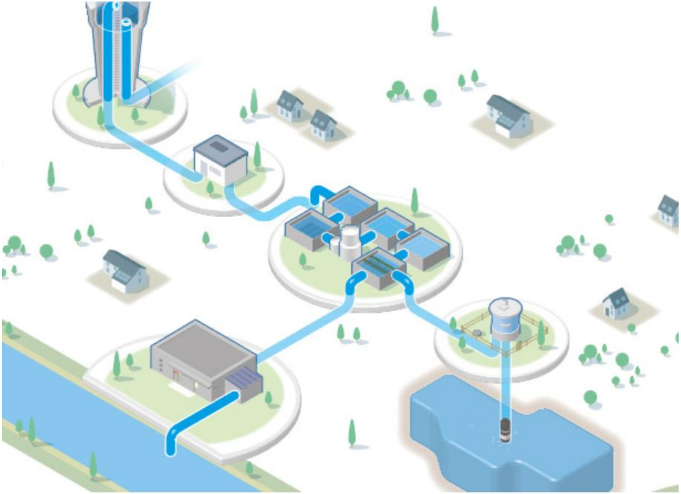
Combining Deep Learning and Variable Splitting for Pump Scheduling

A. Tavakoli, S. Demassey, V. Sessa

20.10.2023

Overview

Drinking water network distribution



Problem definition

Water network distribution: a *nonlinear potential-driven flow network*, represented as a directed graph $G = (\mathcal{J}, \mathcal{A})$
nodes $\mathcal{J}$:

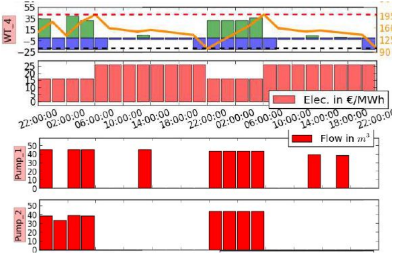
- tanks and reservoirs $\mathcal{C}$
- demands $\mathcal{S}$

arcs $\mathcal{A}$:

- pumps and valves, control arcs: $\dot{A} = \mathcal{A}_K \cup \mathcal{A}_V$
- pipes: $\mathcal{A}_L$

Pump Scheduling Problem

Pump scheduling problem: planning pumps and valves operations over a discretized time horizon $\forall t \in \mathcal{T} = \{0, 1, ..., T-1\}$, ($x_{ta} \in \{0,1\} \forall a \in \dot{\mathcal{A}}\ t \in \mathcal{T}$), at a minimum operation cost, given a water demand (D_{ts}) and an electricity tariff C_t.



Pump scheduling problem is solved routinely, it is reasonable to assume we can have a big collection of solved instances for each network

Mathematical formulation

$$
\begin{aligned}
(\mathcal{P}): \min_{x,q,H} \ & \sum_{t\in\mathcal{T}} c_t(x_t, q_t) = \sum_{t\in\mathcal{T}} \sum_{a\in\dot{\mathcal{A}}} (c_t^0 x_{ta} + c_t^1 q_{ta}) && (1a)\\
\text{s.t: } & (q_t, h_t) \in \mathcal{E}(H_t, D_t, x_t), && \forall t \in T \quad (1b)\\
& q_{tj}^{\dot{\mathcal{C}}} = \sigma_j(H_{(t+1)j} - H_{tj}), && \forall j \in \dot{\mathcal{C}}, \forall t \in \mathcal{T} \quad (1c)\\
& \underline{H}_{tj} \leq H_{tj} \leq \overline{H}_{tj}, && \forall j \in \dot{\mathcal{C}}, \forall t \in \mathcal{T} \cup \{T\} \quad (1d)\\
& x_t \in \mathcal{X}_t \subseteq \{0,1\}^{\mathcal{A}} && \forall t \in \mathcal{T}. \quad (1e)
\end{aligned}
$$

x_t: denotes the status of arcs
H_t: denotes the levels of the tanks
D_t: demand profile
q_t: flows passing through arcs
$q_{tj}^{\dot{\mathcal{C}}}$: inflow of the tanks
h_t: head drop at each arc
$\mathcal{E}$: network analysis problem (NAP)

