# Network-Level Infrastructure Management Using Approximate Dynamic Programming

Kenneth D. Kuhn[1]

**Abstract:** This research introduces the use of approximate dynamic programming to overcome a variety of limitations of distinct infrastructure management problem formulations. The form, as well as the parameters, of a model specifying the long-term costs associated with alternate infrastructure maintenance policies are learned via simulation. The introduced methodology makes it possible to manage large heterogeneous networks of facilities related by budgetary restrictions and resource constraints as well as by dependencies in maintenance costs or deterioration. In addition, the methodology is particularly well suited to consideration of multiple types of infrastructure condition data at the same time, including continuous-valued data and relevant historical data. Introduced techniques will prove valuable when high-quality deterioration and cost estimation models are available but are ill suited for use in a Markov decision problem framework. Computational studies show that the introduced approach is able to find an optimal solution to a relatively simple infrastructure management problem, and is able to find increasingly good solutions to a more complex problem.

**DOI:** 10.1061/(ASCE)IS.1943-555X.0000019

**CE Database subject headings:** Infrastructure; Markov process; Maintenance; Computer programming.

**Author keywords:** Infrastructure management; Approximate dynamic programming; Markov decision problem; Infrastructure maintenance.

## Introduction

The infrastructure management problem involves selecting maintenance actions to perform on a set of infrastructure facilities over a planning horizon. The costs government agencies incur performing maintenance are balanced against the costs the public incur using facilities that deteriorate if not properly maintained. Past research has formulated the infrastructure management problem as a Markov decision problem (MDP) (Golabi et al. 1982; Camahan et al. 1987; Gopal and Majidzadeh 1991; Golabi and Shepard 1997; etc.). This framework allows for relatively quick discovery of socially optimal policies describing what maintenance actions to take as a function of infrastructure condition, for problems of limited size.

There are several shortcomings of the Markovian approach, as typically applied. Data relevant to how infrastructure will deteriorate are summarized by selecting one element in a discrete set of condition rating states. Solution techniques become intractable as this set grows. It thus becomes difficult to manage large networks of facilities jointly or to consider complex relationships among relevant variables. Examples of relationships known to exist but frequently overlooked include the impact of historical conditions on future deterioration (Robelin and Madanat 2007) and economies of scale in maintenance contracting (Dekker et al. 1997).

This research applies approximate dynamic programming (ADP), as outlined in Powell (2007), to the infrastructure management problem. The *value function* central to maintenance action selection in MDP formulations of the problem is approximated, learned via simulation. The approximation can subsequently be used to guide maintenance decision making. The specific methodology introduced here allows learning which form, as well as what parameter values, are appropriate when approximating the value function. The methodology is able to consider multiple types of infrastructure condition data simultaneously, continuous-valued infrastructure condition data, and relevant historical data. The methodology is also able to consider complex relationships between facilities, including realistic dependencies in maintenance costs and deterioration models. Finally, the methodology scales much better than past formulations of the infrastructure management problem, making it possible to manage large heterogeneous networks of related facilities.

## Infrastructure Management Problem

Sample formulations of infrastructure management problems are presented next. These formulations model infrastructure management as MDPs, as is common in the literature (Golabi et al. 1982; Camahan et al. 1987; Gopal and Majidzadeh 1991; Golabi and Shepard 1997; etc.). An effort has been made to use the terminology and notation of earlier works in the field of infrastructure management.

## Sample Markov Decision Problem Formulations

### Single-Facility Management

Assume we are managing one infrastructure facility over the next $T$ years. Each year, one action $a$ out of a set $A$ of possible main-

[1]Lecturer, Dept. of Civil and Natural Resources Engineering, Univ. of Canterbury, Private Bag 4800, Christchurch 8041, New Zealand. E-mail: kenneth.kuhn@canterbury.ac.nz

Note. This manuscript was submitted on April 28, 2009; approved on July 31, 2009; published online on August 3, 2009. Discussion period open until November 1, 2010; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Infrastructure Systems***, Vol. 16, No. 2, June 1, 2010. ©ASCE, ISSN 1076-0342/2010/2-103–111/$25.00.

tenance actions is performed on the facility. The set $A$ includes a "do nothing" option allowing for the possibility no maintenance action is performed. The condition of the facility in question is rated each year, and assigned a state $i$ in the discrete set $I$. $I$ may include "like new," "unusable," and a number of states representing conditions between these two extremes.

The social cost of the facility being in state $i$ with maintenance action $a$ applied is defined as $c(i,a)$. This cost accounts for agency costs performing the relevant maintenance and user costs the public incur using the facility. User costs incorporate diverse considerations, for example accident risk and vehicle wear and tear as a function of roadway condition. Costs are discounted using a discount factor $\alpha$ $[\alpha=1/(1+r)$ where $r$ is the discount rate]. The goal will be to minimize the sum of discounted costs over time, i.e., to find the social optimal maintenance policy. This formulation was chosen for its simplicity, but other researchers have considered other objectives. Golabi et al. (1982) considered maximizing level of service subject to agency budget constraints or minimizing public agency expenditures subject to level of service requirements.

*Transition probabilities*, $p(j|i,a)$ terms, capture the probability that the facility will be in condition state $j$ 1 year after being in state $i$ with action $a$ applied. These values serve as the model of deterioration. A number of research papers describe how transition probabilities can be derived from empirical condition data (Mishalani and Madanat 2002; Chu and Durango-Cohen 2007; etc.). Given these definitions, Bellman's equation is as follows:

$$V_t(i) = \min_{a \in A}\left[ c(i,a) + \alpha \sum_{j \in I} p(j|i,a) V_{t+1}(j) \right] \tag{1}$$

Assume *salvage values*, terms representing the costs of ending the planning period with the facility in a given condition state, are known for all condition states $i$ in $I$. Set $V_T(i)$ terms equal to the salvage values. Then work backward through the planning horizon, using Eq. (1) to find costs from year $T-1$ onward, then from year $T-2$, and so on. For year 0, solve Eq. (1) given the current (assumed known) condition of the facility to provide optimal social costs through the planning horizon. The actions that have been chosen along the way to minimize costs provide the optimal maintenance policy, describing what maintenance action to take as a function of year and facility condition.

