Deep reinforcement learning for rehabilitation planning of water pipes network

Zaharah A. Bukhsh
Eindhoven University of Technology
Eindhoven, Netherlands
z.bukhsh@tue.nl

Nils Jansen
Radboud University
Nijmegen, Netherlands
n.jansen@science.ru.nl

Hajo Molegraaf
Rolsch Assetmanagement
Enschede, Netherlands
hajo.molegraaf@rolsch.nl

ABSTRACT

Cost-effective asset management is an area of interest across several industries, for example, manufacturing, transportation, and infrastructure. In this paper, we develop a deep reinforcement learning (DRL) framework in order to automatically learn an optimal rehabilitation policy for continuously deteriorating water pipes. The DRL agent interacts with a simulated environment of multiple pipes that have distinct characteristics in terms of length, material, and failure rate. We train the agent to learn an optimal policy with minimal average costs and maximum reliability level for assets under-consideration. The learned policy shows improvements over standard preventive and corrective planning approaches. We found that deep reinforcement learning effectively devises optimal policies for dynamic environments; however, simulation of the environment is critical for complex assets (such as bridges, tunnels, underground utilities) that are subject to various environmental and operational stresses.

KEYWORDS

Reinforcement learning, DRL, DQN, Infrastructure management, Water pipe, maintenance planning

1 INTRODUCTION

Reliable water distribution systems (WDS) are paramount for functioning societies. Such systems are subject to deterioration around the globe due to budget cuts, lack of maintenance, and increase in urbanization [6]. Different kinds of *maintenance strategies*, such as recurring schedules (planned) or run-to-failure (corrective), are implemented to keep these assets from failing. However, such strategies do not promise an effective solution at minimal cost and a desired level of services. Replacing an asset according to a predefined plan results in loss of its useful functioning life, whereas replacement after failure can cause large consequential damage [22]. This work seeks to (automatically) learn an optimal rehabilitation policy for continuously deteriorating water pipes.

The recent success of deep neural networks as high-capacity function approximators, such as deep Q networks, policy gradient methods, has stimulated enormous progress in addressing sequential decision-making problems. The integration of deep learning and reinforcement learning, as *deep reinforcement learning* (DRL) framework, has been applied to various applications, including board games [19], video games [11], robotic control [14], and optimal routing [1]. DRL harnesses powerful general-purpose representations to learn and characterize feedback in a long-term horizon [21]. Besides games and standard optimization problems such as knapsack or traveling salesman, the application-oriented studies of DRL for pragmatic real-world problems remain scarce.

This paper develops and implements a DRL framework to automatically devise an optimal rehabilitation policy for water pipes network under economic and reliability requirements. We model agent and environment interactions as a Markov decision process (MDP) [13], see Figure 1. Note that we do not explicitly create such an MDP within our framework but use it merely to gain the required insights into the problem at hand. The agent observes the (simulated) environment state S at time t and performs an action A_t. The agent receives a reward R_t as a feedback signal, and the environment moves to the next state represented as S_{t+1}. Having the MDP formulation, it is assumed that the state transition follows the Markovian property, meaning that future states depend only on the current state S_t and action A_t irrespective of the previous states and actions taken by the agent. The core objective of the agent is to learn to maximise the expected cumulative reward in a definite time horizon.

We choose a (dueling) deep Q-network as a learning algorithm [24]. The deep Q `agent` follows a trial-and-error approach to actively interact with the simulated (water pipes) `environment` in the form of `actions`. The `environment` simulates the physical properties of water pipes in terms of *gradual deterioration* in case of no action and *improvement in reliability state* due to maintenance or replacement. The `agent` can choose among `actions` including do nothing, maintain or replace. The `agent` learns using the feedback (i.e., `reward`) signal from environment to devise an optimal policy.

