DATA-DRIVEN PUMP SCHEDULING FOR COST MINIMIZATION IN WATER NETWORKS

Jyotirmoy Bhardwaj, Joshin Krishnan, Baltasar Beferull-Lozano

WISENET Center, Department of Information & Communication Technology,
University of Agder, Grimstad, Norway
{jyotirmoy.bhardwaj, joshin.krishnan, baltasar.beferull}@uia.no

ABSTRACT

Pumps consume a significant amount of energy in a water distribution network (WDN). With the emergence of dynamic energy cost, the pump scheduling as per user demand is a computationally challenging task. Computing the decision variables of pump scheduling relies over mixed integer optimization (MIO) formulations. However, MIO formulations are NP-hard in general and solving such problems is inefficient in terms of computation time and memory. Moreover, the computational complexity of solving such MIO formulations increases exponentially with the size of the WDN. As an alternative, we propose a data-driven approach to estimate the decision variables of pump scheduling using deep neural networks (DNN). We evaluate the performance of our trained DNN relative to a state-of-the-art MIO solver, and conclude that our DNN based approach can be used to minimize the pump switching and cost incurred due to dynamic energy in a given WDN with much lower complexity.

***Index Terms*—Pump scheduling, mixed-integer formulation, deep neural networks, dynamic energy cost, water-energy nexus.**

I. INTRODUCTION

Pump scheduling is an integral part of water distribution network (WDN) management. As per energy statistics of US, WDNs and treatment plants consume approximately 4% of total produced energy [1]. Pumps consume a significant amount of energy, and optimal pump scheduling can save the energy consumption by 10%-20% [2]. The empirical flow and energy constraints imposed by WDN are non-convex, and the decision variables (pumps and valve control) are binary [3]. Hence, most approaches construct the pump scheduling problem as a *mixed-integer optimization* (MIO) problem with an objective to minimize the energy cost of water dispatch to consumers [4] [5]. In general, MIO is an $\mathcal{NP}$-hard non-convex problem, and computing the decision variables of such problems takes substantial memory and computation time with the current approaches [6]. Moreover, the computational complexity further increases with the ever growing expansion of WDNs.

In contrast, we propose a data-driven approach, in which the decision variables of MIO formulations are learned from the data set of a WDN, and bypass the need of any MIO solver. This approach is motivated by the observation that a feed-forward deep neural network (DNN) can estimate the decision variables of MIO problems with high accuracy. Such data-driven approaches exploit the repetitive patterns of problem instances, and reduce the MIO formulations to a neural network prediction, and once the DNN model is learned, it can speedup the computation time to solve MIO problem for new problem instances [7].

This paper has two major contributions. First, we propose a MIO framework for pump scheduling in a WDN aiming at minimizing the energy cost, given the time-ahead dynamic electric energy prize. The proposed optimization framework also ensures that the WDN parameters are within the admissible operating range, which is a difficult task to achieve by a human operator especially for large-scaled WDNs.The second contribution focuses on a data-driven pump scheduling strategy based on training a feed forward DNN. We solve an offline MIO problem for given network constraints using a MIO solver, obtain the values of decision variables, and train a feed-forward DNN using those values. We benchmark the performance of DNN against the state-of-the-art MIO solver Gurobi [8] using experiments conducted over synthetic data sets, which shows that our approach bypass the need of a solver.

The rest of the paper is structured as follows. Section II formulates an MIO problem in WDN with an objective to minimize the pump switching and cost incurred due to dynamic energy. Section III proposes a model-free data-driven approach for estimation of decision variables in the WDN. Section IV and Section V respectively, present the experimental results and conclusion.

II. MIO FOR PUMP SCHEDULING

Model: Consider a WDN modeled as a directed graph $\mathcal{G}$:=($\mathcal{N}, \mathcal{P}$), where $\mathcal{N}$ and $\mathcal{P}$ denote the sets of nodes and edges (pipes) of the WDN, respectively. A typical WDN

This work was supported in part by IKTPLUSS funded Project "Data-driven cyber-physical networked systems for autonomous cognitive control and adaptive learning in industrial urban water environments (INDURB)", led by WISENET Center, University of Agder, Norway.

