THE PUMP SCHEDULING PROBLEM: A REAL-WORLD SCENARIO FOR REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Reinforcement Learning (DRL) has demonstrated impressive results in domains such as games and robotics, where task formulations are well-defined. However, few DRL benchmarks are grounded in complex, real-world environments, where safety constraints, partial observability, and the need for hand-engineered task representations pose significant challenges. To help bridge this gap, we introduce a testbed based on the pump scheduling problem in a real-world water distribution facility. The task involves controlling pumps to ensure a reliable water supply while minimizing energy consumption and adhering to the system's constraints. Our testbed includes a realistic simulator, three years of high-resolution (1-minute) operational data from human-led control, and a baseline RL task formulation. This testbed supports a wide range of research directions, including offline RL, safe exploration, inverse RL, and multi-objective optimization.

1 INTRODUCTION

The pump scheduling problem involves determining when to operate pumps to meet fluctuating water demands while efficiently respecting operational constraints. Scheduling strategies vary with the characteristics of the system, such as the network topology or variations in electricity prices. For example, in regions with variable energy prices, pumps can run during off-peak hours, with storage tanks supplying water during peak periods (Candelieri et al., 2018). Although energy dominates pump operation costs (Nault & Papa, 2015; Abdelsalam & Gabbar, 2021), constraints such as limiting pump switching frequency to protect infrastructure and ensuring water storage turnover for quality must also be addressed.

Several optimization techniques, such as Bayesian Optimization (BO) (Candelieri et al., 2018), Branch-and-Bound (BB) algorithms (Costa et al., 2016; Menke et al., 2016), Genetic Algorithms (GA) (Abiodun & Ismail, 2013; Luna et al., 2019), and Model Predictive Control (MPC) (Castelletti et al., 2023), have been applied to this problem. However, these approaches have important limitations. The population-based optimization of GA and the combinatorial search of BB incur high computational costs, restricting their application to small-scale networks using precomputed daily schedules based on historical water consumption patterns, without real-time adaptability. BO enables fine-grained decisions, but lacks dynamic foresight to consider the impact of actions in future states. MPC, while predictive, relies on accurate system models, which are vulnerable to unmodeled dynamics, hampering its robustness in complex networks. Recent works (see *e.g.* Hajgató et al., 2020; Seo et al., 2021) explore Deep Reinforcement Learning (DRL), highlighting its potential as a data-driven solution with advantages in real-time adaptability and scalability for complex water distribution systems.

Reinforcement learning (RL) (see details in Sutton & Barto, 2018) is a decision-making framework in which an agent learns from interactions with the environment to maximize cumulative returns. Some works have extended RL applications to diverse domains, including autonomous driving (Liu et al., 2021), robotics (Gu et al., 2017; Nair et al., 2018), video games (Lample & Chaplot, 2017; Mnih et al., 2013; Wurman et al., 2022; Vinyals et al., 2019), dialogue systems (Jaques et al., 2019), and more (Levine et al., 2020). Despite its success, RL research is based primarily on virtual environments such as games (Machado et al., 2018) and control simulators (Tassa et al., 2018). In these virtual scenarios, the reward function that shapes the agent's objective(s) and the observation

space that defines the agent's possible knowledge about the environment's state are given. In contrast, real-world problems often pose significant challenges, such as defining these representations to support agent decisions (Nair et al., 2018). Despite the potential of RL, relatively few real-world based scenarios are accessible to the research community.

This work aims to help bridge the gap between real-world applications and RL by introducing a testbed based on the pump scheduling problem in a water facility. We provide a real-world system simulator and logged sensor data regarding pump operation, collected over three years at one-minute intervals from a human-led operation. Alongside these, we detail the operational constraints defined by hydraulic experts and describe the current human-led scheduling strategy. We also established a baseline representation of the RL task, including observation features and a reward function, and highlighted research opportunities enabled by these tools. Our goal is for this testbed to serve as a real-world benchmark for diverse RL approaches, including offline RL, safe exploration, inverse RL, multi-objective RL, and state representation learning. The main contributions are summarized below:

- We release an RL testbed grounded in real-world conditions, comprising a water distribution system simulator, three years of human-led operation data sampled at one-minute intervals, and the implementation of offline RL algorithms.
- We describe the operational characteristics of the water distribution system, including key constraints, the current human scheduling strategy, and a baseline RL task formulation that allows agents to minimize energy consumption while respecting safety and operational requirements.
- We benchmark state-of-the-art offline RL algorithms using the proposed simulator and task formulation, demonstrating that policies trained solely from logged data can surpass human-led operation in energy efficiency by up to 6%.

2 Water Distribution System Overview and Operational Constraints

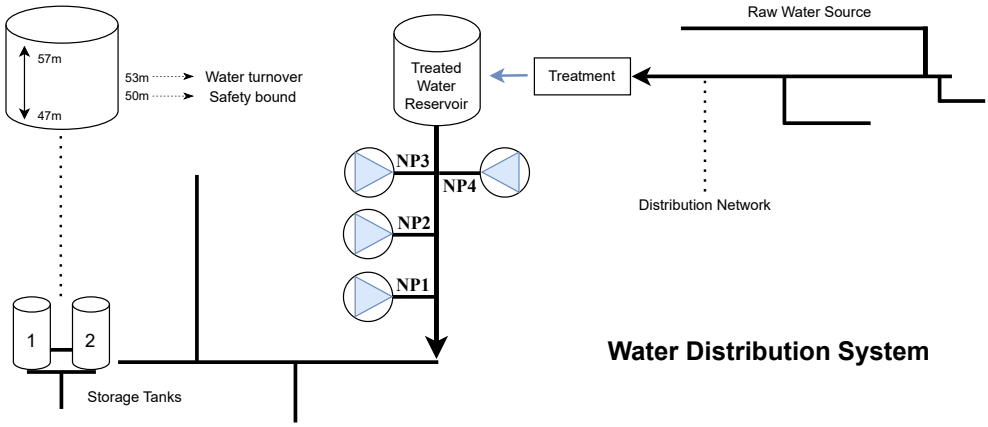


Figure 1: Water distribution system overview. The system has four pumps with fixed speed (ON/OFF) and two elevated water storage tanks.

Figure 1 provides an overview of the water distribution system analyzed in this work. The system draws and treats raw water from wells, which is then stored in a reservoir. In the water utility, four distribution pumps (NP1 to NP4) of varying capacities are available to pump water through the network to two elevated storage tanks. These pumps operate using start/stop control; speed or throttle control is not employed, and pumps operate individually, although parallel operation is used during periods of exceptionally high demand.

The pump scheduling considers multiple factors, including water demand forecasts, energy consumption, water quality, and supply security, to determine which pump is most suitable at any given

time. The tanks are approximately 47 meters above the pump station, have identical dimensions, and are effectively treated as a single tank with a total storage volume of $16\,000$ m³ and a maximum water level of 10 meters. This results in a geodetic height differential of $47 - 57$ meters between the pumps and the storage tanks.