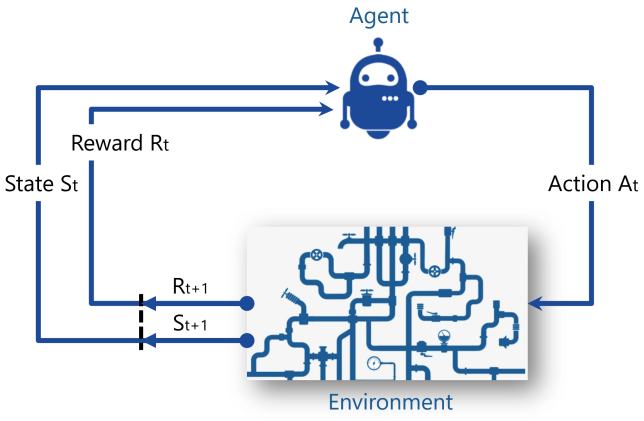

Figure 1: The agent–environment interaction in a Markov decision process

Specifically, the DRL `agent` seeks to maximize the expected cumulative `reward` in a definite time-horizon following sequence of actions. We construct the `reward` function with inverse values since we seek to minimise the overall intervention cost while improving the structural performance of the pipes.

Our work offers two main contributions. First, we provide a novel and scalable solution for optimal rehabilitation of water distribution system under economic and reliability requirements. Second, we adapt a standard DRL framework and DQN methods to learn an optimal policy for the multiple pipes, simultaneously, having distinct physical characteristics. The rest of the paper is structured as follows: Section 2 provides an overview of related work. The problem formulation and details of the deep Q-network are introduced in Section 3. The experiments and results are provided in Section 4. Section 5 presents concluding remarks and a future outlook.

2 RELATED WORK

2.1 Rehabilitation planning of water pipes

The optimal scheduling of rehabilitation of water distribution systems (WDS) has been actively studied since 1979 with the pioneering work of Shamir and Howard [18]. The reported methods in the literature employ various optimization techniques and mathematical modelings such as genetic algorithm [28], dynamic programming [9], integer linear programming [16], and multi-criteria decision analysis methods to facilitate decision-maker in the planning of repair and replacement of water pipes. Advanced data-driven technologies such as artificial neural networks (ANN) are adapted by a few studies [26] to predict the number of pipe failures and to discover the influencing factors. Similarly, in the realm of sequential decision-making, Markov decision processes (MDP) are used to model the deterioration of water networks [20].

Despite the vast research interest, it is noted that existing planning methods are site-specific [10] and do not include comprehensive criteria (such as economic, social impact) of large-scale pipe networks [17]. Besides, traditional programming approaches, either mathematical or optimization, do not scale to accommodate a large number of continuously changing states, and their (computational) complexity grows exponentially with the problem size [8].

2.2 Deep reinforcement learning applications

With the success of the Deep Q network for playing Atari video games [11], the field of deep reinforcement learning (DRL) has gained enormous traction. DRL has outperformed human experts on several board [19], card [12] and video games [2]. It has also been applied to solve complex robotic movement [14], vision control [5] and routing tasks [1]. Nevertheless, many opportunities (and challenges) of DRL in solving problems in diverse domains such as asset management, health, manufacturing, and transportation have largely remained under-explored.

A few notable studies pursue to address sequential decision-making problems from diverse application domains using the DRL framework. Zheng et al. [29] conducted a comprehensive study to learn dynamic tax policies by economic simulations to balance equality and productivity in socio-economic settings. Hubbs et al. [7] solve a chemical production scheduling process to account for uncertainty using the actor-critic policy gradient algorithm. Cals et al. [4] introduces an approach to solve order batching and sequencing problem in a warehouse using the proximal policy optimization algorithm. Similarly, the sequencing problem to avoid bus bunching is addressed by Wang and Sun [23] using a multi-agent framework. Wei et al. [25] devise an optimal structural maintenance policy for bridge components using DQNs. Specficaly for WDS, **?**] proposed a DRL-based approach to control the watr pump speed using the real-time measurement data.

3 APPROACH

3.1 Problem formulation

The objective is to (automatically) learn an optimal rehabilitation policy for continuously deteriorating water pipes. We define an optimal policy based on its economic and reliability aspect. Specifically, the goal is to find optimal intervention moments such that we incur the minimum average economic cost and maximum average reliability of water pipes within a planning time period. We model the rehabilitation planning problem of water pipes as a finite Markov decision process (MDP) within the DRL framework. In the following, we briefly introduce each component of the DRL framework (see Figure 1).