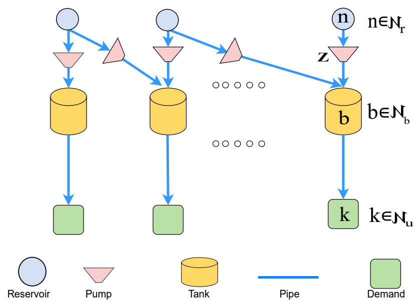

Fig. 1: Schematic of a WDN.

mainly includes the following types of nodes: **a)** reservoirs, the primary source of water; **b)** tanks, the nodes that store water from the reservoirs; and **c)** users, the points of water demands. Let $\mathcal{N}_r \subset \mathcal{N}$ and $\mathcal{N}_b \subset \mathcal{N}$ be respectively the subsets of nodes corresponding to the reservoirs and the tanks, and $\mathcal{N}_u := \mathcal{N} \setminus \mathcal{N}_r \cup \mathcal{N}_b$ be the subsets of nodes denoting the users. An edge between node n and node k of the graph is denoted by the directed pair $(n,k) \in \mathcal{P}$; and let $\mathcal{P}_a \subset \mathcal{P}$ be the subset of edges that host the pumps. In our model, we assume that the pumps are deployed only between the reservoirs and the tanks, and the water flow from tanks to users is generated by means of gravity; hence, $(n,b) \in \mathcal{P}_a \implies n \in \mathcal{N}_r, b \in \mathcal{N}_b$ as shown in Fig. 1.

Flow Conservation: Let q_t^k be the net water outflow rate from the node $k \in \mathcal{N}$ at the time index t, where $t = 1, 2, \dots, T$. The water flow from node n to node k at time t is denoted as $q_t^{nk} \geq 0$, with $q_t^{nk} = 0$ when there is no flow. Further, we assume a directed graph, meaning that $q_t^{kn} = 0$ when $q_t^{nk} \geq 0$. Considering the flow conservation, the net water outflow at node k, can be written as

$$q_t^k = \sum_{(k,n)\in\mathcal{P}} q_t^{kn} - \sum_{(n,k)\in\mathcal{P}} q_t^{nk}, \quad \forall t \in \mathcal{T}, \tag{1}$$

where $\mathcal{T} = \{0, 1, \dots, T-1\}$ is the set of time indices. Further, the water flow is constrained as

$$0 \leq q_t^{nk} \leq q^{\max}, \quad \forall\ (n,k) \in \mathcal{P}, \quad t \in \mathcal{T}, \tag{2}$$

where $q^{\max}$ is the maximum possible flow on any edge, as the flow is constrained by physical properties of the pipe such as length, diameter and friction coefficient. This capacity bound can be obtained from empirical head-loss equations such as Hazen-Williams or Darcy-Weisbach equations [9].

Tank formulation: We assume that each tank $b \in \mathcal{N}_b$ receives water from the reservoirs $n \in \mathcal{N}_r$ through the pump-hosting edges $(n,b) \in \mathcal{P}_a$ and they supply water to the rest of the network. The dynamics of the water storage tank can be modelled as

$$v_{t+1}^b = v_t^b + \tau\left(\sum_{(n,b)\in\mathcal{P}_a} q_t^{nb} - \sum_{(b,n)\in\mathcal{P}\setminus\mathcal{P}_a} q_t^{bn}\right), \tag{3}$$

where v_t^b is the amount of water stored in tank b at time index t and $\tau > 0$ is the sampling time interval. We also bound v_t^b to avoid the overflow of the tank and to meet any unexpected user demand:

$$v^{\min} \leq v_t^b \leq v^{\max},\ b \in \mathcal{N}_b,\ \forall t \in \mathcal{T}. \tag{4}$$

Where, $v^{\min}$ and $v^{\max}$ is the minimum and maximum water volume in a tank respectively.

Pump Formulation: We introduce a binary variable $z_t^{nb} \in \{0,1\}$ to denote the pump switching in the edge $(n,b) \in \mathcal{P}_a$. When $z_t^{nb} = 1$, the pump is ON and the water flows from node n to node b. When $z_t^{nb} = 0$, the pump is OFF and there is no water flow between n and b. For the pump-hosting edges, (2) can be rewritten by including the pump switching as

$$0 \leq q_t^{nb} \leq q^{\max} z_t^{nb},\ \forall (n,b) \in \mathcal{P}_a,\ t \in \mathcal{T}. \tag{5}$$

Switching Constraints: Frequent switching of pump between ON and OFF states is not a desirable phenomenon in WDN, as it increases the transients in the network. Hence, we restrict the number of the pump switching over a predetermined time horizon (T_s). To constrain the switching, we introduce the binary toggle variables $d_t^{nb} \in \{0,1\}$, $\forall t \in \mathcal{T}$, $(n,b) \in \mathcal{P}_a$, where d_t^{nb} indicates a toggle in the state of the pump, i.e., $d_t^{nb} = 1$ when $z_t^{nb} \neq z_{t+1}^{nb}$, whereas $d_t^{nb} = 0$ implies no switching, that is the pump state is same at time indices t and $t+1$.