In the water distribution system, human-led operation follows a strategy (policy) to handle pump scheduling. As shown in Figures 2(d), 2(e), 2(f), the operation fills the tank to a high level before the peak in water consumption (see Figures 2(a), 2(b), 2(c)), and then they let it decrease throughout the day to provide water turnover in the tanks and keep its quality. Safety protocols require maintaining a minimum tank level of 3 meters to ensure operational robustness against unexpected events. Figures 2(g), 2(h), 2(i), show the average daily pump switch per month. Each pump switch, either `ON` to `OFF` or `OFF` to `ON`, increments a counter by $+1$. Thus, we could say that the current pump operation generally uses each pump at most once a day. Although it is difficult to measure the impact of a strategy in preserving the system's assets, the idea is to minimize the amount of switching and provide a distribution of pump usage. Finally, in Figures 2(j), 2(k), 2(l), we show the electricity consumption, where, given the pump settings, we observe the prioritization of pump NP2.

3 The Pump Scheduling Problem as a Reinforcement Learning Task

Real-world RL problems, such as pump scheduling, often exhibit partial observability: the agent receives noisy or incomplete observations given the limited number of sensors and hidden variables influencing the system's dynamics. Theoretically, such problems are Partially Observable Markov Decision Processes (POMDPs) (Sutton & Barto, 2018), with observations $o_t \in \mathcal{O}$ generated from states $s_t \in \mathcal{S}$ via an unknown observation function $\Omega : \mathcal{S} \mapsto \mathcal{O}$. However, practical approaches often cast these tasks as MDPs by approximating the Markov property. Techniques like frame stacking (Mnih et al., 2015) and recurrent neural networks (*e.g.* LSTMs) (Hausknecht & Stone, 2015) enrich observations with temporal context, enabling agents to infer hidden dynamics from history. In this work, we adopt this approximation, defining the observation space $\mathcal{O}$ explicitly while noting that the observation function Ω remains unspecified. The task is thus formulated as a POMDP but approximated as an MDP with the tuple $(\mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where:

- $\mathcal{O}$ is the observation space that approximates the hidden state space $\mathcal{S}$;
- $\mathcal{A}$ is the action space;
- $\mathcal{P} : \mathcal{O} \times \mathcal{A} \times \mathcal{O} \mapsto [0, 1]$ is the transition probability, giving the likelihood of moving from observation o to o' after action $a \in \mathcal{A}$;
- $\mathcal{R} : \mathcal{O} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function;
- $\gamma \in [0, 1]$ is the discount factor, balancing immediate and future rewards.

The agent's objective is to learn a policy $\pi : \mathcal{O} \mapsto \mathcal{A}$ that maximizes the expected discounted reward $J(\pi) = \mathbb{E}[\sum_{t=0}^{T-1} \gamma^t r(o_t, a_t)]$, where T is the length of the episode and $r(o_t, a_t)$ is the reward given observation $o_t \in \mathcal{O}$ and action $a_t \in \mathcal{A}$. Based on the daily cycle of water consumption, we set $T = 1440$ timesteps, corresponding to one day of operation with decisions taken at 1-minute intervals. The observation space $\mathcal{O}$ includes sensor-derived features such as tank level and water consumption, while the action space includes four pumps, each with different flow rates Q (m³/h) and power consumptions kW (kW/h), and a no-operation option (NOP with $Q, kW = 0$). The reward function aims to supply water while maintaining safe tank levels and limiting energy use and pump switching.

Observation space: The observation at time t, denoted $o_t \in \mathcal{O}$, comprises the following features: the tank level $l_t \in [47, 57]$ (meters), the water consumption c_t (m³/h) at time t, the time of day (step) $t \in [0, 1439]$ (minutes), the month $m \in 1, \ldots, 12$, the previous action $a_{t-1} \in \mathcal{A}$, the cumulative run time of pumps $p_t \in [0, 1439]^4$ (minutes) and the turnover of the water tank $w(t) \in \{0, 1\}$ within the current episode. These features are designed to provide the agent with sufficient information about the hidden state to select actions that optimize the reward function. Observations are updated by simulating every 1-minute timestep, and the features are chosen to:

- Capture the current state of the system via the level of the tank (l_t) and the consumption of water (c_t);
- Model temporal consumption patterns using the time of day (t) and month (m) to reflect daily and seasonal variations;
- Promote balanced pump usage and minimize switching by tracking the previous action (a_{t-1}) and cumulative runtime (p_t);
- Promote daily turnover in the water stored in the tanks to preserve its quality through the binary feature (w_t).

Action space: The action space $\mathcal{A}$ represents the agent's control options on the pump system, defined as $\mathcal{A} = \{$ NP1, NP2, NP3, NP4, NOP$\}$. The actions correspond to activating one of four pumps, each with different flow rates (Q, in m³ / h) and power consumption (kW, in kW/h), ordered as NP1 $>$ NP2 $>$ NP3 $>$ NP4 in both metrics, or selecting no operation (NOP), where $Q, kW = 0$.

Reward function: The reward function, defined in Eq.1, is designed to balance multiple subgoals: (i) maximize pump operation efficiency, (ii) maintain safe tank levels, (iii) penalize frequent pump switching and distribute usage across pumps, and (iv) promote daily water turnover in the tanks.

$$r_t = \begin{cases} \exp\left(-\frac{kW_t}{Q_t}\right) - C\psi_t + Dw(t) - \ln\left(p_t(a_t) + P\right) & \text{if } Q_t > 0, \\ C\psi_t + Dw(t) - \ln\left(p_t(a_t) + P\right) & \text{if } Q_t = 0\,, \end{cases} \tag{1}$$

where:

- $\exp\left(-\frac{kW_t}{Q_t}\right)$ promotes energy efficiency by favoring higher flow Q_t per power kW_t;
- $-C\psi_t$ enforces safety constraints, penalizing tank levels outside safe bounds;
- Dw_t encourages daily water turnover to maintain quality;
- $-\ln(p_t(a_t) + P)$ discourages prolonged pump use and frequent switching, where p_t is the cumulative runtime of the active pump (selected by a_t) and P is a switching penalty.

The constants C and D are weighting parameters (set both as 10), and the terms ψ_t, w_t, and P are defined as:

- **Tank level constraint**:

$$\psi_t = \begin{cases} \min\left(|l_t - 50|, 1\right) & \text{if } l_t < 50 \text{ (shortage risk)}, \\ 1 & \text{if } l_t = 57 \text{ (overflow)}, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

- **Water turnover**:

$$w_t = \begin{cases} 1 & \text{if } w_{t-1} = 0 \text{ and } 50 \leq l_t < 53, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

- **Switching penalty**:

$$P = \begin{cases} 1 & \text{if } a_t = a_{t-1} \text{ or } Q_t = 0 \text{ or } p_t(a_t) = 0, \\ 30 & \text{otherwise (pump switch)}, \end{cases} \tag{4}$$

Although we show in the following sections that this task formulation can lead to policies that satisfy the objectives, its hand-crafted design may not ensure optimality (Hayes et al., 2021), motivating future exploration of state representation and reward learning techniques discussed with further details in Section 6.