Mathematical formulation

network analysis problem $\mathcal{E}$ for each time step:

$$(q_t, h_t) \in \mathcal{E}(H_t, D_t, x_t), \qquad \forall t \in \mathcal{T}$$

compact representation of conservation of flow, capacity of the arcs, and head-flow relationship in arcs

$$q_{tj} = D_{tj} \qquad \forall j \in \mathcal{S}$$
$$\phi_a(q_{ta}) + \underline{V}_{ta}(1 - x_{ta}) \leq h_{ta} \leq \phi_a(q_{ta}) + \overline{V}_{ta}(1 - x_{ta}) \qquad \forall a \in \mathcal{A}$$

in general: $\phi(q) = Aq|q| + Bq + C$ nonlinear and nonconvex constraint

for a given D_j and H_t, for each configuration x_t we have a unique flow through arcs and correspondingly head drops

Caveats

The optimization problem $\mathcal{P}$ has three major issues:

- **discrete**: binary decisions $x \in \{0,1\}^{|A^C| \times T}$
- **non-convex**: pressure-flow relation $h = \phi(q)$
- **large scale**: network size $\times T$

$\mathcal{P}$ is a large scale non-convex Mixed-Integer Non-linear Program (MINLP)

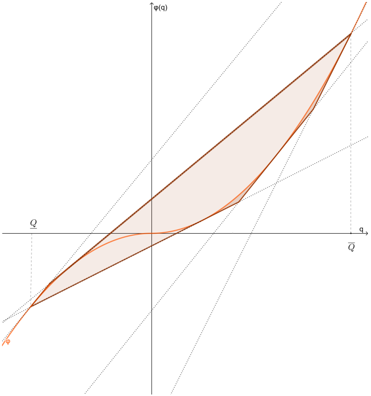
how to tackle this problem?

solve/approximate $\mathcal{P}$

- evolutionary search, e.g., metaheuristics
- Systematic search, e.g., branch-and-cut applied to a more tractable formulation, typically a MILP, Lagrangian relaxation, etc

flow-head relationship in arcs

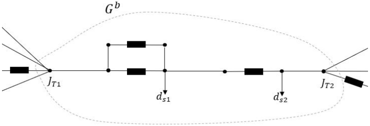
$\phi(q) = Aq|q| + Bq + C$ and its relaxation in approximation or branch and check algorithm (Bonvin et al)



Constructive properties

- **Temporal decomposition**: flow and pressure in arcs at t' are described by a nonlinear function g:
$$(q_t, h_t) \in \mathcal{E}(x_t, D_t, H_t)$$
by relaxing levels of the tanks $H_{t'j} \in [\underline{H}_{t'j}, \overline{H}_{t'j}] \forall j \in \dot{\mathcal{C}}$, $(q_{t'}, h_{t'})$ are independent of decision made at $t \neq t'$.

- **Graph decomposition**: decomposition of the network graph G along reservoirs into subgraphs G_b for branch b: $A_b \cap A_{b'} = \emptyset$, $J_b \cap J_{b'} \subseteq R$, (q_t, h_t) in branch b is fully independent from $G \setminus G_b$.



Alternating Direction Method

ADM:solving complex optimization problems by decomposing them into more manageable sub-problems

the classical ADM is a solution method for optimization problems with a clear partition of the decision variables in two sets:

$$P : \min\{f(u,v) : u \in \mathcal{U}, v \in \mathcal{V}\}.$$

The main steps of ADM are presented in the following algorithm.

Algorithm Standard Alternating Direction Method

Input: choose the initial values (u^0, v^0)
Output: a partial minimum solution (u^*, v^*)
for $k = 0, 1, \ldots$ **do**
  Compute: $u^{k+1} \in \arg\min_u\{f(u, v^k) : u \in U\}$
  Compute: $v^{k+1} \in \arg\min_v\{f(u^{k+1}, v) : v \in V\}$
  if $\|u^{k+1} - u^k\| < \varepsilon$ and $\|v^{k+1} - v^k\| < \varepsilon$ **then**
    break
  end if
end for

nonseparable Alternating Direction Method

ADM can be extended for nonseparable and quasi-separable cases by slightly changes in the formulations. This can be realized through:

- Alternating Direction Method of Multipliers (ADMM)
- Penalty Alternating Direction Method (PADM)

ADMM: well established work, requires linear coupling constraints to converge to feasible solution (Boyd et al)
PADM: develop initially for large scale network optimization (Geißler et al.).
convergent partial minimum of (P_{PADM}) concept may violate some coupling constraints; On the other hand, the convergence proof in the PADM framework requires weaker assumptions:
as long as the solution of one of the subproblems is always unique, the inner loop converges to a partial minimum.

