Structural Safety 80 (2019) 46–55

Contents lists available at ScienceDirect

Structural Safety

journal homepage: www.elsevier.com/locate/strusafe

Model-free reinforcement learning with model-based safe exploration: Optimizing adaptive recovery process of infrastructure systems

Milad Memarzadeh[a,*], Matteo Pozzi[b]

[a] Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720, USA
[b] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Keywords:
Reinforcement learning
Extreme events
Resilience
Infrastructure systems

ABSTRACT

Extreme events represent not only some of the most damaging events in our society and environment, but also the most difficult to predict. Model-based predictions of the disruptions induced by extreme events on urban infrastructure systems are often unreliable, as these events are unlikely by their very definition. Specifically, characterizing the effect of such disruptions to the urban infrastructure using a parameterized model is a difficult task. On the other hand, model-free approaches based on recent advancements in reinforcement learning can model the complex dynamics of urban society and infrastructure under the risk of extreme events explicitly without relying on any specific physics-based mechanism. However, these approaches usually require performing random exploration of the effects of management actions on the system (typically in the post-event situation) to allow for an acceptable approximation to the optimal management policy. When dealing with costly infrastructure systems and important communities, this random exploration can be unacceptable and risky. In this paper, we propose a method called Safe Q-learning, which is a model-free reinforcement learning approach with addition of a model-based safe exploration for near-optimal management of infrastructure system pre-event and their recovery post-event. Our method requires the decision-maker to model the structure of the state space of the problem, and a suitable equilibrium of the system (optimum functionality pre-event). This information is usually available for urban systems, as they spend long time in optimum equilibrium before the occurrence of such events. We show on several examples of infrastructure management how the proposed approach is able to achieve near-optimal performance without the risk due to random exploration.

1. Introduction

Resilience is a key aspect in the behavior of complex dynamical systems such as our built environment, and it indicates the system's capacity to withstand the disruptions caused by extreme events and to recover from it [17,8,5]. The definition of engineering resilience is concerned with disturbances that threaten the functional stability of engineering systems, and quick recovery to normal levels of functionality after a disruption [36]. In such definition, the resilience is measured usually based on four metrics: robustness, or the strength of the system to withstand a disturbance without functional degradation, redundancy, or substitutability of the system's components, resourcefulness, and rapidity, or the capacity to restore system to the normal functionality in a timely manner [3]. Quantification of such metrics requires a specific knowledge on the dynamics of the system as well as the risks and costs associated with actions that a decision-maker can take [11]. However, in many scenarios, the reaction mechanisms and the effects of an extreme event on a system is highly uncertain and it is hard to parameterize such mechanism and evaluate these metrics.

Exhaustive reviews of previous research on definition and quantification of resilience are presented by Hosseini, Barker, and Ramirez-Marquez [14] and Koliou et al. [15]. Here we focus on the few most recent studies. The literature that has studied how to quantify resilience largely focuses on either solutions based on network theory or parametric approaches based on reliability and risk analysis. On one side, network theory is used to identify the changes within the network of infrastructure systems to quantify its reliability, vulnerability and recover-ability in occurrence of an external disturbance [31,41,35,40,12]. For example, Ramirez-Marquez et al. [31] models the restoration of links among different components in the system, however, the recovery of the components themselves and its economic cost is not studied. Similarly, Zhang et al. [41] focuses on the design phase rather than the actual operation and maintenance of infrastructure systems under the risk of such disturbances. On the other hand, parametric non-linear

* Corresponding author.
E-mail address: miladm@berkeley.edu (M. Memarzadeh).

https://doi.org/10.1016/j.strusafe.2019.04.003
Received 23 October 2018; Received in revised form 28 April 2019; Accepted 29 April 2019
Available online 09 May 2019
0167-4730/ Published by Elsevier Ltd.

models have been used to describe the recovery from the disturbances based on historical data [32,1,25,8,27]. Given the peculiarity of each extreme event and the lack of sufficient historical observations, the generalization of such parametric approaches is limited. Moreover, these methods do not specifically provide a decision optimization platform for recovering after the distrubance caused by extreme event, that can incorporate the risk associated to the future events and lessons learned from observations of previous ones [16,30].

Decision theory and reinforcement learning, on the other hand, can incorporate the risk associated to such rare events prior to their occurrence and identify the optimal recovery strategy post-event. Recently, approaches based on decision theory and reinforcement learning have been utilized to quantify and mitigate the effect of extreme events on interdependent infrastructure systems [11,26,13]. For example, Faber et al. [11] provides a decision-theoretic framework to quantify the resilience and sustainability of infrastructure systems under the risk of extreme events. Nozhati et al. [26] and Gomez and Baker [13] utilize model-based reinforcement learning approaches to optimize the recovery process post-event from disturbances caused by extreme events for a network of interdependent infrastructure systems under the seismic hazard.

As mentioned before, model-based reinforcement learning approaches require some knowledge of the dynamics of the system, of costs related to actions taken by the manager and utilities, and effects on the system's functionality. To overcome such limitations, model-free reinforcement learning approaches has been employed [9,10], however not in relation to extreme events and to optimize the recovery process. Most of these models have been developed for regular operation and maintenance of infrastructure systems. A main reason for this is the high risk of applying model-free methods to critical infrastructure systems. Almost all of model-free approaches require performing random exploration (i.e., random strategies taken by the manager to maximize the speed of learning) to guarantee the convergence of the solution to the optimal policy, which may be too risky in application to critical urban infrastructure.

In this paper, we develop a model-free reinforcement learning approach with addition of a model-based safe exploration for near-optimal management of infrastructure system pre-event and their recovery post-event Our method requires the decision-maker to model the structure of the state space of the problem, and to define a suitable equilibrium of the system (i.e., an optimal level of functionality pre-event). This information is usually available for urban systems, as they spend long time in a stable configuration, which can be assumed as an optimum equilibrium, before occurrence of the event. The advantages of the proposed method are: (1) it can achieve a near-optimal performance without the risk due to random exploration, (2) the method is non-parametric (similar to other model-free reinforcement learning approaches), and as a result does not make significant assumptions about the parametric functions used to model the dynamics or the effect of extreme events, (3) similarly, it does not require any prior knowledge of the utility functions and economic costs of the actions taken by the manager, and (4) is usually computationally more efficient than model-based approaches and capable of real-time decision-making under uncertainty.

2. Methods

Before getting into details of the proposed method, we summarize the fundamentals of sequential decision optimization and reinforcement learning.

