \rfloor} G(y - y^*) + \lfloor \underline{F} \rfloor, \quad (10)$$

where y^* denotes an optimal solution of the OBTT problem $\min_{y \in O} f(y)$, $\underline{F} = f(y^*)$ is the optimum value, and G is the gradient of f at y^* . If relaxation O is an LP, then gradient G is directly derived from an optimal dual solution.

4. COMPUTATIONAL EXPERIMENTS

We empirically evaluate and compare the effect of the BT techniques and inequalities on a benchmark set [4] made of two networks – one reportedly easy (S='Simple FSD' with 3 parallel pumps, 1 tank) and one reportedly hard (P= real network 'Poormond' with 7 pumps, 4 gate valves, 5 tanks) – and on 3 horizon lengths ($T = 12, 24, 48$). Results below are given for each set

(denoted as, e.g., S24 or P48) as the average of the values obtained for the 5 instances. All algorithms were implemented in Python using the LP and MILP default solvers from Gurobi 9.1.1 and executed on a personal computer (Intel(R) Core(TM) i7-10810, 6 cores at 1.6 GHz, 32GB). The details on the test instances, algorithms, parameters, and results can be found at: <https://github.com/sofdem/gopslpnlpbb/tree/icac22>.

The variable bounds were initialized using the procedure described in Section 3.1. We use these non-trivial bounds (called B0) as the baseline to start the OBBT procedure described in Sections 3.2 to 3.4 (called B1) and to evaluate their impact. Precisely, in B1, we solve the auxiliary optimization problems in turn up to 3 consecutive times within a given time limit. For the multi-period relaxation O in Section 3.3, the RCVX MILP relaxation has been considered for instances of network S, and we relaxed the integrality constraints and restricted the horizon to 6 periods for network P.

4.1. Impact on feasibility in the approximated model

In Table 1, we show the effect of the preprocessing B1 on the tightness of the MILP approximation APWL and on the feasibility of its solutions w.r.t. (P). We measure the feasibility gap for a given pumping schedule as the accumulated absolute difference between the tank levels computed by the hydraulic simulator on this schedule and the approximated ones. The fitting of the nonlinear functions ϕ are chosen according to [2], with either 3 or 7 pieces, leading to two MILP models, namely APWL3 and APWL7. Unsurprisingly, APWL7 in time limit of 1 hour provides more accurate solutions (feasibility gap is 63% smaller in average) but it shifts the difficulty of obtaining near-feasible solutions to finding any approximate solution within the time limit for the larger instances, e.g.: APWL3 provides a feasible solution for the 5 instances of P48, whereas APWL7 only for 2. Preprocessing alleviates this difficulty: both models find feasible solutions for almost all instances with bounds B1, but for only half of them with bounds B0. For those instances where finding feasible solutions is not a critical issue, B1 also helps to improve the quality of the solutions: the best cost declines 1.4% for S24 and 0.7% for P48. Furthermore, the number of feasible solutions is significantly higher (111%), giving more versatility to the decision-making.

4.2. Impact on optimality in the exact method

We assess the effect of preprocessing and cut generation on the exact method discussed in [4] based on the MILP relaxation RCVX. For unbiased comparison, we let branch-and-bound run for 1 hour on model RCVX built by using bounds B1, say RCVX1, and on model RCVX including also the constraints (9)-(10), say RCVX2,

described in Sections 3.4-3.5. Finally, we let the branch-and-bound algorithm run for an additional hour (i.e., the time limit for preprocessing B1) on model RCVX built with bounds B0, say RCVX0.

The three models solve the smallest and simplest instances S24 in a few seconds, but preprocessing speeds up the optimality proof thanks to a significant 59% reduction of the size of the search tree. As shown in Table 2, the search reduction and consequent speed-up are even more noticeable on the largest instances S48, as preprocessing appears necessary to close the optimality gap: RCVX0 proves optimality for only 1 instance, against 3 and 4 instances respectively for RCVX1 and RCVX2. In the larger network P, the number of nonconvex constraints is considerably higher even for the shortest horizon $T = 12$. As a consequence, solving exclusively MILP relaxation RCVX0 on P12 is possible in a few minutes, whereas the exact method applied to this relaxation does not even find a feasible solution in 2 hours. By mitigating the accuracy of the relaxation, preprocessing allows the exact method with model RCVX1 not only to find feasible solutions for all instances but also to close the optimality gap in one case. With model RCVX2, we were able to solve 4 among 5 instances to their optimality and shrink the gap for the remaining instance from 5.7% to 1.4%. Finding a feasible solution is less an issue in problems P24 and P48, as shorter time steps provide more flexibility to the scheduling task. Still, the method with model RCVX0 takes an hour to obtain the first feasible solution, and it leaves one instance in P24 without any solution. Preprocessing in model RCVX1 allows finding a feasible solution also on this instance, and it reduces the optimality gap from 3.7% to 2.8% on average for the 9 other instances. The additional constraints in model RCVX2 allow shrinking the gap to 2.3% and to get the first feasible solutions in just a few minutes. It is worthy to highlight here that our approach improved the results presented so far in the literature for the Simple and Poormond networks.