Penalty Alternating Direction Method

suppose that coupling constraints can be represented as $g_i(u, v) = 0 \ \forall i$ and $h_{i'}(u, v) \geq 0 \ \forall i'$. The PADM is defined as:

$$P_{PADM} : \min\{f(u, v) + \sum_{i=1}^{m} \rho_i |g_i(u, v)| + \sum_{i=1}^{p} \mu_i [h_i(u, v)]^- : u \in \mathcal{U}, v \in \mathcal{V}\}.$$

and

$$\phi(u, v, \rho, \mu) = \min\{f(u, v) + \sum_{i=1}^{m} \rho_i |g_i(u, v)| + \sum_{i=1}^{p} \mu_i [h_i(u, v)]^-\}$$

denotes the objective function of the PADM

PADM algorithm

Algorithm Standard Penalty Alternating Direction Method

Input: choose the initial values $(u^0, v^0, \rho^0, \mu^0)$
Output: a partial minimum solution (u^*, v^*)
for $j = 0, 1, \dots$ **do**
    for $k = 0, 1, \dots$ **do**
        Compute: $u^{k+1} \in \arg\min_u \{\phi(u, v^k, \rho^j, \mu^j) : u \in U\}$
        Compute: $v^{k+1} \in \arg\min_v \{\phi(u^{k+1}, v, \rho^j, \mu^j) : v \in V\}$
        if $\|u^{k+1} - u^k\| < \varepsilon$ and $\|v^{k+1} - v^k\| < \varepsilon$ **then**
            break
        end if
    end for
    choose a new penalty parameter $\mu^{j+1} \geq \mu^j$ and $\rho^{j+1} \geq \rho^j$
end for

if the solution of one of the subproblem is always unique, then always we have convergence in the inner loop.
The convergence condition is by far easier to be met; however, the concept of partial minumum is rather weak and its framework it deos not mean feasible

Variable splitting in pump shceduling problem

$(\mathcal{P})$ can be decomposed temporally after dualizing the storage time-coupling constraints as in Lagrangian relaxation, or after penalizing their violations as in the following model. Given ℓ_1 penalty and multipliers, we define

$$(L_\rho): \min_{x,q,H} \{l(x,q,H,\rho) : (1b), (1d), (1e)\},$$

$$\text{with } l(x,q,H,\rho) = \sum_{t \in T} \Big(c_t(x_t, q_t, H_t) + \sum_{j \in \dot{\mathcal{C}}} \rho_{tj} d_{tj}(q,H) \Big),$$

$$\text{and } d_{tj}(q,H) = |H_{(t+1)j} - (H_{tj} + q_{tj}^{\dot{\mathcal{C}}})| \quad \forall t \in T, j \in \dot{\mathcal{C}}.$$

if separable as T independent subproblems, it remain difficult to solve solving each subsystem $\mathcal{E}(H_t, D_t, x_t)$ when H_t is variable is hard

Variable splitting in pump scheduling problem

Assumption (A0). The steady-state sub-problems with known initial state variables(levels of the tanks)

$$(\mathcal{P}_t(H_t)) : \min_{x_t, q_t} \{ f_t(x_t, q_t) \mid (q_t, h_t) \in \mathcal{E}(H_t, D_t, x_t), x_t \in \mathcal{X}_t \subseteq \{0,1\}^{\mathcal{A}} \}$$

are easy for all $t \in \mathcal{T}$ and $H_t \in \mathbb{R}^{\dot{\mathcal{C}}}$, and for any linear function f_t.

Variable splitting algorithm

Algorithm Partial storage/control splitting for $(\mathcal{P})$

1: **Input:** $i = 0$, tank profiles $H^0 \in \mathbb{R}^T$, penalty ρ^0, tolerance $\varepsilon, \varepsilon' > 0$
2: **Output:** a feasible solution (x, q, H) of $(\mathcal{P})$
3: **for** $k = 0, 1, \ldots$ **do**:
4:     **while** $\|H^{i+1} - H^i\|_\infty \geq \varepsilon'$ **do**:
5:         $(x^{i+1}, q^{i+1}) \in \arg\min_{(x,q)}\{l(x, q, H^i, \rho^k) : (1b), (1e)\}$
6:         $H^{i+1} \in \arg\min_H\{l(x^{i+1}, q^{i+1}, H, \rho^k) : (1d)\}$
7:         **if** $d_{tj}(q^{i+1}, H^{i+1}) < \varepsilon \; \forall t \in T, j \in$ **then**
8:             **return** $(x^{i+1}, q^{i+1}, H^{i+1})$
9:         **end if**
10:         $i \leftarrow i + 1$
11:     **end while**
12:     update ρ
13: **end for**

we do not dualize/penalize here the nonlinear coupling constraint (1b). Instead, we propose to relax the coupling equilibrium constraints (1b) in the second subproblem (Line 6) and keep the storage capacities (1d) as the only