### Network-Level Management

Next, imagine managing a network of $N$ infrastructure facilities. As in the single-facility case, the condition of the network will be described via a discrete set of condition states, here labeled $X$. The condition of the network is typically described by a combination of the condition states of each of the individual facilities to be managed. Likewise, the set of potential maintenance actions, $B$, typically includes possibilities for combinations of actions taken on each of the individual facilities to be managed. There are often fixed budgets for what the responsible public agency may spend on maintenance across various subsets of the facilities to be managed. There may also be limited materials, personnel, or other resources available for use. Chan et al. (2003) noted that operational and resource constraints may vary by region within a set of facilities jointly managed. The set $B$ of possible (system-level) actions that may be taken is often limited by constraints on various subsets of the set of facilities to be managed.

The costs of having the network of facilities in state $x$ and performing action $b$ is labeled $d(x,b)$. Costs may or may not be additive across the set of facilities being managed. Durango-Cohen and Sarutipand (2007) note that performing maintenance on adjacent sections of pavement at the same time is likely to decrease both public agency expenditures and driver inconvenience. Let $q(y|x,b)$ be the probability of the network being in state $y$ 1 year after being in state $x$ with action $b$ applied. $q(y|x,b)$ likely equals the product of individual facility transition probability terms, although this is not required.

Given all of the aforementioned, the network-level infrastructure management problem can be formulated as

$$V_t(x) = \min_{b \in B}\left[ d(x,b) + \alpha \sum_{y \in X} q(y|x,b) V_{t+1}(y) \right] \tag{2}$$

The multifacility formulation, based on Eq. (2), could then be solved in exactly the same manner as the single-facility formulation, based on Eq. (1).

### Shortcomings and Mitigation Strategies

There are several shortcomings of the approaches introduced earlier. First, the condition of any infrastructure facility or set of facilities is described by selecting one of a handful of possible condition rating states. Some measures of an infrastructure facility's condition are continuous, like the reliability index of a bridge deck (Robelin and Madanat 2007), and would have to be discretized. In addition, there may be multiple sources of data regarding infrastructure condition. For example, the deterioration of a section of roadway may be described in terms of the area covered by cracks on its surface or the depths of ruts that develop in vehicle wheel paths. The manner in which data are combined/discretized must be established prior to optimization, before it is clear which conditions are critical to decision making. It is also worth noting that consideration of multiple sources of data, each one of which has a range of possible values, can lead to a quite large space of possible overall condition rating states. Generally speaking, discretization results in a loss of information inversely proportional to the number of discrete states employed.

In addition, the aforementioned approach employs the Markovian assumption. Information relevant to how facilities will deteriorate are summarized in the current state of the facilities. Evidence on this point is mixed. Mishalani and Madanat (2002) found that historical data were useful for predicting future deterioration on bridge decks in one of two cases investigated. In cases where historical data are relevant, it is possible to incorporate such data into "current" condition state definitions using a technique known as state space augmentation. Robelin and Madanat (2007) formulated an infrastructure management problem using condition states for bridge decks that included information on the current state, the last action performed, and the time since the last action was performed. Again, using larger numbers of discrete condition ratings states to summarize historical data reduces information loss.

The biggest drawback to the aforementioned approach is that it scales poorly. Assume our network of $N$ facilities is comprised of facilities each of which is described as being in one of the set $I$ of condition rating states. Then the set $X$ of networkwide conditions contains $|I|^N$ states. Eq. (2) would have to be evaluated for each of the $|I|^N$ condition states, for each year from year 0 to year $T-1$. Each evaluation of Eq. (2) would require summing across the $|I|^N$ condition states for each of the $|A|^N$ potential combinations of actions. The value function $V$ would have to store $T*|I|^N$ values. Public agencies generally manage large numbers of infrastructure facilities. For example, Golabi et al. (1982) noted that the state of Arizona managed 7,400 mi of highway treating each mile as a separate "facility." It would be impossible to manage a system of anywhere near 7,400 facilities using the formulation introduced earlier.

Golabi et al. (1982) were able to manage Arizona's network of 7,400 mi of highway using aggregation. Instead of considering what action to take on each specific facility, the writers consider what action(s) to take on sets of facilities similar in terms of their structure and condition. The problem is then solved via linear programming. Assume we are managing a set of $N$ independent, homogeneous facilities, with associated costs and transition probabilities for each facility individually given by $c(i,a)$ and $p(j|i,a)$ terms. Let $f_t(i,a)$ represent the fraction of all the facilities being managed that are in condition state $i$ with action $a$ applied in year $t$. These fractions are the decision variables. Let $x_0(i)$ be the fraction of facilities initially in state $i$, assumed to be given. The linear program to be solved is then as follows:

$$\min_f \sum_{t=0}^{T} \sum_{i \in I} \sum_{a \in A} \alpha^t f_t(i,a) c(i,a) N \tag{3}$$

subject to

$$\sum_{a \in A} f_t(j,a) = \sum_{i \in I} \sum_{a \in A} f_{t-1}(i,a) p(j|i,a), \quad \forall\, j \in I,\; t \in \{1,2, \ldots T\} \tag{4}$$

$$\sum_{a \in A} f_0(i,a) = x_0(i), \quad \forall\, i \in I \tag{5}$$

$$f_t(i,a) \geq 0, \quad \forall\, i \in I,\; a \in A,\; t \in \{0,1, \ldots T\} \tag{6}$$

Note that if the $x_0(i)$ terms sum to 1, constraint (5) ensures $f_0(i,a)$ terms sum to 1 and then constraint (4) ensures $f_t(i,a)$ terms sum to 1 in any year $t$. All facilities are accounted for, every year. It would be possible to add additional (linear) budget or resource constraints. Analysis of dual variables could then tell us the marginal value of increased budgets or extra resources. By using fractional variables, the size of the problem becomes independent of the number of facilities being managed. In addition, we are able to solve the problem using relatively fast linear programming techniques.

The Golabi et al. formulation raises certain concerns. Dependencies between facilities in maintenance costs or deterioration models are not considered. The aforementioned formulation requires translating the model output, for example selecting which sections of roadway comprise the 60% that are to be reconstructed. In addition, the new formulation remains reliant on being able to describe the condition of a facility by assigning it one value in a small and discrete set. Finally, the optimal solution is based on tracking the expected distribution of facilities over time. Eqs. (1) and (2) implicitly survey all possible realizations of deterioration, examining a range of possible future maintenance needs. The Golabi et al. formulation only considers one "sample path" of deterioration, which will almost surely not be observed in practice. The optimal maintenance strategy as well as associated terms (like dual variable values) identified using the Golabi et al. framework will be out of date as soon as the distribution of facilities by condition state fails to match the expected distribution.