2.1. Sequential decision optimization

Decision theory has long been employed to optimize the operation and maintenance of infrastructure systems. Madanat [19] is among the early work of framing the optimal management of infrastructure systems as a model-based sequential decision-making problem under uncertainty. Since then, this family of approach is utilized for monitoring and control of many infrastructure systems such as transportation assets [33,20,21], structural components subject to fatigue [18], and corrosion [28], and wind turbines [6,22,23]. In such setting, the condition state of the infrastructure system is described by a variable, $s \in \mathbf{S}$, and the actions that manager can take to maintain the system are described by another variable, $a \in \mathbf{A}$. The deterioration behavior (i.e., the dynamics) of the system's state is modeled stochastically, i.e., $s_{t+1} = f(s_t, a_t) + \zeta_s^t$, where t denotes time steps, f defines the dynamics function, and ζ_s^t is a random variable modeling the uncertainty in predicting the future system state. The quality of actions taken by the manager is quantified by a utility function, which maps the states and actions spaces to real-valued numbers: $\mathbf{U}: \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$. The utility function quantifies the immediate costs/benefits for the decision-maker. The manager's goal is to find a policy that minimizes the long-term cumulative cost of operation and maintenance of the system over its entire life span, $\sum_{t=0}^{T} \gamma^t u(s_t, a_t)$, where T is the life-span of the system (also known as the management time horizon), and γ is the discount factor, relating the future costs to their net present value. In such setting, the manager's policy for maintaining the system is a mapping from state space to the action space: $\pi: \mathbf{S} \rightarrow \mathbf{A}$. At each time step t, the manager observes the current condition state (e.g., using sensors or by performing visual inspections), s_t, and identifies what action to take according to a specified policy , i.e., $a_t = \pi'(s_t)$. For such arbitrary policy $\pi'(s)$, the value of maintaining the system according to that policy starting from each specific state, s (i.e., the initial condition), $V^{\pi'}(s)$, can be calculated as follow,

$$V^{\pi'}(s) = u(s, \pi'(s)) + \gamma \sum_{s' \in \mathbf{S}} p(s'|s, \pi'(s)) V^{\pi'}(s') \tag{1}$$

where $p(s'|s, \pi'(s))$ is the probability of of reaching state s' at next step, if the current state is$s$, and policy π' is adopted. This probability is related to function f and the distribution of variables ζ. Formulating the problem in this way, the manager is interested in finding a policy (i.e., amaintenance strategy) that minimizes the operation cost over its entire life-span, i.e., the policy that minimizes the value function in Eq. (1)[2]:

$$\pi^*(s) = \text{argmin}_{a \in \mathbf{A}} \left[u(s, a) + \gamma \sum_{s' \in \mathbf{S}} p(s'|s, a) V^*(s') \right] \tag{2}$$

The above formulation is well-known as Markov decision processes (MDPs) [34] in decision theory and dynamic programming can be used to solve Eq. (2) and find the optimal policy. Fig. 1 shows the probabilistic graphical representation of MDPs.

2.2. Reinforcement learning

Reinforcement learning is an approach for sequential decision optimization when the models describing dynamics of the system and the utilities (or cost functions) are either uncertain or unknown. Hence, it is concerned with adaptive management and control in an uncertain environment. In such context, there is a decision-maker that controls a system within an uncertain environment by taking actions (i.e., planning) and adopts her actions based on receiving new information from the environment (i.e., learning). In the MDP setting defined in Section 2.1, the uncertainty is about the functions describing the dynamics of the system (e.g., the deterioration behavior of infrastructure components) and characterizing the utilities (cost of actions taken by the manager). The goal of the decision-maker is to interact with the environment by taking actions and adjust the policies based on the information she gets from the environment and hoping to identify the optimal policy in this process. Depending on how one formulates the problem, reinforcement learning is categorized into two main

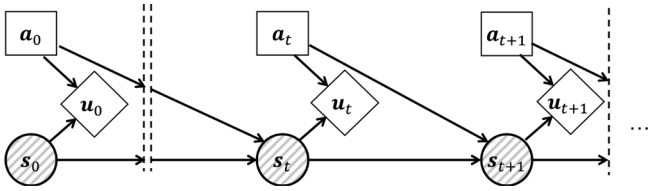

Fig. 1. Probabilistic graphical model of Markov decision processes. Circles show random variables, squares show decision variables, and diamonds show the utility variables. Shaded circles are the variables that are fully observable.

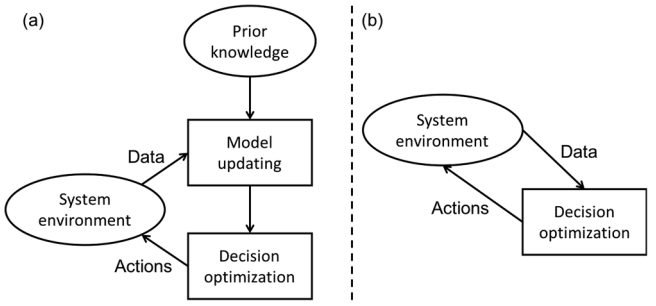

Fig. 2. Graphical representation of the steps in (a) model-based approaches and (b) model-free methods.

categories of model-based and model-free approaches. Fig. 2 illustrates the differences among these two. As mentioned before, the purpose of this paper is to propose a method that does not rely on specific mechanisms or function to describe the resilience and recovery of the infrastructure systems under the risk of extreme events. As a result, we will focus on the model-free reinforcement learning going forward; interested readers can refer to Wiering and Otterlo [22,38] for detailed description of model-based reinforcement learning.

Model-free reinforcement learning approaches learn the policy directly from the observations, without any explicit inference of the model (e.g., dynamics of the system, and utilities). They tend to learn better policies faster, when the prior knowledge is weak. In the next sections, we first review a classical model-free reinforcement learning approach and then will explain the fundamentals of deep reinforcement learning, before going to the proposed method of *Safe Q-learning*.

2.3. Q-learning

One of the classical model-free reinforcement learning approaches is Q-learning [37]. In Q-learning, the decision-maker represents the quality of taking each possible action, $a \in \mathbf{A}$, starting from any state, $s \in \mathbf{S}$ as a table, $Q(s, a)$, and updates this table by interacting with environment through taking actions and receiving observations. The Q-value can be defined using Eq. (2) as follow:

$$Q(s, a) = u(s, a) + \gamma \sum_{s' \in \mathbf{S}} p(s'|s, a) V^*(s') \tag{3}$$

At each time step t, given the current state of the system, s_t, the manager takes an action a_t, and receives two observations, the immediate utility u_t, and next resulting state, s_{t+1}. Based on this, she can update the Q-table as follow,

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [u_t + \gamma \min_{a \in \mathbf{A}} (Q(s_{t+1}, a)) - Q(s_t, a_t)] \tag{4}$$

where α is the learning rate. Once this inference on the Q-table is done, the next optimal action can be found as follow,

$$a^*_{t+1} = \text{argmin}_{a \in \mathbf{A}} Q(s_{t+1}, a) \tag{5}$$

Although, in practice, it is recommended to adopt an ϵ-greedy (or variations of that such as adaptive ϵ-greedy exploration or Boltzmann approach) action selection [34], to allow the agent to explore the unknown areas of the Q-table that can lead to better policies for future management. This random exploration is essential for convergence of the policy to the optimal one, otherwise the algorithm might iterate on a sub-optimal policy. However, when dealing with costly infrastructure systems under the risk of extreme events, random exploration is not acceptable. In Section 2.5 we explain how we replace this random exploration with a safer one.

There are two fundamental limitations to the classical Q-learning method: (1) in a high-dimensional problem, it is hard to maintain and perform inference on the large Q-table efficiently; and (2) this approach fails to provide any estimation of the Q-value for those areas of state space that are not observed.

2.4. Deep Q-learning

To overcome both limitations presented above, we adopt the Deep Q Network (DQN) approach [24] for solving operation and maintenance of infrastructure systems in a model-free setting. The idea here is to use a function approximator to estimate the Q-function, $Q(s, a;\theta) \approx Q^*(s, a)$, where $Q^*(s, a)$ is the exact Q-function, and θ lists the parameters of the function approximator. The function approximator can be any linear [10] or non-linear function, although in the deep reinforcement learning, a non-linear function approximator such as neural network is used. This neural network function approximator with weights θ is called Q-network. It can also generalize the estimation and predication to those areas of the state space that are unobserved, and can also work with high-dimensional problems efficiently. The interested reader should refer to [4] for further details on neural networks.