Variable Splitting algorithm

- first subproblem: for a fixed $H_t = H_t^i$, we find the optimal solution for each single step penalized problem $(\mathrm{P}_t(H_t))$
  (even enumeration!)
- second subproblem: solving an LP

The major deficiency of such algorithm is that there is no guarantee to end up a feasible solution

hypothesis: given relatively close optimal tank profile H for initialization is likely to end up finding the optimal configuration

Observation: Initialization of the decomposition with arbitrary state variables almost never end up to feasible solution

Decomposition guided by a learning algorithm

- Optimal operation of the water network distribution carries out at daily basis.
- In practice at each day for a given demand and tariff profile (D, C) the optimization problem is solved

A supervised learning problem.
a hypothesis function $\mathcal{H}$: $(D, C) \in \mathbb{R}^{TS} \times \mathbb{R}^{T} \to H(D, C) \in \mathbb{R}^{TR}$ an optimal solution of the problem $(\mathcal{P})$ with input (D, C). minimizing the regression loss:

$$\mathcal{L}_{loss}(\mathcal{H}(D, C), H_{(D,C)}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathcal{H}(D_i, C_i) - H_i||_2^2.$$

Idea: providing a near optimal tank profile for the decomposition algorithm to search locally around approximated state profile

Decomposition guided by a learning algorithm

- the approximation of the optimal state variable here resembles learning sequential data
- using approximation of Bayesian neural network (Monte Carlo dropout) to have several approximation for each given input data: multi start $\rightarrow$ more chance to find optimal solution
- Instead of approximating the discrete control variables, approximating the continuous variables
  - allowing smoother moves in the neighborhood exploration
  - multi-start mechanism
  - fixing the levels in the first subproblem and regaining tractability
- Scaling mechanism: to tackle scheduling problems with high resolution for which collecting pre-solved instances is very time consuming; by doing this we mitigate the limitation of supervised learning

Experimental results

van Zyl network (van Zyl et al) is characterized with 15 pipes, 2 symmetrical pumps, a boost pump parallel to a check valve. It has two taks and two demand nodes.
Data Generation: We have generated instances drawn from realistic consumption data. each instance i originally has $T = 48$ (0.5h) resolution
data collection: build the dataset of a given hydraulic network $\mathcal{G}$ by only considering coarse-time pump scheduling instances (with typically $T = 12$) with branch and check algorithm with some bound tightening (Tavakoli et al.) at the preprocessing $\rightarrow$ leading to 2113 dataset capturing seasonality of the demand and various electricity tariff.
benchmark is available at:

https://github.com/sofdem/gopslpnlpbb/tree/benchs%26nets

Experimental results

Limitation and our solution:
scaling: for finer resolution ($T = 24$, $T = 48$) we cannot gather a collection of data since solving each of them takes unbearable computational effort.

1. resample these input data by averaging the consecutive time steps to obtain its $T = 12$ version
2. approximate the solution of this resampled instance with our learning algorithm, to have $H(D, C)$
3. linear interpolate the tank profile predicted by the deep learning model

Experimental results

Baseline: We compare the results of the hybrid approach (**HA**) with different penalty initialization $\rho = 50, 2$ (**HA50, HA2**) with branch-and-check algorithm with and without preprocessing (BC, BCpre) over 50 test instances for *van Zyl* $T = 12, 24, 48$ (VZ12, VZ24, VZ48)

Penalty update:

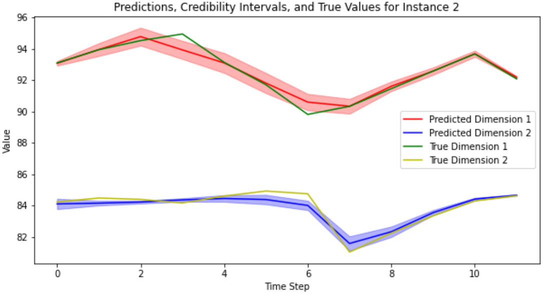
$$(\rho_{ti})^{a+1} = \begin{cases} 5\xi e^{(\frac{-a}{10})}\rho_{ti}^{a} + 1 \text{ if } d_{ti}(q^{b+1}, H^{b+1}) > \varepsilon_a \\ 2\xi e^{(\frac{-a}{10})}\rho_{ti}^{a} + 1 \text{ otherwise} \end{cases}$$

Experimental set up:All algorithms are implemented in Python and experiments

are executed on an Intel(R) Xeon(R) 6148 2.40GHz and 128 GB memory. The deep learning model is built using Tensorflow API version 2.12.0 on Jupyter notebook in Google Colab with GPU A100.

