Decision making for road infrastructures in a network based on a policy gradient method

Kotaro Sasai, Luc E Chouinard, Gabriel J Power, David Conciatori, Nicolas Zufferey

Cite this article
Sasai K, Chouinard LE, Power GJ, Conciatori D and Zufferey N
Decision making for road infrastructures in a network based on a policy gradient method.
Infrastructure Asset Management,
https://doi.org/10.1680/jinam.23.00045

Research Article
Paper 2300045
Received 26/09/2023; Accepted 30/04/2024
First published online 15/05/2024

Infrastructure Asset Management

Decision making for road infrastructures in a network based on a policy gradient method

Kotaro Sasai
Department of Civil Engineering, McGill University, Montreal, QC, Canada (corresponding author: kotaro.sasai@mail.mcgill.ca)

Luc E. Chouinard
Department of Civil Engineering, McGill University, Montreal, QC, Canada

Gabriel J. Power
Département de Finance, Assurance et Immobilier, Université Laval, Quebec, QC, Canada

David Conciatori
Département de Génie Civil et de Génie des Eaux, Université Laval, Quebec, QC, Canada; Laboratoire ICube CNRS UMR 7357, Département Génie Civil, INSA de Strasbourg, Strasbourg, France

Nicolas Zufferey
Geneva School of Economics and Management, University of Geneva, Geneva, Switzerland

Developing proper maintenance and rehabilitation investment plans is vital for prolonging the service life of road infrastructures while preserving the required service level under capital constraints. This paper proposes a reinforcement learning approach for determining an optimal policy of selecting maintenance, repair and rehabilitation alternatives for a network of road infrastructure facilities. The proposed approach is based on a policy gradient method and overcomes the computational complexity of optimisation problems due to a large number of possible combinations of network conditions and maintenance, repair and rehabilitation alternatives. The developed optimal management policy takes into consideration interdependencies among infrastructure facilities in a road network. Numerical studies on concrete bridge decks in road networks are performed to demonstrate the advantage, feasibility and capability of the proposed approach.

Keywords: infrastructure planning/Markov decision process/rehabilitation, reclamation & renovation/reinforcement learning/transport planning

Notation

a	action
$b(s)$	baseline function for state s
G_t	return following time t
$h(s, a, \boldsymbol{\theta})$	preference for selecting action a in state s based on $\boldsymbol{\theta}$
I	number of iterations
$J(\boldsymbol{\theta})$	performance measure for policy $\pi_{\boldsymbol{\theta}}$
M	mini-batch size
$q_\pi(s, a)$	value of state-action pair s, a under policy π
r	reward
s, s'	states
$v_\pi(s)$	value of state s under policy π
$\boldsymbol{w}$	weight parameter for a state value function
$\boldsymbol{x}$	feature vector
$z_{(m)}$	episode in a mini-batch m
α	step size parameter
γ	discount rate
$\boldsymbol{\theta}$	parameter for a policy
π	policy

Introduction

Decision making in road infrastructure management is a complex process for determining what types of maintenance, rehabilitation and replacement (MR&R) actions should be selected and when and where the selected actions should be performed. Planning an optimised MR&R strategy for road infrastructures is a daunting challenge involving the evaluation of MR&R projects – construction costs, project risks and improved service level after implementation – and that of their ramifications – traffic disruptions, accident risks, environmental impacts and so on. The importance of MR&R planning has been exacerbated by the global phenomenon of ageing infrastructure and concomitant loss of structural health. The need for effective decision-support tools is well documented in the literature (Frangopol and Liu, 2007; Kulkarni and Miller, 2003; Uddin *et al*., 2013; Vanier, 2001).

It has been a common practice to formulate an infrastructure-management problem as a Markov decision process (MDP) because of the use of condition states for the interpretation of inspection and evaluation data, which are ubiquitous in bridge- and pavement-management systems (Golabi and Shepard, 1997; Kulkarni and Miller, 2003). The MDP framework is amenable to analytical solutions for the optimisation of infrastructure-management activities through operations research methods, such as linear programming and dynamic programming (Camahan *et al*., 1987; Golabi *et al*., 1982; Gopal and Majidzadeh, 1991; Jesus *et al*., 2011; Smilowitz and Madanat, 2000). These programming methods are computationally efficient in determining an optimal management policy for a road network of limited size, but they perform less well when the road network is large, due to the 'curse of dimensionality' unless they are restructured to subnetworks for analytical purposes.

Studies have been proposed to deal with optimisation problems beyond the capabilities of traditional formulations of programming methods. A common approach in the road-infrastructure-management context is the use of metaheuristic methods such as genetic algorithms (e.g. Chan *et al*., 1994; Fwa

et al., 1994; Morcous and Lounis, 2005). Their computational efficiency is appealing; metaheuristic methods can explore various situations involving interdependencies among road infrastructure facilities in a network. However, metaheuristic methods inherently lack mathematical bases and do not guarantee near-optimal solutions. Other common approaches are modified programming methods such as bi-level programming (Chu and Chen, 2012; Hajibabai *et al*., 2014; Ng *et al*., 2009). These modified programming methods provide mathematical foundation for offering optimal solutions; however, they are usually NP-hard (Bard, 2013). The third approach is approximate solution methods from reinforcement learning. These approximate methods are MDP-based techniques, which aim to overcome the challenges associated with high dimensions faced by programming methods. Unlike programming methods, which are often referred to as tabular methods because they aim to fill and update tables of state-action pairs, approximate solution methods aim to generalise sensible decisions for large state and action spaces. This type of approach has recently gained more popularity in the literature (examples are introduced in the following section), mostly due to its MDP-based mathematical bases and efficiency in updating parameters to determine a good approximate solution. Reinforcement learning with approximation methods has the potential to be applied to various road-infrastructure-management problems to which traditional formulations are not applicable.

Background

Reinforcement learning is an area of machine learning concerned with how an agent must act in an environment to maximise the cumulative reward. The agent has little or no prior knowledge on which actions to take and discovers which actions yield the most reward by trial and error. The actions taken by the agent may affect not only the immediate reward but also the environment and, through that, all subsequent rewards. Trial-and-error search and delayed rewards are the two most distinguishing features of reinforcement learning (Sutton and Barto, 2018). Figure 1 shows the agent–environment interaction in the reinforcement learning framework. The goal of reinforcement learning is for the agent to learn an optimal policy. A policy is a decision making rule mapping from perceived states of the environment to available actions. The optimal policy is a policy that maximises the cumulative reward or more generally the reward function.

