Water Resources Management
https://doi.org/10.1007/s11269-025-04290-8

Rethinking Urban Water Network Design: A Reinforcement Learning Framework for Long-Term Flexible Planning

Lydia Tsiami[1,2] · **Christos Makropoulos**[1] · **Dragan Savic**[2,3]

Received: 1 March 2025 / Accepted: 11 June 2025
© The Author(s) 2025

Abstract
Effectively planning the design of water distribution networks (WDNs) for their whole lifecycle is a complex task for water utilities due to the dynamic nature of WDNs, their long planning horizons, and the deep uncertainty that characterises key design parameters such as future water demand and population growth. Existing flexible design methods, which attempt to address these challenges, rely on static heuristic approaches and predefined decision pathways, requiring re-optimisation whenever new information becomes available. As additional scenarios are introduced, these methods also suffer from exponential increase in complexity, limiting their ability to adapt to emerging information and efficiently explore a wide range of future possibilities. In this work, we introduce a deep reinforcement learning (DRL) framework for the flexible, long-term design of WDNs. By formulating the least-cost staged design problem as a Markov Decision Process and training an agent using Proximal Policy Optimisation, our approach learns cost-effective, sequential interventions across multiple construction stages and future scenarios without relying on predefined decision trees. We evaluate our method on a modified New York Tunnels benchmark across three design tasks, ranging from static single-stage to flexible multi-stage design. Our results show that the DRL agent performs comparably to state-of-the-art heuristics for static and staged deterministic tasks. In the flexible design task, it autonomously devised adaptive strategies, clustered similar scenarios, and maintained high sample efficiency as the number of stages and scenarios increased. These findings highlight DRL as a promising alternative for the lifecycle design of WDNs, establishing a new paradigm for long-term water network planning under deep uncertainty.

Keywords Reinforcement Learning · Deep Uncertainty · Staged Optimisation · Markov Decision Process · Proximal Policy Optimisation · Flexible Design

✉ Lydia Tsiami
lydia_tsiami@mail.ntua.gr

[1] Department of Water Resources and Environmental Engineering, School of Civil Engineering, National Technical University of Athens, Heroon Polytechneiou 9, 157 80 Athens, Greece

[2] KWR Water Research Institute, Nieuwegein, The Netherlands

[3] Centre for Water Systems, University of Exeter, Exeter, UK

Published online: 05 August 2025

1 Introduction

Planning the design of water distribution networks (WDNs) for the long term is an essential but complex task for water utilities. This complexity arises from two main challenges. First, WDNs are dynamic systems that evolve and require multiple construction interventions throughout their lifecycle, which spans several decades. As networks age, their pipes deteriorate, water losses increase, and components fail. Moreover, urban development and demographic variations make the demands placed upon the network increase. As a result, interventions are necessary to maintain performance and ensure uninterrupted water supply. However, these interventions are capital-intensive, constrained by limited budgets, and irreversible (once a pipe is laid, it remains for decades). Furthermore, interventions are interdependent; modifications to one part of the network can influence the performance elsewhere (Basupi and Kapelan 2015a). Given these constraints, prioritising interventions (deciding when and where to implement them) is critical, as decisions made today impact both immediate and long-term performance.

Second, most critical drivers of change necessary to describe the network in the future, such as urban development, population variations and consumer behaviour, are difficult to forecast. This is because they are influenced by factors such as climate, socio-economic conditions, and technology, which are characterised by the so-called 'deep uncertainty' (Walker et al. 2013). Deep uncertainty occurs when decision makers "do not know or cannot agree upon (1) the appropriate models to describe interactions among a system's variables, (2) the probability distributions to represent uncertainty about key parameters in the models, and/or (3) how to value the desirability of alternative outcomes" (Lempert 2002). This uncertainty complicates long-term planning, making it even more challenging for water utilities to anticipate future conditions and optimise strategic decisions.

Despite these challenges, utilities often design WDNs based on a "best guess" of future demand, assuming the network construction occurs in a single phase. This traditional static approach may often lead to either overdesigned or underdesigned infrastructures, which require costly, reactive, and inefficient interventions to align with the actual requirements of the future.

To address this, the research community has moved toward more realistic staged design approaches. Rather than assuming the network will be built all at once, staged design divides the planning horizon (network lifecycle) into multiple construction stages, aiming to identify an optimal sequence of contiguous interventions that will be implemented throughout the network's lifecycle. This approach allows water companies to balance current needs with the expected future growth of the network. By staging interventions, decision-makers can defer costs and make more informed decisions about future interventions as new information becomes available, ensuring that budgets are allocated more efficiently over time.

The literature identifies three categories of staged optimisation: 1) deterministic, 2) robust and 3) flexible (Tsiami et al. 2022). Deterministic approaches (such as Creaco et al. 2014; Halhal et al. 1999; Minaei et al. 2020; Tanyimboh and Kalungi 2008) assume that the future is known. While they account for the dynamic nature of the network and allow deferring decisions until new information becomes available, the initial investment is based on a single predefined scenario, which is a limitation when dealing with uncertainty. Robust approaches (such as Creaco et al. 2015; Cunha et al. 2020; Dell'Aira et al. 2021; Sirsant and Reddy 2021; Zhou and Hu 2009) aim to identify a solution that works for a range of

scenarios. Uncertainty is addressed by building in additional system redundancy; however, this redundancy may lead to overdesign if the considered scenarios diverge significantly, which is common in long-term planning under deep uncertainty.

Flexible optimisation offers a more effective alternative. Instead of implementing a rigid design, it identifies a set of initial interventions that allow the network to adapt to different scenarios with minimal modifications. Scenarios are represented as decision trees, where each branch maps out an optional path for decision-makers. In this way, utilities can implement the first-stage solution that is adaptive to a range of possible futures without overdesigning. In the literature, several studies explore flexible design methodologies. Works like (Huang et al. 2010) and (Tsegaye et al. 2020) show that flexible designs are more cost-effective than static approaches. Kang and Lansey (2014) developed a flexible strategy for the design of a decentralised integrated water and wastewater supply system, minimising regret across multiple scenarios. Marques et al. published a series of papers (Marques et al. 2015a, b, 2018) that focused on network expansion using Real Options theory to develop flexible designs optimising various objectives, including regret (Marques et al. 2015a), carbon emissions (Marques et al. 2015b), and many-objective optimisation across four different criteria (Marques et al. 2018). Basupi and Kapelan (2015a, b) took a different approach by developing a decision-tree-based framework, but instead of tailoring each pathway to an individual scenario, they defined demand thresholds that triggered the interventions. With this approach, each solution pathway was robust across a range of scenarios, rather than being tied to specific ones.

All of the aforementioned flexible optimisation approaches rely exclusively on heuristic algorithms, such as genetic algorithms (Huang et al. 2010; Kang and Lansey 2014; Basupi and Kapelan 2015b; Tsegaye et al. 2020) and simulated annealing (Marques et al. 2015a, b, 2018). While heuristic algorithms can efficiently find near-optimal solutions for predefined scenarios, they are also inherently static. Whenever new scenarios arise or additional information becomes available, the entire optimisation must be repeated from scratch. Moreover, these methods require an explicitly defined decision tree structure, where each intervention pathway must be predetermined. If each pathway corresponds to a unique scenario, the search space grows exponentially, making optimisation computationally challenging when considering a large number of plausible future scenarios. Even clustering scenarios together, as in Basupi and Kapelan's (2015a, b) approach, introduces challenges: (1) the number of decisions or available options at each stage still needs to be defined, a process which may require extensive manual tuning, and (2) the demand thresholds (or other intervention triggers) need to be explicitly optimised. This process may become impractical if a large number of diverse scenarios need to be considered.