In the Q-network representation, the goal is to estimate the optimal target function, $Q^*(s, a) = u(s, a) + \gamma \min_{a'} Q^*(s', a')$, where s is the current state the decision-maker takes action a and pays cost $u(s, a)$, and ends up in state s' in the next time step. As a result, the parameters (i.e., the neural network weights) at each iteration i, i.e., θ_i, need to be adjusted to reduce the mean-squared error (or any other appropriate error measure) in the Bellman equation, where the optimal target values are substituted with approximate target values $y = u + \gamma \min_{a'} Q(s', a';\theta_i^-)$, using parameters θ_i^- from some previous iteration. As a result, the loss function at time i, $\mathscr{L}_i$, can be written as,

$$\mathscr{L}_i(\theta_i) = \mathbb{E}_{s,a,u,s'}[(y - Q(s, a;\theta_i))^2] \tag{6}$$

An important note here is that the target depends on the network weights (this is in contrast with respect to supervised learning). At each stage of the optimization, the weights are kept fixed from the previous iteration, θ_i^-, to update the i-th loss function $\mathscr{L}_i(\theta_i)$[24]. To optimize the above loss function, we use the stochastic gradient descent after every time step, which results to a form similar to the well-known Q-learning algorithm. Again, we note that the deep Q-learning needs the random exploration to converge to the exact optimal policy, which is not acceptable in application to critical infrastructure systems. In the next section we describe the proposed method to introduce safe exploration.

2.5. Safe Q-learning

One of the fundamental issues with Q-learning (either in the deep or in the classic version), as mentioned above, is related to the selection of actions from highly uncertain areas of state space (which is usually the case after occurrence of an extreme event). For example, for a system that has been in an equilibrium state for a long time, that area of the Q-table is well-known, while the other areas can be highly uncertain. If an extreme event pushes the system to those areas, how can we take advantage of the structure of the state space to turn the random exploration into a safe one?.

To solve this issue, we propose to add a term to the action-selection strategy in Eq. (5), which we call the “momentum towards the system’s

Safe Q-learning algorithm

Initialize memory **D**
Initialize Q-function **Q** with random weights θ
Initialize target Q-function $\hat{\mathbf{Q}}$ with weights $\theta^- = \theta$
Specify momentum function, $\mathcal{M}$ and η
Repeat (for each episode i):
  Initialize s_0 at time $t = 0$
  Repeat (for each time step t):

$$a_t = \underset{a \in A}{\operatorname{argmin}}\left[Q(s_t, a; \theta) + \eta^i \mathcal{M}(a, s_t, \mathbf{S}^*)\right]$$

Take action a_t, observe u_t and s_{t+1}

Store transition (s_t, a_t, u_t, s_{t+1}) in memory **D**

Sample random mini-batch of transitions (s_j, a_j, u_j, s_{j+1}) from **D**

Set $y_j = u_j + \gamma \min_{a'} \hat{Q}(s_{j+1}, a'; \theta^-)$

Perform a gradient descent step on $\left(y_j - Q(s_j, a_j; \theta)\right)^2$ with respect to the network parameters θ

Every C steps reset $\hat{\mathbf{Q}} = \mathbf{Q}$
Until t is terminal.

Fig. 3. Safe Q-learning algorithm.

equilibrium". We refer to equilibrium as system's optimal state before the occurrence of the event. Let us call this equilibrium of the system pre-event or state spaces around it a safe region, $\mathbf{S}^* \in \mathbf{S}$. Now, we revise the action selection strategy as follow:

$$a^* = \text{argmin}_{a \in \mathbf{A}}[Q^*(s, a) + \eta \mathcal{M}(a, s, \mathbf{S}^*)] \tag{7}$$

where $\mathcal{M}(a, s, \mathbf{S}^*)$ is the momentum towards the safe region, $\mathbf{S}^*$, from the current state, s, under continually taking action a, and η is a decay factor that controls the effect of momentum (e.g., it reduces its contribution when sufficient learning has been performed). It should be noted that the definition of momentum needs assumptions about dynamics of the systems and the structure of the state space, and it is related to a model-based approaches. Basically, we are combining model-free reinforcement learning with a model-based exploration to make the approach safer for application to critical infrastructure systems. A major limitation of the proposed method, in its current form, is that the definition of the momentum function $\mathcal{M}$ is problem-dependent, however, as we see later in the results, in the investigated problems, simple linear functions are sufficient to obtain near-optimal policies. Now we are ready to provide the full algorithm of the *Safe Q-learning* method proposed here in Fig. 3.

There are some fundamental differences in the algorithm with respect to the classic Q-learning approach that are worth detailed explanation. First, the reader can note that we store observations in a memory set **D** and each time we sample a mini-batch of transitions from this memory to use for fitting the model (this is sometimes referred to as experience replay [24]). The reasons are: (1) by using this, the previously observed transitions (especially the rare ones) will be re-sampled and as a result the learning process would be more efficient; and (2) Due to the high correlation among the transitions in a sequence, the Q-network function might diverge, if we fit the model on the sequential data. Re-sampling transitions from the memory breaks this correlation and as a result it reduces the variance of the updates and helps the convergence. The second difference is using a separate network for generating the targets y_j in the Q-learning update, i.e., $\hat{\mathbf{Q}}$. Then, every C steps we replace $\hat{\mathbf{Q}}$ with the learned **Q**. This helps the stability of the learning and avoids oscillations or divergence of the policy.

3. Results and discussion

We evaluate the performance of the proposed method, *Safe Q-learning*, comparing it to that of the regular Q-learning with different exploration approaches, and to the control with perfect knowledge on the system dynamics and utilities on multiple examples.

3.1. System under the risk of extreme events

In this section, we consider the management of a set of infrastructure components under the risk of extreme events. The system is made up by similar components, up to $N = 100$, exposed to extreme events. The system supplies a service to society, to meet its demand. We discretize time in weeks (meaning that Δt is 7 days). Demand, d, is unknown and modeled as a log-normal distribution, $d \sim \mathcal{LN}(\lambda_d, \zeta_d)$. System state defines the number of functioning components, so that there are n_t number of functioning components at time step t. Components deteriorate, and they are prone to failure when extreme event occurs. The change Δn_t in the number of functioning components from time step t to $t + 1$ is given by three contributions:

$$\Delta n_t = n_{t+1} - n_t = \Delta n_t^{(a)} - \Delta n_t^{(d)} - \Delta n_t^{(e)} \tag{8}$$

where, $\Delta n_t^{(a)}$ is the decision variable and defines the number of components to be repaired (or replaced), $\Delta n_t^{(d)}$ is the number of components damaged by deterioration and $\Delta n_t^{(e)}$ of those damaged by extreme events. Binary variable e_t defines the occurrence of an extreme event at time step t, and is Bernoulli distributed with rate ϕ, which is uncertain and modeled as $\phi \sim \text{Beta}(\alpha_\phi, \beta_\phi)$. Fig. 4 shows the graphical illustration of the dynamical system in this example. If an event occurs, the