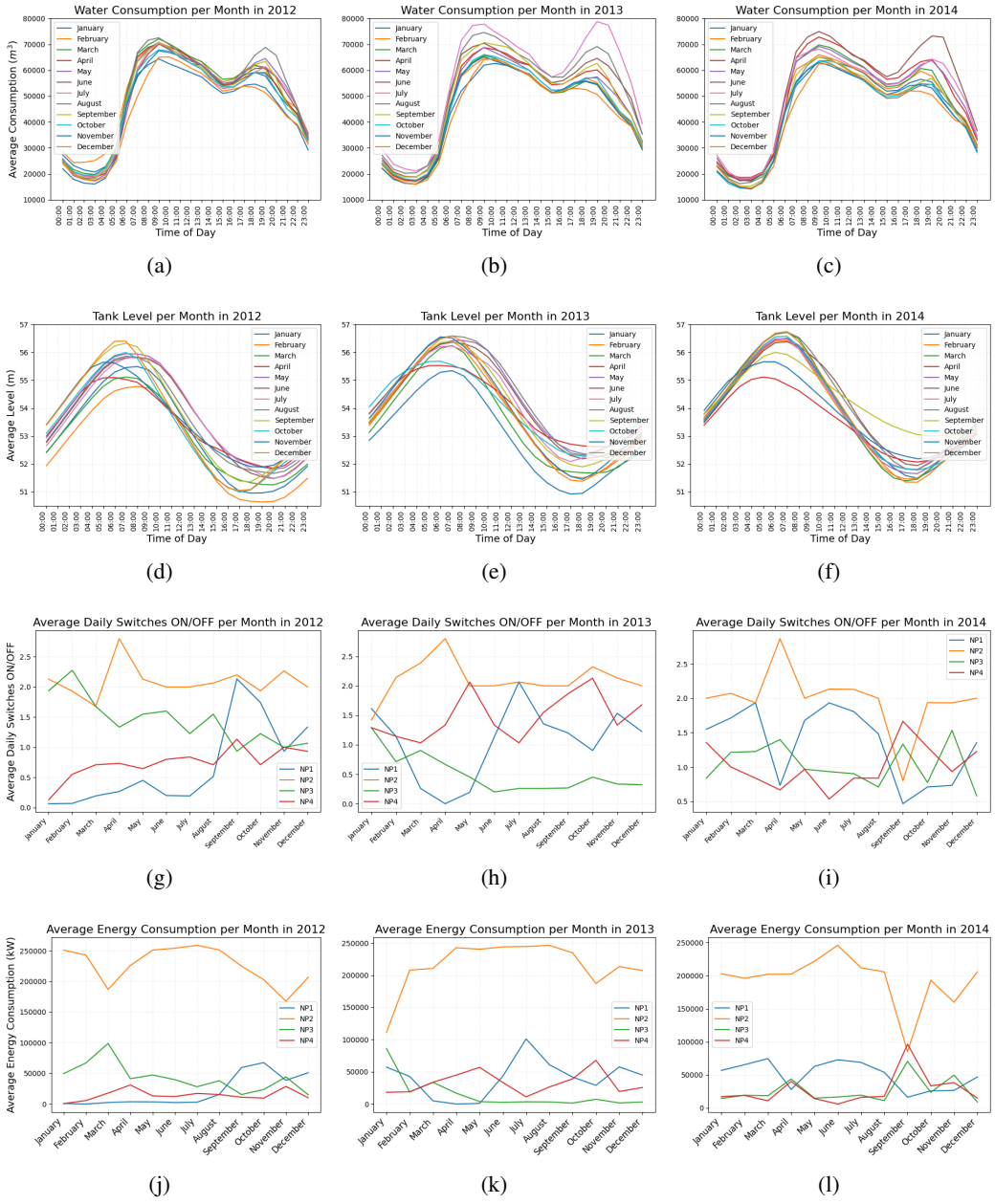


Figure 2: Overview of human-led water system operation. (a–c) Water consumption. (d–f) Tank level profiles. (g–i) Pump switching. (j–l) Electricity usage.

4 INTEGRATING THE WATER SYSTEM SIMULATOR INTO THE RL LOOP

In RL, an agent selects an action a given an observation o, receives a reward r, and transitions to a new observation o'. This process forms a transition tuple $\tau = \langle o, a, r, o' \rangle$, and a complete episode consists of a sequence of T such transitions, *i.e.* episode $= \sum_{i=0}^{T-1} \tau_i$. Figure 3 illustrates how this interaction loop is modeled for the pump scheduling problem using our simulator.

The simulator takes logged water consumption data at time t as input, since this cannot be synthetically generated during new interactions. It also requires an initial condition for the tank level and other features that define the observation o. In the provided codebase, actions are derived from logged human data, although any control policy can be employed. A pump $\in \{\text{NP1}, \text{NP2}, \text{NP3}, \text{NP4}\}$ is considered `ON` if either its power consumption $kW_{NP\#}(t)$ or flow rate $Q_{NP\#}(t)$ are nonzero. If both values are zero for all pumps, the corresponding action is defined as NOP (no operation).

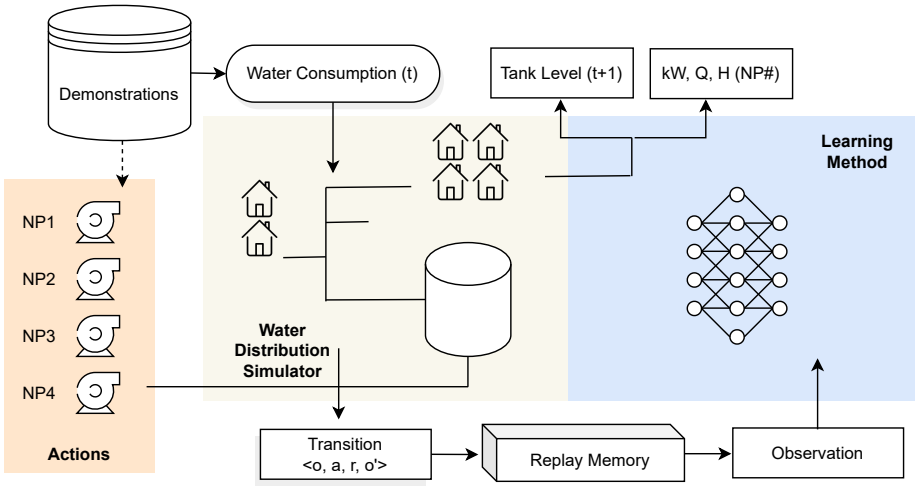


Figure 3: The dynamics of a POMDP and the RL learning process through the water distribution system simulator.

When an action a is applied to the hidden state s, the simulator computes and returns the values of $Q_{NP\#}$, $kW_{NP\#}$, and $H_{NP\#}$ for the given timestep t (see Appendix C.2 for details). These outputs are then used to compute the reward $r(t)$. In the next timestep, a new tank level is observed, forming the next observation o'. This process creates a loop of transitions that are stored in a replay buffer (Lin, 1992), following the Experience Replay paradigm. These collected transitions can be used to learn a new policy π, different from the human-led control strategy, by evaluating the demonstrations through the task representation proposed in Section 3. To learn such a policy, we include some offline RL algorithms in our codebase, which are benchmarked in the following section.