Durango-Cohen (2004) previously highlighted the "simplifications that are necessary to model deterioration" in a Markovian context. The writer described a facility using its current condition rating state and current maintenance activity. Estimates of future costs are associated with each possible description and are updated over time. A discrete state space is again used. This state space would have to be quite large if complex descriptions of facility condition were used or if multiple, related facilities were to be jointly managed. Costs are estimated separately for each possible combination of condition state and maintenance action. The amount of data required to set cost estimates accurately grows rapidly with the state space. This is especially troubling since data on infrastructure condition accumulate slowly. The techniques introduced in this paper extend some of the techniques of Durango-Cohen (2004), although the prior work considered infrastructure management in the absence of deterioration or cost estimation models while this work considers infrastructure management in cases where high-quality deterioration and cost estimation models are available but are not readily adaptable for use in a Markovian framework.

## Approximate Dynamic Programming for Infrastructure Management

ADP offers a methodology capable of addressing the shortcomings of the approaches for infrastructure management introduced earlier. In particular, ADP addresses the curses of dimensionality (Powell 2007). An ADP formulation of the infrastructure management problem is introduced in this section. Rather than use a limiting Markovian model of deterioration and solving Eq. (2) a (potentially impossibly large) number of times to find the value function $V$ and a maintenance policy simultaneously, we use simulation to learn a validated and relatively simple model of $V$ which can subsequently be used to select a maintenance policy.

### *Estimating V*

We begin by generalizing the notion of a condition state. Let the condition of a network of facilities, $\mathbf{x}$, contain any and all data relevant to the prediction of future deterioration or estimation of future costs associated with the network. $\mathbf{x}$ likely contains real-valued, historical, and multidimensional data.

We next modify the definition of the value function, $V$. Above $V(x)$ represented the value of being in condition state $x$ before a maintenance action has been selected. Now let $V(\mathbf{x},b)$ represent the value of being in state $\mathbf{x}$ and having selected, but not performed, the maintenance action $b$. The appeal of a "postdecision" value function has been noted in past ADP research (Powell 2007; Simão et al. 2009) and the artificial intelligence community (Sutton and Barto 1998) and allows us to avoid calculating summations across (impossibly large) sets of condition states possible as required by Eqs. (1) and (2).

It is common in the ADP literature to approximate the value function as a linear combination of basis functions. Here $\hat{V}(\mathbf{x},b)$, an estimate of the value of being in state $\mathbf{x}$ and having elected to perform action $b$, would be $\sum_{s=1}^{S} w_s \phi_s(\mathbf{x},b)$. The $\phi_s$ functions are fixed and capture important attributes of the conditions of the facilities being managed and the actions selected while being easy to calculate. $w_s$ terms are weights that will be refined via simulation. Note that in order to define the estimate of the value function we are required only to set $S$ parameters. This is considerably simpler than setting the value function for each condition state $x$ in $X$, for each of $T$ years through the planning horizon as required by Eq. (2).

Techniques introduced earlier required special care in setting up the states of MDP formulations of the infrastructure management problem. The Markovian assumption could be mitigated by redefining condition states to include historical data, provided one knew which historical data were important and how it could be included. Here, decision making is done by examining $\phi_s(\mathbf{x},b)$ values. Therefore, the key becomes carefully setting up the $\phi_s$ functions.

In actuality, it may be difficult to select one set of $S$ basis functions, or even one number $S$ of basis functions to select. Simão et al. (2008) considered several different models of the value function simultaneously in one application of ADP. That work was based on treating similar states as one, at different "levels of aggregation," rather than the more standard functional approximation of the value function introduced earlier. The approach of Simão et al. (2008) is here modified for use given a number of alternate functional models of the value function.

Assume all potential models of the value function are linear combinations of basis functions (themselves not necessarily linear), although this is not required. Weights are assigned to each possible set of basis functions, and the weights vary as ADP solution techniques are applied. Weight can be added to the more accurate models of the value function, as we learn more about the value function. Alternatively (or perhaps additionally), more weight can be attached to more complex models of $V$ as more simulation data become available. We will learn which form and what specific parameter values to use when combining, discretizing, and otherwise manipulating infrastructure condition data as we learn which maintenance actions are optimal.

Let there be $K$ sets of basis functions with set $k$ in $\{1,2,\ldots,K\}$ containing $S_k$ functions. The subscript $k$ is added to the different weights and basis functions to indicate to which set each belongs. Let $g_k$ be the weight attached to the set of basis functions indexed by $k$. $g_k$ values will be updated in a manner identified later. Given these definitions, our overall best estimate of $V(\mathbf{x},b)$ is

$$\hat{V}(\mathbf{x},b)=\sum_{k=1}^{K} g_k \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x},b) \tag{7}$$

### *Learning w and g*

$\phi_{k,s}$ functions are specified initially and learning is done by refining the weights attached to the particular functions and sets of functions. Learning proceeds during the course of a large number of simulation runs, each involving simulating managing infrastructure facilities over a planning horizon. Say that in year $t$ of simulation run $z$, we take action $b_t^z$ when our network of facilities is characterized by condition state (or condition data) $\mathbf{x}_t^z$. Model $k$ would then approximate the future costs of managing the network of facilities to be $\Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z)$. The following (simulated) year we find the network in condition state $\mathbf{x}_{t+1}^z$ and apply action $b_{t+1}^z$. We would like to refine $w_{k,s}$ terms so that, for any model $k$, $\Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z)$ moves closer to a new estimate of future costs based on what was observed during the simulation: $d(\mathbf{x}_t^z,b_t^z)+\alpha V(\mathbf{x}_{t+1}^z,b_{t+1}^z)$. Note that while $V(\mathbf{x}_{t+1}^z,b_{t+1}^z)$ is unknown, we do have the ready approximation $\hat{V}(\mathbf{x}_{t+1}^z,b_{t+1}^z)$.