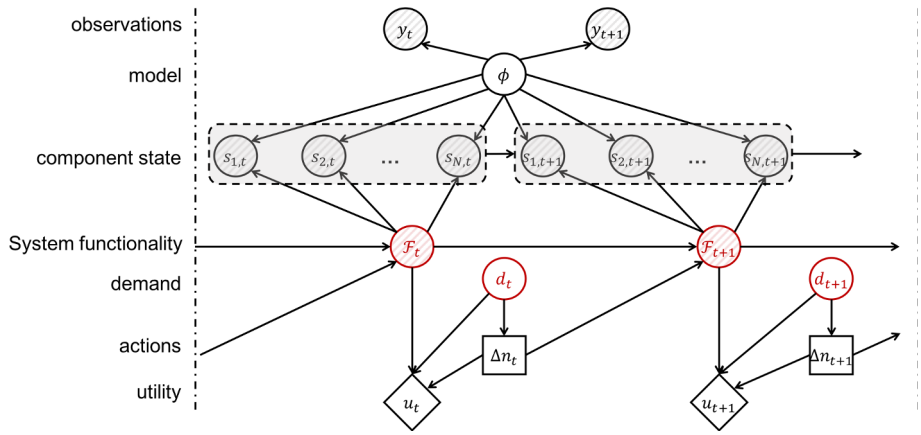

Fig. 4. Probabilistic graphical model of the example of set of infrastructure systems. Circles show random variables, squares show decision variables and diamonds show the utility variables. Shaded circles are the variables that are fully observable. $s_{k,t}$ is the state of component k at time t, y_t is the observations of the extreme event, $\mathcal{F}_t$ is the functionality of the system and d_t is the demand.

functioning components might get damaged according to an uncertain probability p_e, that defines the event's intensity and is beta-distributed: $p_e \sim \text{Beta}(\alpha_e, \beta_e)$. Then, we model the effect of extreme event on the functioning components as follow,

$$\begin{cases} \Delta n_t^{(e)} | [e_t = 0] = 0 \\ \Delta n_t^{(e)} | [e_t = 1] \sim \text{Binomial}\ (n_t, p_e) \end{cases} \tag{9}$$

Failure of components due to deterioration and its effect on the functionality of the systems is also similarly modeled as, $\Delta n_t^{(d)} \sim \text{Binomial}(n_t - \Delta n_t^{(e)}, p_d)$.

The utility function $\mathbf{U}$ is the sum of two cost contributions: repairing/replacing cost $\mathbf{C}_R$, and expected cost of insufficient functionality to meet the demand $\mathbf{C}_F$. replacing cost $\Delta n^{(a)}$ components is formalized as follow,

$$C_R(\Delta n^{(a)}) = c_0 I[\Delta n^{(a)} > 0] + c_r [\Delta n^{(a)}]^{\nu} \tag{10}$$

where I is the indicator function and c_r, c_0, and ν are the model parameters. Expected cost for insufficient functionality to meet the demand, $\mathbf{C}_F$, is a function of lacking components $\Delta n_{\text{lack}} = d - n_t$ is defined as,

$$C_F(n_t) = \mathbb{E}_d[\max\{c_{\text{pen}}(\Delta n_{\text{lack}})^{\mu}, 0\}] \tag{11}$$

where, c_{pen} and μ are model parameters.

3.1.1. Momentum definition

We define momentum function for this example as follow,

$$\mathscr{M}(a, s, \mathbf{S}^*) = \mathscr{B}(a, s, \mathbf{S}^*) - \text{Pen}(a) \tag{12}$$

where $\mathscr{B}(a, s, \mathbf{S}^*)$ is the loss for taking action a in the current state s, while the safe region is $\mathbf{S}^*$, and $\text{Pen}(a)$ is a penalty for expensive actions. For simplicity, we define the penalty function the same as the $\mathbf{C}_R$ defined in Eq. (9), where $\Delta n^{(a)} = a$. Since the state corresponds to the number of functioning components, we define the loss function, $\mathscr{B}$, as follow,

$$\mathscr{B}(s, a, \mathbf{S}^*) = (1 - \gamma^{\tau(s,a|\mathbf{S}^*)}) \widetilde{V}(\mathbf{S}^*) \tag{13}$$

where $\tau(s, a|\mathbf{S}^*)$ is the expected time to reach $\mathbf{S}^*$ from s, while continually taking action a, and $\widetilde{V}(\mathbf{S}^*)$ is an estimate of the value starting from the safe region. This estimate can be based on historical data. Function $\mathscr{B}(s, a, \mathbf{S}^*)$ measures how much value is lost before the system reaches the optimum equilibrium region $\mathbf{S}^*$. For this problem, it is reasonable to assume expected time as a linear function of the state space structure,

$$\tau(s, a|\mathbf{S}^*) = \left\lceil \frac{\mathbf{S}^* - s}{a} \right\rceil \tag{14}$$

3.1.2. Q-network implementation

The deep Q-network is implemented (Fig. 3) using Keras and tensorflow libraries in R, and we refer the reader to Chollet and Allaire [7] for further details on the implementation. Parameter $C = 20$, the number of samples of mini-batches of transition is set to 10, the decay factor for the momentum function is $\eta = 99.5\%$. The architecture of the neural network is as follow,

- Dense(40 neurons, activation = 'tanh')
- Dense(20 neurons, activation = 'tanh')
- Dense(10 neurons, activation = 'tanh')
- Dense(number of actions neurons, activation = 'linear')

Adam optimizer with a learning rate 0.001 is used for optimization using stochastic gradient descent and mean-squared errors is used as a metric.

3.1.3. Performance comparison

The parameters of Log-normally distributed uncertain demand are set to $\lambda_d = \log 60$ and $\zeta_d = 10\%$. The rate of the extreme event rate of occurrence is fixed so that the expected value and the coefficient of variation of ϕ are $0.2\%\text{w}^{-1}$ (corresponding to the return period of 10 years) and 70%, respectively, meaning that $\alpha_\phi = 2$, $\beta_\phi = 1000$. The intensity of the extreme event and its effect on system's functionality, p_e is assumed to have an expected value and the coefficient of variation of 50%, meaning $\alpha_e = \beta_e = 1.5$. p_d is 0.2%, so that the expected annual number of degraded components is 10 when $n = N$. Utility parameters are defined as follow: $c_0 = \$4\text{K}$, $c_r = \$10\text{ K}$, $\nu = 2$, $c_{\text{pen}} = \$1\text{ K}$, and $\mu = 2$. The discount factor is $\gamma = 99\%$ per week. The implemented Q-network, random exploration rate, and learning rate for classic Q-learning approach are the same as Section 3.1.

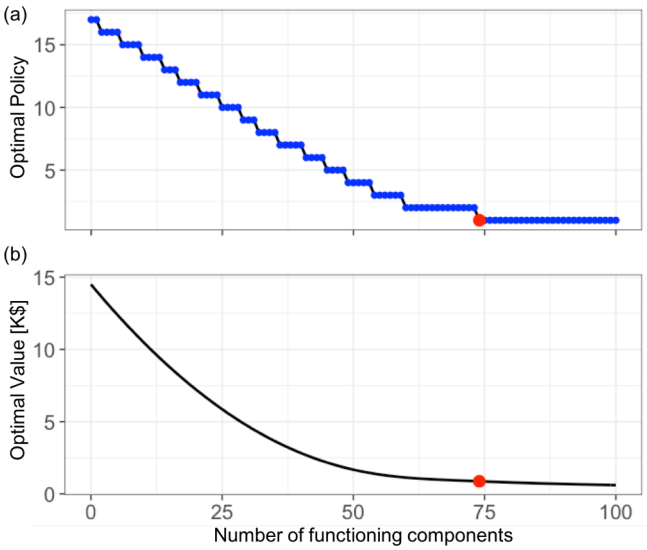

Fig. 5. The (a) optimal policy and (b) long-term expected cost (value), i.e., value of managing the system.