States: The state of the environment is represented by a matrix $\mathcal{S} = m \times n$, where m represents the number of pipes considered for rehabilitation planning and n is the number columns representing the characteristics of each pipe. For $m = i$, the vector $n_i = \langle age_i, mat_i, L_i, FR_i, aux_i, pf_i, rl_i \rangle$ defines the state of the pipe i, where age_i, mat_i, L_i are the age, material and length of a pipe, respectively. FR_i is the failure rate concerning the material of the pipe. aux is an auxiliary variable for the age that will represent the change in the physical state of pipes resulting from interventions (actions). Finally, pf_i is a probability of failure and rl_i is a reliability level, both, elicited from Equation 1.

Actions: We denote the action space as $\mathcal{A} = \{a_1, a_2, ..., a_m\}$ where, as before, m is the number of pipes. The objective of the agent is to find an optimal action depending on a given state, at each timestep, such that there is a minimum average economic cost and maximum reliability of assets within a planning horizon. The action $a_i^t \in [0, 1, 2]$ represents the discrete actions for an agent at each timestep for a pipe i, where $a_i^t = 0$ suggests *no maintenance* action, $a_i^t = 1$ represents the action *maintain*, and $a_i^t = 2$ denotes the *replacement* action.

Reward function: The DRL-based agent seeks to maximize the expected cumulative reward in a definite time-horizon following a sequence of actions [21]. Accordingly, we design the reward function to represent our user-specific objective. We construct the reward function with inverse values since we seek to minimise the overall intervention cost while improving the structural performance of the pipes. We denote the rewards as $\mathcal{R} = \{r_1, r_2, ..., r_m\}$ where m is number of total pipes under consideration. The reward

function is presented below:

$$R(s_t, a_t, s_{t+1}) = \begin{cases} 0 & \text{if } a_i^t = \text{Do nothing} \\ -0.8 & \text{if } a_i^t = \text{Replace} \\ -0.5 & \text{if } a_i^t = \text{Maintain and } pf > 0.5 \\ -1 & \text{if } a_i^t = \text{Maintain and } pf \leq 0.5 \\ -1 & \text{if } a_i^t = \text{Do nothing and } pf \geq 0.9 \end{cases}$$

where a is an action, t is a timestep and i represents an individual pipe. We introduce a penalty of -1 to discourage unnecessary maintenance actions. Similarly, to ensure a sufficient reliability level of pipes at all times, a penalty is given if the system is near failure, yet the agent chooses the action of doing nothing.

Simulated environment: The environment simulates the physical characteristics of multiple water pipes. Depending on the material, each pipe has a specific failure rate, which is obtained from [27]. We estimate the failure probability of water pipes using the exponential (Poisson) distribution [3] represented as:

$$pf_t = 1 - rl_t = 1 - e^{-\mu\, aux_t} \tag{1}$$

where pf_t is the probability of failure at time t, rl_t is the reliability of an asset, and μ is the failure rate, and aux_t is an auxiliary variable for the current age of the pipe. We opted for an auxiliary variable for the age to illustrate the improvement in the physical condition of the pipes resulting from interventions (i.e., actions). Depending on the age, material, and length, the pipes experience a certain deterioration induced by environmental and operational stresses in a different manner.

Dynamics (Time to transition): At any timestep, the agent receives a representation of the environment's state in the form of state $s_t \in \mathcal{S}$ where $s_t = \langle age, mat, L, FR, aux, pf, rl \rangle$. The agent responds with an action $a_t \in \mathcal{A}$, receives a reward r_t and moves to next state $s_{t+1} \in \mathcal{S}$. In the finite MDP, the next s_{t+1} and r_{t+1} have a discrete probability distribution dependent only the current s_t and a_t [21]. The dynamics function (also referred as state transition probability function) is defined as:

$$p(s', r|s, a) = \mathbb{P}(s_{t+1} = s', r_{t+1} = r|s_t = s, a_t = a) \tag{2}$$

In other words, the next state and reward are only dependent on the current state and action irrespective of all the previously visited states, thus respecting the Markovian property.

