Incorporating Network Considerations into Pavement Management Systems: A Case for Approximate Dynamic Programming

Aditya Medury [*]and Samer Madanat[†]

February 27, 2013

Abstract

The objective of infrastructure management is to provide optimal maintenance, rehabilitation and replacement (MR&R) policies for a system of facilities over a planning horizon. While most approaches in the literature have studied the decision-making process as a finite resource allocation problem, the impact of construction activities on the road network is often not accounted for. The state-of-the-art Markov decision process (MDP)-based optimization approaches in infrastructure management, while optimal for solving budget allocation problems, become internally inconsistent upon introducing network constraints. In comparison, approximate dynamic programming (ADP) enables solving complex problem formulations by using simulation techniques and lower dimension value function approximations. In this paper, an ADP framework is proposed, wherein capacity losses due to construction activities are subjected to an agency-defined network capacity threshold. A parametric study is conducted on a stylized network configuration to infer the impact of network-based constraints on the decision-making process.

1 Introduction

The objective of transportation infrastructure management is to provide optimal maintenance, rehabilitation and replacement (MR&R) policies for a system of facilities (roads, bridges, tunnels, etc.) over a planning horizon. While most approaches in the literature have studied it as a problem of optimal allocation of limited financial resources, the interdependence between facilities, as introduced by a unifying network configuration, is often not accounted for. Dekker et al. (1997) suggests that interactions between individual components of an infrastructure system can be classified into three different types: economic dependence (benefits/costs associated with joint maintenance), structural dependence (set of facilities collectively determining system performance such as connectivity or capacity) and stochastic dependence (presence of correlated deterioration factors like environment, loading). The work presented in this paper focuses on incorporating the structural interdependence among facilities into the MR&R decision-making process.

The implementation of MR&R activities on road networks can result in significant delays to travelers due to loss in network capacity, detours, etc. According to one estimate, more than 60 million vehicles per hour per day of capacity were lost due to work zone activity on the National Highway System over a two week period in the United States in 2001 (Wunderlich and

[*] Graduate Student Researcher, Dept. of Civil and Environmental Engineering, Univ. of California, Berkeley, 116 McLaughlin Hall, Berkeley, CA 94720 (corresponding author). E-mail: amedury@berkeley.edu

[†] Professor, Dept. of Civil and Environmental Engineering, Univ. of California, Berkeley, 110 McLaughlin Hall, Berkeley, CA 94720. E-mail: madanat@ce.berkeley.edu

Hardesty 2003). Given that the impact of scheduling work zones, especially in saturated traffic flow conditions, can be severe, it is important to systematically address and incorporate these user concerns within the decision-making process.

2 Literature Review

Discrete-state, discrete-time Markov decision process (MDP)-based frameworks have been widely used in infrastructure management, especially in the context of uncertainty in the underlying facility performance models. One of the first instances of using MDP frameworks for infrastructure management was the development of the Arizona pavement management system (Golabi et al. 1982). The LP-based approach utilized *randomized* policies to effectively accommodate budget constraints within the MR&R decision-making problem. Randomized policies are probabilistic in nature, wherein the optimal policy for a facility in a given condition state is defined as a probability distribution function across two or more actions. Randomized policies are also proven to be optimal in a constrained MDP setting (Kallenberg 1994). However, one of the limitations is that due to its probabilistic nature, the randomized policies do not directly translate into facility-specific recommendations. In order to address this issue, Medury and Madanat (2012) extended the LP-based approach to provide facility-specific policies, while retaining the optimality of the original problem formulation. Other applications of randomized policies include the bridge management system, Pontis (Golabi and Shepard 1997), Smilowitz and Madanat (2000), Kuhn and Madanat (2005) and Madanat et al. (2006), among others.

Other efforts to obtain deterministic facility-specific policies have led researchers to develop bottom-up approaches, wherein individual facility-level MDPs are coupled with each other through a common budget constraint. Such approaches either rely on decomposing the system-level MDP problem into facility-level and system-level problems (Sinha et al. 1988; Yeo et al. 2012), or utilize Lagrangian-multiplier based algorithms to decouple system-level MDP into multiple facility-specific MDPs (Ohlmann and Bean 2009). The benefit of obtaining facility-specific policies is that inter-facility interactions, such as network-based constraints, can also be modeled within the optimization formulation. However, since the bottom-up approaches are heuristic in nature, the optimality of the solutions cannot be guaranteed.

In recent times, reinforcement learning/approximate dynamic programming (ADP) algorithms have also been applied to infrastructure management problems. Durango-Cohen (2004) applied some learning techniques to facility-level problems under imperfect deterioration information scenarios. Gao and Zhang (2009) and Kuhn (2010) utilized ADP frameworks to provide system-level MR&R policies in the context of budget allocation problems. Through the use of simulation techniques, in conjunction with value function approximations, ADP allows for complex constraints to be included into the decision-making framework. However, one of the limitations of the approach is that theoretical optimality guarantees cannot be provided on the solutions thus obtained. In addition, while Gao and Zhang (2009) and Kuhn (2010) demonstrate the applicability of ADP-based approaches to infrastructure management, its performance vis-a-vis other MDP-based approaches is not evaluated.

Based on the overview of the existing MDP-based frameworks, it is observed that most system-level MR&R decision-making methodologies consider the resource allocation problem without recognizing the presence of an underlying network configuration. While network effects have previously been incorporated into MR&R decision-making in the context of deterministic performance models (Ouyang 2007, Durango-Cohen and Sarutipand 2007, Ng et al. 2009), these approaches are not applicable for stochastic rates of deterioration.

On the operations side of infrastructure management, studies on work zone scheduling of maintenance activities have sought to incorporate the interaction between individual facilities at the system level. Fwa et al. (1998) formulated an integer programming problem to provide an optimal schedule of activities which minimizes traffic delays caused by the resulting lane closures. Chang et al. (2001) employed a Tabu search-based methodology for an optimal work zone schedule. Hajdin and Lindenmann (2007) used the notion of corridors as a bundle of maintenance activities on various assets (roads, bridges, tunnels, etc.) in order to determine optimal corridor lengths on a highway network that minimize user and agency costs. However, while these formulations highlight the interdependence between individual facilities at the system-level, the work zone scheduling problems do not concern themselves with the selection of MR&R activities, which also involves trade-offs associated with deferring maintenance.

Finally, at the policy level, state agencies utilize different construction options, which trade off duration, safety and loss of road capacity, based on the extent of closure undertaken. For example, urban freeway repair projects in California traditionally used seven or ten-hour nighttime closures because daytime closures were seen to cause unacceptable delays to weekday peak travel. However, nighttime closures are also associated with adverse impacts, such as poor safety for road users and construction crews, and longer closure times (Lee and Ibbs 2005). In recognition of these drawbacks, decision-support tools such as Construction Analysis for Pavement Rehabilitation Strategies (CA4PRS) have been developed in order to assess multiple highway rehabilitation strategies such as continuous (round-the-clock) operations during 55 hour weekend closures, or 72 hour weekday along with night-time closures, with the help of traffic simulation models. Such analysis is currently absent for system-level MR&R decision-making.