Reinforcement learning (RL), a machine learning subfield focused on sequential decision-making within uncertain environments, is a promising solution to the problem. Unlike heuristic approaches, RL offers a more dynamic decision-making framework that learns generalisable policies and adapts to new information (Nagabandi et al. 2019). Recently, RL has achieved notable breakthroughs in sequential decision-making problems under uncertainty. RL agents have demonstrated human (or even superhuman) performance in complex strategy games within stochastic environments (Mnih et al. 2013; Silver et al. 2016; Vinyals et al. 2019; Berner et al. 2019), and have been used in design tasks such as chip (Mirhoseini et al. 2021) and wind turbine blade design (Wang et al. 2023). In RL, an agent learns by interacting with the environment, and its goal is to select actions that maximise a cumulative

reward (Sutton and Barto 2018). One distinctive RL characteristic is that agents can adapt to unforeseen circumstances, making it a promising approach to the long-term planning of WDNs under deep uncertainty.

Although RL has been successfully applied to real-time control in water engineering problems (Castelletti et al. 2013; Hu et al. 2020, 2023; Hajgató et al. 2020), its potential in the (short-term and long-term) design of WDNs has only recently emerged. A few works have applied RL for maintenance planning (Kerkkamp et al. 2022; Bukhsh et al. 2023) and post-disaster rehabilitation (Fan et al. 2022), but its use in flexible design under uncertainty remains relatively unexplored, and is the focus of this study.

Building on our prior work, which demonstrated preliminary promising results in applying RL to WDN design (Tsiami et al. 2024), this study presents a more comprehensive analysis. We formally define flexibility and flexible optimisation in the context of WDNs and show how the design task can be formulated as a Markov Decision Process (MDP), and therefore solved using RL. We propose the use of Deep Reinforcement Learning (DRL) to train an agent to design WDNs under multiple future scenarios and devise flexible planning strategies. Our methodology was first validated on the static classic problem and the deterministic staged optimisation problem before it was extended to flexible optimisation, using the well-studied New York Tunnels (NYT) benchmark. The proposed algorithm achieved comparable performance to state-of-the-art heuristic methods on the static and staged deterministic tasks, demonstrated strong sample efficiency as the number of future scenarios and construction stages increased, and did not require a predefined decision tree in the flexible optimisation setting.

By introducing DRL into the design process, we aim to establish a new paradigm for long-term WDN planning and encourage the adoption of methodologies better suited to navigating the challenges of deep uncertainty.

2 Materials and Methods

2.1 Flexible Optimisation

Flexibility in a WDN refers to the network's ability to adjust to a range of plausible future scenarios and accommodate any new, different or changing requirements through a sequence of performance-efficient, cost-effective, and contiguous interventions over its lifecycle. The key in flexible design is to identify a set of initial interventions, or an initial network configuration, that meets current demands while also allowing the network to adjust with few modifications to a set of plausible future scenarios. To achieve this, a staged optimisation approach is necessary, where the planning horizon is divided into multiple construction stages, each addressing scenario-specific requirements at the time of implementation. The most common objective of flexible optimisation, and the one adopted in this work, is to minimise the total net present cost (NPC) of the network's lifecycle across all considered future scenarios. The scenarios are organised in a scenario tree, with each branch corresponding to a plausible future. Similarly, the interventions for each stage and each scenario are mapped onto a solution tree, which outlines the sequence of design changes needed to adapt to each scenario efficiently.

Formally, suppose a set of scenarios $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{|\Xi|}\}$, where each scenario $\xi \in \Xi$ represents a plausible future described by a set of uncertain parameters (such as water demand, topological changes, and electricity prices). For a planning horizon with N construction stages, the objective of flexible optimisation can be expressed as in Eq. (1):

$$\min_{x_1, x^{i(\xi)}} \left[C(x_1) + \sum\nolimits_{\xi \in \Xi} \left(P(\xi) \sum\nolimits_{i=2}^{N} \frac{C\left(x^i(\xi)\right)}{(1+r)^{\tau_i - \tau_1}} \right) \right] \tag{1}$$

where x^1 is the initial interventions or network configuration vector at the first stage (common to all scenarios), and $x^i(\xi)$ represents the interventions at stage i for scenario ξ. $P(\xi)$ is the probability (uniform if equally likely) or weight assigned to scenario ξ. $C\left(x^i(\xi)\right)$ is the cost of implementing interventions $x^i(\xi)$ at stage i, where $i = 1, \ldots, N$. The discount rate is denoted by r, and τ_i is the time associated with stage i (in years).

The flexible optimisation problem is also subject to:

$$a_{z,\xi}\left(x^i(\xi)\right) = 0, \quad z \in Z, \quad Z \geq 0, \quad \forall i = [1, \ldots, N], \quad \forall \xi \in \Xi \tag{2}$$

$$b_{j,\xi}\left(x^i(\xi)\right) \geq 0, \quad \mathrm{j} \in \mathrm{J}, \quad J \geq 0, \quad \forall i = [1, \ldots, N], \quad \forall \xi \in \Xi \tag{3}$$

$$c_{k,\xi}\left(x^i(\xi)\right) \leq 0, \quad k \in K, \quad \mathrm{K} \geq 0, \quad \forall i = [1, \ldots, N], \quad \forall \xi \in \Xi \tag{4}$$

where Eqs. (2)-(3) are different types of problem-dependent constraints (e.g. mass balance, energy conservation, minimum or maximum nodal pressure or flow requirements). Figure 1 illustrates the concept of flexible optimisation.

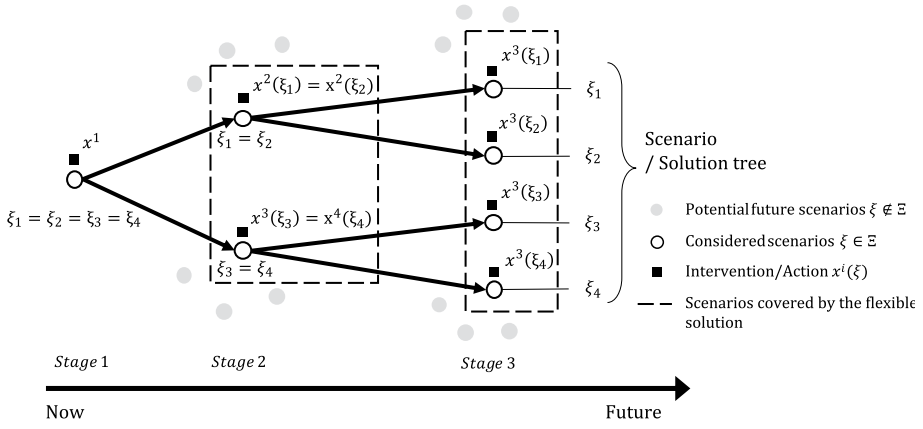


Fig. 1 Schematic representation of the scenario and solution tree of a flexible design. The common initial solution x^1 branches into multiple trajectories, representing distinct plausible future scenarios $\xi_1, \xi_2, \xi_3, \xi_4$. Each path corresponds to a complete sequence of interventions over the planning horizon. Circles denote future scenarios considered at each construction stage, while squares indicate the corresponding interventions. Dashed boundaries outline the range of scenarios covered by the flexible solution at each construction stage

2.2 Modelling Flexible Optimisation as a Markov Decision Process

In this section, we present our approach to flexible design of WDN using DRL. The problem is formulated as a sequential decision-making problem, where a strategy to develop agents that learn to generate flexible designs over an entire planning horizon, for multiple construction stages and under multiple scenarios, is proposed. Our formulation focuses on pipe modifications to minimise installation costs. This configuration was selected as it is the most common in the strategic planning of WDNs and allows for a clear demonstration of the proposed approach, but it can be extended to include other network components.