Let $e_k$ be an estimate of the distance between our original estimate of future costs under model $k$ and the best available approximation 1 simulated year later

$$e_k=d(\mathbf{x}_t^z,b_t^z)+\alpha\hat{V}(\mathbf{x}_{t+1}^z,b_{t+1}^z)-\sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z),\quad \forall\, k\in\{1,2,\ldots,K\} \tag{8}$$

Here, we use a gradient-based approach to move $\Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z)$ closer to $d(\mathbf{x}_t^z,b_t^z)+\alpha\hat{V}(\mathbf{x}_{t+1}^z,b_{t+1}^z)$. Note that $\nabla_{w_k}\Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z)$ is simply $[\phi_{k,1}(\mathbf{x}_t^z,b_t^z),\phi_{k,2}(\mathbf{x}_t^z,b_t^z),\ldots,\phi_{k,S_k}(\mathbf{x}_t^z,b_t^z)]^T$, which is quick and easy to calculate. The distance we move when refining $w_{k,s}$ terms is determined by a step size, as is common in learning problems (Sutton and Barto 1998). Assume the step size varies by simulation run, and is here $\gamma_z$. Then we update each $w_{k,s}$ according to the following rule:

$$w_{k,s}\leftarrow w_{k,s}+\gamma_z e_k\phi_{k,s}(\mathbf{x}_t^z,b_t^z),\quad \forall\, k\in\{1,2,\ldots,K\},\quad s\in\{1,2,\ldots,S_k\} \tag{9}$$

In addition to updating the weights associated with particular basis functions, at a higher level we update weights associated with different models. The weights reflect how much relative importance to assign to the different models when calculating $\hat{V}$. As earlier, note that for each model $k$ we want $\Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z,b_t^z)$ to be close to what we observe in the simulation and approximate one (simulated) year later: $d(\mathbf{x}_t^z,b_t^z)+\alpha\hat{V}(\mathbf{x}_{t+1}^z,b_{t+1}^z)$. Here we update the weights on the different models, putting relatively more weight on relatively more accurate models as determined by "error" terms. Again, we use a step size determined by simulation run. In addition, we want to ensure the sum of the model weights adds to 1 to avoid biasing our overall estimate. Thus, we update $g_k$ according to the following two-step process

$$g_k\leftarrow g_k+\gamma_z\left(1-\frac{e_k}{\max\limits_l e_l}\right),\quad \forall\, k\in\{1,2,\ldots,K\} \tag{10}$$

$$g_k\leftarrow \frac{g_k}{\sum\limits_l g_l},\quad \forall\, k\in\{1,2,\ldots,K\} \tag{11}$$

Note that choosing how to update the weights, both for basis functions and models of value functions, is an art and not a science. Eqs. (8)–(11) define one, but definitely not the only way weights could be updated.

### *Overall Framework*

Given the definitions introduced earlier, an ADP approach to infrastructure management will proceed as follows and as depicted in Fig. 1. We take as given a way to quickly obtain costs, $d(\mathbf{x},b)$ terms, and to simulate taking any action on a set of facilities in any condition to yield the condition of the facilities the following year. Previously, cost and deterioration models had to be exceedingly simple so that they could be incorporated into small linear formulations of maintenance policy optimization problems. Such limitations are no longer necessary. In the end, or at any stage of the algorithm, weights can be multiplied by the basis functions to provide an approximation of the value function. The value function approximation can then be used for maintenance decision making. For example when observing condition state $\mathbf{x}$, take maintenance action(s) $b$ that minimizes $\Sigma_{k=1}^{K} g_k \Sigma_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x},b)$ (assuming optimization is possible; more on this later).

An effort was made to keep the notation used in Fig. 1 consistent with that of Eqs. (7)–(11). However, it is worth noting that it is only necessary to store two states, two actions, and two approximations of the value function at any given time [$\mathbf{x}_{t-1}^z$ and $\mathbf{x}_t^z$, $b_{t-1}^z$ and $b_t^z$, $\hat{V}(\mathbf{x}_{t-1}^z,b_{t-1}^z)$ and $\hat{V}(\mathbf{x}_t^z,b_t^z)$]. Memory requirements

Fix basis functions $\phi_{k,s}(\mathbf{x},b)$ for all $k \in \{1,2,...K\}, s \in \{1,2,...S_k\}$

Set initial values for $g_k$, $w_{k,s}$ for all $k \in \{1,2,...K\}, s \in \{1,2,...S_k\}$

For each sample path $z \in \{1,2,...Z\}$:

  Set $\mathbf{x}_0^z$ according to the (given) initial condition of the network of infrastructure facilities

  $b_0^z = \underset{b \in B}{\operatorname{argmin}} \sum_{k=1}^{K} g_k \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_0^z, b)$

  $\hat{V}(\mathbf{x}_0^z, b_0^z) = \sum_{k=1}^{K} g_k \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_0^z, b_0^z)$

  Compute $\mathbf{x}_1^z$ based on simulating action $b_0^z$ in state $\mathbf{x}_0^z$

  For each year $t \in \{1,2,...T\}$:

    $b_t^z = \underset{b \in B}{\operatorname{argmin}} \sum_{k=1}^{K} g_k \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z, b)$

    $\hat{V}(\mathbf{x}_t^z, b_t^z) = \sum_{k=1}^{K} g_k \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_t^z, b_t^z)$

    $e_k = d(\mathbf{x}_{t-1}^z, b_{t-1}^z) + \alpha\hat{V}(\mathbf{x}_t^z, b_t^z) - \sum_{s=1}^{S_k} w_{k,s}\phi_{k,s}(\mathbf{x}_{t-1}^z, b_{t-1}^z)$ $\quad \forall k \in \{1,2,...K\}$

    Update $w_{k,s} <= w_{k,s} + \gamma_z\ e_k\ \phi_{k,s}(\mathbf{x}_{t-1}^z, b_{t-1}^z)$ $\quad \forall k \in \{1,2,...K\}, s \in \{1,2,...S_k\}$

    Update $g_k <= g_k + \gamma_z(1 - \frac{e_k}{\max_l e_l})$ $\quad \forall k \in \{1,2,...K\}$

    Update $g_k <= \frac{g_k}{\sum_l g_l}$ $\quad \forall k \in \{1,2,...K\}$

    Compute $\mathbf{x}_{t+1}^z$ based on simulating action $b_t^z$ in state $\mathbf{x}_t^z$

**Fig. 1.** Pseudocode description on solving an ADP formulation of an infrastructure management problem

are minimal and are controlled by how many basis functions are considered. In addition, almost all the steps outlined earlier are relatively quick and easy computationally.