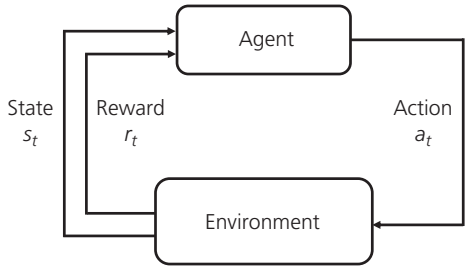

Figure 1. Agent–environment interaction in reinforcement learning

MDPs are a classical formalisation of sequential decision making pertaining to determining the optimal policy and a mathematically idealised form of the reinforcement learning problem. An MDP is a five-tuple (S, A, P_a, R_a, T), where S is the state space; A is the action space; P_a is the transition probability from state s at time t to state s' at time $t + 1$ when action a is selected; R_a is the immediate reward received by transitioning from state s to state s' due to an action a; and T is the set of decision epochs for finite horizon problems. A value function, $v_\pi(s)$, gives the value of a state s, under a policy denoted as π. $v_\pi(s)$ is the expectation of cumulative rewards for the remaining periods:

$$v_\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s\right] \quad (1)$$

where $\mathbb{E}[\cdot]$ denotes the expected value given that the decision maker follows policy π at decision epoch t and γ is the discount factor $0 \leq \gamma \leq 1$. Value functions satisfy particular recursive relationships, known as Bellman equations:

$$v_\pi(s) = \sum_a \pi(as)\sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_\pi(s')\right] \quad (2)$$

The objective of an MDP is to find an optimal policy π^*, which maximises the value function:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} v_\pi(s) \quad (3)$$

for all $s \in S$.

Consider a road network of N road infrastructure facilities and that the condition state of each facility is described by one of the set C of condition rating states. Let X denote the networkwide aggregated condition describing the conditions of all N facilities. Then, the set X would contain $|C|^N$ states, and Equation 2 would have to be evaluated for each of $|C|^N$ states. Moreover, Equation 2 requires evaluation of the whole action space with $|A|^N$ possible combinations. Therefore, obtaining an exact solution of an MDP for a road network might be infeasible for public agencies since agencies generally manage a large number of infrastructure facilities in a network. In addition, a large-scale MDP such as a road-infrastructure-network-management problem requires not only a designed algorithm to deal with the large state space but also network-level consideration such as underlying network configurations. Dekker *et al*. (1997) propose three classifications of interdependencies between infrastructure facilities in a network: economic dependencies (the benefits/costs of management on an

individual facility are affected by the conditions of other facilities), structural dependencies (network system performance such as connectivity and capacity is collectively determined by facilities) and stochastic dependencies (correlated deterioration factors such as environment and loading conditions exist among facilities).

Approximate solution methods for large-scale MDPs have been adapted to infrastructure-management problems. Durango-Cohen (2004) proposes a value function approximation approach using temporal-difference learning applicable to imperfect information on facility transition rates between states. Kuhn (2010) proposes an approximate dynamic programming model using value function approximation in which the value functions are approximated by a linear combination of basis functions. Medury and Madanat (2013) also propose an approximate dynamic programming approach in which the value functions of state-action pairs are generalised by a set of linear and separable functions. The study emphasises structural interdependencies (set of facilities collectively determining system performance such as connectivity or capacity) between infrastructure facilities by imposing a network capacity constraint on the MR&R activity selection.

The above examples use function approximation methods that aim to produce a good approximation of the value functions over large subsets of the entire state space. Methods that instead learn a parameterised policy that can select actions without necessarily consulting a value function are policy gradient methods. One advantage of parametrising policies is that the approximate policy can be driven to produce a stochastic optimal policy while it can also approach a deterministic policy according to parametrisation, allowing more flexibility tailored to the problem context. Moreover, the policy gradient theorem, which provides mathematical bases for calculating the performance gradient with respect to the policy parameters, adds a theoretical advantage to policy gradient methods. One drawback is that policy gradient methods often suffer from high variance in the parameter update process and converges to a local rather than global optimum.

Han *et al*. (2021) present a proximal policy optimisation model, a policy gradient algorithm that improves training stability compared with 'vanilla' policy gradient update. Proximal policy optimisation uses a clipped surrogate objective, which discourages policy updates for extremes for better rewards while retaining similar performance. It is also worth noting that in the study, a Markov state transition model was developed based on a deep artificial neural network. While the number of prior studies on applying policy gradient methods to network-level optimisation problems in infrastructure management was limited, policy-gradient-based approaches had shown promising applicability to similar engineering problems. For instance, Andriotis and Papakonstantinou (2019) demonstrate an application of a deep artificial neural network based policy gradient approach to a truss bridge whose structural system comprises a number of subsystem components.

In this study, an approach to determining the optimal MR&R actions based on policy gradient methods is proposed. A stochastic gradient-ascent algorithm, Reinforce (Williams, 1992), is used to update parameters pertaining to both parameterised policy and state value functions. The proposed approach is aimed at determining an optimal management solution for a complex network of infrastructures, which should consider interdependencies. The following section provides the derivation of the proposed approach.

Policy gradient method

Policy gradient methods rely on optimising parameterised policies with respect to the expected return (long-term cumulative reward) by gradient, which may be sometimes a more direct and natural approach than value function approximation. The parameterised policy $\pi(a|s,\boldsymbol{\theta}) = \Pr\{A_t = a|S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}$ describes the probability that action a is taken at time t given that the environment is in state s at time t with parameter $\boldsymbol{\theta}$. Let $J(\boldsymbol{\theta})$ be any policy objective function. Policy gradient methods seek to maximise $J(\boldsymbol{\theta})$. Therefore, policy gradient algorithms search for a local optimum in $J(\boldsymbol{\theta})$ by ascending the gradient of the policy $\nabla J(\boldsymbol{\theta}_t)$ with respect to parameters $\boldsymbol{\theta}$.

$$4. \quad \theta_{t+1} = \theta_t + \alpha \nabla_{\theta} J(\boldsymbol{\theta}_t)$$

where $\nabla_{\theta} J(\boldsymbol{\theta}_t)$ is the policy gradient or a stochastic estimate whose expectation approximates the gradient of $J(\boldsymbol{\theta})$ with respect to its argument θ_t. To ensure that the policy can be parameterised and its gradient is calculable, $\pi(a|s,\boldsymbol{\theta})$ must be differentiable with respect to its parameters – that is, $\nabla\pi(a|s,\boldsymbol{\theta})$ must exist and be finite for all s, a, $\boldsymbol{\theta}$, where $\nabla\pi(a|s,\boldsymbol{\theta})$ is a vector notation of partial derivatives of $\pi(a|s,\boldsymbol{\theta})$ with respect to the components of $\boldsymbol{\theta}$.