5 BENCHMARKING OFFLINE RL ON REAL-WORLD DEMONSTRATIONS

In offline (batch) RL, the learning process is restricted to logged data, such as demonstrations gathered from human interactions with a system. This constraint often arises from the difficulty in creating accurate simulators or safety concerns related to real-world exploration (Levine et al., 2020). For example, an autonomous vehicle cannot employ trial-and-error learning in the environment due to safety risks and high costs. As a result, offline RL methods (Fujimoto et al., 2019b; Agarwal et al., 2020; Kumar et al., 2020; Jaques et al., 2019) rely

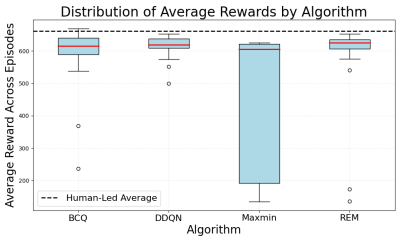


Figure 4: Benchmark results.

on the ability to exploit and generalize from static datasets to learn policies. Recently, benchmarks have been proposed to evaluate offline RL approaches in various scenarios (Gulcehre et al., 2020; Fu et al., 2020; Qin et al., 2022). However, these scenarios often lack real-world constraints, such as safety, and the demonstrations, typically collected from multiple expert RL policies, offer sample diversity compared to datasets derived from a single source, such as human-led control.

In real-world tasks, the behavior policy, that is, the policy from which samples are derived, often exhibits a conservative approach considering exploration, and consequently, sample diversity. As illustrated in Figures 2(d),2(e), and 2(f), human control of pump operations follows a predictable pattern to manage tank levels. Consequently, critical scenarios such as overflow or water shortages are underrepresented in the logged data. This makes pump scheduling an ideal case for assessing the performance of offline RL algorithms using demonstrations from human interactions with real-world systems. To establish a baseline, we benchmark several algorithms (see Appendix D for details)—BCQ (Fujimoto et al., 2019a), DDQN (van Hasselt et al., 2016), Maxmin Q-learning (Lan et al., 2020), and REM (Agarwal et al., 2020)—based on the experimental setup outlined in Appendix E. Using three years of data, we present the results in Figure 4, with one year of data on water consumption reserved for policy evaluation (2012) and two years of data allocated for training ($2013 - 2014$). The results reflect the average cumulative return across episodes, calculated using the reward function defined in Equation 1.

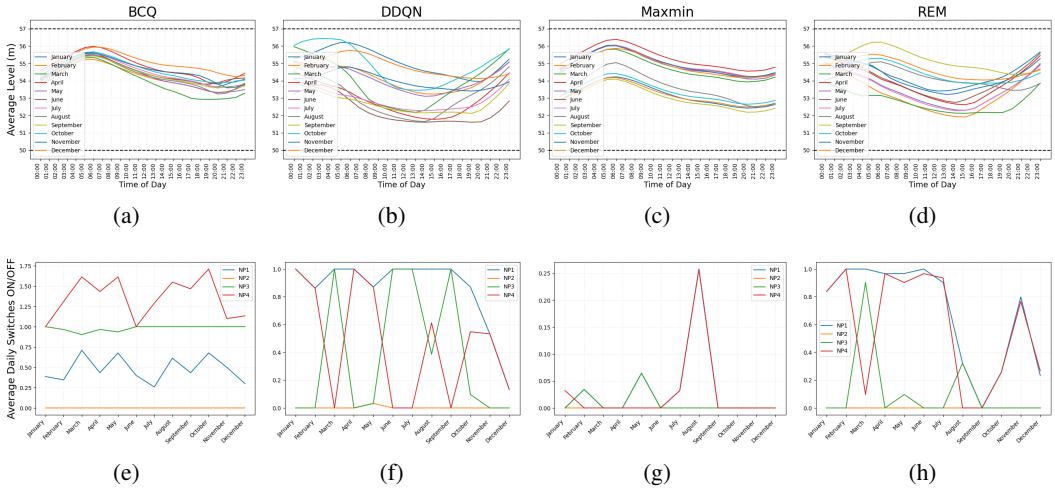


Figure 5: Tank levels (top row) and average daily pump switches (bottom row) for the best-performing policy of each offline RL algorithm: BCQ, DDQN, Maxmin Q-learning, and REM. All policies successfully maintain tank levels within safety limits while enforcing conservative switching patterns compared to human-led control.

To obtain the results shown in Figure 4, we computed the average cumulative reward across 20 independently trained models for each offline RL algorithm. This was necessary to account for the inherent stochasticity of the learning process (Henderson et al., 2018). The dashed black line indicates the average cumulative reward of the behavior policy, *i.e.*, the performance of the human-led control strategy evaluated under the proposed reward function. The results show that the median performance (red line) of all RL algorithms falls below the behavior policy. However, the best-performing models within each set remain competitive. This performance gap can likely be attributed to the challenges of offline RL: the learning process is restricted to demonstrations, making it difficult for policies to generalize beyond the observed state-action distribution. Algorithms such as BCQ and REM exhibit more consistent results, possibly because BCQ constrains action selection to logged data via behavior cloning, while REM stabilizes learning through full-network ensemble updates. Notably, DDQN achieves competitive performance, despite its lower computational burden, while the worst performance of Maxmin Q-learning might be explained by its overly conservative updates.

Figure 5 illustrates the behavior of the best-performing policy (based on average cumulative reward) for each algorithm. As shown in Figures 5(a),5(b),5(c), and 5(d), all learned policies maintained tank levels within safety bounds while minimizing pump switching frequency, often adopting more conservative behavior than the human-led baseline. In terms of energy consumption, the best policies achieved results comparable to or better than real-world control. Specifically, the relative energy savings compared to human-led control were: BCQ: 4.9%, DDQN: -0.4%, Maxmin: 5.9%, and REM: 1.3%, where positive values indicate energy reduction relative to the expert baseline.

6 Research Opportunities Enabled by the Testbed

This testbed offers the RL community an opportunity to move beyond synthetic benchmarks by engaging with challenges inherent in real-world decision tasks. Through its combination of logged human demonstrations, safety constraints, and long-term operational dynamics, it enables research advances in representation learning, safe exploration, inverse RL, multi-objective optimization methods, areas where existing benchmarks fall short.

Representation learning and reward design. Real-world problems like pump scheduling lack a predefined state representation or a reward function that produces optimal policies. In our testbed, the observation space was defined *ad hoc*, based on the information (variables) provided by the sensors and the frequency at which these measurements are recorded. However, some variables may be uninformative or redundant, while incorporating additional variables or extending their historical context could improve policy efficiency. Some works (see *e.g.* Lesort et al., 2018; Merckling et al., 2020; Westphal et al., 2024) have addressed the challenge of identifying more informative, low-dimensional state representations. Similarly, reward engineering poses difficulties, as defining a reward function that balances multiple, potentially conflicting objectives is complex. One potential solution is inverse RL (Ng & Russell, 2000; Abbeel & Ng, 2004), which infers a reward function from the demonstrations. More recently, some approaches have completely bypassed the reward design by conditioning the learning process on achieving specific goals, such as reaching designated states (see *e.g.* Ghosh et al., 2019).

Multi-objective RL. In real-world water distribution systems, the foremost priority is to guarantee a reliable supply of high-quality water to consumers. Once this requirement is satisfied, operational objectives such as minimizing energy consumption, reducing equipment wear, and managing risk become relevant. These competing goals create a natural multi-objective optimization problem. However, in most RL applications, such objectives are aggregated into a single scalar reward using fixed weightings, masking the underlying trade-offs, and limiting flexibility under changing operational conditions. For example, consider a scenario in which planned maintenance is required. To proceed, operators must fill the storage tanks in advance to ensure an uninterrupted supply. In such cases, a static reward function may fail to prioritize this temporary yet critical goal, as it cannot dynamically adjust to the objective importance change. Multi-objective RL (MORL) (Roijers et al., 2013; Hayes et al., 2021) addresses this limitation by representing rewards as a vector of cumulative returns, one for each objective. This enables explicit prioritization, allowing policies to adapt preferences over time without retraining.