The only potential difficulty arises from having to select maintenance actions to perform. Revisiting the problems described in the second section of this paper, the set of actions to take across our set of facilities may include possible combinations of facility-specific actions. If there are $|A|$ actions that may be taken on any facility, we may have to survey $|A|^N$ combinations of actions. We derive much comfort from the fact that we select the basis functions that sum to create $\hat{V}(\mathbf{x},b)$, and thus largely control how easy or difficult it is to find optimal actions to take. In the computational studies presented later in this paper, appropriate definition of basis functions makes potentially difficult problems relatively easy to solve exactly or approximately. This brings up another relevant point: it is not essential that optimal maintenance actions be selected. Learning will continue regardless of what action is taken, although what we learn will be based on how we make decisions.

It is worth noting here that the approach introduced in this paper faces a challenge often referred to as "exploitation versus exploration" (Sutton and Barto 1998; Powell 2007). This challenge has been noted before in the field of infrastructure management (Madanat et al. 2006). We are learning only about actions we take. Inaccuracies in initial cost estimates can lead us to initially favor suboptimal infrastructure maintenance policies. The suboptimal policies will continue to be favored if these policies produce results better than the (inaccurate) estimated costs of alternate policies. We may never learn actual optimal maintenance policies. One way to avoid this potential pitfall is to use an "*e*-greedy strategy," selecting random maintenance actions to take a certain percentage of the time [as in Sutton and Barto (1998); Madanat et al. (2006)]. We are not actually performing random acts of maintenance, just evaluating them in a simulation so that we may learn more about the underlying structure of our problem.

## Computational Studies

This section describes particular implementations of the ADP approach to infrastructure management, as introduced earlier. The goal will be to illustrate how this approach may be applied. We begin with a relatively simple management problem where past research has already identified the optimal solution, so that we may check how well the ADP approach performs in learning the solution. This example is then extended to create a problem whose optimal solution would be difficult to find using traditional infrastructure management techniques.

### *Relatively Simple Example*

We begin by managing three lane-yards of pavement, with each lane-yard treated as a separate facility. Each facility deteriorates independently, one each deteriorating at slow, medium, and fast rates. The various deterioration models are taken from Durango and Madanat (2002) which was itself based on Camahan et al. (1987). The pavement sections are assigned a condition state between 1 and 8 where state 1 represents "unusable" and 8 represents like new. Seven different management actions may be taken on any facility in any 1 year, going from Action 1—do nothing and increasing in intensity up to Action 7—roadway reconstruction. The effect of performing a maintenance action on the condition of a facility is considered to be a random variable, and assumed to follow the truncated normal distribution. Table 1 lists the mean and standard deviation of the effect of maintenance action on pavement condition state.

In previous research (Durango and Madanat 2002), condition data were discretized. Table 1 was used to calculate transition probabilities which served as the deterioration model during decision making. Here the continuous nature of the condition data are maintained when simulating deterioration.

There are agency costs associated with performing maintenance and user costs associated with the condition state of a facility. In order to maintain consistency with earlier descriptions of the infrastructure management problem provided in this paper, agency and user costs have been summed to provide social costs. Social costs by (discrete) condition state and maintenance action are provided in Table 2 based on Durango and Madanat (2002). Certain costs are said to be infinite, meaning that such conditions are to be avoided. In practice, assigning large costs to these conditions ensures they are never encountered when following a rea-

**Table 1.** Mean and Standard Deviation of Action Effects on Condition State

| | Action | | | | | | |
|---|---|---|---|---|---|---|---|
| Deterioration rate | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Slow | (−0.25,0.3) | (0.50,0.3) | (1.75,0.3) | (3.00,0.3) | (4.25,0.3) | (5.50,0.3) | (8.00,0.3) |
| Medium | (−0.75,0.5) | (0.00,0.5) | (1.00,0.5) | (2.00,0.5) | (3.00,0.5) | (4.00,0.5) | (6.00,0.5) |
| Fast | (−1.75,0.7) | (−0.50,0.7) | (0.25,0.7) | (1.00,0.7) | (1.75,0.7) | (2.50,0.7) | (4.00,0.7) |

sonable maintenance policy. In the computational studies done here, the continuous-valued condition of a facility was rounded to an integer value which was then used to assign costs.

Note that each facility is independent, in terms of its associated costs and deterioration model. This means that optimal maintenance across the set of facilities is achieved by determining the policies that minimize social costs for each of the three facilities in turn, and applying these policies to the relevant facilities. Once condition data have been discretized, it is possible to use Eq. (1) to find social cost-minimizing policies for each of the three types of facilities. This has been done previously, in Durango and Madanat (2002). Table 3 contains the long-term (infinite-horizon) cost-minimizing policies, identifying what action to take on a facility as a function of the current condition and deterioration rate of the facility.

An ADP algorithm was provided the data contained in Tables 1 and 2, deterioration models and cost figures, and tasked with finding the data contained in Table 3, the long-term cost-minimizing policy. Chosen basis functions indicated whether or not the facility deteriorating at the associated rate (say fast), was in the associated condition state range (say between 4.5 and 5.5), with the associated maintenance action selected (say 2). There were 168 total basis functions (three deterioration rates × eight condition states × seven maintenance actions). This is quite a large number of basis functions for a relatively simple problem, but was a convenient structure for this problem. The decision of what maintenance action to take on any facility reduced to comparing the weights associated with the different maintenance options and the (known) condition state of the facility and the (known) deterioration rate of the facility. The weights were arbitrarily all set to 0 at the start of the ADP exercise. The weights became estimates of the long-term costs of performing different maintenance action on facilities.

Ten-thousand sample paths were evaluated. Each sample path only involved 2 years of management. The facilities started in randomly generated initial conditions with random actions being applied. This was done to ensure that we learned the values associated with a range of different condition data sets and maintenance actions applied. The following year the condition was evaluated, the best maintenance action (based on current weights) selected, and weights updated according to Eqs. (8) and (9).

Computations were performed in the statistical software R. Running all 10,000 sample paths took roughly 3 s. These three seconds represent the time it takes to learn appropriate value functions, a task that could be accomplished *offline* of actual infrastructure management. Subsequent selection of maintenance actions based on infrastructure condition would take hundredths or thousandths of a second. It is believed that computation times would be significantly lower if the computations were performed using a simpler software language, although the quoted computation times are more than fast enough for practical infrastructure management applications.