A natural and common kind of parameterisation is forming parameterised numerical preferences $h(s,a,\boldsymbol{\theta}) \in \mathbb{R}$ for each state-action pair. Probabilities of actions being selected in each state are found according to preferences. Soft-max in action preferences is formed as

$$5. \quad \pi(as, \boldsymbol{\theta}) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}$$

The action preferences can be parametrised arbitrarily. For instance, linear-in-feature preference can be constructed as

$$6. \quad h(s, a, \theta) = \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x}(s, a)$$

using feature vectors $\boldsymbol{x}(s, a)$.

Parametrising policies according to the soft-max in action preferences results in some favourable characteristics. An

advantage of such parametrisation is that the approximate policy can reach a deterministic policy, while ε-greedy action selection is always associated with the ε probability of selecting a random action. The action-value estimates would converge to their corresponding true values. Another advantage of using soft-max in action preferences is being able to select actions with arbitrary probabilities. For instance, in a road-infrastructure-management problem, an MR&R action on an individual facility given its state may not be strictly better than the other MR&R alternatives in a deterministic manner due to the large state space of the road network conditions. Rather, an MR&R action may be optimal for certain percentage of time. Therefore, the best approximate policy can be stochastic. In addition, policy gradient methods are advantageous over value function approximation in terms of better convergence properties in high-dimensional space while they may suffer from a tendency to converge to a local optimum and involve high variance in estimating parameters. Various policy gradient algorithms for addressing these issues have been proposed in the computer science/machine learning/artificial intelligence literature (Agarwal *et al*., 2020).

Policy gradient algorithms are developed based on the policy gradient theorem, which provides an analytic expression for the gradient of the objective function with respect to the policy parameter $\boldsymbol{\theta}$:

$$7. \quad \nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\sum_{a} q_{\pi}(S_t, a)\, \nabla\, \pi(a \mid S_t, \boldsymbol{\theta})\right]$$

where $q_{\pi}(s, a)$ is the value of taking action a in state s under policy π. The proof of the theorem can be seen in prior literature (e.g. Sutton and Barto, 2018). Equations 4 and 7 with some calculations bring about a stochastic gradient-ascent algorithm:

$$8. \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha^{\theta} G_t \frac{\nabla \pi(A_t \mid S_t, \boldsymbol{\theta})}{\pi(A_t \mid S_t, \boldsymbol{\theta})}$$

where G_t represents the return (the sum of future rewards) following time t and α^{θ} is the step size or learning rate for updating $\boldsymbol{\theta}$. Policy gradient algorithms in which parameters are updated following Equation 8 are referred to Reinforce. The update increases the parameter vector in a direction proportional to the return. This renders the parameter to move most in the directions that favour actions that yield the highest return. In addition, the update is inversely proportional to the action probability. This avoids actions that are selected frequently to be at an advantage. Reinforce can be classified as a Monte Carlo algorithm and therefore is suitable for an episodic case because the update uses the complete return from time t, which is the sum of future rewards up until the end of the episode.

Reinforce has good theoretical convergence properties since the expected update over an episode is in the same direction as the gradient of the objective function. However, as a Monte Carlo method, high variance in parameter updates can be an issue. To counter this, adding a baseline is known to reduce the variance. By adding an arbitrary baseline $b(s)$, Equation 7 can be rewritten as

$$9. \quad \nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\sum_{a} [q_{\pi}(S_t, a) - b(s)]\, \nabla\, \pi(a \mid S_t, \boldsymbol{\theta})\right]$$

which can then be used to form a generalised Reinforce by rewriting Equation 8:

$$10. \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha^{\theta}[G_t - b(S_t)] \frac{\nabla \pi(A_t \mid S_t, \boldsymbol{\theta})}{\pi(A_t \mid S_t, \boldsymbol{\theta})}$$

The baseline can be any function given that it does not vary with the action a. A natural and common choice for the baseline is an estimate of the state value, $v(S_t, \boldsymbol{w})$, where $\boldsymbol{w}$ is a weight vector. As Reinforce uses a Monte Carlo method to learn the policy parameter $\boldsymbol{\theta}$, it can also use a Monte Carlo method to learn the state value weights $\boldsymbol{w}$.

$$11. \quad \boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \alpha^{\boldsymbol{w}}[G_t - v(S_t, \boldsymbol{w})]\, \nabla\, v(S_t, \boldsymbol{w})$$

A pseudocode using Reinforce with a state value function as the baseline is provided in Algorithm 1. In this study, mini-batch sampling is added to pursue even better stability in estimating the parameters. In the algorithm, $\boldsymbol{x}$ denotes a state vector describing all the important attributes and features in the network in time t. In the presented modified Reinforce algorithm, a stochastic gradient ascent is employed to refine iteratively the policy parameters for optimal decision making in a reinforcement learning framework. The algorithm starts with the initialisation of policy parameters $\boldsymbol{\theta}$ and state value weights $\boldsymbol{w}$, essential for defining the policy $\pi(a|s,\boldsymbol{\theta})$ and the state value function $v(S_t, \boldsymbol{w})$, respectively. Additionally, step sizes for both $\boldsymbol{\theta}$ and $\boldsymbol{w}$, a mini-batch size M and the number of iterations I are specified. In each iteration, the algorithm generates episodes using the current policy and computes the average reward G_t for each state in these episodes. This computation is pivotal, as it captures the expected return from a state, discounted over future steps. Subsequently, for each decision epoch within an episode, the algorithm updates the state value weights $\boldsymbol{w}$ using the difference between the computed average reward and the estimated value from $v(S_t, \boldsymbol{w})$. This step is critical for refining the state value function towards more accurate estimations. In parallel, the policy parameters $\boldsymbol{\theta}$ are adjusted based on the gradient of the policy log-probability, scaled by the discounted reward. This dual updating mechanism of both $\boldsymbol{w}$ and

Algorithm 1. Pseudocode for updating parameters by stochastic gradient ascent

Initialise:
Policy parameter θ in a differentiable parameterised policy $\pi(a|s,\boldsymbol{\theta})$
State value weights $\mathbf{w}$ in a differentiable parameterised state value function $\hat{v}(S_t, \boldsymbol{w})$
Step sizes $\alpha_{(\theta)} > 0$ and $\alpha_{(w)} > 0$
Mini-batch size $M \in \mathbb{N}^+$
Number of iterations $I \in \mathbb{N}^+$
$i \leftarrow 0$
while $i \leq I$ **do**
  for each $m \in \{1, 2, \ldots, M\}$ **do**
    Generate an episode $z_{(m)} = (S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T)$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
    Calculate average reward at time t, G_t, for the mini-batch:
    $$G_t = \frac{1}{m}\sum_{m}\sum_{k=t+1}^{T}\gamma^{k-t-1}R_{k,m}$$
  end for
  for each decision epoch $t \in \{1, 2, \ldots, T\}$ **do**
    $\delta = G_t(x) - \hat{v}(S_t, \boldsymbol{w})$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_{(w)}\delta \nabla \hat{v}(S_t, \boldsymbol{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{(\theta)}\gamma^t \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$
  end for
  $i \leftarrow i + 1$
end while

$\boldsymbol{\theta}$ converges to a policy that maximises the expected cumulative reward over time, encapsulating the essence of the policy gradient method in reinforcement learning.