An exploration challenge. One constraint in controlling the system is to avoid switching the pump operation too frequently to minimize mechanical wear and extend the pump's lifespan. However, exploration strategies such as *epsilon-greedy* (Sutton & Barto, 2018) can lead to frequent action switches, which conflict with the need for repeated consecutive actions, potentially causing premature convergence to suboptimal policies. Moreover, approaches that prioritize visiting novel states (Bellemare et al., 2016; Subramanian et al., 2016; Tang et al., 2017) risk violating safety constraints, such as tank level limits, by driving the system into unsafe configurations. Thus, the pump scheduling problem can be a setting for evaluating exploration strategies that aim to improve sample efficiency while limiting state visitation to those that meet system safety constraints (García & Fernández, 2015). In particular, our episodic formulation is *state-persistent*, as the initial tank level of each episode depends on the final state of the previous episode, reflecting the continuity of the real world. This persistence of the state requires a robust policy to varying initial conditions, a challenge absent in most existing RL benchmarks (Co-Reyes et al., 2020).

Learning from demonstrations. In this work, we evaluated the performance of offline RL algorithms, such as BCQ and REM, using a dataset of state-action pairs derived from expert demonstrations. These demonstrations reflect a consistent and safe human behavior, and consequently, a concentrated distribution: undesirable or risky system configurations, such as tank overflows or shortages, are rarely encountered. This makes the testbed particularly well-suited to studying offline RL challenges where generalization from narrow expert behavior is required. The testbed also enables the evaluation of Behavior Cloning (BC) (Hussein et al., 2017) approaches, which correspond to solving a maximum likelihood problem to mimic the agent's actions. BC can serve as a pre-training step to accelerate policy learning, although mitigating the distributional shift when the learned policies begin to generate new trajectories remains a critical issue (Nair et al., 2018; Nakamoto et al., 2023). Beyond imitation learning, demonstrations can also support alternative approaches, such as building dynamics models or guiding uncertainty-aware planning, as explored in model-based methods (see *e.g.* Yu et al., 2020; Kidambi et al., 2020).

Continuous action-space. In its current configuration, the control space of the water distribution simulator is discrete: a finite set of pumps operate at fixed speeds, with the additional option of turning off all pumps. However, as discussed in Appendix F, the simulator can be extended to support variable-speed pumps, enabling continuous control of flow rates. This opens the door to evaluating RL approaches designed for continuous action spaces, such as actor-critic methods (Sutton & Barto, 2018). Such extensions would allow researchers to investigate smooth control policies with fine-grained actuation.

Together, these research directions underscore the testbed's versatility as a platform to advance RL research in realistic environments.

7 Conclusion

We introduce a real-world based RL testbed for the pump scheduling problem, containing a validated simulator and three years of high-resolution sensor data collected from human-led control. The simulator models system dynamics based on hydraulic principles, enabling end-to-end policy evaluation. We formally cast the problem as a POMDP, defining the observation and action spaces, reward function, and discussing the constraints to control the system, which guide the design of the task representation. Using the proposed POMDP, we benchmark offline RL algorithms, showing that policies trained solely from demonstrations can match or exceed expert performance in energy efficiency while respecting safety requirements.

Limitations. A key limitation lies in the finite nature of the dataset: unlike synthetic benchmarks such as Gymnasium (Towers et al., 2024), our testbed does not allow unlimited data generation, which restricts the training. However, this constraint reflects a common real-world challenge and positions the testbed as a platform for studying generalization, offline RL, and data-efficient learning. We view this work as a foundation for further development, and we encourage the community to expand the benchmark by addressing additional challenges and settings not covered in the current release.

References

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

Abdelazeem A. Abdelsalam and Hossam A. Gabbar. Energy saving and management of water pumping networks. *Heliyon*, 7(8):e07820, 2021. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2021.e07820. URL https://www.sciencedirect.com/science/article/pii/S240584402101923X.

Folorunso Taliha Abiodun and Fatimah Sham Ismail. Pump scheduling optimization model for water supply system using awga. In *2013 IEEE Symposium on Computers Informatics (ISCI)*, pp. 12–17, 2013. doi: 10.1109/ISCI.2013.6612367.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 104–114. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/agarwal20c.html.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf.

Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Antonio Candelieri, Raffaele Perego, and Francesco Archetti. Bayesian optimization of pump operations in water distribution systems. *Journal of Global Optimization*, 71(1):213–235, 2018.

Andrea Castelletti, Andrea Ficchì, Andrea Cominola, Pablo Segovia, Matteo Giuliani, Wenyan Wu, Sergio Lucia, Carlos Ocampo-Martinez, Bart De Schutter, and José María Maestre. Model predictive control of water resources systems: A review and research agenda. *Annual Reviews in Control*, 55:442–465, 2023.

John D. Co-Reyes, Suvansh Sanjeev, Glen Berseth, Abhishek Gupta, and Sergey Levine. Ecological reinforcement learning. *CoRR*, abs/2006.12478, 2020. URL https://arxiv.org/abs/2006.12478.

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.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Hellmann Dieter-Heinz. *Lecture Notes on Centrifugal Pumps and Compressors*. Technical University of Kaiserslautern, 2004.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019a.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019b. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL http://jmlr.org/papers/v16/garcia15a.html.

Dibya Ghosh, Abhishek Gupta, Justin Fu, Ashwin Reddy, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals without reinforcement learning. *ArXiv*, abs/1912.06088, 2019.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396. IEEE Press, 2017. doi: 10.1109/ICRA.2017.7989385. URL https://doi.org/10.1109/ICRA.2017.7989385.

Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez, Konrad Zolna, Rishabh Agarwal, Josh S Merel, Daniel J Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. Rl unplugged: A suite of benchmarks for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7248–7259. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/51200d29d1fc15f5a71c1dab4bb54f7c-Paper.pdf.

Gergely Hajgató, György Paál, and Bálint Gyires-Tóth. Deep reinforcement learning for real-time optimization of pumps in water distribution systems. *Journal of Water Resources Planning and Management*, 146(11):04020079, nov 2020. doi: 10.1061/(asce)wr.1943-5452.0001287.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (AAAI-SDMIA15)*, November 2015.

Conor F. Hayes, Roxana Radulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel de Oliveira Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *CoRR*, abs/2103.09568, 2021. URL https://arxiv.org/abs/2103.09568.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16669.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), apr 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL https://doi.org/10.1145/3054912.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Àgata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind W. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *CoRR*, abs/1907.00456, 2019. URL http://arxiv.org/abs/1907.00456.

Nathan Jay, Noga Rotman, Brighten Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3050–3059. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jay19a.html.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2140–2146. AAAI Press, 2017.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Timothée Lesort, Natalia Díaz Rodríguez, Jean-François Goudou, and David Filliat. State representation learning for control: An overview. *CoRR*, abs/1802.04181, 2018. URL http://arxiv.org/abs/1802.04181.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Long Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.*, 8:293–321, 1992.

Haochen Liu, Zhiyu Huang, and Chen Lv. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. *CoRR*, abs/2102.09243, 2021. URL https://arxiv.org/abs/2102.09243.