Generally, in RL, an agent learns an optimal decision-making policy by interacting with an environment over discrete time steps. At each step, the agent observes the current state, selects an action, receives feedback on the quality of its decision in the form of a reward, and transitions to a new state based on the system dynamics. Through repeated interactions, the agent's goal is to refine its policy to maximise the expected cumulative reward it receives over time.

Given a network with n nodes and m pipes, the staged cost-minimisation problem can be expressed as a sequential decision task. Starting from an initial network design, an agent takes $L = \sum_{i=1}^{N} l_i$ actions, where l_i is the number of upgrades at construction stage i, to minimise the total NPC while meeting pressure requirements under all future scenarios. This process can be formulated as an MDP, a mathematical framework used to model sequential decision-making tasks and the foundation for structuring problems in RL. An MDP is defined by the tuple (S, A, T, R), where S is a state space, A a set of possible actions, T a transition function and R a reward function. This task is treated as episodic with L actions being the length of each episode. More details about each component of the MDP are provided below.

2.2.1 State Space

The agent chooses an action based on the state $s_t \in S$ provided by the environment. The state must include sufficient information for the agent to make well-informed decisions while avoiding unnecessary complexity or irrelevant data. To achieve this, the state is designed to include the minimum necessary details about the current network configuration and pressure conditions. Specifically, s_t is a vector obtained by concatenating the difference between the hydraulic head and the minimum required head at each node and the diameters of all network pipes.

2.2.2 Action Space

At each step, the agent takes an action given the state s_t. We define the action as a two-dimensional tuple $a_t = (a_{1t}, a_{2t})$ where $a_{1t} \in A_1$ selects one pipe in the network to modify and $a_{2t} \in A_2$, chooses its new diameter from a discrete set of commercially available pipe diameters.

2.2.3 State Transition Dynamics

The environment uses the EPANET (Rossman 2000) hydraulic solver to update the network's state. When an action a_t is taken, the network is modified, and a new state s_{t+1} is calculated. Since EPANET deterministically computes the effects of each action (new pressures and diameters), the transition function $T(s_{t+1}|s_t, a_t)$ is deterministic within each construction stage.

Once a feasible design is achieved for the current stage, the environment transitions to the next stage under a specific future scenario. This stage-to-stage transition introduces stochasticity as new, scenario-specific conditions (e.g., demand changes) are applied. These transitions depend upon the predefined set of future scenarios and their associated probabilities (if considered), and introduce stochasticity into each episode.

During training, the agent is exposed to this set of predefined scenarios, allowing it to develop a generalisable policy that performs well across all of them. Because the agent experiences these scenarios repeatedly, it learns to optimise its decision-making strategy to maximise cumulative reward across the entire set. Figure 2 illustrates the agent-environment interaction loop.

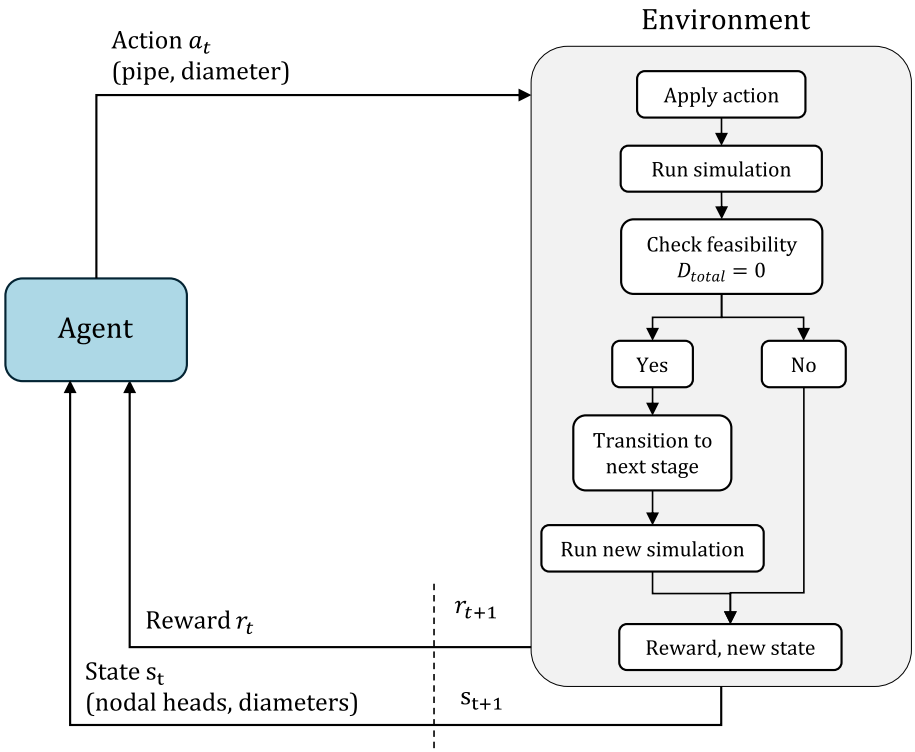


Fig. 2 The agent-environment interaction loop for the flexible design of WDNs. At each time step, the agent selects an action, which modifies the network. The environment simulates the resulting state using the EPANET hydraulic solver and checks for feasibility (i.e., whether the total pressure deficit $D_{total} = 0$). If the current configuration is feasible, the system transitions to the next construction stage under a specific scenario. The updated state and corresponding reward are then returned to the agent

2.2.4 Reward Design

In RL, the agent's primary objective is to maximise its return. Since the total NPC can only be evaluated once all stages are complete and in our formulation the solution is developed incrementally, relying solely on an end-of-episode reward would result in a sparse feedback signal. To address this, we combined intermediate and final rewards to provide a denser learning signal to the agent.

Reward Components

1. Pressure Deficit Penalty

At each step, the agent receives a penalty (Eq. 5) proportional to the total pressure deficit D_{total} of the network at each step:

$$r_{penalty} = -c_1 D_{total}, \quad if\ D_{total} > 0 \tag{5}$$

where $c_1 > 0$ a scaling coefficient. This penalty encourages the agent to prioritise actions that reduce the pressure deficit, guiding it towards feasible designs.

2. Cost-Effectiveness Reward

If the agent identifies a feasible design for stage i (that is when $D_{total} = 0$), it receives a positive reward $(r_{ce})_i$ (Eq. 6) based on the cost-effectiveness of the design up to the current stage i.

$$(r_{ce})_i = c_2 (C_{max} - C_{disc,i})^{c_3} \tag{6}$$

where c_2>0 is a cost scaling coefficient, C_{max} is the maximum possible cost of the network, $C_{disc,n}$ is the present cost of the network, starting from the beginning of the planning horizon up to the current stage i with $i \in [1, N]$. By taking the difference $C_{max} - C_{disc,i}$, the agent is receiving larger rewards for more cost-efficient designs. $c_3 \geq 1$ is an exponentiation factor which determines how strongly the cost differences are amplified in the reward.

At $i = 1$ the $C_{disc,1}$ represents the cost of the first construction stage, while at $i = N$, $C_{disc,N}$ corresponds to the total NPC of the network over the entire planning horizon, which is the optimisation objective.

Since cost-effectiveness can only be fully assessed at the final stage $i = N$, intermediate rewards $r_{ce,i}$ for $i < N$ provide incomplete information about the final solution. To address this, a dynamic weighting factor is applied to emphasise the increasing importance of rewards at later stages, where more complete information is available. The weighted reward at stage i, when $D_{total} = 0$ is:

$$r_{t,i} = w_i (r_{ce})_i \tag{7}$$

where $w_i = 2i$, a linearly increasing weight for stage i, emphasising the importance of rewards closer to the final construction stage.