The objective of this research is to extend the state-of-the-art Markov decision process (MDP)-based methodologies in infrastructure management to accommodate the user concerns pertaining to the adverse impacts of MR&R activities. In particular, the use of ADP to model complex problems in infrastructure management is motivated, since randomized policies, while optimal for budget allocation problems, are restrictive in their modeling capabilities. In the following sections, the system-level MR&R decision-making problem is first formulated as a MDP. Subsequently, the merits and demerits of randomized policy-based frameworks and ADP are compared, so as to identify the most suitable methodology for incorporating network constraints. The two frameworks are first presented in a general fashion, so as to simplify the problem formulation and facilitate a holistic understanding of the approaches. Finally, a parametric study is conducted on a stylized network to infer the impact of network-based constraints on the decision-making methodologies.

3 System-level MDP Problem

An exact solution to a finite horizon system-level MDP problem can be obtained by employing a backward recursive dynamic programming approach, which is based on Richard Bellman's *principle of optimality* (Bellman 1954). The optimality rule states the following: *an optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions.* In other words, the principle alludes to the Markovian assumption, wherein the optimal policy in a given year of the planning horizon is only dependent on the state of the system in that year.

In the context of MR&R decision-making, the problem is formulated as follows: if the state of the pavement network at the beginning of year t is represented by $\mathbf{s}_t$, then the optimal cost-to-go from year t to the end of the planning horizon, T, can be recursively obtained using the following set of equations:

$$V_\tau(\mathbf{s}_\tau) = \min_{\mathbf{a}_\tau \in \mathbb{A}_c^N} \left[c(\mathbf{s}_\tau, \mathbf{a}_\tau) + \alpha \sum_{\mathbf{s}' \in \mathbb{S}^N} P(\mathbf{s}'|\mathbf{s}_\tau, \mathbf{a}_\tau) V_{\tau+1}(\mathbf{s}') \right] \qquad \forall \mathbf{s}_\tau \in \mathbb{S}^N, \quad \tau = T, \ldots, t, \qquad (1)$$

$$a_\tau(\mathbf{s}_\tau) = \arg\min_{\mathbf{a}_\tau \in \mathbb{A}_c^N} \left[c(\mathbf{s}_\tau, \mathbf{a}_\tau) + \alpha \sum_{\mathbf{s}' \in \mathbb{S}^N} P(\mathbf{s}'|\mathbf{s}_\tau, \mathbf{a}_\tau) V_{\tau+1}(\mathbf{s}') \right] \qquad \forall \mathbf{s}_\tau \in \mathbb{S}^N, \quad \tau = T, \ldots, t, \qquad (2)$$

$$V_{T+1}(\mathbf{s}_{T+1}) = \bar{V}(\mathbf{s}_{T+1}) \qquad \forall \mathbf{s}_{T+1} \in \mathbb{S}^N, \qquad (3)$$

where,

$V_\tau(\mathbf{s}_\tau)$: value function associated with state, $\mathbf{s}_\tau$, in year τ, representing the minimum expected cost-to-go from year τ to the end of the planning horizon,
$a_\tau(\mathbf{s}_\tau)$: the optimal policy associated with state, $\mathbf{s}_\tau$, in year τ,
$\bar{V}(\mathbf{s}_{T+1})$: salvage value associated with state, s_{T+1}, at the end of the planning horizon,
$c(\mathbf{s}_\tau, \mathbf{a}_\tau)$: unit costs associated with implementing the action set, $\mathbf{a}_\tau$, when the network is in state, $\mathbf{s}_\tau$,
$P(\mathbf{s}'|\mathbf{s}_\tau, \mathbf{a}_\tau)$: one-year transition probability of being in state, $\mathbf{s}'$, in year $\tau + 1$, given that the network is in state, $\mathbf{s}_\tau$, in year τ, and action set, $\mathbf{a}_\tau$, is implemented,
α: discount amount factor,
N: number of facilities in the network,
$\mathbb{S}^N$: feasible state space of the network,
$\mathbb{A}_c^N$: constrained feasible action space of the network (constraints can include finite budget availability, scheduling, network flow, etc.).

Using equations 1-3, the multi-stage MR&R decision-making problem can be optimally solved as follows: at first, the optimal policies and costs-to-go are determined for all states in the final year of the planning horizion, year T, using the agency-chosen salvage values for year $T + 1$ (equation 3). Once the optimal costs-to-go for year T are calculated, the optimal costs-to-go and policies for year $T - 1$ are obtained by using equations 1 and 2. Similarly, the policies for the preceding years are solved for by recursing backwards in time.

The advantage of a backward recursive dynamic programming approach is that it guarantees an optimal policy. However, its limitation is that it suffers from the *curse of dimensionality*, wherein the computational costs associated with solving the Bellman equation increase exponentially with the size of the network. A major consideration in this regard is that equation 1 is required to be solved over the entire state space, which increases exponentially with the number of facilities in the network. For instance, if $|\mathbb{S}| = 8$, the number of condition states possible for $N = 5$ are $8^5 = 32{,}768$; whereas for $N = 10$, the combinations rise up to $8^{10} = 1{,}073{,}741{,}824$. Other factors contributing towards an increase in computational complexity include, looping over the entire state space to calculate the expectation of the future cost-to-go $\left(\sum_{\mathbf{s}' \in \mathbb{S}^N} P(\mathbf{s}'|\mathbf{s}_\tau, \mathbf{a}_\tau) V_{\tau+1}(\mathbf{s}')\right)$, and determining the optimal action from a feasible action space, $\mathbb{A}_c^N$, which also increases exponentially with N. Consequently, such an approach is not suitable for a system-level MR&R decision-making problem.

4 Simultaneous Network-Level Optimization Framework

In the context of budget allocation problems, Golabi et al. (1982) developed a linear programming (LP)-based MDP framework which provides optimal policies for a system-level MR&R decision-making problem. As indicated previously, a limitation of this approach is that the recommended policies are probabilistic or *randomized* in nature. Consequently, these policies need to be translated into facility-specific decisions by either using engineering judgment, or additional sub-routines.

In order to resolve this issue, Medury and Madanat (2012) proposed a simultaneous network optimization (SNO) approach, which extends the LP-based formulation to allow for facility-specific policies for the current year of implementation. The underlying problem formulation utilizes the randomized policies for estimating the future costs, while associating facility-specific binary integer variables with each MR&R action for the current year of decision-making. The resulting optimization formulation for a budget allocation problem in a given year t of the planning horizon can be represented as follows:

$$\begin{aligned} \min_{x,w} & \sum_{i=1}^{N} \sum_{a \in \mathbb{A}} c\left(\mathbf{s}_t(i), a\right) x_{i,a,t} \\ & + \; N \left(\sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} \left(\sum_{\tau=t+1}^{T} \alpha^{\tau-t} c(s,a) w_{s,a,\tau} + \alpha^{T+1-t} \; \tilde{V}(s) w_{s,a,T+1} \right) \right), \end{aligned} \tag{4}$$

subject to

$$\sum_{a \in \mathbb{A}} x_{i,a,t} \qquad = 1 \qquad \forall i = 1, \ldots, N, \tag{5}$$

$$\frac{1}{N} \left(\sum_{i=1,\ldots,N | \mathbf{s}_t(i) = s'} x_{iat} \right) \qquad = w_{s',a,t} \qquad \forall r \in \mathbb{S}, \forall a \in \mathbb{A}, \tag{6}$$