After every 100 sample paths had been run, basis function weights were saved. Fig. 2 shows how the weights changed as the algorithm progresses. Seven of the weights are tracked, representing the long-term costs of performing each of the seven different maintenance actions on a facility deteriorating at a medium rate and having a condition rating between 4.5 and 5.5. Table 3 indicates the optimal action for a facility deteriorating at a medium

**Table 2.** Social Costs ($/Lane-Yard) by Condition State and Action

| | Action | | | | | | |
|---|---|---|---|---|---|---|---|
| Condition | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 2 | 25.00 | 27.00 | 35.40 | 37.31 | 41.11 | 44.92 | 50.97 |
| 3 | 22.00 | 23.40 | 30.78 | 32.69 | 36.49 | 40.30 | 47.97 |
| 4 | 14.00 | 14.83 | 21.15 | 23.06 | 26.86 | 30.67 | 39.97 |
| 5 | 8.00 | 8.65 | 12.73 | 14.64 | 18.43 | 22.25 | 33.97 |
| 6 | 4.00 | 4.31 | 6.20 | 8.11 | 11.91 | 15.72 | 29.97 |
| 7 | 2.00 | 2.15 | 4.00 | 5.91 | 9.71 | 13.52 | 27.97 |
| 8 | 0.00 | 0.04 | 1.90 | 3.81 | 7.61 | 11.42 | 25.97 |

**Table 3.** Optimal Maintenance Action as a Function of Condition and Deterioration Rate

| | Condition | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Deterioration rate | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Slow | 7 | 6 | 5 | 4 | 4 | 3 | 2 | 2 |
| Medium | 7 | 7 | 6 | 5 | 4 | 4 | 3 | 2 |
| Fast | 7 | 7 | 7 | 6 | 5 | 4 | 4 | 3 |

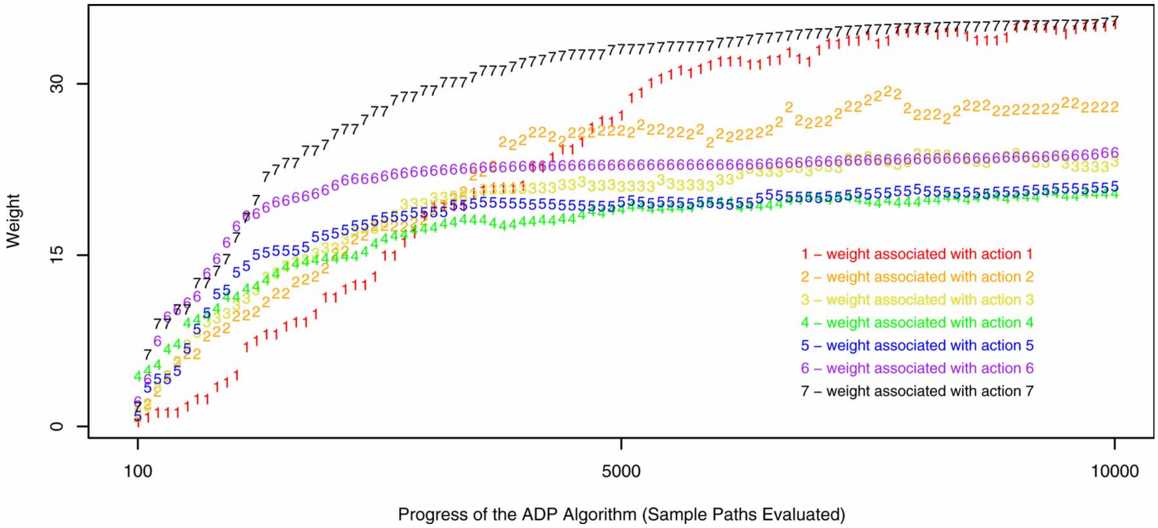


**Fig. 2.** Progression of the ADP algorithm in fixing weights, Trial 1

rate in Condition State 5 is Action 4. This action is also consistently associated with the minimum weight, i.e., long-term cost, after 3,000 sample paths had been evaluated.

Basis function weights were used to generate recommended maintenance policies at different stages of the ADP algorithm's progression. Simulations were run, estimating associated social costs (note that the use of continuous-valued condition data made it difficult to find closed form expressions for expected costs). Twenty-thousand simulation runs were performed to evaluate each policy, with each run consisting of 100 years management of the three lane-yards of pavement. Fig. 3 shows the average social cost, across simulation runs, encountered when using generated maintenance policies. The figure is separated into graphs *a*, at left, and *b*, at right. Graph *a* shows all estimated cost figures, including some excessive costs encountered as our algorithm begins to learn its value function. Graph *b* focuses on data after 2,500 sample paths had been evaluated, making it easier to see how social costs evolve once cost estimates reach a certain level of maturity. Simulation runs were also conducted using the optimal maintenance policy shown in Table 3. Solid horizontal lines on Fig. 3 shows the average social cost encountered when using this policy.

Note that after around 2,500 sample paths were evaluated, the ADP algorithm was able to find a maintenance policy that limited social costs to close to their minimum values. After 7,000 sample paths have been evaluated, the ADP algorithm was able to manage the infrastructure network essentially at minimal social cost. This simple example has shown that an ADP algorithm is able to learn the social optimal infrastructure maintenance policy found by MDP formulations, when one exists.

### More Complex Example

We now turn to a more complex infrastructure management problem. Three-hundred lane-yards are to be managed, with 100 each deteriorating at slow, medium, and fast rates. Resource and budget constraints link the facilities. One constraint requires that at most 10 lane-yards of pavement may be reconstructed in any