The weighting guarantees that the reward at the final stage $i = N$ has the highest magnitude, aligning the return with the overall objective and encouraging long-term cost efficiency. At the same time, all intermediate rewards help reduce reward sparsity by providing the agent with frequent signals of progress throughout an episode.

3. Final Reward Function

Combining all components, the reward received by the agent at each time step t of the episode is given by Eq. 8:

$$r_{t,i} = \begin{cases} r_{penalty,t}, & if\ D_{total,t} > 0 \\ w_i(r_{ce})_i & if\ D_{total,t} = 0 \end{cases} \tag{8}$$

3 Application

3.1 Case Study

We apply our methodology to a well-studied benchmark in the literature: the New York Tunnels network (Fig. 3). Originally introduced by Schaake and Lai (1969), the network consists of 21 pipes and 20 nodes and has been extensively used to evaluate different optimisation methods. Due to its manageable size, this case study serves as an illustrative example, allowing for in-depth analysis of the results. In the original formulation of the problem, the network does not meet the minimum required nodal pressures (under a single demand

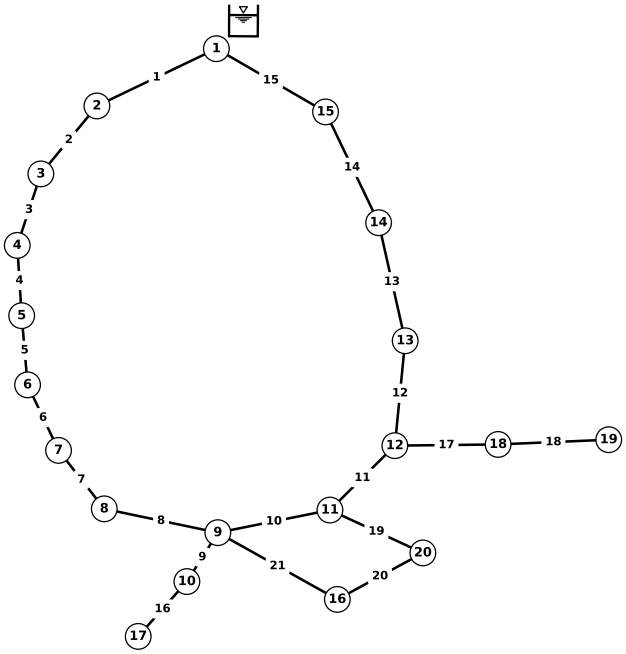


Fig. 3 The New York tunnels network layout

condition), requiring reinforcement through pipe duplication. The optimisation objective is to minimise the cost of installing these parallel pipes while ensuring the minimum pressure requirements are met throughout the network.

Each pipe in the network has 16 possible configurations: no duplication or duplication with one of 15 commercially available diameters (ranging from 0.91 m to 5.18 m). Despite the relatively small size of the network, the original problem is computationally challenging, as the total number of possible network configurations is $16^{21} = 1.9 * 10^{24}$.

3.2 Experiments

To assess the effectiveness of our RL-based approach, we evaluated it on three design tasks of increasing complexity. The first two tasks serve to validate the agent and provide a direct comparison with established heuristic algorithms in the literature. The third task addresses the full flexible optimisation problem, which is the primary focus of this work.

Task 1: Traditional Static Optimisation ($N = 1, and\ |\Xi| = 1$) This task involves a single construction stage (N=1), representing the traditional static optimisation problem, usually solved by utilities in practice, where the network is considered to be built in one construction stage and there is no future uncertainty (usually utilities design with a single scenario which represents a "best-guess" of the future). We used the original NYT problem setup, as described above and compared the agent's performance against the best solutions reported in the literature.

Task 2: Staged Deterministic Optimization ($N > 1, and\ |\Xi| = 1$) In this task, we expand the problem to include multiple construction stages ($N > 1$) while maintaining a single future scenario ($|\Xi| = 1$). Although decisions are made under a best-guess future, they must now be planned incrementally across stages. The goal is to evaluate whether the RL agent can learn a sequence of contiguous, cost-effective actions that optimise the network over the full planning horizon.

We based our setup on the staged framework proposed by Cunha et al. (2020) who adapted the NYT problem by introducing a 60-year planning horizon divided into three construction stages (N=3), each 20 years apart. At each stage: (1) network demands increase uniformly by 0–10%, and (2) the Hazen-Williams coefficient decreases by 2.5 per decade to account for pipe ageing. Cunha et al. generated 50 future demand scenarios using a uniform distribution. From these, they identified cost-optimal solutions for 10 reference scenarios using a 4% discount rate. These 10 reference solutions were then ranked using multicriteria decision analysis to find the most robust design.

We adopted the 10 reference scenarios and corresponding cost-optimal solutions as baselines, treating each scenario as an independent deterministic problem.

Task 3: Flexible Optimisation ($N > 1, and\ |\Xi| > 1$) This task is aligned with the flexible optimisation problem formulation. Unlike the previous tasks, where future conditions were assumed to be known, multiple scenarios ($|\Xi| > 1$) are introduced in this setting, requiring the agent to devise a flexible strategy that can adapt to all of them with as few modifications as possible.

For the flexible design task, we slightly modified the same 10 scenarios introduced by Cunha et al. (2020). Flexible designs require an initial solution that can adapt to multiple future scenarios with minimal modifications. However, since the 10 reference scenarios have different initial demands, we introduced an extra construction stage ($N=4$) with a common starting point for all scenarios, where the first-stage demand matches the original NYT problem conditions. The demand increase trajectories used for both tasks are summarized in Table 1.

All network configuration data, including nodal demands, pipe roughness values, and other system parameters, are obtained from Schaake and Lai (1969) and Cunha et al. (2020). Full details on the benchmark problem, including pipe costs and minimum pressure requirements, are provided in the Supplementary Information (Online Resource 1; Tables S1-S2).

3.3 RL Agent Training Details and Hyperparameters

To train the agent for designing flexible WDNs, we employed Proximal Policy Optimisation (PPO) (Schulman et al. 2017), a state-of-the-art policy gradient algorithm that follows an actor-critic architecture. A custom reinforcement learning environment was developed following the Gymnasium API standards (Towers et al. 2024) and integrated with WNTR (Klise et al. 2020) for hydraulic simulations. The PPO algorithm was implemented using Stable-Baselines3 (Antonin et al. 2021).

To stabilise training, rewards were normalised and state observations were min–max scaled to the [0,1] range. Each episode begins from a fixed initial network configuration, where no pipes are duplicated. The PPO hyperparameters were primarily adopted from the default configuration reported on (Schulman et al. 2017), with the clipping parameter and the entropy bonus coefficient empirically tuned for improved performance. The reward function parameters (c_1, c_2, c_3) were experimentally tuned to balance the relative scales of intervention costs and pressure deficit, ensuring that cost remains the dominant signal while maintaining stable learning dynamics. The total number of training steps varied depending on the specific task, with further details provided in the Results section. All implementation details and hyperparameters are provided in the Supplementary Information (Online Resource 1; Table S3).