Application

This section demonstrates the implementation of the proposed policy gradient approach in road infrastructure management. In this application, concrete bridge decks are selected as the primal target of management because they are often the most expensive elements of bridge systems in terms of durability (Cady and Weyers, 1984). In addition, partial or full closures of bridges can affect the overall network performance and result in considerable economic loss, as they represent the most vital and critical links of a road network. The assumed region of application is Quebec, Canada. The climate of Quebec is in general cold and humid. In such an environment, the corrosion of reinforcing steel due to the use of de-icing chemicals in winter, as well as freezing and thawing cycles, is a critical factor affecting degradation of concrete bridge decks in addition to loads from traffic.

In this application, four types of MR&R actions are considered, and their unit costs are shown in Table 1: do nothing, minor repair, major repair and replacement. Examples of minor repair include surface cleaning, minor sealing of joints, repair of cracks and local repairs on the slab above the reinforcement. Major repair primarily comprises localised repair of the concrete to a depth below the reinforcement in addition to major sealing of joints and epoxy injection to fill cracks. Replacement is the construction of a new slab with new reinforcement. The cost data for each MR&R activity are obtained from the Ministère des Transports du Québec (MTQ, 2021) for typical activities in Quebec. It is assumed that only one MR&R activity can be carried out on a bridge deck at each decision epoch. Note that the durations of bridge closures assume small bridges or viaducts. Therefore, applying the numbers to larger bridges or viaducts would lead to an optimistic scenario.

Table 1. Costs of MR&R actions

Action	Unit cost: US$/m^2	Bridge closure
Do nothing	0	N/A
Minor repair	800	Half-closure, 2 days
Major repair	1500	Half-closure, 7 days
Replacement	2000	Full closure, 14 days

Yearly condition state transition probability matrices, which describe the rate of deterioration, are obtained from the paper by Zhang *et al*. (2020) and shown in Table 2. The matrices are estimated using TransChlor, which uses hourly climate data (precipitation, temperature, relative humidity, solar radiation) to replicate the application of de-icing salts within a given climatic region Conciatori *et al*. (2010). The deterioration process is divided into four stages. At stage 1, the structural health of the deck is like new; it refers to the beginning of the service life until the initiation of corrosion. Stage 2 starts when corrosion initiates and lasts until a surface crack width reaches 0.8 mm. Stage 3 is the final phase of the propagation of corrosion up until the onset of concrete cover spalling. The four stages defined herein are compatible with visual inspection and condition assessment procedures and therefore are applicable to a wide range of infrastructure facilities. The effects of interventions are reflected in the transition probability matrices. It is assumed that stage 4 is too severe for minor repair; therefore, minor repair on a stage 4 deck has no effect. Major repair is assumed to restore stage 2

Table 2. Transition probability matrices of concrete bridge decks

Do nothing				
	1	2	3	4
1	0.68	0.32	0	0
2	0	0.89	0.11	0
3	0	0	0.91	0.09
4	0	0	0	1

Minor repair				
	1	2	3	4
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	0	1

Major repair				
	1	2	3	4
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	1	0	0

Replacement				
	1	2	3	4
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0

considering that the reinforcing steel is not replaced after a major repair. After replacement, decks are assumed to be like new regardless of their stage before the intervention.

In the following sections, results of numerical studies are presented for a single facility, a small network and a large network. In the single-facility case, the objective is to show that the policy gradient approach can determine a policy acceptably similar to, if not the same as, the one determined by dynamic programming, an analytical approach that explores the entire state space and is thus theoretically the best approach in terms of precision. The small-network case is composed of three bridges in a road network. The policy gradient approach is intended to show its capability to account for interdependencies among the facilities and determine the optimal set of MR&R actions for the network. The results are again compared with dynamic programming to prove their validity. Lastly, the large-network example is presented to show the applicability of the policy gradient approach to a large-scale problem to which traditional methods are less suitable. For all three cases, the decision epoch is every 5 years, meaning that a 1-year transition occurs according to a selected MR&R action, and the deck deteriorates following the transition probabilities for do nothing for the remaining 4 years. The assumed service life is 100 years; therefore, 20 decision epochs are considered. The optimal policies are determined as an infinite horizon problem since the assumed service life is long enough to be regarded as such. The discount factor is 0.98.

In this study, economic and structural dependencies within infrastructure networks are highlighted in the sections headed ‘Small network’ and ‘Large network’. Economic dependencies arise when the management of one facility is influenced by the condition of others, while structural dependencies relate to how individual facilities collectively impact the overall performance of the network. The authors’ approach can be easily expanded to accommodate stochastic dependencies, which involve correlated deterioration factors among facilities. Future expansions of this work could adapt the method to encompass these stochastic aspects, enhancing its comprehensiveness in infrastructure management. This potential for broader applicability underscores the depth and versatility of this approach.

Single facility

Consider managing a bridge on a highway primarily focusing on the deterioration of its deck. The bridge is assumed to have three lanes in each direction and a deck area of 10 000 m^2. The impact of interventions is estimated from the results of traffic simulations using the Emme software (INRO, 2022) on a real bridge in Quebec that has similar design, characteristics and role. The estimated daily economic losses from half-closure and full closure of this bridge are US$70 000 and US$326 000, respectively.

The optimal policies determined by the policy gradient approach and dynamic programming are shown in Table 3. Both methods

Table 3. Optimal policy for a single facility

Policy gradient				
Stage	DN	Minor	Major	Replace
1	1	0	0	0
2	0.58	0.30	0.12	0
3	0.27	0.58	0.16	0
4	0	0	1	0
Dynamic programming				
Stage		Optimal action		
1		Do nothing		
2		Do nothing		
3		Minor repair		
4		Major repair		

DN, do nothing

identify the same policies: the optimal actions when the bridge deck is in states 1, 2, 3 and 4 are, respectively, do nothing, do nothing, minor repair and major repair. Replacement was not selected as the best action for any stage of degradation, as it was less cost effective than the other MR&R alternatives. As discussed in the section headed 'Policy gradient method', an advantage of the policy gradient approach is to be able to describe the extent of preferences with probabilities. For instance, while dynamic programming determines the single optimal action for each stage, policy gradient estimates, for instance, for stage 2, 0.58 for do nothing, 0.30 for minor repair and 0.12 for major repair according to soft-max in action preferences. These numbers represent the attractiveness of the alternatives and provide additional pieces of information crucial for decision making.