Tiago Luna, João Ribau, David Figueiredo, and Rita Alves. Improving energy efficiency in water supply systems with pump scheduling optimization. *Journal of cleaner production*, 213:342–356, 2019.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *J. Artif. Intell. Res.*, 61:523–562, 2018. URL https://doi.org/10.1613/jair.5699.

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*, 30(14):5333–5349, 2016.

Astrid Merckling, Alexandre Coninx, Loic Cressot, Stephane Doncieux, and Nicolas Perrin. State representation learning from demonstration. In *Machine Learning, Optimization, and Data Science*, pp. 304–315, Cham, 2020. Springer International Publishing. ISBN 978-3-030-64580-9.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, 2018. doi: 10.1109/ICRA.2018.8463162.

Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning, 2023. URL https://arxiv.org/abs/2303.05479v1.

Johnathan Nault and Fabian Papa. Lifecycle assessment of a water distribution system pump. *Journal of Water Resources Planning and Management*, 141(12):A4015004, 2015.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/file/c3e0c62ee91db8dc7382bde7419bb573-Paper.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Carl Pfleiderer and Hartwig Petermann. *Strömungsmaschinen*. Springer-Verlag Berlin Heidelberg, 2005.

Rong-Jun Qin, Xingyuan Zhang, Songyi Gao, Xiong-Hui Chen, Zewen Li, Weinan Zhang, and Yang Yu. Neorl: A near real-world benchmark for offline reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24753–24765. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9cd828eb8dc81a84fb6bf89a94263e1b-Paper-Datasets_and_Benchmarks.pdf.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Int. Res.*, 48(1):67–113, October 2013. ISSN 1076-9757.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Giup Seo, Seungwook Yoon, Myungsun Kim, Changho Mun, and Euiseok Hwang. Deep reinforcement learning-based smart joint control scheme for on/off pumping systems in wastewater treatment plants. *IEEE Access*, 9:95360–95371, 2021. doi: 10.1109/ACCESS.2021.3094466.

Viswanath Sivakumar, Tim Rocktäschel, Alexander H. Miller, Heinrich Küttler, Nantas Nardelli, Mike Rabbat, Joelle Pineau, and Sebastian Riedel. MVFST-RL: an asynchronous RL framework for congestion control with delayed actions. *CoRR*, abs/1910.04054, 2019. URL http://arxiv.org/abs/1910.04054.

Kaushik Subramanian, Charles L. Isbell, and Andrea L. Thomaz. Exploration from demonstration for interactive reinforcement learning. AAMAS '16, pp. 447–456, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450342391.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3a20f62a0af1aa152670bab3c602feed-Paper.pdf.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2024. URL https://arxiv.org/abs/2407.17032.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI Conference on Artificial Intelligence*, 2016. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349277. doi: 10.1145/3061639.3062224. URL https://doi.org/10.1145/3061639.3062224.

Charles Westphal, Stephen Hailes, and Mirco Musolesi. Information-theoretic state variable selection for reinforcement learning, 2024. URL https://arxiv.org/abs/2401.11512.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, 2022.

Yaodong Yang, Jianye Hao, Mingyang Sun, Zan Wang, Changjie Fan, and Goran Strbac. Recurrent deep multiagent q-learning for autonomous brokers in smart grid. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 569–575. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/79. URL https://doi.org/10.24963/ijcai.2018/79.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.

A Appendix

B Related Works

Below, we outline related approaches in the literature on optimizing pump scheduling and control using RL.

Control using Reinforcement Learning. In (Yang et al., 2018) the authors use Deep Recurrent Q-Networks (Hausknecht & Stone, 2015) in the scenario of a smart grid. The objective is to develop a pricing strategy to maximize the broker's (agent) profits. A reward-shaping strategy is proposed once customers are clustered according to their consumption patterns and managed by their respective sub-brokers. Then, a mechanism for credit assignment was necessary to indicate the contribution of each sub-broker to the global return. Wei et al., 2017 propose a *data-driven* solution to control the HVAC (heating, ventilation, and air conditioning) system. The objective is to control the environment temperature by handling numerous disturbances with real-time data. The possible actions are a discrete set of airflow rates divided by building zone. Then, the approach splits each zone, controlling it with distinct neural networks. Sivakumar et al., 2019 propose a network control strategy that acts asynchronously with the environment. The delay, which the authors call δ, corresponds to the policy lookup time when selecting an action. Meanwhile, the network transmits data in the interval $[t, t + \delta]$. Thus, the transitions depend on the state and action in some time step t and the previous action in $t - 1$. In (Jay et al., 2019), the problem of network congestion control is also addressed, and a testbed is released. In (Bellemare et al., 2020), RL is applied to a stratospheric balloon flight controller that must handle a partially observable environment and continuous interaction. Degrave et al., 2022 use RL for nuclear fusion control, which achieved a variety of plasma configurations.

Pump Scheduling. In (Costa et al., 2016), the authors adopted a branch-and-bound algorithm that interacts with the hydraulic simulator EPANET [1] to evaluate decisions at each timestep corresponding to 1 hour in a horizon length of 24 hours. The proposed algorithm aims to find a solution with minimum electricity consumption. Menke et al., 2016 also used a branch-and-bound approach through an algorithm with several steps for the branching procedure. The objective function adopted

[1] https://www.epa.gov/water-research/epanet

is in contact with the one presented in this work. The authors consider pumps with fixed speed (ON/OFF), intending to minimize the power consumption while penalizing switches in pump operation. In (Seo et al., 2021), the authors also applied DRL to control a real-world scenario of a wastewater treatment plant. Unlike our case study, in their system, the electricity price has different tariffs throughout the day, which adds another constraint to the proposed strategy.

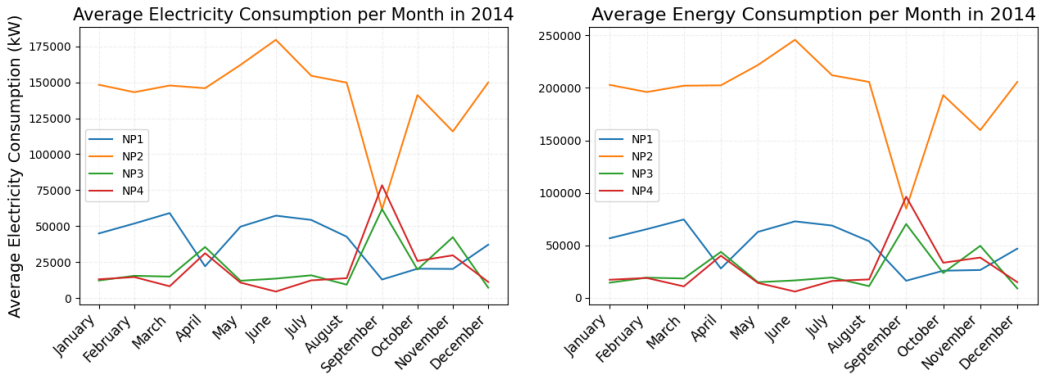
C Details of the Water Distribution System Simulator

C.1 Modeling Electricity Consumption with the Simulator

Figure 6 shows the electricity consumption of the measured (real-world) data and the simulator. Note that the range of values between the measured and simulated data differs. The reason is that the pump efficiency is not considered in Equation 5. Thus, we obtain the values for the measured data by taking the hydraulic power (P_h) and dividing it by the efficiency η:

$$P_h(kW) = Q\rho gh/(3.6\ 10^6), \text{where} \tag{5}$$

- Q is the flow (m^3/h)
- ρ is the density of fluid (kg/m^3)
- g is the acceleration of gravity $(9.81m/s^2)$
- h is the differential head (m)



(a) (b)

Figure 6: Average electricity consumption calculated using measured data 6(a) and simulator 6(b).