We observed that the switching constraints involving continuous and binary variables can be formulated with *mixed logical dynamics* [10] [11]. To this end, we introduce an auxiliary variable $\gamma_t^{nb} \in \{-1, 0, 1\}$ to model the updates of z_t^{nb} as a function of d_t^{nb}:

$$z_{t+1}^{nb} = z_t^{nb} + \gamma_t^{nb}, \tag{6}$$

where γ_t^{nb} is given by

$$\gamma_t^{nb} = \begin{cases} -1, & \text{if } d_t^{nb} = 1 \wedge z_t^{nb} = 1 \\ +1, & \text{if } d_t^{nb} = 1 \wedge z_t^{nb} = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The logic relationship between (6) and (7) can be added to an optimization framework using the following linear inequalities [11]:

$$\mathbf{B}[\gamma_t^{nb}\ z_t^{nb}\ d_t^{nb}]^\top \leq \boldsymbol{b}, \tag{8}$$

where $\mathbf{B} = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & -1 \\ 1 & 2 & 2 \\ -1 & -2 & -2 \end{bmatrix}$ and $\boldsymbol{b} = [0\ 0\ 3\ 1]^\top$.

Using the toggle d_t^{nb}, the total number of switching s_t^{nb} over a time window T_s, can be written recursively as

$$s_{t+1}^{nb} = s_t^{nb} + d_t^{nb} - d_{t-T_s}^{nb}, \quad \forall t \in \mathcal{T}. \tag{9}$$

To control the number of switchings over T_s, we impose following constraint:

$$s_t^{nb} \leq \overline{s^{nb}}, \quad \forall t \in \mathcal{T}, \tag{10}$$

where $\overline{s^{nb}}$ is the maximum number of switching for the pump in the edge $(n,b) \in \mathcal{P}_a$. Finally, we assume the following initializations for the variable z_t^{nb}, v_t^{nb}, and $s_t^{nb}, \forall (n,b) \in \mathcal{P}_a$:

$$z_0^{nb} = z_{\text{init}}, \quad v_0^{nb} = v_{\text{init}}, \quad s_0^{nb} = s_{\text{init}}. \tag{11}$$

Let $\boldsymbol{z} \in \{0,1\}^{T|\mathcal{P}_a|}$ and $\boldsymbol{q} \in \mathbb{R}_+^{T|\mathcal{P}_a|}$, where $\mathbb{R}_+$ is the set of non-negative real numbers, be the vector obtained by stacking z_t^{nb} and q_t^{nb} in the lexicographical order of t, n, and b. A similar stacking of other variables $d_t^{nb}, s_t^{nb}, \gamma_t^{nb}$, and v_t^{nb} is done to obtain the vectors $\boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\gamma}$, and $\boldsymbol{v}$ respectively having length $T|\mathcal{P}_a|$.

Mixed-Integer Optimization Framework: We assume that the time-ahead *dynamic energy cost* $\{\pi_t\}_{t=1}^T$ for the pump operation is given. Then, the total cost associated with pumping is given by $f_o(\boldsymbol{z}) = \sum_{t=0}^{T-1} \sum_{(n,b)\in\mathcal{P}_a} {z_t}^{nb}\pi_t$. We propose a MIO framework with the following objectives: **i)** to compute the optimal switching trajectories for the pumps that minimize $f_o(\boldsymbol{z})$ and **ii)** to ensure that while optimizing the switching strategy, all the WDN parameters are within the admissible operation range given by the equations (3),(4),(5),(6),(8),(9), (10), and (11). The proposed MIO framework is

$$\begin{aligned} \text{minimize} \quad & f_o(\boldsymbol{z}) = \sum_{t=0}^{T-1} \sum_{(n,b)\in\mathcal{P}_a} {z_t}^{nb}\pi_t \\ \text{over} \quad & \{\boldsymbol{z}, \boldsymbol{q}, \boldsymbol{v}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\gamma}\} \\ \text{subject to} \quad & (3),(4),(5),(6),(8),(9),(10),(11). \end{aligned} \tag{12}$$

It is to be remarked that the switching constraint (10) controls the number of switchings over a specified time interval, which is a tedious task in manually operated WDNs.