Table 1 Demand increase scenarios for the deterministic staged task (Task 2), obtained directly from Cunha et al. (2020), and the flexible task (Task 3), which introduces an additional construction stage common to all scenarios. Each scenario represents a different demand growth trajectory over the network's lifecycle

Scenario name	Task 2 Demand increase (%)			Scenario name	Task 3 Demand increase (%)			
	$\tau=0$	$\tau=20$	$\tau=40$		$\tau=0$	$\tau=20$	$\tau=40$	$\tau=60$
T2-S01	0	0	0	**T3-S01**	0	0	0	0
T2-S02	0	3	4	**T3-S02**	0	0	3	4
T2-S03	0	5	1	**T3-S03**	0	0	5	1
T2-S04	5	1	1	**T3-S04**	0	5	1	1
T2-S05	4	3	3	**T3-S05**	0	4	3	3
T2-S06	3	3	9	**T3-S06**	0	3	3	9
T2-S07	9	1	3	**T3-S07**	0	9	1	3
T2-S08	8	6	1	**T3-S08**	0	8	6	1
T2-S09	6	9	9	**T3-S09**	0	6	9	9
T2-S10	10	10	10	**T3-S10**	0	10	10	10

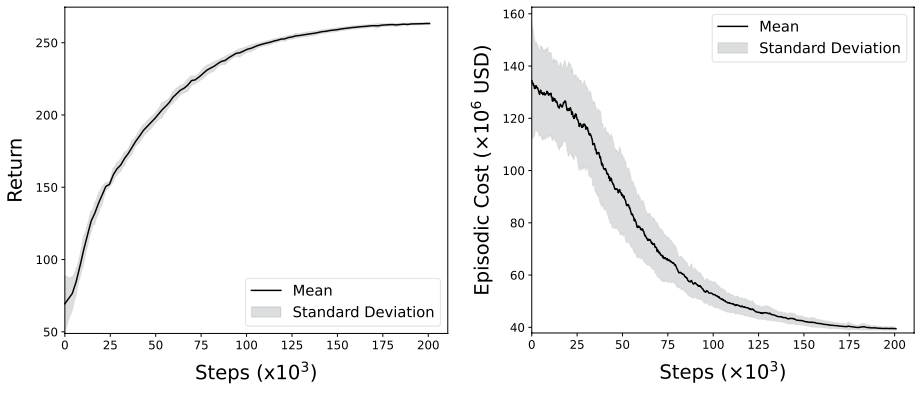
4 Results

4.1 Task 1: Static Optimisation

For the classic WDN design task, we trained the agent 10 times, each with a different random seed. Each training run consisted of 200,000 steps, with the agent completing on average 23,422 episodes per training session. The average cost of the solution produced by the agent was 39.32 ± 0.44 million USD. For the best-performing model, the total number of episodes completed was 23,460, and the best solution found cost 38.81 million USD. The agent first identified this optimal solution after completing 11,552 episodes during training.

In comparison, the best-known solution in the literature for the NYT problem was first found by Maier et al. (2003) at 38.64 million USD. Our agent's best solution is 0.4% more expensive, which shows that it can achieve comparable performance. A detailed comparison between the best-known solution from the literature and our solution is provided in the Supplementary Information (Online Resource 1; Table S4).

The mean learning curve across the 10 training runs of the agent is presented in Fig. 4(a). It shows the return (or cumulative episodic reward) over the time steps. The curve shows that the agent learned quickly at the beginning of the training. After a steep increase, the policy started to converge at 125,000 time steps, and at 160,000 steps, the moving average of the reward reached a plateau and remained almost constant beyond that point. Figure 4(b) further depicts the mean episodic cost of the network across the 10 random seeds. Early in training, the agent produced high-cost designs, but as it refined its policy, it gradually found lower-cost solutions. This reflects the agent learning that cost-efficient designs yield higher rewards. As expected, improvements in return directly correlated with reductions in network cost, aligning with the learning objective.



(a) (b)

Fig. 4 Mean learning curve and mean episodic cost over timesteps for Task 1. The solid lines represent the mean across 10 random seeds, while the shaded regions indicate one standard deviation. (**a**) Average episodic return over time, smoothed using a 100-episode moving average. (**b**) Mean cost of the network at the end of each episode. The cost curve is exponentially smoothed with a weight of 0.9 for improved readability

4.2 Task 2: Staged Deterministic Optimisation

For the deterministic staged optimisation task, the agent was trained independently for each of the demand increase scenarios using 10 different random seeds.

Table 2 compares the agent's performance with the results from the baseline simulated annealing (SA) approach from Cunha et al. (Cunha et al. 2020). The agent's performance summarised in Table 2 is the best one obtained across the 10 different random seeds.

Across all scenarios, the agent identified cost-efficient staged designs under the evolving demand requirements. Compared to SA, the agent produced solutions that were on average 1% more expensive with the cost differences ranging from 0.8% to 3.2% higher. However, in scenarios 9 and 10, the agent found solutions that were marginally (0.2% and 0.9% respectively) less expensive.

Interestingly, the agent required no more than 250,000 steps to converge on the three-stage optimisation task, compared to 200,000 steps for the simpler, classic single-stage task. Despite the exponentially larger search space (from 6^{21} to $6^{21\times3}$), this relatively modest step increase was likely due to how DRL agents learn. In the staged optimisation setting, once the agent identifies a feasible design for the current construction stage, the environment transitions to the next stage with updated demands. However, at the next construction stage, the agent does not need to solve an entirely new task from scratch. Since the policy (i.e., the mapping from states to actions) is represented via function approximation (in our case, a neural network within PPO), high-level decision patterns can be transferred and refined across stages, thus reducing redundant exploration and accelerating convergence. Figure 5 illustrates the learned policy of the agent for one of the 10 scenarios, showing how it incrementally modifies the network by adjusting one pipe at a time.

Table 2 Cost comparison of staged solutions. The baseline heuristic algorithm used for comparison is simulated annealing (Cunha et al. 2020). The reported costs for the RL agent correspond to the best cost obtained after training the agent separately for each scenario across 10 random seeds

Scenario	Demand increase (%)			Total demand increase (%)	Cost SA ($\times10^6 USD$) (Cunha et al. 2020)[1]	Cost RL agent ($\times10^6 USD$)
	$\tau=0$	$\tau=20$	$\tau=40$			
T2-S01	0	0	0	0	**40.9**	**40.9**
T2-S02	0	3	4	7.12	**45.4**	46.2
T2-S03	0	5	1	6.05	**46.3**	46.7
T2-S04	5	1	1	7.11	**51.4**	51.8
T2-S05	4	3	3	10.33	**52.6**	53.1
T2-S06	3	3	9	15.64	**53.4**	55.1
T2-S07	9	1	3	13.39	**60.7**	61.7
T2-S08	8	6	1	15.62	**62.5**	63.6
T2-S09	6	9	9	25.94	64.2	**64.1**
T2-S10	10	10	10	33.10	75.5	**74.8**

[1]Cost reported up to one decimal point in (Cunha et al. 2020)

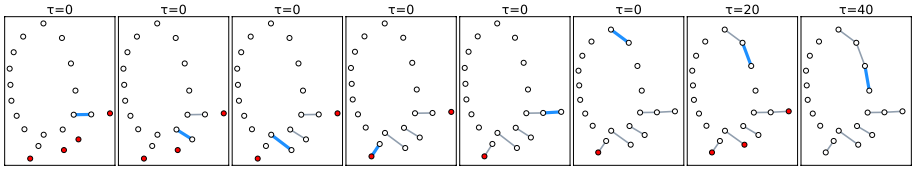


Fig. 5 Learned policy for scenario 6, demonstrates how the agent incrementally modifies the network by adjusting one pipe at a time. In the sixth frame, the agent reaches a feasible solution for the first construction stage, triggering a transition to the next stage with updated demands. For stages at $\tau=20$ and $\tau=40$, only a single pipe modification is necessary. Red nodes indicate locations that do not meet the minimum pressure requirements after an action is applied

4.3 Task 3: Flexible Optimisation

For the flexible optimisation task, the agent successfully found feasible solutions for all 10 demand scenarios and devised automatically a flexible strategy that adapts across them. Trained using 10 random seeds, the best-performing training run produced a flexible strategy with an average cost of 50.67 million USD across the 10 demand scenarios.