Fig. 5 reports (a) the optimal policy and (b) value (i.e., expected sum of discounted cost) if the decision-maker has perfect knowledge of the true models describing the dynamics of the system, the probability of extreme events and their effects, as well as the utility functions including penalties of not meeting the society's demand and costs associated with replacements, as derived from Eqs. (1) and (2). The red dot in the figure shows the equilibrium in the state space, where the system has an optimal performance and as a result the methods should keep the system around this equilibrium to guarantee the optimal operation and maintenance and meeting the demand of the society.

Fig. 6 shows the convergence of the learned Q-table according to the

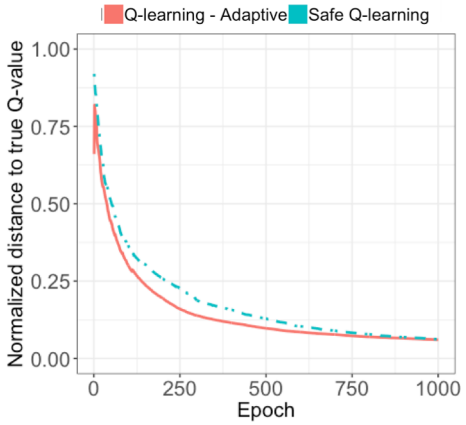

Fig. 6. The Euclidean distance between the true Q-table and the Q-table learned by Q-learning with adaptive ϵ-greedy exploration and Safe Q-learning approaches.

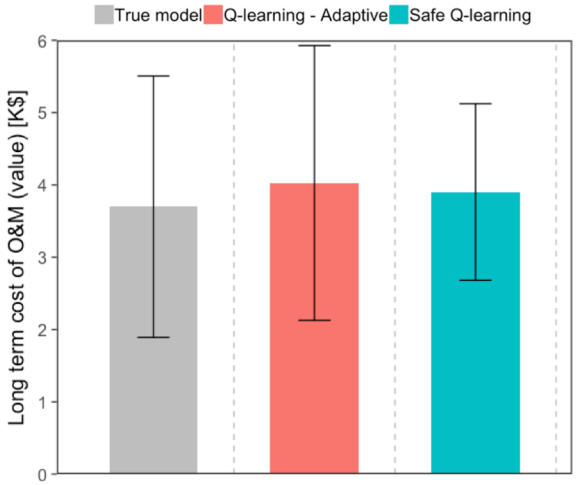

Fig. 7. Expected long-term utility of maintaining the system based on 100 independent simulations according to the policies assigned by each approach.

two methods of regular Q-learning with adaptive ϵ-greedy exploration and *Safe Q-learning* to the true one. We implement the adaptive ϵ-greedy exploration as follow:

$$\epsilon_i = \eta_\epsilon^{(i-1)} \times \epsilon_0 \tag{15}$$

where, $i \in \{1, \ldots, 1000\}$ denotes each episode of training, $\epsilon_0 = 0.2$, ϵ_i is the exploration probability at iteration i, and $\eta_\epsilon = 0.999$ is the decay factor. Fig. 7 shows the corresponding long-term expected utility of maintaining the system according to each model after the learning has done, based on 1000 independent simulations. As it can be seen, the proposed method performs slightly better than the regular Q-learning. Although the difference in the performance is not obvious here, we design an experiment in the next section, where the difference in performance is more significant and random exploration hurts the decision-maker far more than in this example.

After learning the policy, we plot a realization of the recovery process according to all models, after occurrence of an extreme event which takes the number of functioning components down to $n = 36$. Fig. 8 shows that the two methods recover the system to the safe region around the equilibrium level similar to the optimal policy. The blue dashed line represents the expected value of the demand by the society.

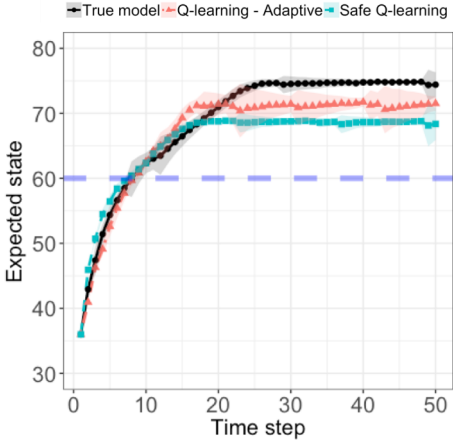

Fig. 8. Recovery process from an extreme event that takes the system down to 36 functioning components, according to the policy learned by each method, in comparison to the optimal recovery process. The blue dashed line represents the expected value of the demand by the society. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Due to the stochasticity in dynamics and demand and risk of failure caused by disturbances, all methods keep the number of components at a safe distance above the expected value of the demand.

3.2. Reservoir water irrigation problem

In this section, we present an example to show the significance of the proposed method compared to the regular Q-learning, where the random exploration hurts. Imagine the manager is in charge of a reservoir that irrigates water for agricultural use (this example is inspired by the reservoir operation example of Yakowitz [39]). The state of the system is the water level of the reservoir, that we discretize and normalize so that $s \in [0, 1]$. The capacity of the reservoir is fixed and set to $K = 0.8$. The water level changes either due to irrigation for agricultural use and by yearly rainfalls. The manager is in charge of selecting how much of the water to distribute for agricultural use, which defines the action space, i.e., $a \in [0, 1]$. The dynamics of the water level is defined according to following function:

$$s_{t+1} = \zeta_t\left(s_t + s_t\left(1 - \frac{s_t}{K}\right) - a_t\right) \tag{16}$$

where, $s \in \mathbf{S}$ denotes the water level of the reservoir, $a \in \mathbf{A}$ is the amount of water used for agricultural purposes, with subscript t denoting time steps, K is the capacity of the reservoir, and ζ_t captures the inherent stochasticity in the water level dynamics and follows a truncated Normal distribution with a unity mean and known standard deviation, which is assumed to be $\sigma^s = 0.1$ (the truncation is to avoid negative values). The manager receives rewards proportional to the amount of water the she distributes for agricultural use. The utility function is defined as $U(s_t, a_t) = \min(s_t, a_t)$. The choice of the utility is arbitrary and including the cost of irrigation would not change the results.

3.2.1. Momentum definition

In this example we define the momentum function as follow,

$$\mathscr{M}(a, s, \mathbf{S}^*) = \gamma^{\tau(s,a|\mathbf{S}^*)}\widetilde{V}(\mathbf{S}^*) \tag{17}$$

where $\tau(s, a|\mathbf{S}^*)$ is the expected time that takes to reach equilibrium level $\mathbf{S}^*$ from s, while continually taking action a, and $\widetilde{V}(\mathbf{S}^*)$ is the best estimate of the value (i.e., long-term expected rewards of management) for the safe state region based on historical data. Function $\mathscr{M}$ here defines the approximate value of performance in the current state, s, if action a is continually taken until the system reaches the optimal region, $\mathbf{S}^*$. We define the expected time as a linear function of the state space structure,

$$\tau(s, a|\mathbf{S}^*) = \lceil |s - \bar{\mathbf{S}}^* - a| \rceil \tag{18}$$

where it basically defines the distance between the current state, s, and the expected state-value of safe region of the state space, $\bar{\mathbf{S}}^*$, after a amount of water is irrigated.