Since the variables $\boldsymbol{z}$ and $\boldsymbol{d}$ are binary, and $\boldsymbol{\gamma}$ and $\boldsymbol{s}$ are integers, problem (12) is a MIO, which is an NP-hard nonconvex problem. Despite MIO being intractable, there are several algorithms that can be used to solve the problem approximately, among which the branch and bound algorithm and the cutting plane method are commonly used. Notably, such formulations can be solved using MIO solvers such as GLPK, Gurobi [8], etc.

III. A DATA-DRIVEN APPROACH

To this end, Section II formulates a MIO problem, where the decision variable $\boldsymbol{z}$ is computed to obtain the optimal trajectory of pump switching. Computing the decision variables of such problems is inefficient in terms of memory and computation time. In this section, we describe model-free data-driven approach to estimate the decision variables of the MIO formulation for WDN. We solve many problem instances of (12) using a standard MIO solver. The obtained solutions are used to train a feed-forward DNN which bypasses the need of a computationally expensive solver for WDN predictions.

III-A. DNN for Estimation of Decision Variables

A feed-forward DNN architecture consists of L *layers*, which define a composition of functions of the form $\hat{f}(\Theta) = h_L(h_{L-1}(\cdots h_1(\Theta))$, where Θ is the input of the DNN, $l = 1$ is the input layer, $l = 2, \cdots, L-1$ are the hidden layers and $l = L$ is the output layer. Each layer depends on the previous layer by $\boldsymbol{y_l} = \boldsymbol{h_l}(\boldsymbol{y_{l-1}}) = \sigma_l(\boldsymbol{w_l}\boldsymbol{y}_{l-1} + \boldsymbol{b}_l) \in \mathbb{R}^{N_l}$, where $\mathbb{R}$ is the set of real numbers, σ is a non-linear activation function, $\boldsymbol{w}_l$ is the weight of the neural network, and N_l is the number of nodes in layer l. The weights $\boldsymbol{w}_l$'s are obtained by training the DNN using the training data sets obtained by solving different problem instances of (12) with the help of the Gurobi MIO solver.

We use rectified linear unit (ReLU) and leaky rectified linear unit (Leaky ReLU) as activation for hidden layers. For the outer layer, sigmoid activation function is used motivated by the nature of the output decision variable and the cost is computed using binary cross entropy. Further, we apply batch normalization at the hidden layers to standardize the preactivation distribution and reduce the internal covariance shift [12]. We apply $\{v_o, z_o, s_o, d_o, \pi_t, q_t^k\}$ at the input layer of DNN for $T = \{12, 15, 18, 20, 24\}$, and the estimates of the decision variables $\{\hat{\boldsymbol{z}}_t\}$ are obtained at the output layer $l = L$ for each T.

III-B. Suboptimality

Let $f_o(\boldsymbol{z}^\star)$ denotes the optimal value of the objective function obtained by solving the MIO problem (12), whereas $f_o(\hat{\boldsymbol{z}})$ denotes the value of the objective function computed through the trained DNN model. We define the suboptimality Υ_o as

$$\Upsilon_o = \frac{|f_o(\boldsymbol{z}^\star) - f_o(\hat{\boldsymbol{z}})|}{f_o(\boldsymbol{z}^\star)} \tag{13}$$

To evaluate the performance of the DNN based approach, we consider that the estimated solution is accurate if suboptimality $\Upsilon_o \leq \epsilon$, where ϵ is the error tolerance.

IV. EXPERIMENTAL RESULTS

We use the WDN shown Fig. 2a, which consists of two reservoirs, four fixed-speed pump, two water storage tanks, and two points of water demand (users), all connected through loss-less pipes. We use the WDN parameters as: $v^{\min} = 30\,000\,\mathrm{m}^3$, $v^{\max} = 100\,000\,\mathrm{m}^3$, and $q^{\max} = 20\,000\,\mathrm{m}^3/\mathrm{h}$. We initialized using $z_{\text{init}} = 0$, $T_5 = 10$, $v_{\text{init}} = 5000\,\mathrm{m}^3$, and $s_{\text{init}} = 3$. In addition, we considered five different time horizons $T = \{12, 15, 18, 20, 24\}$. We consider the sampling time of $\tau = 1\,\mathrm{h}$, $\forall\ T$. The electric energy cost π_t, $\forall t \in \mathcal{T}$ and the demand