C.2 Further Details on the Water System Simulator

The simulator illustrated in Figure 1 is developed to model the operational behavior of distribution pumps in a real-world water network facility located in Germany. The calculation of a pump's operating point is based on the intersection between its characteristic curve (hydraulic head versus flow rate) and the system curve.

The system curve represents the amount of hydraulic head required to deliver a given flow rate through the network. It consists of two components (Figure 7(b)): a static component, independent of flow rate, determined by the difference in geodetic height (*e.g.* when pumping water into an elevated storage tank) and a dynamic component, which reflects friction losses in the piping system and increases quadratically with flow rate. The operating point of a pump is determined by the intersection of its hydraulic head curve with the system curve.

The system curve in this water distribution network is not fixed. The static head varies with the current tank level, while the slope of the dynamic component depends on hydraulic losses influenced by

the instantaneous water demand. Higher water demand lowers the overall system pressure, resulting in a flatter system curve; conversely, lower demand increases system pressure, steepening the curve. The relationship between tank level, water demand, and system curve characteristics was derived from empirical measurements.

The simulator can model pump operations over arbitrary periods. The simulation process requires specifying the expected water demand for each time step, along with the initial tank level and the pump schedule. At each time step, the simulator computes the current system curve based on tank level and water demand, determines whether a pump is active, and calculates its operating point as the intersection between the pump and system curves. If the pump flow rate exceeds the water demand, the tank level increases accordingly; if the flow rate is insufficient, the tank level decreases. This process is repeated iteratively, updating the system curve and tank state at each step, to simulate continuous operation over the defined period.

C.3 Logged Data: List of Variables

Table 1 shows a list of variables included in the logged data to reproduce this work. The dataset contains three years in 1-minute timesteps of raw data gathered through sensors from a real-world system. This dataset provides, at some timestep t, information about the water tank level, and also pump information regarding electricity consumption *kW*, water flow Q, and hydraulic head H.

Table 1: List of Variables in the Water Facility Dataset

Variable Name	**Units/Scale**	**Description**
`../Tank1/{Month}/'HB_Niveau_N_pval'`	Meters	Water tank level
`../WaterConsumption/{Month}/'Netzverbrauch_pval'`	m^3/h	Water consumption
`../NP/NP{#}/KW/'NP_{#}_SIMEAS_P_Leistung_pval'`	kW/h	kW consumption
`../NP/NP{#}/Q/'NP_{#}_Volumenfluss_pval'`	m^3/h	Water flow (Q)
`../NP/NP{#}/H/'NP_{#}_Druck_Druckseite_pval'`	Meters	Hydraulic head (H)

D Offline RL Algorithms Used in Benchmarking

Learning policies constrained to expert demonstrations can be challenging due to the distributional shift issue and overly optimism in the face of uncertainty (Fujimoto et al., 2019b; Levine et al., 2020; Ostrovski et al., 2021). In this section, we further detail how the offline RL algorithms we benchmark tackle this issue.

- **Double DQN (DDQN)** (van Hasselt et al., 2016) mitigates the overestimation bias present in standard Q-learning by decoupling action selection and evaluation during target computation. Specifically, the action that maximizes the Q-estimation is selected using the current network parameters θ_i, but its value is estimated using the target network parameters θ_{i-1}. DDQN is widely used as a baseline in both online and offline RL benchmarks. The loss δ_i used to train the Q-network is given by:

$$\delta_i = \mathbb{E}\left[R(s,a) + \gamma Q\left(s', \arg\max_{a'} Q(s',a';\theta_i);\theta_{i-1}\right) - Q(s,a;\theta_i)\right]^2. \tag{6}$$

- **Random Ensemble Mixture (REM)** (Agarwal et al., 2020) mitigates overestimation by averaging Q-values across an ensemble of estimators using randomly sampled convex combination weights. At each training step, a Dirichlet-distributed weight vector $\boldsymbol{\alpha}$ is sampled such that $\sum_{k=1}^{K} \alpha_k = 1$ and $\alpha_k > 0\ \forall k$. This mixture is used to compute both the target and predicted Q-values. The loss δ_i is given by:

$$\delta_i = \mathbb{E}\left[R(s,a) + \gamma \max_{a'} \sum_{k=1}^{K} \alpha_k Q_k(s',a';\theta_{i-1}^k) - \sum_{k=1}^{K} \alpha_k Q_k(s,a;\theta_i^k)\right]^2. \tag{7}$$

- Although not originally designed for offline settings, **Maxmin Q-learning** (Lan et al., 2020) mitigates overestimation bias by taking the minimum predicted Q-estimation across a set of Q-networks for each action: $Q^{\min}(s,a) = \min_{j\in\{1,\dots,N\}} Q_j(s,a)$, $\forall a \in \mathcal{A}$. During training, a single network $k \in N$ is randomly selected for each update step, but the target is computed using the minimum over all N models:

$$\delta_i = \mathbb{E}\left[R(s,a) + \gamma \max_{a'} Q^{\min}(s',a';\theta_{i-1}) - Q_k(s,a;\theta_i)\right]^2, \quad k \in \{1,\dots,N\}. \tag{8}$$

- **Batch-Constrained Deep Q-Learning (BCQ)** (Fujimoto et al., 2019b;a) mitigates overestimation by constraining the policy to select only actions that are likely under the behavior policy. In the discrete-action version, BCQ uses a BC model $G_\omega(a \mid s)$ trained on the dataset to estimate the action distribution. At each timestep, actions are filtered by a threshold τ, and only those with a sufficiently high likelihood are considered for the target computation. The loss is defined as:

$$\delta_i = \mathbb{E}\left[R(s,a) + \gamma \max_{a' \text{ s.t. } \frac{G_\omega(a'|s')}{\max_{\hat{a}} G_\omega(\hat{a}|s')} > \tau} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\right]^2. \tag{9}$$

E Experimental Setup

Reproducibility remains a major challenge in DRL research (Henderson et al., 2018; Qin et al., 2022), often due to the lack of open-source code, limited access to training data, and insufficient documentation of implementation details such as hyperparameters. To promote reproducibility and transparency, we detail the key hyperparameters, architectural choices, and implementation decisions used in our benchmark experiments.

Partial Observability: Since the pump scheduling problem is naturally partially observable, we employ a Long Short-Term Memory (LSTM) network (Hausknecht & Stone, 2015) to enrich the observation space with temporal information. Specifically, we stack four sequential observations using an LSTM layer.

Prioritized Experience Replay (PER): Experience Replay (Lin, 1992) decorrelates sequential observations by storing transitions and sampling them in mini-batches (Mnih et al., 2013; 2015). Rather than sampling uniformly, we adopt Prioritized Experience Replay (PER) (Schaul et al., 2016), which prioritizes transitions based on their temporal-difference (TD) error δ.