$$\sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} w_{s,a,\tau} \qquad = 1 \qquad \forall \tau = t + 1, \ldots, T, \tag{7}$$

$$N \left(\sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} c(s,a) w_{s,a,\tau} \right) \qquad \leq B_\tau \qquad \forall \tau = t, \ldots, T, \tag{8}$$

$$\sum_{s' \in \mathbb{S}} P\left(s' | s, a\right) w_{s',a,\tau} \qquad = \sum_{a \in \mathbb{A}} w_{s,a,\tau+1} \qquad \forall s \in \mathbb{S}, \forall \tau = t, \ldots, T, \tag{9}$$

$$x_{i,a,t} \in \{0,1\}, w_{s,a,\tau} \qquad \in [0,1] \qquad \begin{aligned} & \forall i = 1, \ldots, N, \forall s \in \mathbb{S}, \\ & \forall a \in \mathbb{A}, \forall \tau = t, \ldots, T + 1, \end{aligned}$$

where,

$x_{i,a,t}$: 1 if action, a, is selected for facility i; 0 otherwise (t refers to the current year),

$w_{s,a,\tau}$: fraction of the network in state, s, to which action, a, is applied in year τ, where τ is representative of all the future years.

B_τ: agency's annual budget in year τ.

Equation 4 refers to the objective function, which is to minimize the expected system-level costs over the planning horizon; equation 5 ensures that exactly one action (including do-nothing) is chosen for every facility in the current year of implementation; equation 6 relates the binary integer variables, $x_{i,a,t}$, to the randomized policy, $w_{s,a,\tau}$; equation 7 constrains the randomized policies to sum up to one; equation 8 constrains the selection of MR&R activities to be within the annual agency budget; and finally, equation 9 refers to Chapman-Kolmogorov equation.

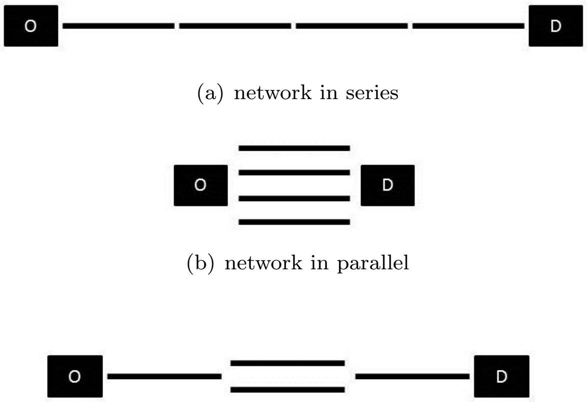
In terms of implementation, SNO needs to be solved at the beginning of every year, since facility-specific policies are only available for the first year for the planning horizon. SNO is optimal for budget allocation problems, as it utilizes randomized policies to estimate the expected future costs. It also offers an advantage over the LP-based Golabi et al. (1982) framework, in that additional facility-dependent constraints can be imposed on the policies obtained for the first year of implementation. However, these constraints cannot be simultaneously incorporated in the future time periods, as the randomized policies are probabilistic in nature. Hence, while SNO is optimal for budget allocation problems, it provides sub-optimal, but feasible, policies upon the inclusion of facility-dependent constraints, like network-based constraints.

5 Incorporating Network Considerations

The recognition of an over-arching network configuration introduces several challenges, as well as opportunities, for system-level MR&R decision-making. In this paper, network-induced inter-facility interactions are examined in the context of structural interdependence among facilities, wherein MR&R activities on individual facilities leads to a cumulative effect on the capacity of the network.

Existing system-level MR&R decision-making paradigms do not adequately account for the impacts of construction activities on the road traffic. Consequently, the measures taken by agencies to mitigate travel time increases occur at the project-level, i.e., once the MR&R activities have already been determined. However, in order to effectively address these user concerns at the system-level, the network configuration can provide insights into determining how each facility affects the system performance (such as the capacity or connectivity of the network).

For instance, consider a system comprising of four road segments. Without explicitly identifying the individual pavements within the network, it is impossible to gauge the impact of the proposed maintenance activities. However, it can be seen that if all the facilities in the system are arranged in series, as shown in figure 1(a), then each facility is critical for the functioning of the network. As a result, a partial/complete road closure during peak hours of traffic will adversely affect the traffic. In comparison, if all the road segments are in parallel (figure 1(b)), the network exhibits a very high level of redundancy. Consequently, potential road closures can be accommodated by rerouting traffic through parallel routes. A more realistic network would perhaps comprise of links in both series and parallel, as shown in figure 1(c). Hence, in order to better mitigate the impact of construction activities on road users, the identification of optimal system-level MR&R policies should capture the relation between the road segments in a systematic manner.



(a) network in series

(b) network in parallel

(c) network with road segments in a combination of series and parallel

Figure 1: Different types of network configurations

6 Approximate Dynamic Programming

Approximate dynamic programming is a MDP-based modeling framework which seeks to overcome the dimensionality issues associated with traditional dynamic programming methods. It employs an algorithmic strategy of stepping forward through time, which obviates the need to loop through the entire state space in future time periods. Researchers have applied ADP techniques to diverse problems, ranging from learning how to play backgammon (Tesauro 1995) to large-scale vehicle fleet management (Simão et al. 2009). The most notable references pertaining to ADP include Bertsekas and Tsitsiklis (1996), Sutton and Barto (1998) and Powell (2007).

The term "approximate" in ADP refers to the fact that the value functions need to be estimated using simulations, and may not always converge to the optimal values. In comparison, a backward recursive dynamic programming approach guarantees optimal policies, as it exactly computes the value functions for all years. However, the advantage of ADP is that it provides a framework to solve high-dimensional MDP problems which cannot be attempted otherwise using analytical techniques. It also allows for complex interactions, such as the network considerations discussed in section 5, to be represented within the decision-making problem. As a result, while attempting to solve MDP problems using ADP, the inherent trade-off lies in sacrificing optimality for computational tractability.

There are many strategies available for estimating value functions and the resulting policies in the ADP literature. The details of the particular methodology undertaken in this work are outlined in the following subsections: firstly, the need for approximating the value function with lower dimensional parametric functions is motivated. Subsequently, the procedures for stepping forward through time are detailed, and finally, the learning algorithm for updating the value function approximation is discussed.

6.1 Value Function Approximation

In a backward recursive dynamic programming approach, the cost-to-go function, $V_t(\mathbf{s}_t)$, is stored using a standard look-up table representation, wherein each element of the state space ($\mathbb{S}^N$) is uniquely identified. A look-up table representation in the context of an ADP framework has several shortcomings. Firstly, the memory requirements can be unsurmountable for high-dimensional problems. Secondly, calculating the future cost-to-go requires looping over the entire state space, which leads to computational inefficiencies. Thirdly, the learning rate associated with a look-up table representation can be very slow, since each element of the state space must be visited multiple times in order to develop a good estimate of its cost-to-go. Consequently, there is a need to approximate the value function so as to make its representation scalable from the point of view of computational efficiency as well as memory allocation.