In our case of water pipes, at each timestep, the age of the pipe is increased by one year and the aux_t is updated depending on the chosen actions as follows:

$$aux_{t+1} = \begin{cases} aux_t + 1 & \text{if } a_i^t = \text{Do nothing} \\ aux_t - 10 & \text{if } a_i^t = \text{Maintain} \\ aux_t = 1 & \text{if } a_i^t = \text{Replace} \end{cases}$$

The auxiliary variable aux_{t+1} is

- incremented with one (year) in case of *no intervention*, depicting the increase in age and thus increase in pf
- reduced by 10 (years) as a result of *maintenance* to represent the improvement in physical state of the pipe, and
- set to one to depict the brand-new condition state of the pipe resulting from *replace* action.

Intuitively, we use the aux_{t+1} variable in order to avoid modification of original age values of the assets. The updated aux_{t+1} is used in Equation 1 to calculate the pf and rl at each timestep in order to represent the physical characteristics of the assets.

3.2 Deep Q-Network

The agent interactions with the environment generate a sequence of trajectories, which are denoted as:

$$s_0, a_0, r_1, a_1, s_2, a_2, r_2, s_3, a_3, r_3, \ldots$$

The goal of an agent is to find an optimal policy (sequences of actions) which maximizes the expected accumulated (discounted) return G_t, which is discounted sum of rewards. represented as $G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2}...$, where r_t is the reward received at time t, and $\gamma \in [0, 1]$ is a discount factor. The optimal Q-function (action-value function) must provide the maximum action values at all the states determined by Bellman optimality equation as follows [21]:

$$Q^*(s, a) = \mathbb{E}_{s'}[r + \gamma max_{a'} Q^*(s', a')] \tag{3}$$

where r is the immediate reward received, a' is the action to select a state s' that returns maximum reward. $\mathbb{E}$ is the expected value of a random variable given that agent follows the policy π. The optimal policy π is derived by Equation 3 in an iterative update such that the agent starts in state s and takes the highest return action a and follows the policy π for all future steps.

Q-learning may fall short for complex systems beyond standard grid examples. The seminal work of Mnih et al. [11] proposed to utilize deep neural networks as function approximator for estimating Q-functions for high-dimensional states spaces, in particular, to update a set of parameters θ to learn optimal Q-function for a policy π such that $Q^*(s, a) \approx Q(s, a|\theta)$. The network achieves this by minimizing the following loss function at each step k:

$$\mathcal{L}(s, a|\theta_i) = \mathbb{E}_{s,a,r,s'\ p(.)}[y_k - Q(s', a'; \theta_k))^2] \tag{4}$$

$$y_k = r + \gamma max_{a'} Q(s', a'; \theta_{k-1}) \tag{5}$$

where p represents the uniform distribution over the transitions tuple s, a, r, s' collected by agent for interacting to environment.

For a long time, utilizing neural networks for reinforcement learning remained an open research question due to the instability of network training and convergence [15]. The Atari DQN introduced *fixed target network* and *experience replay* training techniques. In *fixed target network*, the θ_{i-1} is kept frozen for certain iterations and is updated in predefined intervals. This is to avoid frequent updates, which improves the network's convergence capability. The second technique *experience replay* offers training stability and data efficiency. It recommends storing agent transition (s, a, r, s') into a circular data structure called the replay buffer. In each training iteration, instead of only the latest transition, the batch of transitions is sampled from the replay buffer to compute the loss and gradient.

To balance the exploration and exploitation trade-off, the DQN implements a epsilon-greedy policy which select greedy action (i.e. $a = \arg\max_{a \in \mathcal{A}(s)} Q(s, a)$) with probability 1 - ϵ and a random uniformly distributed action with ϵ probability.