Figure 6 presents the best flexible strategy found by the agent. Since all observed demand scenarios start with the same initial demand conditions, it was expected that the agent would start by identifying a common initial ($\tau=0$) solution for all ten scenarios. Interestingly, in the early construction stages, when demand increases between scenarios within a narrower range, the agent clusters scenarios into subgroups and applies the same actions within each subgroup. As the planning process progresses and the scenarios diverge more significantly, the agent's strategy becomes increasingly scenario-specific.

For example, in the second construction stage, the agent grouped the scenarios into four clusters based on demand growth (0%, 3–6%, 8%, and 9–10%). In the third stage, the number of unique actions increased to nine, and by the fourth and final stage, the agent adopted a fully differentiated strategy with 10 distinct interventions, one for each scenario.

Notably, the agent autonomously discovered and devised this flexible strategy, clustering scenarios without any predefined rules or heuristics. By optimising for the expected cumulative reward, the agent naturally learned a policy that minimises expected costs across all 10 scenarios while reducing unnecessary modifications.

Table 3 compares the flexible strategy with the scenario-optimal solutions. The scenario-optimal solutions were obtained by training the agent separately for each scenario using 5 random seeds and selecting the best-performing results. For the four-stage problem under a single scenario, the agent converged after 300,000 training steps.

While flexible solutions are more expensive than their scenario-optimal counterparts, adopting a flexible approach ensures a zero-pressure deficit across all scenarios. If a water utility were to implement a single scenario-optimal solution, the most comparable to the flexible solution choice would be the solution of Scenario 10, which also has zero pressure deficit but is approximately 9 million USD more expensive than the average cost of the flexible solution. Alternatively, selecting a scenario-optimal solution with a similar cost would correspond to Scenario 6, which, when evaluated under all other scenarios, has a total pressure deficit of 36.41 m.

Interestingly, the agent developed a flexible strategy within 1,000,000 training steps. Given that training for a single scenario in the four-stage problem required 300,000 steps,

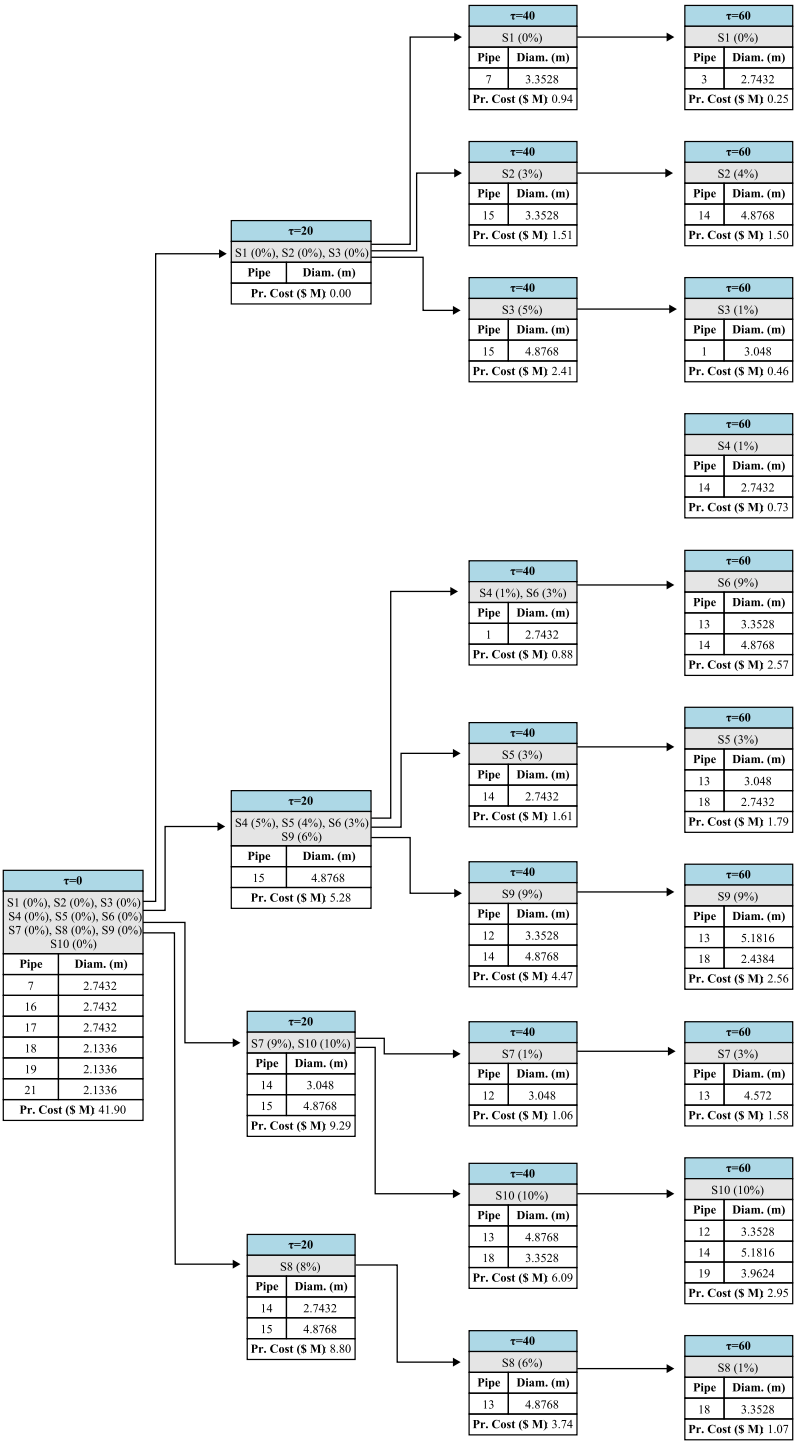


Fig. 6 Flexible strategy developed by the agent, under 10 possible demand increase scenarios

Table 3 Cost comparison of scenario-optimal solutions against the flexible strategy generated by the agent. The scenario-optimal solutions were obtained by training the agent under 5 random seeds and selecting the best-performing result for each scenario. The flexible strategy, in contrast, was trained to adapt across all 10 scenarios with minimal modifications. The deficit column reports the total pressure deficit (m) when each deterministic solution is evaluated across all possible future scenarios

Scenario	Demand increase (%)				Deterministic cost ($\times 10^6$ \$)	Total deficit (m)	Flexible cost ($\times 10^6$ \$)
	$\tau=0$	$\tau=20$	$\tau=40$	$\tau=60$			
T3-S01	0	0	0	0	41.30	180.93	43.09
T3-S02	0	0	3	4	44.28	96.99	44.91
T3-S03	0	0	5	1	44.57	96.59	44.77
T3-S04	0	5	1	1	47.67	70.31	48.79
T3-S05	0	4	3	3	48.73	44.71	50.57
T3-S06	0	3	3	9	49.54	36.41	50.63
T3-S07	0	9	1	3	52.96	25.76	53.82
T3-S08	0	8	6	1	53.77	20.61	55.51
T3-S09	0	6	9	9	53.30	5.98	54.21
T3-S10	0	10	10	10	59.61	0.00	60.22
Avg. Cost					-		**50.65**

this highlights the strong sample efficiency of the DRL algorithm, as it found a generalisable, flexible strategy with only approximately 3.3 times more training steps. Despite the exponential growth in the search space ($16^{21\times4} \rightarrow 16^{21\times31}$), the agent learned a strategy that optimised its performance across all 10 scenarios with minimal additional training effort.