In the analysis presented in Table 3, it is observed that bridges are typically not replaced, a finding consistent with current industry trends where high costs often lead to the deferral of bridge replacement. However, this study employs a time-invariant deterioration model, which does not account for the potential acceleration in deterioration rates following major repairs. The authors acknowledge that this is a simplification of real-world conditions. Future studies could explore the impact of employing a time-variant deterioration model, which might reveal different insights, particularly in terms of the frequency and cost-effectiveness of bridge replacements. Such research could provide a more nuanced understanding of the long-term management strategies for bridge infrastructure, particularly under varying deterioration conditions post-repair.

Small network

Three bridges are considered for the small-network example. Two bridges are added to the one used in the previous example of the single-facility case. They are assumed to be overpasses on arterial roads with two lanes in each direction. Let bridge A be the one from the single-facility case and bridges B and C be the two added bridges. The deck areas of bridges B and C are 7500 and 5000 m^2, respectively. Similar to the single-facility case, the economic losses due to interventions on these bridge decks are estimated using the results of traffic simulation performed using the Emme software for similar real bridges in Quebec. The estimated daily economic losses from half-closure and full closure of individual bridges are US$20 000 and US$93 000 for bridge B and US$26 000 and US$134 000 for bridge C, respectively. The relationship between the deck surface area and economic losses is inversely proportional for bridge C, which is modelled after an existing bridge in proximity to a central business district, therefore playing a critical role for network connectivity.

It is imperative to develop optimal intervention policies that account for interdependencies among infrastructure facilities in proximity to each other. In the following, two common causes of correlated user costs are considered. In the first case, concurrent interventions on multiple bridges in series tend to decrease the total economic losses in comparison with the sum of losses for the bridges considered separately. Conversely, concurrent interventions on multiple bridges in parallel tend to increase the total economic losses in comparison with the sum of losses for the bridges considered separately. In both cases, the change in total economic losses can be obtained considering that the losses from individual bridges are negatively correlated for bridges in series and that they are positively correlated for bridges in parallel. When two bridges are in series, the length of detours and delays in travel times are shorter than the sum of detours and delays when only a single bridge is closed since part of the detours are common. When two bridges are in parallel, the total delays are increased since the bridges cannot be used as mutual detour routes. Therefore, in this example, the feature vector $\boldsymbol{x}$ in Algorithm 1 denotes concurrent interventions on multiple bridges.

Figure 2 shows the evolution of the total costs as a function of the number of iterations for the interventions on a network of three bridges in series and in parallel. In both cases, the correlation coefficients for losses on individual bridges are assumed to be 0.5, which have been observed from simulations using the Emme software on a simplified network. First, the optimal policies for bridges A, B and C are individually determined as in the previous example and used as the initial policies for the network. For the bridges in series, the optimised parameterised policy favours

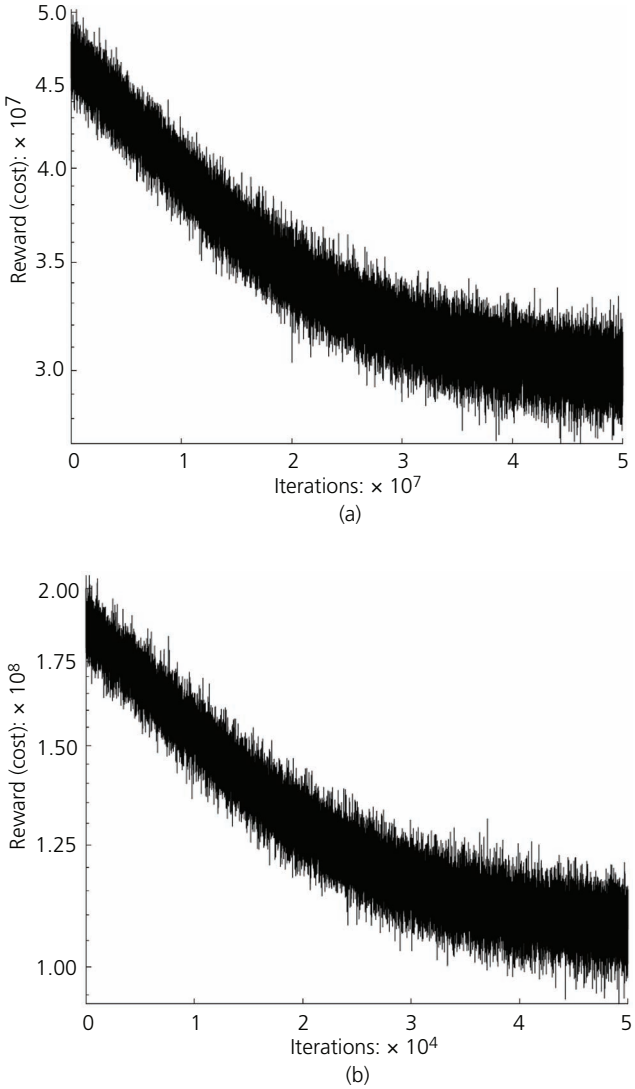

Figure 2. Iterations and simulated costs of the small-network example: (a) bridges in series ($\rho = 0.5$); (b) bridges in parallel ($\rho = -0.5$)

simultaneous interventions on bridges. Conversely, for the bridges in parallel, the optimal policy is to avoid concurrent interventions on multiple bridges. The savings in costs by optimising the intervention times reduces total initial estimates of costs based on optimal interventions on individual bridges by 50% in both cases. The results also suggest that interventions on a larger network should favour concurrent interventions on bridges in series over those in parallel since costs are then reduced by a factor of 3. The differences between the simulated costs at the beginning and the end of the iterations are attributed to adjusting these preferences on conducting concurrent interventions.