Dueling Deep Q-Network is one of the several improvements that has been proposed for a performance enhancement in comparison to using standard DQN. Wang et al., [24] propose to replace

a single stream Q-network into two streams of state-value and advantage functions represented as follows :

$$Q(s,a) = V(s) + (A(s,a) - \frac{1}{\mathcal{A}} \sum_{a'} A(s,a)) \tag{6}$$

The value function $V(s)$ estimates the rewards given the state and chosen action. The advantage function $A(s,a)$ mainly informs the agent about the usefulness of the chosen action compared to the other actions. The rationale behind off-setting the advantage values with its mean is discussed in detail in [24].

It is important to note that the dueling DQN shares the same input and output as the standard DQN. The proposed additions of dueling DQN are implemented as part of the network, and it does not introduce additional algorithmic modifications.

4 EXPERIMENT AND RESULT

4.1 Network implementation

We develop a simulation environment of water pipes using the standard Open AI Gym custom environments. Instead of a single pipe, we simulate 16 independent pipes with varying age, material, length, and failure rates to mimic the real-world planning scenario.

There are several possible ways to obtain Q-values using neural networks. Besides the type and number of network layers, previous approaches seek to directly estimate the scalar values for each state-action pairs [15]. However, these architectures do not scale, and their cost increases linearly with the number of actions. Mnih et al. [11] proposed to use a separate output unit for each action, where the network predicts the Q-values of each action in a single forward pass. To further improve generalisability and efficiency, Wang et al., [24] proposed to estimate the state-value and advantage values before the Q-value. The intuition behind this that the state value does not vary significantly across various actions.

Our network architecture is mainly inspired by [11, 24, 25]. The network takes an input of size $16 \times 7 \times 1$ representing the number of pipes, state variables, and connector for convolution layers. The network consists of four convolution layers with a kernel size of 3×3 (first three layers) and 1×1 (last layer) and stride of 1×1 (first and fourth layer) and 2×2 (second and third layer). All four layers has `tanh` activation function. We apply a global average pooling to the last convolution layer followed by two fully-connected layers. The last fully-connected layer is split into values and advantage stream and fed as an input into an independent fully-connected layer with `tanh` activation function. The output of advantage and value streams are combined using Equation 6 to obtain the Q-value of each action. For training, the network seeks to minimize loss between target Q-value and predicted Q-value based on Equation 4 and Equation 5.

4.2 Training and evaluation

We train an agent to converge to an optimal rehabilitation policy of 100 years, where each timestep is a year. The training is performed for the 10,000 episodes with a replay buffer of size 1000. An episode finishes when the timestep reaches the 100^{th} year. We evaluate the quality of actions in a single trajectory (episode) by computing the discounted sum of rewards G_t, where the discount factor is set to $\gamma = 0.6$. The DQN seeks to obtain the maximum expected cumulative reward; therefore, we inverse the reward function to minimize the cost-related objective. The agent follows the ϵ-greedy behavior policy with ϵ annealed linearly from 1 to 0.01. We used the ADAM optimizer with a learning rate of 0.001 for the training.

Figure 2: The DQN agent performance during training with ϵ-greedy behaviour policy.

The training performance of the (dueling) DQN agent is presented in Figure 2. By the end of every episode, the costs (rewards) are accumulated to gauge the agent's overall learning performance. As it can be noticed, the performance starts to stabilizes after around 2000 episodes. The model high-cost spikes (in blue) represent the exploration characteristic of the ϵ-greedy policy. We report the results in the subsequent section with the optimal parameter configurations.

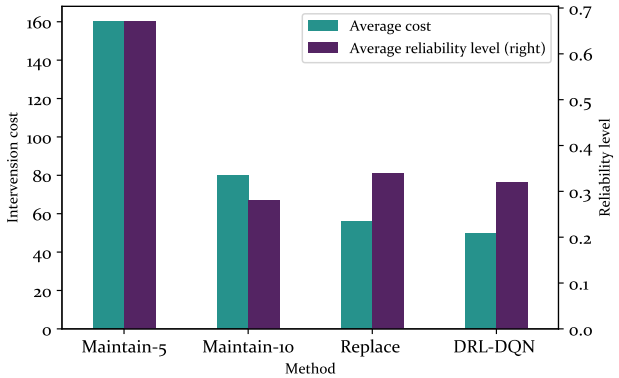

Figure 3: Comparison of DRL-based rehabilitation method with preventive and corrective approach. The graph shows the average cost *(lower values are preferred)* and average reliability level *(higher values are preferred)* for rehabilitation of 16 water pipes for the planning horizon of 100 years.