5 Discussion

We found that DRL presents a promising approach for the long-term design of WDNs under deep uncertainty. The proposed algorithm was validated against state-of-the-art heuristic algorithms for the static and staged deterministic tasks and demonstrated comparable (and in some cases improved) performance. In the deterministic staged optimisation task, the agent successfully generated cost-efficient, contiguous sequential interventions. This capability was further extended in the flexible optimisation task, where the agent also autonomously devised a cost-efficient, adaptable strategy that met the necessary requirements across 10 plausible future scenarios.

Beyond solving all three design tasks, we observed that when multiple future scenarios were considered in the flexible task, the agent automatically clustered scenarios with similar demand increases. For each cluster, it applied the same interventions in the early construction stages. This highlights the agent's ability to identify shared decision pathways without predefined rules, a feature especially valuable for planning under deep uncertainty and multiple plausible futures. By comparison, heuristic methods require a predefined decision tree, where the number of intervention options at each stage is known before training and typically corresponds to the number of scenarios considered. Even when scenarios are grouped to reduce complexity, the clustering process, the number of available decisions, and any intervention thresholds must still be manually defined and optimized.

Another finding is that the agent exhibited high sample efficiency as the number of scenarios and construction stages increased. Although the problem size grew exponentially, the agent did not experience a proportional increase in training effort when transitioning from the static problem to the three-stage and four-stage deterministic problems, and finally to the four-stage problem with 10 future scenarios in the flexible setting. This is because the agent uses knowledge gained in one stage or scenario to improve decision-making in others - a capability that heuristic algorithms lack. Heuristic methods, in contrast, treat each scenario independently, resulting in increased computational overhead as complexity increases. The DRL agent, however, learns high-level decision patterns that can be transferred and refined across multiple stages and scenarios, thereby reducing redundant exploration and accelerating convergence.

However, our approach doesn't come without limitations. First, while DRL achieved comparable and sometimes marginally better solutions, it did not always find the absolute best-performing one, though this was without extensive tuning of the algorithm's hyperparameters and reward shaping. Second, our study focused on single-objective optimisation, whereas real-world planning usually involves multiple competing objectives. How multi-objective or many-objective optimisation can be effectively formulated within a reward-based RL framework remains a challenge, yet recent works have begun to explore this research question (Mossalam et al. 2016). Third, while the agent's capabilities were demonstrated on the NYT benchmark, which is well-studied and allows for in-depth analysis of solutions, the real-world WDNs are significantly larger and more complex. Thus, its scalability to large-scale networks remains an open question, although existing works have successfully applied DRL in problems with large state and action spaces (Vinyals et al. 2019). Additionally, the scenarios that we obtained from (Cunha et al. 2020) assumed uniform demand increases, whereas real networks usually exhibit spatially heterogeneous demand growth and topological changes. Although DRL effectively handled multiple uncertainties, it still required numerous hydraulic simulations to incrementally develop solutions. This limitation could be mitigated by incorporating surrogate models (Tsoukalas et al. 2016; Ashraf et al. 2024; Kerimov et al. 2024) or planning-based DRL techniques (Yan et al. 2022) to improve computational efficiency.

That said, this work represents an initial step towards introducing RL as a new paradigm for WDN design. Rather than aiming to provide a final optimised solution, the goal was to explore and investigate the feasibility of RL-based decision-making for long-term planning under deep uncertainty and open new research directions in this area. Future work will focus on scaling the methodology to larger networks, incorporating a larger number of more realistic and diverse future scenarios, and exploring how new and emerging information can be integrated into the decision-making process, allowing the agent to adapt dynamically over time.

6 Conclusions

We introduced a DRL framework for the long-term design of WDNs, capable of automatically devising flexible strategies over multiple construction stages and future scenarios. Our findings demonstrated that RL performs comparably with state-of-the-art heuristic methods and also exhibited unique advantages, such as the ability to cluster similar scenarios

together without any predefined rules, making it a well-suited approach for planning under uncertainty and a large number of future scenarios. The proposed DRL agent is also sample efficient, as it can transfer knowledge across stages and scenarios, due to its capability to learn high-level patterns that generalise across scenarios and construction stages. As the problem scales and the search space grows exponentially, we found that the agent effectively navigated this increased complexity without requiring a proportional increase in computational effort. While challenges remain (e.g. scalability to larger networks, multi-objective optimisation) our study establishes DRL as a promising approach for WDN planning under deep uncertainty. By shifting towards inherently adaptive methodologies, this work lays the foundation for rethinking traditional WDN planning and encourages further research into scalable DRL approaches capable of adapting to emerging information and dynamic environments. Future research should focus on improving scalability, handling more complex uncertainties, and exploring the agent's ability to adapt to emerging information over time.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11269-025-04290-8.

Author Contributions Conceptualization L.T., C.M., D.S.; Formal analysis L.T.; Funding acquisition D.S.; Investigation L.T.; Methodology L.T., C.M., D.S.; Software L.T., Supervision C.M., D.S.; Validation L.T.; Visualization L.T.; Roles/Writing - original draft L.T.; and Writing - review & editing L.T., C.M., D.S.

Funding Open access funding provided by HEAL-Link Greece. This work is a result of the European Research Council (ERC) funded Water-Futures project (Grant agreement No. 951424).

Data Availability Data available on request from the authors.

Declarations

Ethical Approval Not applicable.

Consent to Participate Not applicable.

Consent to Publish All authors consent to publication of the manuscript.

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Antonin R, Ashley H, Adam G et al (2021) Stable-baselines3: reliable reinforcement learning implementations. J Mach Learn Res 22:1–8

Ashraf I, Strotherm J, Hermes L, Hammer B (2024) Physics-informed graph neural networks for water distribution systems. Proc AAAI Conf Artif Intell 38(20):21905–21913

Basupi I, Kapelan Z (2015) Evaluating flexibility in water distribution system design under future demand uncertainty. J Infrastruct Syst 21(2):04014034. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000199

Basupi I, Kapelan Z (2015b) Flexible water distribution system design under future demand uncertainty. J Water Resour Plan Manag 141:1–14. https://doi.org/10.1061/(asce)wr.1943-5452.0000416

Berner C, Brockman G, Chan B et al (2019) Dota 2 with large scale deep reinforcement learning https://doi.org/10.48550/arXiv.1912.06680

Bukhsh ZA, Molegraaf H, Jansen N (2025) A maintenance planning framework using online and offline deep reinforcement learning. Neural Comput Appl 37:13209–13220. https://doi.org/10.1007/s00521-023-08560-7

Castelletti A, Pianosi F, Restelli M (2013) A multiobjective reinforcement learning approach to water resources systems operation: pareto frontier approximation in a single run. Water Resour Res 49:3476–3486. https://doi.org/10.1002/wrcr.20295

Creaco E, Franchini M, Walski TM (2014) Accounting for phasing of construction within the design of water distribution networks. J Water Resour Plan Manag 140:598–606. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000358

Creaco E, Franchini M, Walski TM (2015) Taking account of uncertainty in demand growth when phasing the construction of a water distribution network. J Water Resour Plan Manag 141:1–13. https://doi.org/10.1061/(asce)wr.1943-5452.0000441

Cunha M, Marques J, Savić D (2020) A flexible approach for the reinforcement of water networks using multi-criteria decision analysis. Water Resour Manag 34:4469–4490. https://doi.org/10.1007/s11269-020-02655-9

Dell'Aira F, Cancelliere A, Creaco E, Pezzinga G (2021) Novel comprehensive approach for phasing design and rehabilitation of water distribution networks. J Water Resour Plan Manag 147:1–11. https://doi.org/10.1061/(asce)wr.1943-5452.0001336