Implementation Choices. Following prior work, we used five Q-networks for REM and three for Maxmin Q-learning after empirical evaluation of these ensemble sizes. We adopt the Huber loss for REM, as suggested by its authors, and the Mean Squared Error (MSE) loss for all other algorithms. For the generative model G_ω used in BCQ, we employ a linear regression model implemented with Scikit-Learn (Pedregosa et al., 2011), selecting its hyperparameters by grid search.

The full list of hyperparameters used is provided in Table 2.

F Extending the Simulator for Variable-Speed Pumps

Speed control is an efficient method for adjusting the operating point of a pump. In this configuration, the motor frequency is varied using a frequency converter, which alters the characteristic curves of the pump, namely the hydraulic head versus flow rate and efficiency versus flow rate relationships. Changes in flow rate and hydraulic head can be described by affinity laws, as given in Equations 10 and 11, where the flow rate (Q) varies proportionally to the pump speed (n) and the hydraulic head (H) varies proportionally to the square of the pump speed:

$$Q \propto n \tag{10}$$

$$H \propto n^2 \tag{11}$$

Hyperparameter	*Value*
Mini-batch size	36
(LSTM, Dense, Dense) nodes	(100, 100, 100)
Update target λ	12000
Optimizer	Adam
Learning rate	0.00003
Discount factor	0.99
α (PER)	0.6
β (PER)	$0.4 \rightarrow 1$
#Q-Networks (REM)	5
#Q-Networks (Maxmin Q-learning)	3
BCQ threshold τ	0.3
L_2 regularization (dense layers)	0.000001
Hardware	GPU V100

Table 2: Hyperparameters and architectural choices used in benchmark experiments.

From Equations 10 and 11, it follows that the hydraulic head changes with the square of the flow rate, as expressed in Equation 12, where Q_1 and H_1 correspond to the flow rate and hydraulic head at the initial reference speed:

$$H_x = \left(\frac{H_1}{Q_1^2}\right) Q_x^2 \tag{12}$$

Thus, under speed variation, all points on the original pump curve move along parabolic trajectories through the origin (Figure 7(a)).

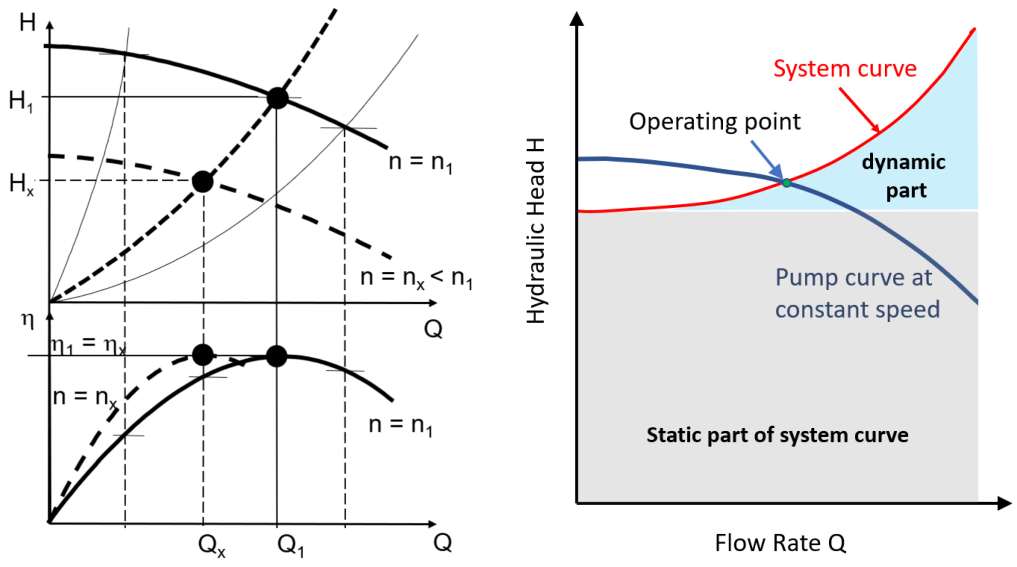


Figure 7: ((a)) Influence of speed control on the pump curve (Dieter-Heinz, 2004). ((b)) Relationship between the pump curve and the system curve.

Changing the speed also modifies the efficiency curve of the pump. If the operating point corresponds to the point of best efficiency at the nominal speed, it remains the point of best efficiency at any other speed. Other points on the efficiency curve shift along quadratic curves while maintaining their position relative to the best-efficiency point. The maximum efficiency value remains

approximately constant. Effects due to variations in the Reynolds number can be incorporated using empirical relationships, such as Ackeret's formula (Pfleiderer & Petermann, 2005):

$$\frac{(1-\eta_{1,\text{opt}})}{(1-\eta_{2,\text{opt}})} = (1-V) + V\left(\frac{Re_2}{Re_1}\right)^{\left(\frac{1}{\alpha}\right)} \tag{13}$$

The change in the pump's operating point due to speed control depends critically on the shape of the system curve. If the system has no static head (*i.e.*, purely dynamic losses), the system curve coincides with the affinity parabola, and the pump can theoretically operate at its point of best efficiency across a range of speeds. However, in practical systems with a nonzero static head, as illustrated in Figure 8, the intersection of the pump curve and the system curve deviates from the ideal affinity parabola. Consequently, the pump operates at reduced efficiency under part-load conditions when the speed is lowered.

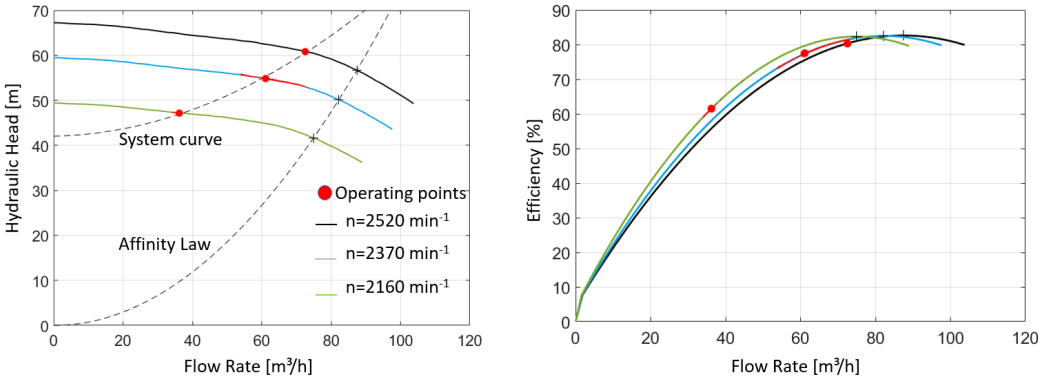


Figure 8: Effect of speed control on the operating point of a pump with nonzero static head.

To enable these effects to be modeled, a speed control function can be incorporated into the main simulator code. This would allow recalculating the pump curve dynamically as the operating speed varies.

G Use of Large Language Models

In preparing this manuscript, we used large language model (LLMs) in the following ways:

1. **Text refinement.** The model was used to improve clarity, grammar, and flow, and to suggest alternative phrasings of drafts. It also helped organize reviewer-style feedback on earlier versions of the text.
2. **Baseline verification.** The model was used to verify whether our implementations of offline RL algorithms were aligned with their respective published descriptions, as well as to refine their code.

All research ideas, methodological contributions, experimental design, and analysis are our own. The LLM was not used to generate new technical content or experimental results. The authors take full responsibility for the content of this paper.