The optimal simulated costs obtained from policy gradient models are compared with those obtained from the dynamic programming model in Figure 3. PG(0) is the policy gradient model based on

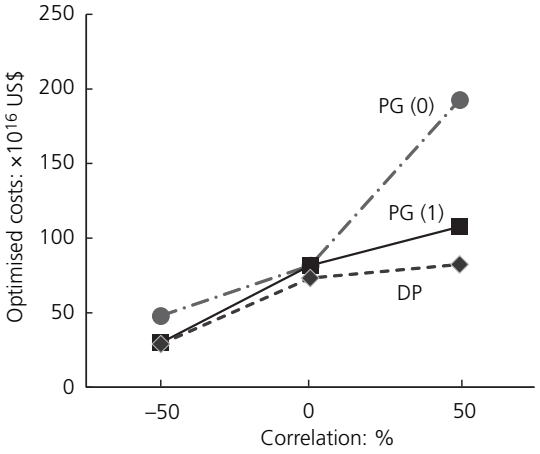

Figure 3. Optimised costs from policy gradient models (PG) and dynamic programming model (DP). PG(0) is calibrated for a single facility, and PG(1) for the network

the policies optimal for individual bridge decks and therefore does not account for interdependencies. PG(1) is the policy gradient model calibrated from PG(0) to take into account interdependencies. DP is the dynamic programming model, which determines the exact solution by exploring all possible combinations of MR&R alternatives and the network condition composed of deterioration stages of individual bridge decks. The figure shows the cost differences between PG(1) and DP, which are approximately US$1 million for the bridges in series and US $25 million for the bridges in parallel. While these differences are not marginal, they are presented to highlight the relative efficiency of different management strategies. Specifically, the comparison between PG(0) and PG(1) demonstrates the applicability and effectiveness of the policy gradient method in a network with interdependencies. This analysis helps underscore that PG(1) is suitably calibrated for network-level optimisation, showing a more cost-effective approach. It is important to note that these cost comparisons are specific to the context of this study and are not meant to be generalised to larger-scale projects or different scenarios.

The differences between the optimised costs of PG(0) and PG(1) are US$18 million for the bridges in series and US$85 million for the bridges in parallel. These differences are much larger than those between PG(1) and DP and demonstrate that PG(1) is capable of identifying near-optimal policies. The differences between PG(0) and PG(1) are much larger for the bridges in parallel. The optimised costs for PG(0) are 60 and 80% higher for the bridges in series and the bridges in parallel, respectively. Because the optimised cost itself is much more expensive for the bridges in parallel, the size of the difference becomes larger. Another plausible explanation is related to the optimal policy. The policies of PG(0) and PG(1) for the bridges in series are similar because the single-facility policy intervenes when the deck is in a progressed deterioration state such as stage 3 or 4. This leads to

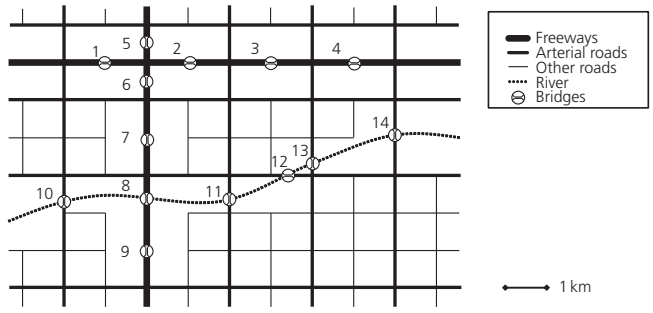

Figure 4. Illustration of the large-network example

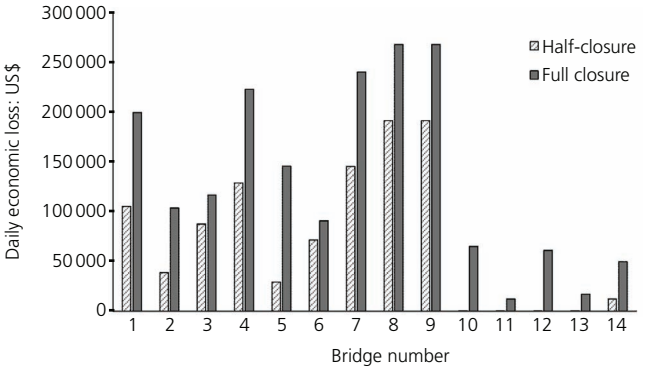

Figure 5. Estimated daily economic loss due to individual interventions

concurrent interventions when multiple bridge decks are deteriorated, which take advantage of the cost reduction. However, the policies are different for the bridges in parallel since PG(1) attempts to avoid concurrent interventions when multiple bridge decks are deteriorated, while PG(0) does not. In the end, the results suggest that the policy gradient approach can incorporate the effects of interdependencies when the model specification is appropriately constructed.

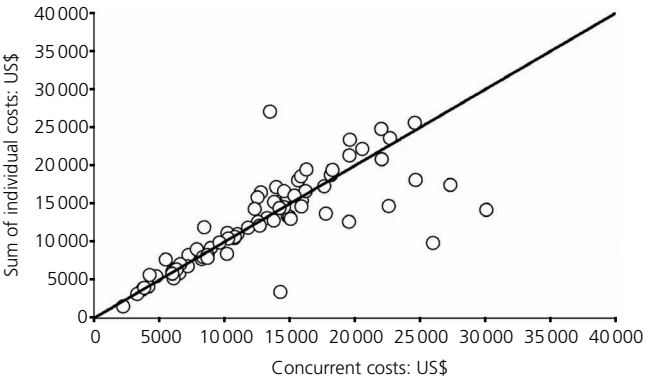

Figure 6. Effect of concurrent two interventions

Large network

For the large-network example, a road network comprising 14 bridges is assumed with nine bridges on highways and five bridges on arterial roads (Figure 4). The study area has dimensions of 8 × 10 km. Bridges 1–4 and bridges 5–9 are assumed to be highway overpasses that are oriented horizontally and vertically on the map, respectively. Bridges 10–14 are assumed to be on arterial roads that cross a river. User equilibrium-based traffic assignment is conducted to estimate traffic redistribution, traffic delays and economic losses due to interventions on the bridge decks. Figure 5 shows the estimated daily economic losses from a closure of each bridge. Highway bridges tend to show large economic losses as a function of interventions. In particular, bridges 7–9 play a significant role in distributing the traffic from the upper part of the map to the lower part and vice versa. The economic losses due to partial closures on bridges 10–13 are not visible in the figure since their values are smaller than US$1000.

An important objective of the proposed approach is to capture the effect of network configurations on optimal sequences of interventions. Figure 6 shows the compound effect of two half-closures in the network for 91 (=14 × 13 ÷ 2) possible combinations of two bridge closures out of 14 bridges. The daily user costs (economic loss) of concurrent two interventions are compared with those of the sum of individual interventions on the two bridges. The straight line in the figure denotes the concurrent costs that are equal to the sum of individual intervention costs. The data points under the line indicate combinations where concurrent costs exceed the sum of individual costs, which should be avoided. These instances are illustrated by interventions on bridges 8 and 10 or 8 and 11 corresponding in both cases to two bridges in parallel relative to the major axis of regional traffic. Conversely, data points above the line correspond to two bridges in series and the benefit of concurrent interventions. These instances are illustrated by interventions on bridges 2 and 3, where the two bridges form a series relationship.