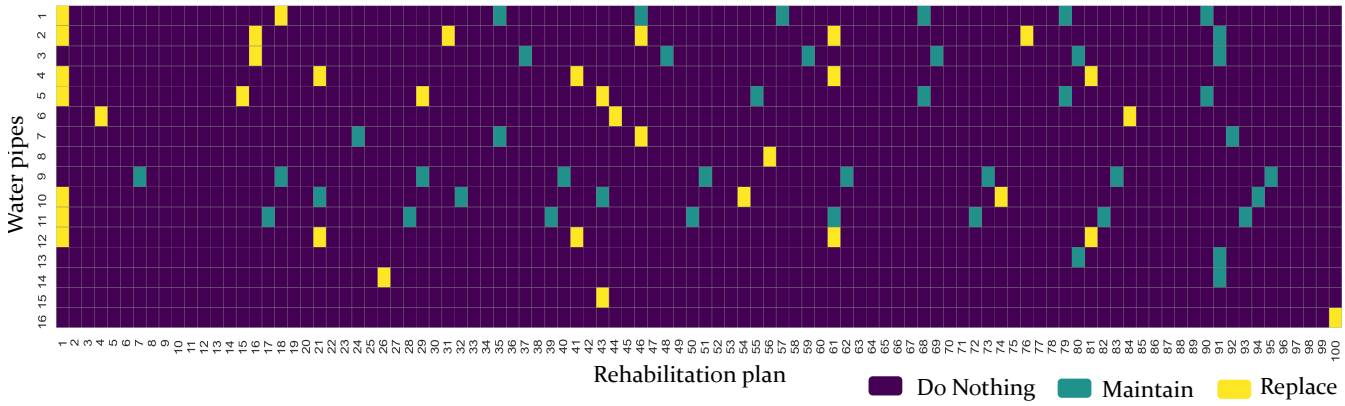

Figure 4: Optimal rehabilitation policy for distinct water pipes (rows) for a specified time period (columns).

4.3 Results

We model the rehabilitation planning problem of water pipes as a finite Markov decision process within the DRL framework. The agent learns an optimal rehabilitation policy as a result of extensive training of agent and environment interactions based on the (dueling) deep q-network. We term a policy `optimal` if an agent returns a minimum (average) cost and maximum reliability level of water pipes network for a planning horizon.

We present the optimal policy of 16 water pipes for the timeframe of 100 years in Figure 4. For discussion purposes, we report the initial state representation in Appendix 1. It is noted that the agent prefers `replace` action for the pipes with a low failure rate (e.g., see for pipe 15 and 16). For most of the pipes with an initial age of more than 40 years, a replacement is proposed at t_1. However, for the overall plan, the agent has proposed `Maintain` action 68 times, followed by `replace` action 25 times only. Since `Do nothing` action incurs no cost, this action is predominant in the rehabilitation plan, which is also aligned with the real planning situation.

Comparison with baselines: We establish baselines with preventive and corrective planning approaches to evaluate and compare the usefulness of employing the DRL framework for rehabilitation planning. The preventive planning approach is based on a recurring schedule where `maintain` action is performed to improve the performance state of assets. In the corrective approach, mainly `replace` action is executed after a failure.

For comparison, we develop a rehabilitation plan of 16 pipes for 100 year with preventive, i.e., Maintain-5 (every five years) and Maintain-10 (every ten years) and corrective (i.e., Replace) approach with similar costs, actions, and resulting performance improvements discussed in Section 3. Figure 3 shows the average costs and reliability level obtained by developing rehabilitation policy with different methods. We prefer the high reliability level with minimum cost. The *maintenance every five years* scenario results in highest average reliability level and highest cost. The ratio of average reliability and cost is noteworthy for the last three methods. The DRL-based approach yields a minimum cost of 50 with an average reliability level of 0.32, which is only 0.02 units less than the replace option and 0.04 better than the planned intervention of every ten years. The comparisons warrant the cost-effectiveness of rehabilitation planning of water pipes network with DRL framework.