Experimental results

- a Deep learning model to map each demand and tariff tuple into the optimal levels of the tank
- to capture epistemic uncertainty of the deep learning model, we introduce monte-carlo sampling
- instead of having one optimal tank profile we can generate different initial tanks' profiles for our decomposition problem



more diversification during local search done by decomposition

Experimental results

		#solved	Med	Mean	std	Min	Max
VZ12	HA50	49	114	254	359	6	1570
	HA2	44	114	305	438	6	1577
	BC	48	39	121	160	1	681
	BCpre	50	125	124	4	116	137
VZ24	HA50	50	183	285	281	18	1257
	HA2	50	169	279	304.	16	1711
	BC	5	425	1097	1215	272	3117
	BCpre	50	755	809	268	601	2430
VZ48	HA50	50	309	776	1294	37	7069
	HA2	49	322	1014	1435	31	5548
	BC	1	-	-	-	-	-
	BCpre	32	1892	2517	1371	1397	6404

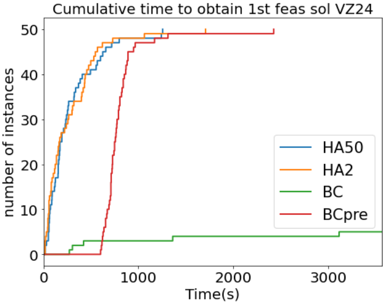
Table: Performance: computation time in seconds.

Experimental results

		#solved	Med	Mean	std	Min	Max
VZ12	HA50	49	6.6	6.6	4.1	0.0	21.2
	HA2	44	4.3	4.6	2.7	0.0	11.3
	BC	48	4.9	5.4	2.9	1.6	12.5
	BCpre	50	3.5	4.3	2.7	0.4	12.4
VZ24	HA50	50	9.6	9.5	4.0	3.3	23.4
	HA2	50	7.6	8.4	3.1	3.4	16.3
	BC	5	11.7	11.1	2.2	7.2	12.6
	BCpre	50	6.5	7.5	6.0	2.4	39.6
VZ48	HA50	50	8.9	9.8	3.9	3.8	21.0
	HA2	49	10.2	10.3	3.9	4.4	19.7
	BC	1	-	-	-	-	-
	BCpre	32	6.4	6.4	1.5	3.4	8.9

Table: Performance: estimated optimality gap in %.

scaling performance of hybrid approach(HA) vs baseline:



number of instances for which we found feasible solution according to time

Proposed hybrid approach **HA** outperforms significantly the baseline branch and check algorithm

Conclusions

- a two-step hybrid approach, consisting a supervised learning algorithm and a local optimization
- Addressing integrality, nonlinearity, and nonconvexity of the problem
- a work can be categorized as the end-to-end learning; however, we have not compromised hard constraints of the problem
- the local optimization algorithm is a tailored direction method enforcing a partial control/storage variable split
- it does a dynamic neighborhood search of the approximated solutions derived from learning
- to increase the chance ending up to a feasible solution, we have selected a learning algorithm providing us several approximations instead of one
- using scaling, the hybrid method is able to retrieve from a low resolution (e.g., scheduling for $T = 12$) obtain good feasible solution for its counterpart $T = 48$

References

- Bonvin, G., Demassey, S., Lodi, A. (2021). Pump scheduling in drinking water distribution networks with an LP/NLP-based branch and bound. Optimization and Engineering, 1-39.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 3(1), 1-122.
- Geißler, B., Morsi, A., Schewe, L., Schmidt, M. (2018). Solving highly detailed gas transport MINLPs: block separability and penalty alternating direction methods. INFORMS Journal on Computing, 30(2), 309-323.
- Tavakoli, A., Sessa, V., Demassey, S. (2022, August). Strengthening mathematical models for pump scheduling in water distribution. In 4th International Conference on Applied Energy (ICAE2022).
- Van Zyl, J. E., Savic, D. A., Walters, G. A. (2004). Operational optimization of water distribution systems using a hybrid genetic algorithm. Journal of water resources planning and management, 130(2), 160-170.

Merci

amirhossein.tavakoli@mines-paristech.fr