The feature vectors $\boldsymbol{x}(s,a)$ are designed to capture whether intervention of the bridges is planned. Therefore, every element of the action preference for a bridge deck in a particular state $h(s, a,\boldsymbol{\theta})$ is described by $\theta_{\text{intercept}} + \sum_i^{13} \theta_i x_i$ as a linear-in-feature preference. Example progresses of the parameter-updating process are shown in Figure 7. Figure 7(a) shows the value of $\boldsymbol{\theta}$ for conducting major repair on bridge 3 when intervention of bridge 4 is also planned. Starting with the initial value of 0, the value monotonically increases. This incremental change results in rendering the major repair action be more likely to be selected, which benefits from saved economic loss from concurrent interventions. Figure 7(b) shows the progress on the value of $\boldsymbol{\theta}$ for conducting major repair on bridge 13 when intervention of bridge 14 is planned. In contrast to Figure 7(a), Figure 7(b) shows another type of example where concurrent interventions are discouraged due to large additional economic losses generated by forcing travellers to take a long detour. Some may find the update

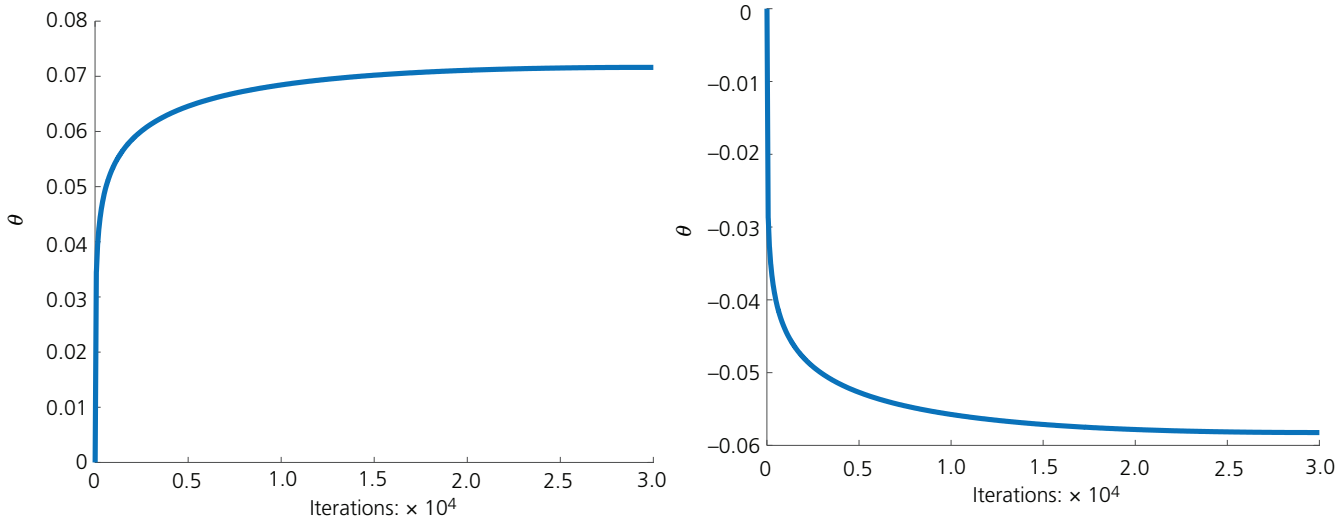

Figure 7. Illustrative examples of progress on parameter θ

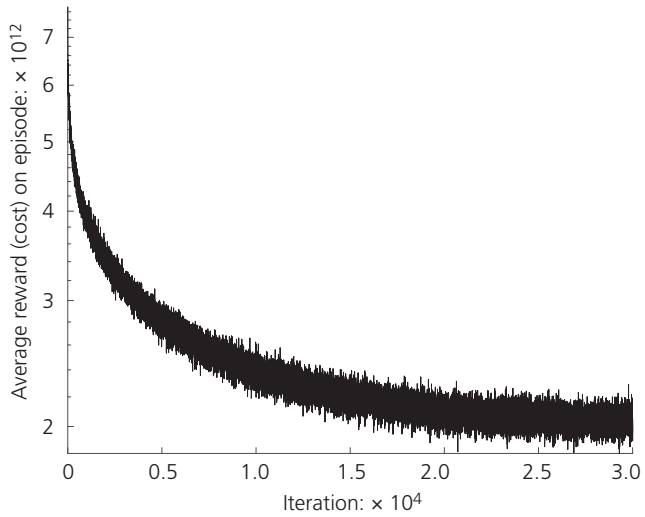

Figure 8. Iterations and simulated costs of the large-network example

process of parameter θ surprisingly smooth; this is mainly a result of applying a large mini-batch size. Using a smaller mini-batch size would make the parameter update process fluctuate, while the time to complete calculation of a simulation episode would be saved.

Therefore, the mini-batch size controls the trade-off between the stability and computational time for the simulations.

Figure 8 presents the progress of the policy gradient model for this large-network example (the figure shows the costs for the entire network for the whole service life). The result of the single-facility case is used as the initial values of the intercept parameters. Therefore, this initial policy already recommends good actions for individual facilities. The network-level consideration such as network configurations, however, is missed by the initial policy. The reduction in costs per episode in Figure 8 is therefore attributed to encouraging concurrent interventions when they are advantageous and avoiding them otherwise.

Conclusion

This paper presents a reinforcement learning approach to network-level road infrastructure management based on the policy gradient method. The proposed approach is able to capture interdependencies among road infrastructure facilities and determine the optimal policy that considers the network configurations. It also resolves the computational complexity of the optimisation problem in road infrastructure management that arises for a large network. The validity and applicability of the proposed approach are demonstrated through computational studies. In these studies, a linear-in-preference formulation is used, and the policy is parameterised to form the soft-max preference in actions. This model formulation provides relative preferences among actions, which can be informative for decision makers when more than one intervention alternative seems competitive.

Future research may address the innate high variance of the policy gradient method in updating the parameters. In this paper, the Reinforce algorithm with estimated state values as the baseline is used to mitigate the issue. In the computer science community, reinforcement learning is an ever-progressing subject, and advanced policy gradient algorithms are proposed to resolve the high variance problem and accelerate the speed of parameter estimation.

In addition, future research may use a reinforcement learning approach to conduct empirical studies that impose strict

restrictions, constraints and challenges particular to real-world examples. In this paper, the computational studies are based on fundamental settings and assumptions that are prevalent among many infrastructure-management projects. It would be interesting to examine the capabilities of reinforcement learning approaches for a wide range of MR&R projects.