5 CONCLUSION AND FUTURE WORK

We present a successful application of deep reinforcement learning (DRL) with a deep Q-network (DQN) agent for the management of the water pipes network. The DQN agent devises an optimal rehabilitation policy that incurs minimum average intervention cost and maximum reliability of the multiple assets. The trained agent effectively and efficiently outperforms classical maintenance planning, mainly preventive and corrective strategies, without the need for explicit expert knowledge and detailed heuristic rules. Besides optimal policy, the DRL framework based on the Markov decision process yields transparency in rehabilitation planning due to distinct definitions of states, actions, transition probability, and reward function.

The future work of this study will extend the application of DRL to include the interconnection among water pipes to reflect the network characteristics, such as the impact on the availability and surrounding households. The goal would be to devise a rehabilitation policy for water network segments instead of individual pipes. We seek to explore diverse deep learning architectures either as a part of value-based or policy-based DRL algorithms in order to capture the spatial information of the water pipes network successfully.

The DRL is a promising framework to model sequential decision-making problems. However, it remains an under-explored research area in asset infrastructure management due to the complexity of modeling simulated environments for multi-component structures. The increasing interest of the community towards digital twin technologies provides a useful testbed to explore and develop DRL-based solutions.

ACKNOWLEDGMENTS

This research has been partially funded by NWO under the grant PrimaVera NWA.1160.18.238.

REFERENCES

[1] Paul Almasan, José Suárez-Varela, Arnau Badia-Sampera, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2019. Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case. *arXiv* (2019), arXiv–1910.

[2] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *CoRR* abs/1912.06680 (2019). arXiv:1912.06680 http://arxiv.org/abs/1912.06680

[3] Alessandro Birolini. 2013. *Reliability engineering: theory and practice*. Springer Science & Business Media.

[4] Bram Cals, Yingqian Zhang, Remco Dijkman, and Claudy van Dorst. 2020. Solving the Order Batching and Sequencing Problem using Deep Reinforcement Learning. *arXiv preprint arXiv:2006.09507* (2020).

[5] Gustavo AP de Morais, Lucas B Marcos, José Nuno AD Bueno, Nilo F de Resende, Marco Henrique Terra, and Valdir Grassi Jr. 2020. Vision-based robust control framework based on deep reinforcement learning applied to autonomous ground vehicles. *Control Engineering Practice* 104 (2020), 104630.

[6] Nehal Elshaboury, Tarek Attia, and Mohamed Marzouk. 2021. Reliability Assessment of Water Distribution Networks Using Minimum Cut Set Analysis. *Journal of Infrastructure Systems* 27, 1 (2021), 04020048.

[7] Christian D Hubbs, Can Li, Nikolaos V Sahinidis, Ignacio E Grossmann, and John M Wassick. 2020. A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering* 141 (2020), 106982.

[8] Jong Woo Kim, Gobong Choi, Jung Chul Suh, and Jong Min Lee. 2015. Optimal scheduling of the maintenance and improvement for water main system using Markov decision process. *IFAC-PapersOnLine* 48, 8 (2015), 379–384.

[9] Y Kleiner, BJ Adams, and JS Rogers. 2001. Water distribution network renewal planning. *Journal of Computing in Civil Engineering* 15, 1 (2001), 15–26.

[10] Zheng Liu, Y Kleiner, B Rajani, L Wang, and W Condit. 2012. Condition assessment technologies for water transmission and distribution systems. *United States Environmental Protection Agency (EPA)* 108 (2012).

[11] 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. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[12] Matej Moravcík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael H. Bowling. 2017. DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker. *CoRR* abs/1701.01724 (2017). arXiv:1701.01724 http://arxiv.org/abs/1701.01724

[13] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

[14] Xinyi Ren, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Abhishek Gupta, Aviv Tamar, and Pieter Abbeel. 2019. Domain randomization for active pose estimation. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 7228–7234.