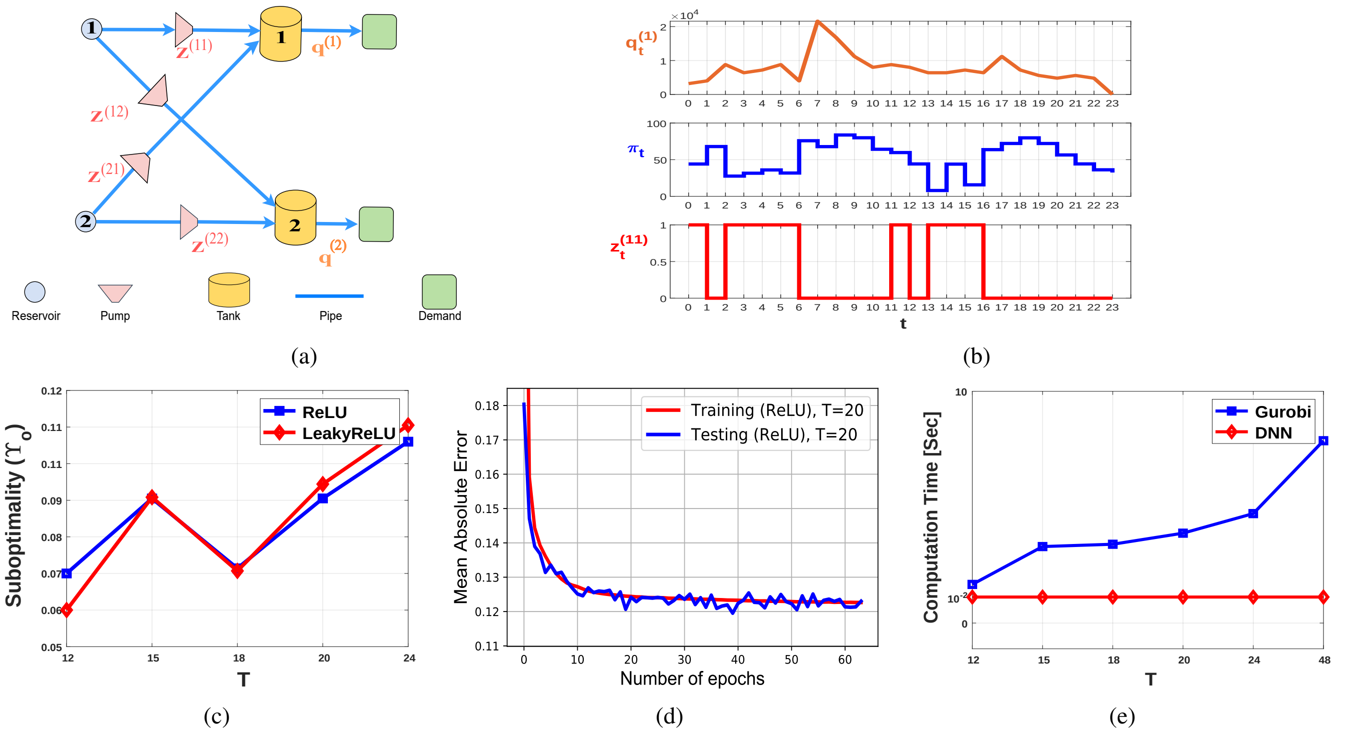

Fig. 2: (2a.) depicts the schematic representation of the WDN, (2b.) presents the pump scheduling $\boldsymbol{z}_t$ and other WDN variables for $T = 24$ with $\tau = 1$hr, (2c.) presents the suboptimality comparison for ReLU and LeakyReLU, (2d.) shows the convergence of DNN for activation ReLU at T=20, and (2e.) presents the comparison of computation time between DNN and Gurobi.

profile of user consumption at the point of water demand $q_t^{bn},\ \forall t \in \mathcal{T},\ (b,n) \in \mathcal{P} \setminus \mathcal{P}_a$ are generated synthetically by assuming a daily pattern of user consumption and electric energy cost for 200000 problem instances for each time horizons T. Also, we added the switching constraints $s^{nb} = 5$ and $T_s = 10$, meaning that total number of switching over a time window of length 10 should not exceed 5.