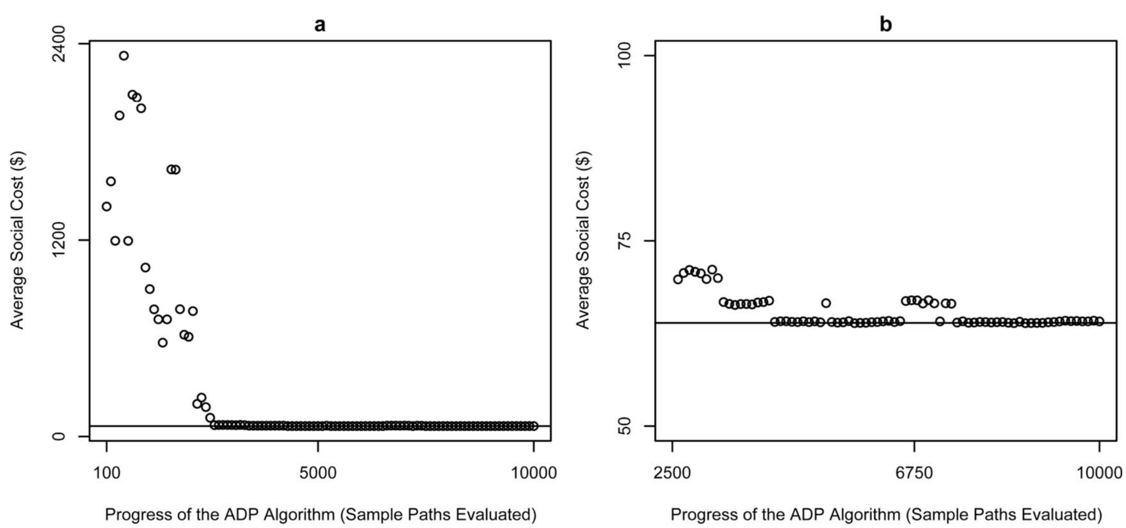


**Fig. 3.** Progression of the ADP algorithm in minimizing social costs, Trial 1

given year. This could reflect a situation where reconstruction requires special resources that are in limited supply. It might also be that reconstruction requires blocking a roadway for a significant period of time and political constraints limit the number of times this may be done per year. Additional constraints require that at most 70 facilities can be maintained using any one of the intermediate maintenance actions between doing nothing and reconstruction. These roughly reflect the fact that maintenance actions require specific resources that may be in limited supply. Given these definitions, we have six resource constraints linking 300 facilities to be managed. Although this problem is not especially complex by the standards of practical infrastructure management, it is not clear how a relevant MDP formulation could be constructed.

An ADP formulation is used here. As in the preceding example, only one model of the value function was used. Extending the previous example, the value function included basis functions that counted how many facilities were deteriorating at the associated rate, were in the associated condition range, with the associated maintenance action selected. As before, there were 168 total basis functions. The results of the previous example were used to initialize weights. Note that these values were based on an implicit assumption that decision making is done in the absence of budget and resource constraints. They are thus imperfect for the new problem where we must consider such constraints.

Our choice of maintenance action had been identified as $\min_{b \in B} \sum_{s=1}^{S} w_s \phi_s(\mathbf{x}, b)$ but note that the chosen basis functions are associated with different deterioration rates (given), conditions (given), and actions (chosen). Thus we can replace the index $s$ with indices $\delta$, $i$, and $a$ over deterioration rates, facility conditions, and maintenance actions, respectively. This yields a new formulation $\min_{b \in B} \sum_{\delta,i,a} w_{\delta,i,a} \phi_{\delta,i,a}(\mathbf{x}, b)$. Now note that $\phi_{\delta,i,a}(\mathbf{x}, b)$ just counts how many facilities are associated with deterioration rate $\delta$, condition $i$, and action $a$. We can replace the function $\phi$ with an indicator variable $\psi$ where $\psi_{f,a}$ indicates if facility $f$ is associated with action $a$, as well as a few (given) parameters: $\delta(f)$, the deterioration rate of facility $f$, and $i(f)$, the condition of facility $f$. Our decision problem, including resource constraints becomes

$$\min_{\psi} \sum_{f} \sum_{a} w_{\delta(f),i(f),a} \psi_{f,a} \tag{12}$$

subject to

$$\sum_{f} \psi_{f,2} \leq 70 \tag{13}$$

$$\sum_{f} \psi_{f,3} \leq 70 \tag{14}$$

$$\sum_{f} \psi_{f,4} \leq 70 \tag{15}$$

$$\sum_{f} \psi_{f,5} \leq 70 \tag{16}$$

$$\sum_{f} \psi_{f,6} \leq 70 \tag{17}$$

$$\sum_{f} \psi_{f,7} \leq 10 \tag{18}$$

$$\sum_{a} \psi_{f,a} = 1 \quad \forall f \tag{19}$$

Note that the new formulation is a classical multichoice multidimensional knapsack problem (MMKP). The term multichoice refers to the fact that we must choose exactly one maintenance action, out of a small discrete set of possibilities, to perform on each facility being managed. The term multidimensional refers to the fact that we have several budget and resource constraints to consider when making this choice. There are numerous efficient algorithms available for exactly or approximately solving MMKPs [see for instance Parra-Hernández and Dimopoulos (2005)]. Our choice of basis functions is not perfect; it does not allow us to fully capture the relationship between facilities that the new resource constraints impose or the continuous nature of the condition data. However, it does allow us to relatively efficiently find feasible, high-quality maintenance strategies by using existing operations research techniques.

Forty-thousand sample paths were generated. As in the preceding example, each sample path consisted of 2 years of management starting with facilities in random initial condition states and with random acts of maintenance being performed. Observations were then made of how the facilities had deteriorated 1 simulated year later and what maintenance actions were then optimal. Eqs. (8) and (9) were used to update weights. MMKP formulations for optimal maintenance action selection were solved to optimality (larger scale or more complex problem formulations might have required heuristic approximate solutions, but would still have been possible in an ADP framework).

Fig. 4 tracks the weights corresponding to those shown in Fig. 2 as the new, more complex problem is evaluated. Note that overall the costs associated with a lane-yard of pavement being (roughly) in Condition State 5 have increased. Resource constraints have made it impossible to maintain facilities enough to keep social costs very low. Estimated costs do decrease, for certain actions, in later stages of the ADP algorithm. This reflects the fact that we are learning optimal maintenance strategies for the new problem, which reduces future costs. The new constraints alter how the different maintenance options compare to one another. Actions 6 and 5 are now preferred to Action 4, Action 7 preferred to Actions 3, 2, and 1, whereas in the earlier example the reverse was true. Limitations on resources available to perform maintenance make it more cost effective to perform more severe maintenance when possible.