5. CONCLUSIONS

This work presents different preprocessing techniques for the pump scheduling problem and gives insights into balancing the preprocessing time and its effect. The results show a solid incentive to employ problem-specific preprocessing for the drinking water distribution network. Therefore, we claim that additional improvements would be possible with more sophisticated, even if time-consuming, preprocessing. This observation demands some techniques for applying an aggressive bound tightening and cut generation to a smartly selected subset of network elements and time steps.

Table 1: Results of solving the APWL MILP model with 3 or 7 pieces, with bounds B0 and B1 on 5 benchmark sets of 5 instances each (#feas: number of instances for which at least 1 feasible solution is computed, avg_feas: the average feasibility gap w.r.t. the tank capacities, #OPT: number of instances solved within a 2% optimality gap, #sol: total number of feasible solutions).

		S24	S48	P12	P24	P48
APWL3_B0	#feas	1	4	0	0	5
	avg_feas	8.3	13.3	3.5	5.9	11.9
	#OPT	5	5	5	3	0
	#sol	2	18	0	0	38
APWL3_B1	#feas	5	5	1	2	5
	avg_fea	1.4	12.4	0.9	2.3	4.7
	#OPT	5	5	5	3	0
	#sol	19	16	1	4	49
APWL7_B0	#feas	2	5	0	0	2
	avg_fea	3.6	3.8	1.2	2.9	6.0
	#OPT	5	5	5	0	0
	#sol	4	17	0	0	7
APWL7_B1	#feas	4	5	2	4	5
	avg_fea	0.8	3.3	0.2	1.1	1.1
	#OPT	5	5	5	0	0
	#sol	11	22	4	19	45

Table 2: Results of the exact method [4] applied to the convex MILP relaxation without preprocessing (RCVX0), after preprocessing B1 (RCVX1), and with additional constraints (RCVX2) (#feas: number of instances for which at least 1 feasible solution is computed, avg_gap: the average optimality gap over instances, #OPT: number of instances solved within an 1e-6% optimality gap).

		S24	S48	P12	P24	P48
RCVX0	#feas	5	5	0	4	5
	avg_gap	0	0.5%	inf	4.1%	3.2%
	#OPT	5	1	0	0	0
RCVX1	#feas	5	5	5	5	5
	avg_gap	0	0.2%	2.3%	2.8%	2.8%
	#OPT	5	3	1	0	0
RCVX2	#feas	5	5	5	5	5
	avg_gap	0	0.0%	0.3%	2.4%	2.1%
	#OPT	5	4	4	0	0

ACKNOWLEDGEMENT

This work has been supported by the French government, through the 3IA Côte d'Azur Investment in the Future project managed by the National Research Agency (ANR) with the reference numbers ANR-19-P3IA-0002.

REFERENCES

[1] Ghaddar, B., Naoum-Sawaya, J., Kishimoto, A., Taheri, N., & Eck, B. (2015). A Lagrangian decomposition approach for the pump scheduling problem in water networks. *European Journal of Operational Research*, 241(2), 490-501.

[2] Menke, R., Abraham, E., Parpas, P., & Stoianov, I. (2016). Exploring optimal pump scheduling in water distribution networks with branch and bound methods. *Water Resources Management*, 30(14), 5333-5349.

[3] Naoum-Sawaya, J., Ghaddar, B., Arandia, E., & Eck, B. (2015). Simulation-optimization approaches for water pump scheduling and pipe replacement problems. *European Journal of Operational Research*, 246(1), 293-306.

[4] Bonvin, G., Demasse, S., & Lodi, A. (2021). Pump scheduling in drinking water distribution networks with an LP/NLP-based branch and bound. *Optimization and Engineering*, 22(3), 1275-1313.

[5] Burschweiger, J., Gnädig, B., & Steinbach, M. C. (2009). Optimization models for operative planning in drinking water networks. *Optimization and Engineering*, 10(1), 43-73.

[6] Puranik, Y., & Sahinidis, N. V. (2017). Domain reduction techniques for global NLP and MINLP optimization. *Constraints*, 22(3), 338-376

[7] Belotti, P., Lee, J. et al. (2009). Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5), 597-634.

[8] Tasseff, B. (2021). *Optimization of Critical Infrastructure with Fluids* (Doctoral dissertation).

[9] Caseau, Y., & Laburthe, F. (1996, September). Cumulative Scheduling with Task Intervals. In *JICSLP* (Vol. 96, pp. 369-383).

[10] Balas, E. (1998). Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1-3), 3-44.

[11] Günlük, O., & Pochet, Y. (2001). Mixing mixed-integer inequalities. *Mathematical Programming*, 90(3), 429-457.