The ADP literature recommends different techniques to work with large state spaces. One such approach relies on the use of aggregation methods to scale down the state and action spaces. An alternate approach is to develop analytical functions using parameters which exploit the structure of the problem in order to capture important attributes of the value function. In this paper, a combination of both approaches is adopted for approximating the value function. In particular, a set of linear, separable basis functions are chosen, which can be summed up to provide an estimate of the future cost-to-go for a given state-action pair, $(\mathbf{s}_t, \mathbf{a}_t)$, in year t:

$$\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t) = \sum_{b=1}^{B_t} \theta_{tb}(\mathbf{s}_t, \mathbf{a}_t)\phi_{tb}(\mathbf{s}_t, \mathbf{a}_t), \tag{10}$$

where,

$\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t)$: expected cost-to-go from year $t+1$ to the end of the planning horizon, when action set, $\mathbf{a}_t$, is chosen for state, $\mathbf{s}_t$; also referred to as the Q-function,

$\phi_{tb}(\mathbf{s}_t, \mathbf{a}_t)$: user-specified basis function which captures specific attributes of the cost-to-go function; also referred to as the Q-factor;

$\theta_{tb}(\mathbf{s}_t, \mathbf{a}_t)$: weight associated with a given basis function, which is iteratively updated within the ADP framework.

Using $\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t)$, system-level MR&R policies can be obtained as follows:

$$\min_{\mathbf{a}_t \in \mathbb{A}_c^N} \quad c(\mathbf{s}_t, \mathbf{a}_t) + \alpha \tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t). \tag{11}$$

Herein, it is noteworthy to distinguish $\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t)$ from $V_t(\mathbf{s}_t)$. $V_t(\mathbf{s}_t)$ represents the optimal expected cost-to-go associated with state, $\mathbf{s}_t$, from year t to the end of the planning horizon. A limitation of approximating the optimal cost-to-go is that it still necessitates a calculation of the expected future cost-to-go for a given action set, $\mathbf{a}_t$. In comparison, the Q-function can be viewed as approximating the expected future cost-to-go using an augmented state, $(\mathbf{s}_t, \mathbf{a}_t)$, which leads to computational savings. The equivalence between $\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t)$ and $V_t(\mathbf{s}_t)$ can be established by comparing equations 11 and 1:

$$\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t) \approx \sum_{\mathbf{s}' \in \mathbb{S}} P(\mathbf{s}' | \mathbf{s}_t, \mathbf{a}_t) V_{t+1}(\mathbf{s}'). \tag{12}$$

It is important to note that $\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t)$ excludes the costs incurred in year t, since the corresponding costs can be easily calculated using $c(\mathbf{s}_t, \mathbf{a}_t)$.

6.2 Stepping forward through time

For a given ADP iteration, n, let the current estimate of the Q-function in year t be defined as $\tilde{Q}_t^{n-1}(\mathbf{s}_t^n, \mathbf{a}_t)$, where $\mathbf{s}_t^n$ is the state of the network. Also, let the optimal action set selected on the basis of equation 11 be defined as $\mathbf{a}_t^n$. Once $\mathbf{a}_t^n$ has been identified, the process of stepping forward in time is undertaken by generating a sample realization of the future state of the system, $\mathbf{s}_{t+1}^n$, using a *Monte Carlo simulation* procedure. The procedure involves using a uniform random number generator to yield a feasible state transition from the transition probability distribution, $P(\mathbf{s}_{t+1}^n | \mathbf{s}_t^n, \mathbf{a}_t^n)$. A symbolic representation for the state transition procedure is:

$$\mathbf{s}_{t+1}^n = \Psi(\mathbf{s}_t^n, \mathbf{a}_t^n, \omega^n), \tag{13}$$

wherein, ω^n represents a vector of uniform random numbers generated from the distribution, U(0,1).

Once a sample state of the system in year $t+1$ is ascertained, the optimal action set for year $t+1$, $\mathbf{a}_{t+1}^n$, can be solved for by using equation 11. Finally, a sequence of state-action pairs realized during an iteration of the planning horizon, $\left[(\mathbf{s}_1^n, \mathbf{a}_1^n), (\mathbf{s}_2^n, \mathbf{a}_2^n), \ldots, (\mathbf{s}_T^n, \mathbf{a}_T^n), \mathbf{s}_{T+1}^n\right]$, is referred to as a *sample path*.

6.3 Updating the value function approximation

The concept of a sample path is fundamental to the idea of learning in approximate dynamic programming (Powell 2007). Its importance lies in the fact that the costs accrued over a sample path can be compared with the costs predicted by the latest estimate of the future cost-to-go, $\tilde{Q}_t^{n-1}(\mathbf{s}_t^n, \mathbf{a}_t^n)$, thus allowing for the weights associated with the Q-factors to be updated.

In this paper, a *temporal difference* (TD) learning algorithm is employed for updating the value function approximation. TD learning is a two-stage procedure which relies on a forward pass to generate a sample path, and a backward pass to update the parameters. More importantly, it generalizes the different types of estimation errors which can be obtained from a given sample path.

For instance, a one-period look-ahead temporal difference error compares the Q-function estimates of the states visited in two successive years of a sample path, as shown below:

$$\delta_t^n = c_(\mathbf{s}_{t+1}^n, \mathbf{a}_{t+1}^n) + \alpha \tilde{Q}_{t+1}^{n-1}(\mathbf{s}_{t+1}^n, \mathbf{a}_{t+1}^n) - \tilde{Q}_t^{n-1}(\mathbf{s}_t^n, \mathbf{a}_t^n), \tag{14}$$

The one-period look-ahead update can also be classified as a *bootstrapping* approach, wherein a value estimate is updated based on an existing value estimate (Sutton and Barto 1998). Dynamic programming is another example of bootstrapping, since the cost-to-go in year t is estimated using the cost-to-go estimate corresponding to year $t+1$. However, a limitation of the one-period look-ahead update is that, unlike dynamic programming, if the initial estimates of the value function approximation in year $t+1$ are poor, the update may lead to a bias in the future cost estimation.

An alternate approach for updating the Q-factors is to compare $\tilde{Q}_t^n(\mathbf{s}_t^n, \mathbf{a}_t^n)$ with the costs realized through the sample path, which can be represented as follows:

$$\tilde{z}_t^n = c(\mathbf{s}_{t+1}^n, \mathbf{a}_{t+1}^n) + \alpha c(\mathbf{s}_{t+2}^n, \mathbf{a}_{t+2}^n) + \ldots + \alpha^{T-t+1} c(\mathbf{s}_{T+1}^n, \mathbf{a}_{T+1}^n). \tag{15}$$

Unlike bootstrapping, $\tilde{z}_t^n$ provides a cost-to-go estimate that is purely based on the Monte Carlo simulation of the sample path. Consequently, $\tilde{z}_t^n$ as an estimate is representative of the costs associated with the current state of the value function approximations. However, a cost-to-go estimate based on the entire sample path does not account for the possibility that the difference between the predicted and realized cost-to-go could have been generated by the sub-optimal nature of the policies corresponding to the future years. In such a case, the Q-factor update of the current policy, $\mathbf{a}_t^n$, may suffer from an overfitting of data.