Fan X, Zhang X, Yu X (2022) A graph convolution network-deep reinforcement learning model for resilient water distribution network repair decisions. Comput Civ Infrastruct Eng 37:1547–1565. https://doi.org/10.1111/mice.12813

Hajgató G, Paál G, Gyires-Tóth B (2020) Deep reinforcement learning for real-time optimization of pumps in water distribution systems. J Water Resour Plan Manag 146:1–11. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001287

Halhal D, Walters GA, Savic DA, Ouazar D (1999) Scheduling of water distribution system rehabilitation using structured messy genetic algorithms. Evol Comput 7:311–329. https://doi.org/10.1162/evco.1999.7.3.311

Hu C, Cai J, Zeng D et al (2020) Deep reinforcement learning based valve scheduling for pollution isolation in water distribution network. Math Biosci Eng 17:105–121. https://doi.org/10.3934/mbe.2020006

Hu S, Gao J, Zhong D et al (2023) Real-time scheduling of pumps in water distribution systems based on exploration-enhanced deep reinforcement learning. Systems 11:56. https://doi.org/10.3390/systems11020056

Huang D, Vairavamoorthy K, Tsegaye S (2010) Flexible design of urban water distribution networks. World Environ Water Resour Congr 2010 Challenges Chang - Proc World Environ Water Resour Congr 2010 4225–4236. https://doi.org/10.1061/41114(371)430

Kang D, Lansey K (2014) Multiperiod planning of water supply infrastructure based on scenario analysis. J Water Resour Plan Manag 140:40–54. https://doi.org/10.1061/(asce)wr.1943-5452.0000310

Kerimov B, Taormina R, Tscheikner-Gratl F (2024) Towards transferable metamodels for water distribution systems with edge-based graph neural networks. Water Res 261:121933. https://doi.org/10.1016/j.watres.2024.121933

Kerkkamp D, Bukhsh Z, Zhang Y, Jansen N (2022) Grouping of maintenance actions with deep reinforcement learning and graph convolutional networks. In: Proceedings of the 14th international conference on agents and artificial intelligence. SciTePress Digital Library, pp 574–585 https://doi.org/10.5220/0000155600003116

Klise, K.A., Hart, D., Bynum, M., Hogge, J., Haxton, T., Murray, R., & Burkhardt, J.B. (2020). Water network tool for resilience (WNTR) user manual.

Lempert RJ (2002) A new decision sciences for complex systems. Proc Natl Acad Sci U S A 99:7309–7313. https://doi.org/10.1073/pnas.082081699

Maier HR, Simpson AR, Zecchin AC et al (2003) Ant colony optimization for design of water distribution systems. J Water Resour Plan Manag 129:200–209. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(200)

Marques J, Cunha M, Savić D (2015a) Using real options in the optimal design of water distribution networks. J Water Resour Plan Manag 141:1–10. https://doi.org/10.1061/(asce)wr.1943-5452.0000448

Marques J, Cunha M, Savić DA (2015b) Using real options for an eco-friendly design of water distribution systems. J Hydroinformatics 17:20–35

Marques J, Cunha M, Savić D (2018) Many-objective optimization model for the flexible design of water distribution networks. J Environ Manag 226:308–319. https://doi.org/10.1016/j.jenvman.2018.08.054
Minaei A, Sabzkouhi AM, Haghighi A, Creaco E (2020) Developments in multi-objective dynamic optimization algorithm for design of water distribution mains. Water Resour Manag 34:2699–2716. https://doi.org/10.1007/s11269-020-02559-8
Mirhoseini A, Goldie A, Yazgan M et al (2021) A graph placement methodology for fast chip design. Nature 594:207–212. https://doi.org/10.1038/s41586-021-03544-w
Mnih V, Kavukcuoglu K, Silver D et al (2013) Playing atari with deep reinforcement learning. 1–9 https://doi.org/10.48550/arXiv.1312.5602
Mossalam H, Assael YM, Roijers DM, Whiteson S (2016) Multi-objective deep reinforcement learning https://doi.org/10.48550/arXiv.1610.02707
Nagabandi A, Clavera I, Liu S et al (2019) Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In: 7th international conference on learning representations ICLR, pp 1–17 https://doi.org/10.48550/arXiv.1803.11347
Rossman LA (2000) Epanet 2 user 's manual. Natl Risk Manag Res Lab Off Res Dev US Environ Prot Agency Cincinnati. https://doi.org/10.1177/0306312708089715
SchaakeJC, Lai D (1969) Linear programming and dynamic programming application to water distribution network design, MIT, 222p
Schulman J, Wolski F, Dhariwal P et al (2017) Proximal policy optimization algorithms. 1–12 https://doi.org/10.48550/arXiv.1707.06347
Silver D, Huang A, Maddison CJ et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529:484–489. https://doi.org/10.1038/nature16961
Sirsant S, Reddy MJ (2021) Optimal design of pipe networks accounting for future demands and phased expansion using integrated dynamic programming and differential evolution approach. Water Resour Manag 35:1231–1250. https://doi.org/10.1007/s11269-021-02777-8
Sutton RS, Barto AG (2018) Reinforcement learning: an introduction, second. MIT Press
Tanyimboh TT, Kalungi P (2008) Holistic planning methodology for long-term design and capacity expansion of water networks. Water Supply 8:481–488. https://doi.org/10.2166/ws.2008.105
Towers M, Kwiatkowski A, Terry J et al (2024) Gymnasium: a standard interface for reinforcement learning environments. 1–10 https://doi.org/10.48550/arXiv.2407.17032
Tsegaye S, Gallagher KC, Missimer TM (2020) Coping with future change: optimal design of flexible water distribution systems. Sustain Cities Soc 61:102306. https://doi.org/10.1016/j.scs.2020.102306
Tsiami L, Makropoulos C, Savic D (2022) A review on staged design of water distribution networks. In: Proceedings - 2nd international join conference on water distribution system analysis (WDSA) & computing and control in the water industry (CCWI). Editorial Universitat Politècnica de València, València https://doi.org/10.4995/WDSA-CCWI2022.2022.14516
Tsiami L, Makropoulos C, Savic D (2024) Staged design of water distribution networks: a reinforcement learning approach. In: The 3rd international joint conference on water distribution systems analysis & computing and control for the water industry (WDSA/CCWI 2024). MDPI, Basel Switzerland, p 111 https://doi.org/10.3390/engproc2024069111
Tsoukalas I, Kossieris P, Efstratiadis A, Makropoulos C (2016) Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget. Environ Model Softw 77:122–142. https://doi.org/10.1016/j.envsoft.2015.12.008
Vinyals O, Babuschkin I, Czarnecki WM et al (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575:350–354. https://doi.org/10.1038/s41586-019-1724-z
Walker WE, Lempert RJ, Kwakkel JH (2013) Deep uncertainty. Encyclopedia of operations research and management science. Springer, US, Boston, MA, pp 395–402
Wang Z, Zeng T, Chu X, Xue D (2023) Multi-objective deep reinforcement learning for optimal design of wind turbine blade. Renew Energy 203:854–869. https://doi.org/10.1016/j.renene.2023.01.003
Yan D, Weng J, Huang S et al (2022) Deep reinforcement learning with credit assignment for combinatorial optimization. Pattern Recognit 124:108466. https://doi.org/10.1016/j.patcog.2021.108466
Zhou Y, Hu T (2009) Flexible design of delivery capacity in urban water distribution system. In: 2009 international conference on management and service science. IEEE, pp 1–4 https://doi.org/10.1109/ICMSS.2009.5304756

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.