## Future Challenges

In the computational studies analyzed earlier, basis functions counted numbers of facilities whose current condition ratings were in given ranges. Further research is needed to define basis functions that better honor the continuous nature of the data and incorporate historical condition data, where such considerations are important. In the computational studies analyzed earlier, the only relationships between facilities considered were constraints on the numbers of times different maintenance actions could be performed networkwide at one time. Further research should examine how more complex relationships, including economic, structural, and stochastic dependencies (Dekker et al. 1997) could be captured. The computational studies undertaken here do not address all of the issues introduced in this paper. However the studies do begin to illustrate the power of the ADP approach to infrastructure management, which is itself able to address the introduced issues.

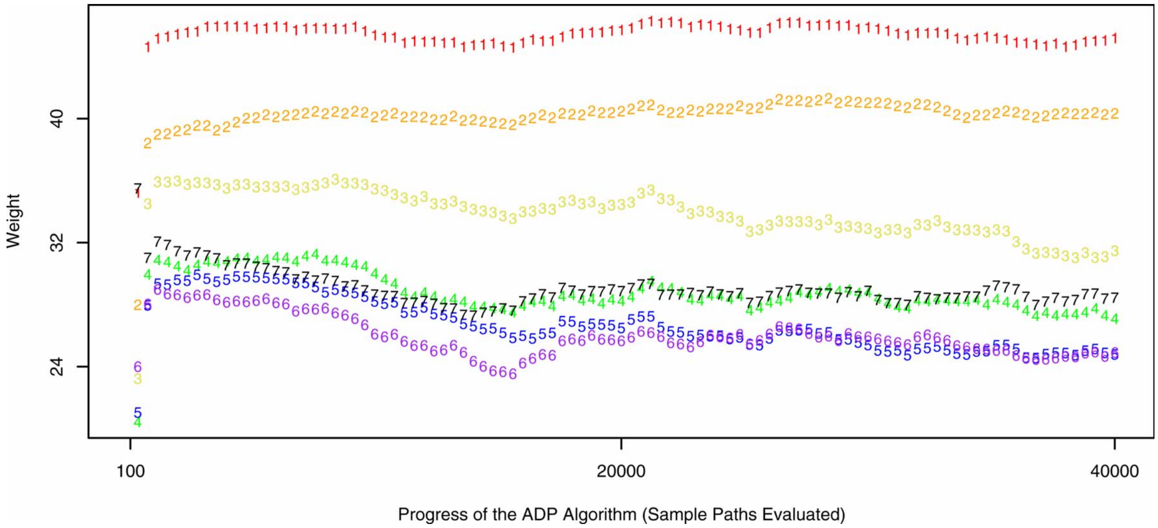


**Fig. 4.** Progression of the ADP algorithm in fixing weights, Trial 2

## Conclusions

This paper introduced the use of ADP for infrastructure management. In comparison to traditional techniques based on MDPs, the introduced methodology was shown to be more flexible in the forms of infrastructure condition data it accepted as well as the format of deterioration and cost estimation models. The new approach enables managing large networks of related facilities. Computational studies showed the new approach was able to both learn optimal infrastructure maintenance policies, when such policies could be confirmed, and to approximate the long-term costs associated with different infrastructure maintenance policies, for more complex problems.

Further research is needed to identify what functions and data sources are relevant when using infrastructure condition and maintenance data to approximate long-term management costs. Related research is also needed to identify how to optimize over the identified functions.

## References


Carnahan, J., Davis, W., Shahin, M., Kean, P., and Wu, M. (1987). "Optimal maintenance decisions for pavement management." *J. Transp. Eng.*, 113(5), 554–572.

Chan, W., Fwa, T., and Tan, J. (2003). "Optimal fund-allocation analysis for multidistrict highway agencies." *J. Infrastruct. Syst.*, 9(4), 167–175.

Chu, C.-Y., and Durango-Cohen, P. (2007). "Estimation of infrastructure performance models using state-space specifications of time series models." *Transp. Res., Part C: Emerg. Technol.*, 15(1), 17–32.

Dekker, R., Wildeman, R., and Duyn Schouten, F. (1997). "A review of multi-component maintenance models with economic dependence." *Mathematical Models of Operations Research*, 45, 411–534.

Durango, P., and Madanat, S. (2002). "Optimal maintenance and repair policies in infrastructure management under uncertain facility deterioration rates: An adaptive control approach." *Transp. Res., Part A: Policy Pract.*, 36(9), 763–778.

Durango-Cohen, P. (2004). "Maintenance and repair decision making for infrastructure facilities without a deterioration model." *J. Infrastruct. Syst.*, 10(1), 1–8.

Durango-Cohen, P., and Sarutipand, P. (2007). "Capturing interdependencies and heterogeneity in the management of multifacility transportation infrastructure systems." *J. Infrastruct. Syst.*, 13(2), 115–123.

Golabi, K., Kulkarni, R., and Way, G. (1982). "A statewide pavement management system." *Interfaces*, 12(6), 5–21.

Golabi, K., and Shepard, R. (1997). "Pontis: A system for maintenance optimization and improvement of US bridge network." *Interfaces*, 27(1), 71–88.

Gopal, S., and Majidzadeh, K. (1991). "Application of Markov decision processes to level-of-service-based systems." *Transp. Res. Rec.*, 1304, 12–17.

Madanat, S., Park, S., and Kuhn, K. (2006). "Adaptive optimization and systematic probing of infra-structure system maintenance policies under model uncertainty." *J. Infrastruct. Syst.*, 12(3), 192–198.

Mishalani, R., and Madanat, S. (2002). "Computation of infrastructure transition probabilities using stochastic duration models." *J. Infrastruct. Syst.*, 8(4), 139–148.

Parra-Hernández, R., and Dimoloulos, N. (2005). "A new heuristic for solving the multichoice multidimensional knapsack problem." *IEEE Trans. Syst. Man Cybern., Part A. Syst. Humans*, 35(5), 708–717.

Powell, W. (2007). *Approximate dynamic programming: Solving the curses of dimensionality*, Wiley, New York.

Robelin, C.-A., and Madanat, S. (2007). "History-dependent bridge deck maintenance and replacement optimization with Markov decision processes." *J. Infrastruct. Syst.*, 13(3), 195–201.

Simão, H., Day, J., George, A., Gifford, T., Nienow, J., and Powell, W. (2009). "An approximate dynamic programming algorithm for large-scale fleet management: A case application." *Transportation Science*, 43(2), 178–197.

Sutton, R., and Barto, A. (1998). *Reinforcement learning*, MIT Press, Cambridge, Mass.