The TD learning algorithm generalizes the bootstrapping and Monte Carlo-based learning approaches shown in equations 14 and 15, as follows:

$$\Delta_t^n = \sum_{\tau=t}^{T} (\alpha\lambda)^{\tau-t} \Big(c(\mathbf{s}_{\tau+1}^n, \mathbf{a}_{\tau+1}^n) + \alpha \tilde{Q}_{\tau+1}^{n-1}(\mathbf{s}_{\tau+1}^n, \mathbf{a}_{\tau+1}^n) - \tilde{Q}_{\tau}^{n-1}(\mathbf{s}_{\tau}^n, \mathbf{a}_{\tau}^n) \Big), \tag{16}$$

$$= \sum_{\tau=t}^{T} (\alpha\lambda)^{\tau-t} \left(\delta_{\tau}^n\right), \tag{17}$$

where,
Δ_t^n: temporal difference error associated with the current estimate of the future cost-to-go, $\tilde{Q}_t^{n-1}\left(\mathbf{s}_t^n, \mathbf{a}_t^n\right)$,
$\lambda \in [0,1]$: heuristic discount amount factor which determines the contribution of the errors associated with policies implemented farther down the sample path.

The learning framework presented in equations 16 and 17 is referred to as TD(λ) learning, wherein the parameter, λ, generalizes the learning process. It can be shown that TD(0) is equivalent to equation 14, whereas TD(1) reduces the problem to estimating the temporal difference error using the costs accrued over the sample path (equation 15). More generally, the artificial discounting introduced by the parameter, λ, allows the algorithm to look into the future, while accounting for the possibility that the policies implemented farther along the sample path may be sub-optimal.

Finally, once the temporal difference error is calculated for a given year t, the weights associated with the Q-factors can be updated using a stochastic gradient algorithm:

$$\theta_{tb}^{n} \longleftarrow \theta_{tb}^{n-1} + \gamma_n \ \nabla_{\theta_{tb}}(\hat{v}_t^n) \ \Delta_t. \tag{18}$$

where,

$$\hat{v}_t^n = c(\mathbf{s}_t^n, \mathbf{a}_t^n) + \alpha \tilde{Q}_t^{n-1}(\mathbf{s}_t^n, \mathbf{a}_t^n), \tag{19}$$

and γ_n represents the stepsize of the stochastic gradient algorithm. The stepsize determines the magnitude of the update made along the direction of the error-minimizing gradient.

There exist various stepsize rules in the stochastic gradient algorithm literature which can be used in order to achieve convergence of the parameters being updated. However, the three basic conditions are as follows:

$$\gamma_n \geq 0, \qquad n = 1, 2, \ldots, \tag{20}$$

$$\sum_{n=1}^{\infty} \gamma_n = \infty, \tag{21}$$

$$\sum_{n=1}^{infty} \gamma_n^2 < \infty. \tag{22}$$

In this paper, a *search-then-converge* stepsize rule is adopted which produces delayed learning. The stepsize formula is represented as follows:

$$\gamma_n = \gamma_0 \frac{\left(\frac{b}{n} + a\right)}{\left(\frac{b}{n} + a + n^{\beta}\right)}, \tag{23}$$

where, γ_0, a, b, and β are parameters to be determined.

This class of stepsize rules is termed "search-then-converge" because they provide for a period of high stepsizes (while the search for optimal policies is taking place) after which the stepsize declines (to achieve convergence). The degree of delayed learning is controlled by the parameters, b and a (Powell 2007).

In conclusion, the proposed ADP framework for solving a system-level MR&R decision-making problem can be summarized using the following pseudo-code:

```
**Step 0.** Initialization:
    **Step 0a.** Initialize $\tilde{Q}_t^0(\mathbf{s}_t, \mathbf{a}_t), \forall \mathbf{s}_t \in \mathbb{S}^N, \forall \mathbf{a}_t \in \mathbb{A}^N, \forall t = 1, \ldots, T+1$.
    **Step 0b.** Choose an initial state $\mathbf{s}_1^0$.
**Step 1.** Do for $n = 1, 2, \ldots, \mathcal{N}$:
    **Step 2.** Do for $t = 1, 2, \ldots, T$:
        **Step 2a.** Solve:
            $\hat{v}_t^n = \min_{\mathbf{a}_t \in \mathbb{A}_c^N} c(\mathbf{s}_t^n, \mathbf{a}_t) + \alpha \tilde{Q}_t^{n-1}(\mathbf{s}_t^n, \mathbf{a}_t)$,
            and let $\mathbf{a}_t^n$ be the optimal solution correponding to $\hat{v}_t^n$.
        **Step 2b.** Compute $\mathbf{s}_{t+1}^n \longleftarrow \Psi(\mathbf{s}_t^n, \mathbf{a}_t^n, \omega^n)$
```

```
Step 3. Initialize $\Delta_T \longleftarrow 0$, $\hat{v}^n_{T+1} \longleftarrow \bar{V}(\mathbf{s}^n_{T+1})$, wherein, $\bar{V}(\mathbf{s}^n_{T+1})$ represents the
        salvage value associated with the end of the planning horizon.
Step 4. For $t = T, T-1, \ldots, 1$, do:
    Step 4a. $\delta_t \longleftarrow \hat{v}^n_{t+1} - \tilde{Q}^{n-1}_t(\mathbf{s}^n_t, \mathbf{a}^n_t)$.
    Step 4b. $\Delta_t \longleftarrow \Delta_t + \delta_t$.
    Step 4c. $\theta^n_{tb} \longleftarrow \theta^{n-1}_{tb} + \gamma_n \; \nabla_{\theta_{tb}}(\hat{v}^n_t) \; \Delta_t$.
    Step 4d. $\Delta_{t-1} \longleftarrow \alpha\lambda\Delta_t$.
```

Herein, $\mathcal{N}$ represents the duration of the training period in which the Q-factor weights are updated. Once the Q-factors have converged to their final values, the facility-specific MR&R policies can be obtained using equation 11.

6.4 Exploration vs. Exploitation

A fundamental challenge within approximate dynamic programming is that, in order to estimate the cost-to-go for a given state-action pair, the state-action combination must be visited as part of the sample path. As a result, a trade-off must be made between visiting a state which leads to the lowest costs (”exploitation”) versus visiting a state in order to obtain information about the value of being in that state (”exploration”) (Powell 2007). Prior research indicates that allowing for intermittent exploration during the training period provides better results than a pure exploitation strategy. In this paper, a ϵ-greedy policy is employed, wherein a sub-optimal policy is randomly selected with a probability, ϵ.

6.5 Comparing SNO with ADP

The advantage of ADP is that the optimal actions selected at every time step of the sample path always satisfy the underlying constraints of the decision-making process. Consequently, the value function approximation updated using the sample path is also consistent with the constraints of the system-level MDP problem. However, since it is a simulation-based approach, ADP does not provide any optimality guarantees. Its performance depends on several factors: choice of basis functions, learning strategy, convergence rate, among others.

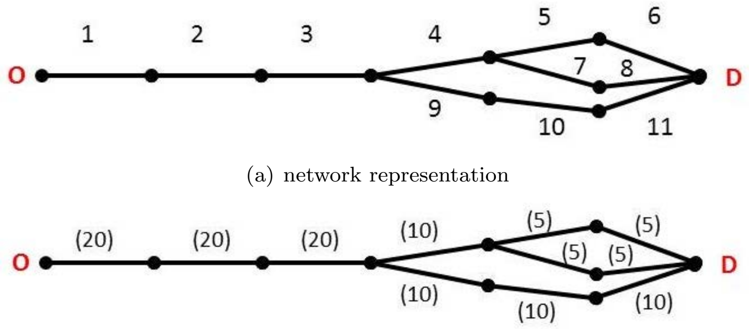
In comparison, SNO is an optimal approach for budget allocation problems. Its limitation arises from the inability to account for network considerations in the future time periods. As a result, the policies recommended by SNO, while feasible, are also sub-optimal.