3.2.2. Performance comparison

Fig. 9 shows the cumulative rewards of maintaining the system based on the policies assigned by the following methods: (1) *Safe Q-Learning*, (2) Q-learning with adaptive ϵ-greedy exploration (labelled as *Q-Learning - Adaptive*), and (3) Q-learning with Boltzmann exploration (labelled as *Q-Learning - Boltzmann*), compared to the optimal policy according to 1000 episodes of training. The Boltzmann exploration weights the exploratory action selection according to their value as follow,

$$p_t(a \mid s) = \frac{\exp(Q_t(s, a)/\text{temp})}{\sum_{a' \in A} \exp(Q_t(s, a')/\text{temp})} \tag{19}$$

where, $p_t(a \mid s)$ denotes the probability of taking action a in state s at time step t, $Q_t(s, a)$ is the best estimation of the Q-value of taking action

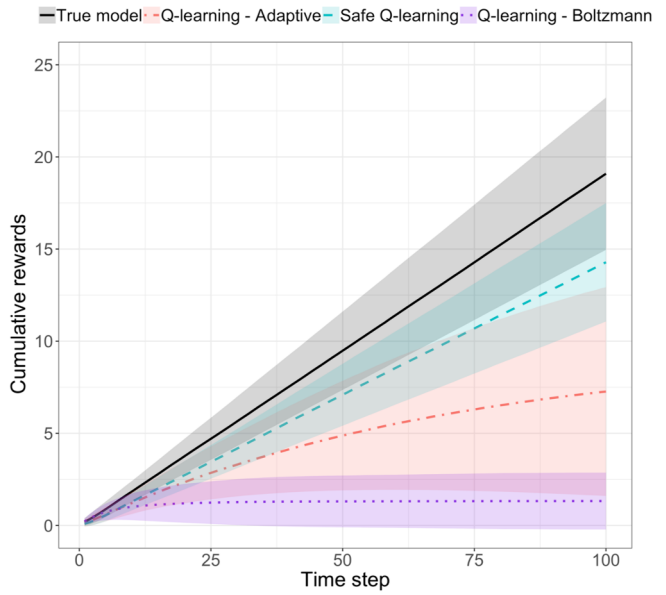

Fig. 9. Cumulative reward of managing the reservoir according the policy suggested by each method compared to the optimal policy.

a in state s at time step t and then following the optimal policy and can be computed according to Eq. (3), and *temp* is a temperature parameter, which is annealed over each episode of training.

As apparent in the graph, the exploration of the Q-learning approach (either the adaptive ϵ-greedy or the Boltzmann approach) hurts the system and results in significant loss in the revenue, while the safe exploration approach proposed here results in significantly better and near-optimal performance (on average safe Q-learning approach results in the 75% of the true model value, see Fig. 10). We also show the long-term expected reward of managing the reservoir under different policies (in Fig. 10), where the Q-learning results in a value significantly less than the ones achieved by the optimal policy and the safe exploration proposed here (the adaptive ϵ-greedy achieves 29.8% of the value of safe Q-learning while, the corresponding number for Boltzmann approach is 6.5%).

Moreover, it can be observed from Fig. 11 that both optimal policy and *Safe Q-learning* approach keep the reservoir's water in a steady level, while Q-learning with both exploration methods result in the decline on the water level in the long run and significant loss of the revenue. The results are based on 1000 episodes of training and we

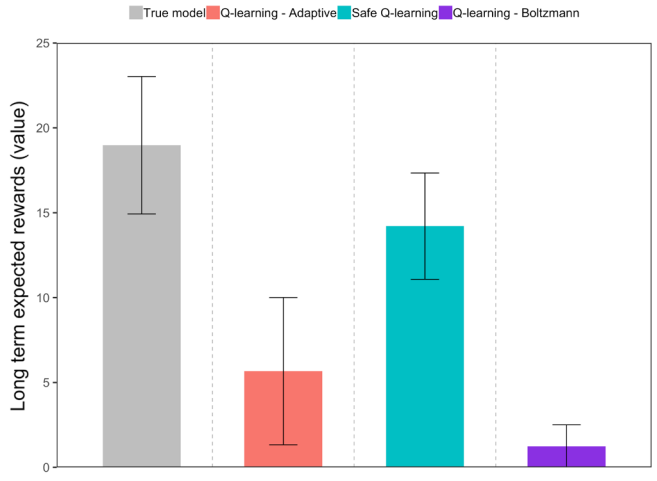

Fig. 10. Long-term expected reward (value) of managing the reservoir according to the policies defined by different methods.

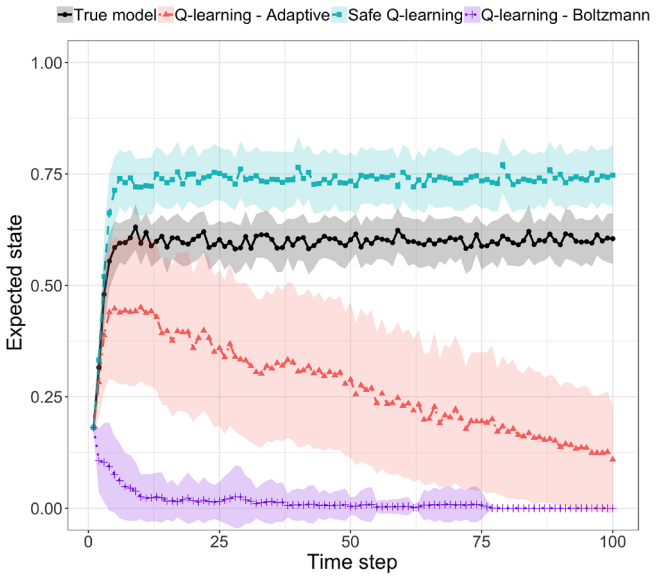

Fig. 11. Expected level of the water in the reservoir in the long run by managing it according to the policies suggested by different methods. The results are based on 1000 episodes of training and we show the mean $+/-$ one standard deviation.

show the mean $+/-$ standard deviation in the figures.

As mentioned before, the performance of *Safe Q-learning* approach relies on a good assumption of the safe region, $\mathbf{S}^*$. In Fig. 12, we evaluate the performance of this approach in a case where the prior information given to the method regarding the safe region of the state space, $\mathbf{S}^*$ is wrong or correct. In the case of correct prior information, the safe region is specified as an area around the true equilibrium of the system $s \approx 0.6$, while in the case of wrong prior information, it is specified as $s \approx 0.15$. It can be seen that the performance of *Safe Q-learning* method proposed here relies on the accuracy of the prior information on the location of the equilibrium and if such information is wrongly provided, it performs even worse than Q-learning.