[15] Martin Riedmiller. 2005. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*. Springer, 317–328.

[16] Dina A Saad, Hany Mansour, and Hesham Osman. 2018. Concurrent bilevel multi-objective optimisation of renewal funding decisions for large-scale infrastructure networks. *Structure and Infrastructure Engineering* 14, 5 (2018), 594–603.

[17] Sattar Salehi, Mohammadreza Jalili Ghazizadeh, Massoud Tabesh, Somayeh Valadi, and Seyed Payam Salamati Nia. 2020. A risk component-based model to determine pipes renewal strategies in water distribution networks. *Structure and Infrastructure Engineering* (2020), 1–22.

[18] Uri Shamir and Charles DD Howard. 1979. An analytic approach to scheduling pipe replacement. *Journal-American Water Works Association* 71, 5 (1979), 248–258.

[19] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[20] Roland Smit, Jasper van de Loo, Martine van den Boomen, Nima Khakzad, Geert Jan van Heck, and ARM Rogier Wolfert. 2019. Long-term availability modelling of water treatment plants. *Journal of Water Process Engineering* 28 (2019), 203–213.

[21] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[22] Rita Ugarelli and Vittorio Di Federico. 2010. Optimal scheduling of replacement and rehabilitation in wastewater pipeline networks. *Journal of Water Resources Planning and Management* 136, 3 (2010), 348–356.

[23] Jiawei Wang and Lijun Sun. 2020. Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework. *Transportation Research Part C: Emerging Technologies* 116 (2020), 102661.

[24] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1995–2003.

[25] Shiyin Wei, Yuequan Bao, and Hui Li. 2020. Optimal policy for structure maintenance: A deep reinforcement learning framework. *Structural Safety* 83 (2020), 101906.

[26] D Wilson, Y Filion, and I Moore. 2017. State-of-the-art review of water pipe failure prediction models and applicability to large-diameter mains. *Urban Water Journal* 14, 2 (2017), 173–184.

[27] BA Wols, A Vogelaar, A Moerman, and B Raterman. 2019. Effects of weather conditions on drinking water distribution pipe failures in the Netherlands. *Water Supply* 19, 2 (2019), 404–416.

[28] Zahra Zangenhmadar, Osama Moselhi, and Sasan Golnaraghi. 2020. Optimized planning of repair works for pipelines in water distribution networks using genetic algorithm. *Engineering Reports* (2020), e12179.

[29] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher. 2020. The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332* (2020).

6 APPENDIX

Table 1 presents the initial state representation of the 16 water pipes. Failure rate with respect to pipe material is obtained from [27]. At t=1, the auxiliary age is equal to the start (given) age. Equation 1 is used to compute failure probability pf_t and and reliability level rl_t. At each time transition, the auxiliary age, failure probability and reliability level are updated depending on the chosen action by the agent.

Table 1: State representation of the water pipes network at timestep t=1.

Pipes	Age (year)	Material	Length (m)	Failure rate (km)	Auxiliary age	Failure probability	Reliability level
1	44	Asbestos cement	2365	0.06	44	0.998	0.002
2	46	Asbestos cement	2732	0.06	46	0.999	0.001
3	6	Asbestos cement	1908	0.06	6	0.497	0.503
4	42	Asbestos cement	1996	0.06	42	0.993	0.007
5	32	Ductile iron	1968	0.09	32	0.997	0.003
6	37	Ductile iron	2915	0.02	37	0.884	0.116
7	25	Ductile iron	2405	0.02	25	0.700	0.300
8	47	Ductile iron	1500	0.02	47	0.654	0.346
9	11	Gray cast iron	2017	0.07	11	0.788	0.212
10	30	Gray cast iron	1679	0.07	30	0.971	0.029
11	31	Gray cast iron	2071	0.07	31	0.989	0.011
12	45	Gray cast iron	1666	0.07	45	0.995	0.005
13	15	PVC	1650	0.015	15	0.310	0.690
14	40	PVC	2365	0.015	40	0.758	0.242
15	22	PVC	2434	0.015	22	0.552	0.448
16	2	PVC	1527	0.015	2	0.045	0.955