7 Parametric Case Study

In this section, the network considerations are formally introduced into the MR&R decision-making framework as mathematical constraints. The structural interdependence between the individual facilities is incorporated by imposing a network capacity constraint on the MR&R activity selection. In particular, an agency-chosen network capacity threshold limits the loss in network capacity due to the implementation of the chosen MR&R policies by an upper bound. The framework also allows for different work zone options to be considered (partial vs. complete closure), wherein time, money and loss in link capacity can be traded-off.

The use of capacity as a performance measure can be interpreted as a supply-based criterion, wherein the agency seeks to provide enough capacity during MR&R activity implementation so that the associated origin-destination demand can be met. While it is recognized that a capacity-based approach does not take into account the demand on individual links, it can be argued that the traffic can be re-routed to maximize capacity utilization, using adequate signages and real-time information dissemination systems.

7.1 Network Representation



(a) network representation

(b) individual link capacities; $C_{max} = 20$ units

Figure 2: Stylized 11-link pavement network for incorporating structural interdependence (individual link capacities expressed in brackets

Let us consider a stylized network configuration consisting of 11 road segments and 10 nodes connecting an origin and a destination, as shown in figures 2(a) and 2(b). Figure 2(b) represents the individual link capacities, using which the resulting network capacity, C_{max}, can be obtained as 20 units. It is to be noted that C_{max} represents the maximum capacity available to the users in the absence of MR&R activities.

In terms of notation, the pavement network is defined as a graph, $G = (N, E)$, wherein the edges, E, represent the road segments, and the nodes, N, representing the points of intersection between any two facilities. An individual road segment is defined as a link, (q, r).

7.2 Value Function Approximation: ADP

In order to implement an ADP methodology, the foremost step is to model the value function approximation for the network under investigation. In this regard, the choice of basis functions plays an important role in determining the quality of the solutions obtained using ADP. The objective is to expoit the properties of the problem such that the learning process is reduced to estimating a few key parameters of interest.

Based on the network configuration depicted in figures 2(a) and 2(b), it is possible to identify groups of road segments which respond similarly to the network capacity constraints. For instance, it can be argued that the groups defined as, $\{1, 2, 3\}$, $\{4\}$, $\{5, 6, 7, 8\}$ and $\{9, 10, 11\}$ have a near-identical policy response towards the network-based constraints. In the case of links, 1-3 and 9-11, since all the road segments are in series, a MR&R activity/construction type chosen for any segment within the group leads to the same loss in capacity of the network. For sections 5-8, the symmetry in the network configuration is also taken into consideration while aggregating the corresponding state-action space. The road segment, $\{4\}$, is considered as a singleton for the purpose of approximating the value function.

Finally, given the Markovian evolution of the system, it can also be assumed that for any given year, t, the optimal policy for a facility, (q, r) is only a function of its current state, and the action-construction type pair under consideration. Consequently, the total number of basis functions chosen for the value function approximation, $|\mathbb{B}|$, can be calculated as follows:

$$|\mathbb{B}| = T|G_N||S||A||L|, \tag{24}$$

where,

T: length of the planning horizon,
$|G_N|$: number of groups having a homogeneous response to the capacity constraints,
$|L|$: number of construction options available (partial and complete closures).

The savings from using a value function approximation, in terms of memory requirement, can be gauged from the fact that for a look-up table representation (equation 1), the state space comprises of $T|S|^N|A|^N|L|^N$ combinations.

Using this Q-function representation, the future cost-to-go associated with implementing the action-construction type combinations, $(\mathbf{a}_t, \mathbf{l}_t)$, when the network is in state, $\mathbf{s}_t$, can be obtained as follows:

$$\tilde{Q}_t(\mathbf{s}_t, \mathbf{a}_t, \mathbf{l}_t) = \sum_{(q,r)\in E} \theta_{t,G_N(q,r),\mathbf{s}_t(q,r),\mathbf{a}_t(q,r)\mathbf{l}_t(q,r)}, \tag{25}$$

where, $G_N(q,r)$ refers to the group number associated with facilitiy, (q,r).

It is important to acknowledge that approximating the value function is not a precise science. The approach involves having some a-priori expectations about the learning process, followed by modeling those beliefs using the approximation function, and evaluating its performance using simulations. Based on the quality of the simulation results, further modifications can be made to the chosen basis functions, before arriving at a suitable value function approximation.

7.3 Problem Formulation: ADP

The objective of the MR&R optimization is to provide optimal policies for each facility, (q,r), in the network, while satisfying the budget restrictions and the network capacity constraints:

$$\begin{aligned} \min_{x,\tilde{x},\tilde{z},y} \sum_{(q,r)\in E} \sum_{a\in\mathbb{A}} \sum_{l\in\mathbb{L}} \big(c(\mathbf{s}_t(q,r),a,l) + u(\mathbf{s}_t(q,r)) \\ + \alpha\ \theta_{t,G_N(q,r),\mathbf{s}_t(q,r),a,l}\big) x_{(q,r),a,l,t}, \end{aligned} \tag{26}$$

subject to

$$\sum_{(q,r)\in E} \sum_{a\in\mathbb{A}} \sum_{l\in\mathbb{L}} c(\mathbf{s}_t(q,r),a,l) x_{(q,r),a,l,t} \leq B \tag{27}$$

$$\sum_{a\in\mathbb{A}} \sum_{l\in\mathbb{L}} x_{(q,r),a,l,t} = 1 \qquad \forall\,(q,r) \in E, \tag{28}$$

$$\sum_{\Delta=1}^{52} \tilde{x}_{(q,r),a,l,\Delta} \geq x_{(q,r),a,l,t} \qquad \forall a \in \mathbb{A}, \forall l \in \mathbb{L}, \quad \forall\,(q,r) \in E, \tag{29}$$

$$\sum_{\Delta - d_{a,l} < \Delta' \leq \Delta} \tilde{x}_{(q,r),a,l,\Delta'} \leq \tilde{z}_{(q,r),a,l,\Delta} \qquad \forall a \in \mathbb{A}, \forall l \in \mathbb{L}, \quad \forall \Delta = 1,\dots,52, \quad \forall\,(q,r) \in E, \tag{30}$$

$$y_{(D,O),\Delta} \geq hC_{max} \qquad \forall \Delta = 1,\dots,52, \tag{31}$$

$$C_{(q,r)}\left[1-\left(\sum_{a\in\mathbb{A}}\sum_{l\in\mathbb{L}}\kappa_{(q,r),a,l}\tilde{z}_{(q,r),a,l,\Delta}\right)\right]\geq y_{(q,r),\Delta} \qquad \forall\,(q,r)\in E,\ \forall\Delta=1,\ldots,52, \quad (32)$$

$$\sum_{r:(r,q)\in E} y_{(r,q),\Delta}-\sum_{r:(q,r)\in E} y_{(q,r),\Delta}=0 \qquad \forall(q,r)\in E,\ \forall\Delta=1,\ldots,52,, \quad (33)$$

$$x_{(q,r),a,l,t},\tilde{x}_{(q,r),a,l,\Delta},\tilde{z}_{(q,r),a,l,\Delta}\in\{0,1\},y_{(q,r),\Delta}\in\mathbb{R}^{+} \qquad \forall a\in\mathbb{A},\forall l\in\mathbb{L},\ \forall\Delta=1,\ldots,52,\ \forall\,(q,r)\in E,$$

where,

$x_{(q,r),a,l,t}$: 1 if the MR&R action-construction type pair (a,l), is selected for facility (q,r) in year t; 0 otherwise,