REFERENCES

Agarwal A, Kakade SM, Lee JD and Mahajan G (2020) Optimality and approximation with policy gradient methods in Markov decision processes. In *Proceedings of Thirty Third Conference on Learning Theory* (Abernethy J and Agarwal S (eds)). PMLR, pp. 64–66.

Andriotis CP and Papakonstantinou KG (2019) Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety* **191**: article 106483, https://doi.org/10.1016/j.ress.2019.04.036.

Bard JF (2013) *Practical Bilevel Optimization: Algorithms and Applications*. Springer, Dordrecht, the Netherlands.

Cady PD and Weyers RE (1984) Deterioration rates of concrete bridge decks. *Journal of Transportation Engineering* **110(1)**: 34–44, https://doi.org/10.1061/(ASCE)0733-947X(1984)110:1(34).

Camahan JV, Davis WJ, Shahin MY, Keane PL and Wu MI (1987) Optimal maintenance decisions for pavement management. *Journal of Transportation Engineering* **113(5)**: 554–572, https://doi.org/10.1061/(ASCE)0733-947X(1987)113:5(554).

Chan WT, Fwa TF and Tan CY (1994) Road-maintenance planning using genetic algorithms. I: Formulation. *Journal of Transportation Engineering* **120(5)**: 693–709, https://doi.org/10.1061/(ASCE)0733-947X(1994)120:5(693).

Chu JC and Chen YJ (2012) Optimal threshold-based network-level transportation infrastructure life-cycle management with heterogeneous maintenance actions. *Transportation Research Part B: Methodological* **46(9)**: 1123–1143, https://doi.org/10.1016/j.trb.2012.05.002.

Conciatori D, Laferrière F and Brühwiler E (2010) Comprehensive modeling of chloride ion and water ingress into concrete considering thermal and carbonation state for real climate. *Cement and Concrete Research* **40(1)**: 109–118, https:doi.org/10.1016/j.cemconres.2009.08.007.

Dekker R, Wildeman RE and Van der Duyn Schouten FA (1997) A review of multi-component maintenance models with economic dependence. *Mathematical Methods of Operations Research* **45(3)**: 411–435, https://doi.org/10.1007/BF01194788.

Durango-Cohen PL (2004) Maintenance and repair decision making for infrastructure facilities without a deterioration model. *Journal of Infrastructure Systems* **10(1)**: 1–8, https://doi.org/10.1061/(ASCE)1076-0342(2004)10:1(1).

Frangopol DM and Liu M (2007) Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost. *Structure and Infrastructure Engineering* **3(1)**: 29–41, https://doi.org/10.1080/15732470500253164.

Fwa TF, Tan CY and Chan WT (1994) Road-maintenance planning using genetic algorithms. II: Analysis. *Journal of Transportation Engineering* **120(5)**: 710–722, https://doi.org/10.1061/(ASCE)0733-947X(1994)120:5(710).

Golabi K and Shepard R (1997) Pontis: a system for maintenance optimization and improvement of US bridge networks. *Interfaces* **27(1)**: 71–88, https://doi.org/10.1287/inte.27.1.71.

Golabi K, Kulkarni RB and Way GB (1982) A statewide pavement management system. *Interfaces* **12(6)**: 5–21, .

Gopal S and Majidzadeh K (1991) Application of Markov decision process to level-of-service-based maintenance systems. *Transportation Research Record* **1304**: 12–18, https://doi.org/10.1287/inte.12.6.5.

Hajibabai L, Bai Y and Ouyang Y (2014) Joint optimization of freight facility location and pavement infrastructure rehabilitation under network traffic equilibrium. *Transportation Research Part B: Methodological* **63**: 38–52, https://doi.org/10.1016/j.trb.2014.02.003.

Han C, Ma T and Chen S (2021) Asphalt pavement maintenance plans intelligent decision model based on reinforcement learning algorithm. *Construction and Building Materials* **299**: article 124278, https://doi.org/10.1016/j.conbuildmat.2021.124278.

INRO (2022) *Emme*. INRO, Montreal, QC, Canada. See https://www.inrosoftware.com/en/products/emme/ (accessed 31/05/2022).

Jesus M, Akyildiz S, Bish DR and Krueger DA (2011) Network-level optimization of pavement maintenance renewal strategies. *Advanced Engineering Informatics* **25(4)**: 699–712, https://doi.org/10.1016/j.aei.2011.08.002.

Kuhn KD (2010) Network-level infrastructure management using approximate dynamic programming. *Journal of Infrastructure Systems* **16(2)**: 103–111, https://doi.org/10.1061/(ASCE)IS.1943-555X.0000019.

Kulkarni RB and Miller RW (2003) Pavement management systems: past, present, and future. *Transportation Research Record* **1853(1)**: 65–71, https://doi.org/10.3141/1853-08.

Medury A and Madanat S (2013) Incorporating network considerations into pavement management systems: a case for approximate dynamic programming. *Transportation Research Part C: Emerging Technologies* **33**: 134–150, https://doi.org/10.1016/j.trc.2013.03.003.

Morcous G and Lounis Z (2005) Maintenance optimization of infrastructure networks using genetic algorithms. *Automation in Construction* **14(1)**: 129–142, https://doi.org/10.1016/j.autcon.2004.08.014.

MTQ (Ministère des Transports du Québec) (2021) *Liste et Prix des Ouvrages d'Infrastructure de Transport 2020–2021*. MTQ, Quebec, QC, Canada (in French).

Ng M, Lin DY and Waller ST (2009) Optimal long-term infrastructure maintenance planning accounting for traffic dynamics. *Computer-aided Civil and Infrastructure Engineering* **24(7)**: 459–469, https://doi.org/10.1111/j.1467-8667.2009.00606.x.

Smilowitz K and Madanat S (2000) Optimal inspection and maintenance policies for infrastructure networks. *Computer-aided Civil and Infrastructure Engineering* **15(1)**: 5–13, https://doi.org/10.3141/1667-01.

Sutton RS and Barto AG (2018) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.

Uddin W, Hudson WR and Haas R (2013) *Public Infrastructure Asset Management*. McGraw-Hill Education, New York, NY, USA.

Vanier DD (2001) Why industry needs asset management tools. *Journal of Computing in Civil Engineering* **15(1)**: 35–43, https://doi.org/10.1061/(ASCE)0887-3801(2001)15:1(35).

Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* **8(3)**: 229–256, https://doi.org/10.1007/BF00992696.

Zhang Y, Chouinard LE, Power GJ, Tandja MCD and Bastien J (2020) Flexible decision analysis procedures for optimizing the sustainability of ageing infrastructure under climate change. *Sustainable and Resilient Infrastructure* **5(1–2)**: 90–101, https:doi.org/10.1080/23789689.2018.1448665.