The MIO problem (12) is solved using the Gurobi solver to obtain the pump trajectory $\boldsymbol{z}^{\star}$. Fig. 2b depicts the obtained pump switching trajectory of a problem instance for a time window of 24 hours. As expected, the pump $z_t^{(11)}$ is in ON state when the energy cost π_t is smaller. The data, which contains all the above mentioned WDN variables and the Gurobi solutions are split as $90\% - 10\%$ for training and testing the DNN. We train the DNN using random initialization with ADAM optimizer. The structure of DNN is constructed by one input layer, 8 hidden layers and one output layer. The hidden layers were consisting of 20, 40, 60, 80, 100, 80, 60, and 40 neurons respectively. In addition, we fixed the value of the Leaky ReLU parameter α as 0.1.

The model performance is evaluated using the suboptimality Υ_o defined in (13) averaged over the test data set and the results given in Fig. 2c shows that the feed-forward DNN can estimate the decision variables of MIO formulation within a tolerance range of $\epsilon \leq 11 \times 10^{-2}$. Fig. 2d shows the convergence of the DNN cost function for $T = 20$. Fig. 2e compares the computation time of the Gurobi solver and the proposed DNN-based solver. Table I presents the percentage of instances for which constraint (5) is satisfied for $\{\hat{z}_t\}$, which ranges between 85% to 90%. We would like to leave a remark that the constraint violation can be addressed by projecting the affected variables of the corresponding problem instances to the feasible region defined by the constraints. The tests were conducted on a 2.7 GHz, Intel Core i7 computer with 8 GB RAM. It is observed that there is an exponential increase in computation time for Gurobi solver when $T \geq 48$, whereas the DNN solves the problem in milliseconds for all values of T.

Table I: Constraint Satisfaction

T	*ReLU*	*LeakyReLU*
12	85.57 %	85.12 %
15	89.5 %	89.28 %
18	81.87 %	81.5 %
20	87.8 %	87.74 %
24	89.5 %	89.88 %

V. CONCLUSION

With ever increasing expansion of WDN, pump scheduling using existing MIO solver is inefficient in terms of memory and computational time. In this work, we propose a MIO formulation to minimize the electric energy cost of WDN while keeping the WDN parameters within a desired admissible range. Further, we propose a computationally efficient method to solve the MIO formulation using DNN, which can bypass the need of using MIO solvers. In a real WDN, given the various interconnected tanks, valves, pump, user demand and dynamic energy price, this approach could compute the decision variables in a computationally efficient and feasible manner, given the availability of necessary data.

VI. REFERENCES

[1] M Fayzul K Pasha and Kevin Lansey, "Strategies to develop warm solutions for real-time pump scheduling for water distribution systems," *Water resources management*, vol. 28, no. 12, pp. 3975–3987, 2014.

[2] Paul F Boulos, Zheng Wu, Chun Hou Orr, Michael Moore, Paul Hsiung, and Devan Thomas, "Optimal pump operation of water distribution systems using genetic algorithms," in *Distribution system symposium*. Citeseer, 2001.

[3] Shen Wang, Ahmad Taha, Nikolaos Gatsis, and Marcio Giacomoni, "Receding horizon control for drinking water networks: The case for geometric programming," *IEEE Transactions on Control of Network Systems*, 2020.

[4] Dariush Fooladivanda and Joshua A Taylor, "Energy-optimal pump scheduling and water flow," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1016–1026, 2017.

[5] Manish K Singh and Vassilis Kekatos, "Optimal scheduling of water distribution systems," *IEEE Transactions on Control of Network Systems*, 2019.

[6] Dimitris Bertsimas and Robert Weismantel, *Optimization over integers*, vol. 13, Dynamic Ideas Belmont, 2005.

[7] Dimitris Bertsimas and Bartolomeo Stellato, "Online mixed-integer optimization in milliseconds," *arXiv:1907.02206*, 2019.

[8] LLC Gurobi Optimization, "Gurobi optimizer reference manual," 2020.

[9] Kevin E Lansey and LW Mays, "Optimal design of water distribution systems," *Water Distribution System Handbook, McGraw-Hill, New York*, 2000.

[10] Alberto Bemporad and Manfred Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[11] Damian Frick, Alexander Domahidi, and Manfred Morari, "Embedded optimization for mixed logical dynamical systems," *Computers & Chemical Engineering*, vol. 72, pp. 21–33, 2015.

[12] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.