$\tilde{x}_{(q,r),a,l,\Delta}$: 1 if the action-construction type pair, (a,l), is started in week Δ for facility (q,r); 0 otherwise,

$\tilde{z}_{(q,r),a,l,\Delta}$: 1 if the action-construction type pair, (a,l), is under implementation in week Δ for facility (q,r); 0 otherwise,

$y_{(D,O),\Delta}$: available network capacity in the presence of MR&R activities in week Δ, represented as a virtual edge connecting destination D and origin O,

$y_{(q,r),\Delta}$: available link capacity for facility (q,r) in week Δ,

$c(\mathbf{s}_t(q,r),a,l)$: agency cost of undertaking the action-construction type pair, (a,l),

$u(\mathbf{s}_t(q,r))$: user costs linked with increase in vehicle wear-and-tear, fuel usage, etc.,

B: annual budget,

$d_{(q,r),a,l}$: duration of implementing the action-construction type pair, (a,l),

h: fraction of the maximum capacity of the network, C_{max}, representing a minimum network capacity threshold,

$C_{(q,r)}$: maximum link capacity associated with facility (q,r), as defined when no MR&R activity is scheduled, and,

$\kappa_{(q,r),a,l}$: loss in capacity associated with implementing the action-construction type pair, (a,l), for facility (q,r).

Herein, equation 26 represents the objective function, defined as minimizing the expected system-level cost-to-go, as per the estimates of the Q-function; equation 27 indicates that the total amount spent on MR&R activities should be within the annual budget; equation 28 assigns exactly one action-construction type pair (including do-nothing) to each facility. The scheduling constraints, as represented by equations 29 - 30, are modeled as a non-preemptive scheduling problem Sousa and Wolsey (1992), wherein the assigned activity is completed in one sequence; equation 29 ensures that only the chosen action-construction type pair is considered for assessing the feasibility of the MR&R scheduling, and equation 30 ensures that the chosen action-construction type pair undergoes a continuous construction period of $d_{(q,r),a,l}$ weeks. Equation 31 guarantees that the network capacity in the presence of MR&R activities does not violate the minimum network capacity threshold; equation 32 represents the loss in link capacity associated with implementing an action-construction type pair on facility(q,r); and equation 33 represents the flow conservation equation for every node in the network.

It is important to note that the optimization routine only imposes the constraints on the policies associated with the current year of decision-making. However, since the sample path is generated by solving the optimization problem in each time period, the underlying budget and network considerations are always satisfied along the sample path. Consequently, the resulting Q-function updates, which are also based on the sample path traversed, are consistent with the constraints of the problem.

7.4 Problem Formulation: SNO

In section 4, the SNO framework was described in the context of budget allocation problems. In order to add the network-based constraints into the formulation, the budget allocation problem can be reformulated similar to the ADP formulation using a facility-specific decision variable identical to the ADP formulation, $x_{(q,r),a,l,t}$. Hence, the network constraints (equations 28 - 33) can be modeled in an identical fashion, and the policies obtained using SNO satisfy the network constraints for the current year of the decision-making, t.

However, the disadvantage of SNO is that its future cost estimation relies on the use of randomized policies which cannot accommodate the network constraints. Consequently, while SNO still provides feasible policies, the optimality of the solutions is no longer guaranteed.

7.5 Scenario Generation: Supplementary Information

The condition state of the facilities is evaluated using an eight point ordinal index, where 1 is the best state and 8 is deemed to be an unacceptable state by the agency. For the purpose of illustration, four types of activities are considered: do-nothing, routine maintenance, rehabilitation and reconstruction. It is assumed that maintenance activities can be implemented overnight and hence lead to no loss in capacity of a road segment. There exist two reconstruction options: a partial road closure which requires 10 weeks of construction time and causes 30% loss in capacity, while a complete road closure can be completed in 2 weeks, but leads to a 100% loss in capacity, i.e., the link is rendered inaccessible. These values are not based on empirical data, but are representative of the kind of trade-offs which can be expected in real-life scenarios.

Table 1: Cost structure for numerical example

Maintenance Activity	Pavement State							
	1	2	3	4	5	6	7	8
	Acceptable							Unacceptable
Do-Nothing	0	0	0	0	0	0	0	0
Maintenance	0.1	0.2	0 .3	0.6	0.8	1.4	2	6.9
Rehabilitation	3.8	3.9	4.1	6.6	9.1	10.7	12.3	21.8
Replacement	26	26	26	26	26	26	26	26
User Costs	0	2	4	8	14	22	25	100

The agency and user cost structure, shown in table 1, is taken from Madanat (1993). Herein, maintenance and rehabilitation activities are prohibitively more expensive as the state worsens, whereas reconstruction incurs a fixed cost. The user cost also increases as the facility deteriorates, and a high penalty cost is imposed when the facility is in the non-permissible condition state ($s = 8$). The transition probability matrices for the different MR&R alternatives are also adapted from Madanat (1993), but suitably modified to reflect the increasing levels of maintenance effectiveness. The planning horizon consists of 15 years and the discount rate is chosen to be 5%. The salvage value at the end of the planning horizon is set equal to the user costs, wherein the user costs can be interpreted as a proxy for the quality of the terminating state of the facility.

The parameters corresponding to ADP's stepsize rule are $a = 300; b = 0.5; \beta = 0.2; \gamma_0 = 0.7;$. The artificial discounting factor, λ, is taken to be 0.4, and the Q-function is trained for 1500 iterations. The algorithms are programmed in C++, and the optimization problems are solved using CPLEX® on a Windows-based OS with a 3.10 GHz processor and 4GB RAM.

In order to compare ADP and SNO, three budget levels are considered, $B = 50; 100; 150$ units. The network capacity threshold, h, is chosen to be 0.75, such that the network constraints are active for the purpose of the investigation. The initial condition of the facilities is uniformly distributed between the states, 1(good), 4(moderate), and 7(poor), within each group, g, so as to capture a range of deterioration levels. The performance of the two methodologies is compared on the basis of implementing each scenario 1000 times using Monte Carlo simulations.