3.3. Reinforced concrete bridge structure

In this section we apply the proposed *Safe Q-learning* approach to a real-world example of maintenance planning of a reinforced concrete bridge structure under the risk of extreme events. The data of this example is obtained from the work by Papakonstantinou and Shinozuka [29]. In this example the condition state of the structure is characterized by two variables: (1) four discrete conditions corresponding to the deterioration of the condition of the structural elements, and (2) 83 state variables corresponding to the deterioration rate of the structure. This results in an augmented state space with 332 states, $|S| = 4 \times 83 = 332$. Four different actions are available to the decision-maker which are *Do Nothing*, *Minor Repair*, *Major Repair*, and *Replace*. Doing nothing results in the structure deteriorating, while repairing improves the condition of the structure and replacement result in a intact structure. Costs for each action are fixed realistically based on the condition level of the structure and are valued according to their expected effect on structure's condition. Costs of minor repairing according to the data is estimated to be \$60, \$110, \$160, and \$280 depending on the condition state of the structure. Similarly, cost of major repairing is estimated at \$105, \$195, \$290, and \$390. Replacement would cost the manager \$820, and the penalty occurred to the manager due to lack of service to the society is estimated to be \$4. 75, \$40, \$120, and \$250 for each condition states. For further details regarding this example, we refer to Papakonstantinou and Shinozuka [29]. It should be noted that in the original example of Papakonstantinou and Shinozuka [29], the effect of extreme events are not

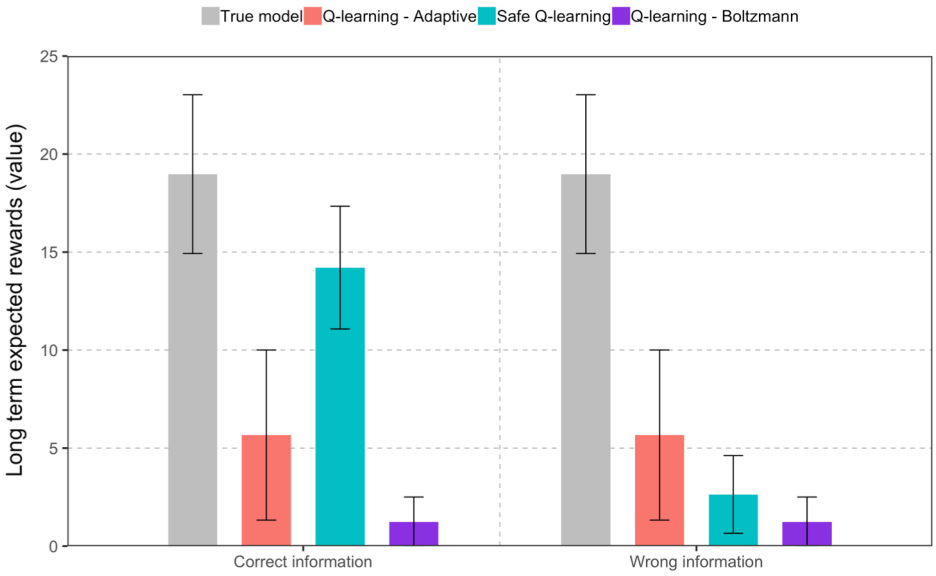

Fig. 12. Comparing the performance of the safe Q-learning approach to Q-learning with either adaptive ϵ-exploration or Boltzmann approach, and the true model where the prior knowledge on the equilibrium of the system given to the safe Q-learning is wrong or correct.

modeled and, in order to add such effect, we have assumed that an extreme event with a probability of 5% can alter the deterioration of the structure significantly. Specifically, we assume that the deterioration rate increases by 50 states in the case that the extreme event occurs (without the extreme event the deterioration rate increases by one state at each time step).

In this example we define the momentum function as follow,

$$\mathscr{M}(a, s, \mathbf{S}^*) = \gamma^{\tau(s,a|\mathbf{S}^*)}\widetilde{V}(\mathbf{S}^*) \tag{20}$$

where $\tau(s, a|\mathbf{S}^*)$ is the expected time that takes to reach equilibrium level $\mathbf{S}^*$ from s, while continually taking action a, and $\widetilde{V}(\mathbf{S}^*)$ is the best estimate of the value (i.e., long-term expected cost of management) for the safe state region based on historical data. We define the expected time as a linear function of the state space structure,

$$\tau(s, a|\mathbf{S}^*) = \frac{1}{a}|s - \bar{\mathbf{S}}^*| \tag{21}$$

Fig. 13 shows (A) the expected state of the structure under management based on each method, and (B) the long-term expected cost of maintaining the structure in a good condition. As expected, our proposed approach keeps the system in a condition state identical to the true model, however with a 33% higher maintenance cost compared to

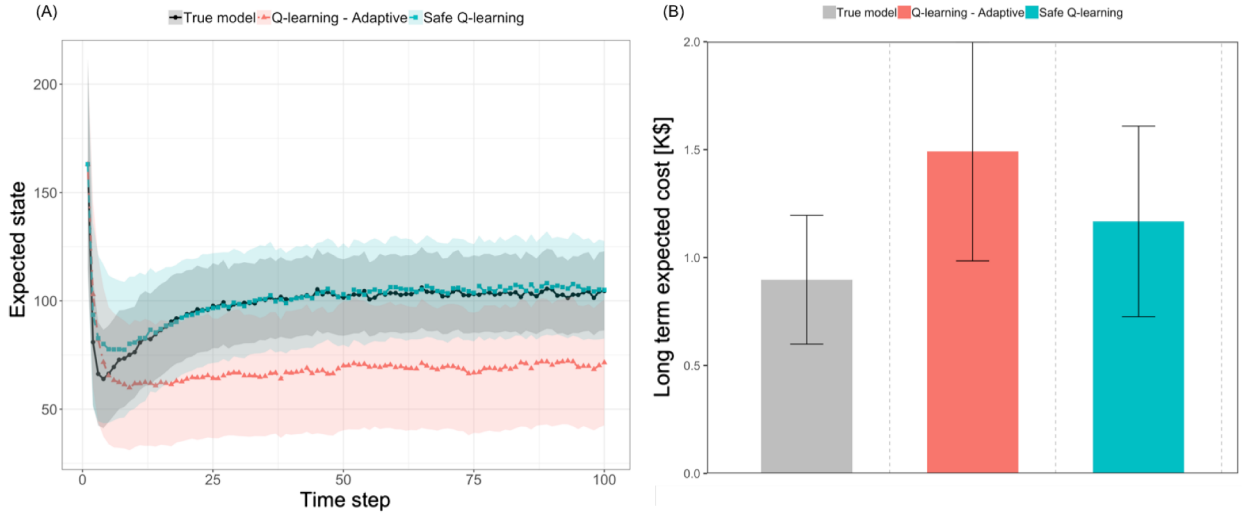

Fig. 13. (A) the expected state and (B) long-term expected cost of maintenance for management of the reinforced concrete bridge structure under the risk of extreme events.

the true model. In this example, Q-learning with adaptive ϵ-greedy exploration results in approximately 70% higher maintenance cost compared to the true model.

4. Conclusions

In this paper, we have proposed a model-free reinforcement learning approach with a model-based safe exploration, *Safe Q-learning*, for optimizing the operation of urban infrastructure under the risk of extreme events and the recovery process post-event. The main advantage of the method, compared to the literature, is replacing the random exploration of the model-free approaches with a safe exploration, which is crucial in application to costly urban infrastructure. Moreover, the proposed method is non-parametric and hence it does not make any assumption about parametric functions used to model the dynamics of the system, the effect of extreme events on the system, and the utilities (which are usually unknown in real-world scenarios). Relying on numerical simulations on several infrastructure management examples, we quantify the performance of the proposed method and we compare it to the optimal performance, assessing its advantage with respect to the traditional method of Q-learning with different exploration strategies.

Although the proposed method here is promising, there are a few caveats that need to be addressed as future direction of this research: (1) The current approach assumes full observability of the condition state of the infrastructure components. In many situations, this assumption might be violated due to existence of the measurement error or incomplete observations of the full state. In those settings, the manager has access to only noisy measurement of the states. Further advancement of the current proposed approach to deal with partial observability of the state in these problems is part of the future work; (2) the deep Q-network such as the one implemented in this paper suffers from instability and convergence issues [24]. The adjustment and tuning of the parameters in the optimization, and the design of the neural network architecture needs careful consideration and currently is a trial and error procedure. Further research needs to be done to improve the stability of these methods and to make the optimization model robust to noise, and the nature of the problem under study that would help generality and flexibility of these methods and their applications; and (3) the momentum function introduced in this paper is problem-dependent and it relies intrinsically on an accurate assumption about the safe region pre-event. Further work needs to be done to make the current formulation robust with respect to the lack of accuracy in the prior knowledge.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1663479. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Barabadi A, Ayele YZ. Post-disaster infrastructure recovery: prediction of recovery rate using historical data. Reliab Eng Syst Saf 2018;169:209–23.
[2] Bellman RE. Dynamic programming. Princeton, NJ: Princeton University Press; 1957.
[3] Berneau M, Chang S, Eguchi R, Lee G, Orourke T, Reinhorn A, Shinozuka M, Tierney K. A framework to quantitatively assess and enhance the seismic resilience of communities. Earthquake Spectra 2003;19:733–52.
[4] Bishop CM. Pattern recognition and machine learning. LLC, Singapore: Springer Science + Business Media; 2006.
[5] Bristow DN, Hay AH. Graph model for probabilistic resilience and recovery planning of multi-infrastructure systems. J Infrastruct Syst 2017;23(3):04016039.
[6] Byon E, Ding Y. Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process. IEEE Trans Power Syst 2010;25(4):1823–34.
[7] Chollet F, Allaire JJ. Deep learning with R. Shelter Island, NY: Manning Publications Co.; 2018. p. 11964.
[8] Cimellaro G, Reinhorn A, Bruneau M. Framework for analytical quantification of disaster resilience. Eng Struct 2010;32:3639–49.
[9] Durango PL, Madanat SM. Optimal maintenance and repair policies in infrastructure management under uncertain facility deterioration rates: an adaptive control approach. Transp Res Part A 2002;36:763–78.
[10] Durango-Cohen PL. Maintenance and repair decision making for infrastructure facilities without a deterioration model. J Infrastruct Syst 2004;10(1):1–8.
[11] Faber MH, Miraglia S, Qin J, Stewart MG. Bridging resilience and sustainability – decision analysis for design and management of infrastructure systems. Sustain Resilient Infrastruct 2018. https://doi.org/10.1080/23789689.2017.1417348.
[12] Goldbeck N, Angeloudis P, Ochieng WY. Resilience assessment for interdependent urban infrastructure systems using dynamic network flow models. Reliab Eng Syst Saf 2019;188:62–79.
[13] Gomez C, Baker JW. An optimization-based decision support framework for coupled pre- and post-earthquake infrastructure risk management. Struct Saf 2019;77:1–9.
[14] Hosseini S, Barker K, Ramirez-Marquez JE. A review of definitions and measures of system resilience. Reliab Eng Syst Saf 2016;145:47–61.
[15] Koliou M, van de Lindt JW, McAllister TP, Ellingwood BR, Dillard M, Cutler H. State of the research in community resilience: progress and challenges. Sustain Resilience Infrastruct 2018. https://doi.org/10.1080/23789689.2017.1418547.
[16] Lee JY, Burton HV, Lallemant D. Adaptive decision-making for civil infrastructure systems and communities exposed to evolving risks. Struct Saf 2018;75:1–12.
[17] Linkov T, Bridges T, Creutzig F, Decker J, Fox-lent C, Kroger W, Lambert J, Levermann A, Nathwani MBJ, Nyer R. Changing the resilience paradigm. Nat Clim Change 2014;4:407–9.
[18] Luque J, Straub D. Risk-based optimal inspection strategies for structural systems using dynamic Bayesian networks. Struct Saf 2019;76:68–80.
[19] Madanat S. Optimal infrastructure management decision under uncertainty. Transp Res Part C 1993;1(1):77–88.
[20] Medury A, Madanat S. Incorporating network considerations into pavement management systems: a case for approximate dynamic programming. Transp Res Part C 2013;33:134–50.
[21] Memarzadeh M, Pozzi M. Value of information in sequential decision making: Component inspection, permanent monitoring and system-level scheduling. Reliab Eng Syst Saf 2016;154:137–51.
[22] Memarzadeh M, Pozzi M, Kolter JZ. Optimal planning and learning in uncertain environments for the management of wind farms. J Comput Civil Eng 2014;29(5):04014076. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000390.
[23] Memarzadeh M, Pozzi M, Kolter JZ. Hierarchical modeling of systems with similar components: a framework for adaptive monitoring and control. Reliab Eng Syst Saf 2016;153:159–69.
[24] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, et al. Humen-level control through deep reinforcement learning. Nature 2015;518:529–33.
[25] Nan C, Sansavini G. A quantitative method for assessing resilience of interdependent infrastructures. Reliab Eng Syst Saf 2017;157:35–53.
[26] Nozhati S, Sarkale Y, Ellingwood B, Chong EKP, Mahmoud H. Near-optimal planning using approximate dynamic programming to enhance post-hazard community resilience management. Reliab Eng Syst Saf 2019;188:116–26.
[27] Ouyang M, Dueas-Osorio L, Min X. A three-stage resilience analysis framework for urban infrastructure systems. Struct Saf 2012;36–37:23–31.
[28] Papakonstantinou KG, Shinozuka M. Optimum inspection and maintenance policies for corroded structures using partially observable Markov decision processes and stochastic, physically based models. Probab Eng Mech 2014;37:93–108.
[29] Papakonstantinou KG, Shinozuka M. Planning structural inspection and maintenance policies via dynamic programing andMarkov processes, Part II: POMDP implementation. Reliab Eng Syst Saf 2014;130:214–24.
[30] Pozzi M, Memarzadeh M. A sequential decision making prospective on resilience. ICOSSAR, 12th International Conference on Structural Safety and Reliability. 2017. p. 2633–40.
[31] Ramirez-Marquez JE, Rocco CM, Barker K, Moronta J. Quantifying the resilience of community structures in networks. Reliab Eng Syst Saf 2018;169:466–74.
[32] Sharma N, Tabandeh A, Gardoni P. Resilience analysis: a mathematical formulation to model resilience of engineering systems. Sustain Resilient Infrastruct 2018;3(2):49–67.
[33] Smilowitz K, Madanat S. Optimal inspection and maintenance policies for infrastructure networks. Comput-Aided Civil Infrastruct Eng 2000;15:5–13.
[34] Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge,

Massachusetts: The MIT Press; 1998.
[35] Tran HT, Balchanos M, Domercant JC, Mavris DN. A framework for the quantitative assessment of performance-based system resilience. Reliab Eng Syst Saf 2017;158:73–84.
[36] Wang C, Blackmore J. Resilience concepts for water resource systems. J Water Resour Planning Manage 2009;135:528–36.
[37] Watkins CJCH. Learning from delayed rewards. Cambrdige, UK: Cambridge University Press; 1989.
[38] Wiering M, van Otterlo M. Reinforcement learning: state-of-the-art. Springer; 2012.
[39] Yakowitz S. Dynamic programming applications in water resources. Water Resour Res 1982;18(4):673–96.
[40] Zhang W, Wang N. Resilience-based risk mitigation for road networks. Struct Saf 2016;62:57–65.
[41] Zhang X, Mahadevan S, Sankararaman S, Goebel K. Resilience-based network design under uncertainty. Reliab Eng Syst Saf 2018;169:364–79.