7.6 Results

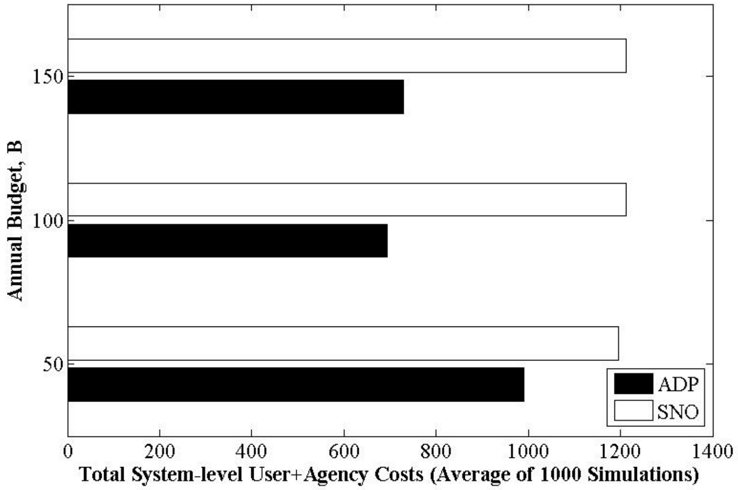
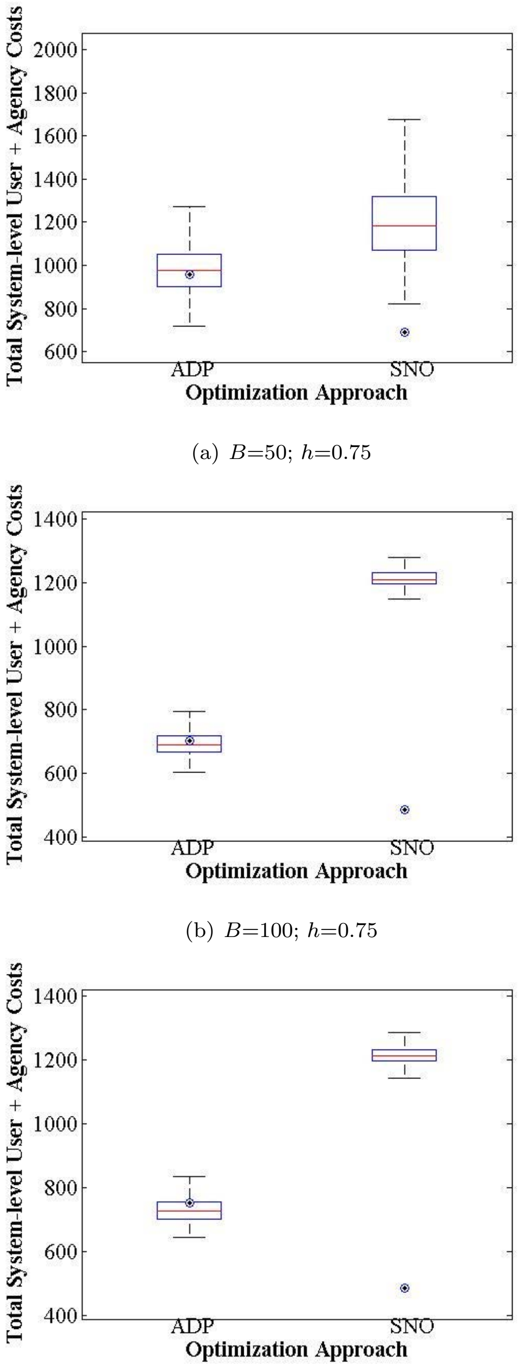


Figure 3: Total costs comparison between SNO and ADP for different budget levels (h=0.75)

Figure 3 represents the simulation-average of the system-level user+ agency costs incurred from using SNO and ADP. The results indicate that, in the presence of network constraints, ADP performs better than SNO (on an average) across all budget levels.

An alternate way of representing the simulation results is to observe the distribution of the realized costs in conjunction with the costs predicted at $t = 1$. A box plot representation, as shown in figure 4, provides information about the different quartiles of the realized cost distribution. The horizontal bars in each box plot represent the sample minimum, lower quartile, median, upper quartile and sample maximum. In addition, a dot is marked on the plot to indicate the expected cost predicted by the optimization approach at t=1. Using this representation, figures 4(a) to 4(c) indicate that ADP's predicted costs align well with the median of the costs realized by the simulation. On the other hand, SNO provides an inaccurate prediction of future costs as the budget increases.

The primary reason for the disparity in SNO's simulated and predicted costs is that the randomized policies corresponding to the future years do not account for the network constraints. In the context of the numerical example, reconstruction activities are required to be excluded for certain road segments due to capacity considerations. However, since the network constraints can only be imposed on the facility-specific MR&R policy selection, SNO defers the activities to the future years, in the form of randomized policies. As the sample path is simulated, the policies predicted for the future years are not realized, and the gap between the predicted and the real-



(a) B=50; h=0.75

(b) B=100; h=0.75

(c) B=150; h=0.75

Figure 4: Box plots comparing ADP and SNO for varying budget levels and $h = 0.75$

ized costs widens. With an increase in the available budget, a greater share of the randomized policies are allocated for reconstruction activities, thus leading to more inaccuracies in the estimation of the future costs. In comparison, ADP updates its value function approximation based on states/actions that are in agreement with the network and budget constraints. Consequently, the costs predicted by ADP at $t = 1$ are consistent with the realized cost distribution.

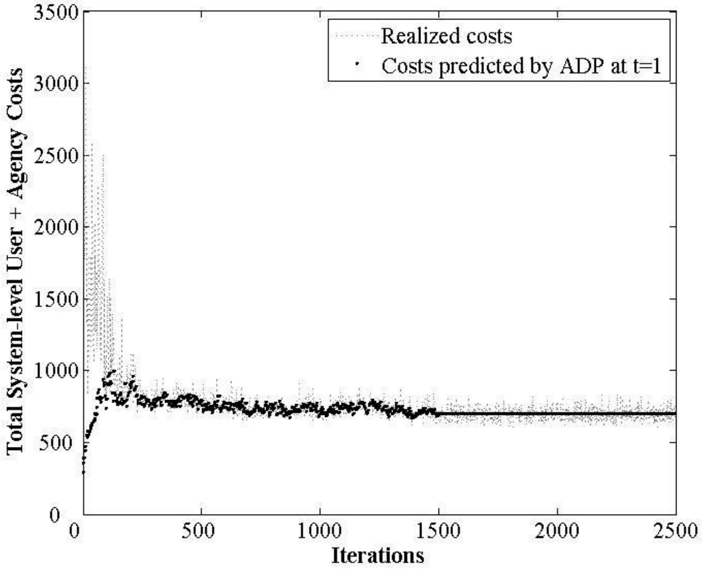


Figure 5: Convergence of the value function approximation for $B = 100$ scenario

The proximity between the predicted and the realized costs is also an indication that the TD(λ) learning algorithm is suitable for updating the Q-function. Figure 5 shows the convergence of the Q-function for $B = 100$ scenario. As can be observed, the realized and the predicted costs converge to their final values after 500-600 iterations.

8 Conclusions and Future Work

In this paper, the state-of-the-art MDP-based approaches were extended to incorporate network-based considerations into MR&R decision-making. The structural interdependence problem was motivated to account for the adverse impact of construction activities on the road network. The results indicate that randomized policies, while ideal for modeling budget allocation problems, do not adequately capture these network-based constraints. In comparison, ADP is able to provide better results by learning the value function parameters using simulation techniques.

In the context of infrastructure management, the use of approximate dynamic programming is a recent phenomenon. While previous research established that MR&R decision-making problems can be modeled using ADP, its performance had not been benchmarked. For a pure budget allocation problem, there exist alternate approaches in the literature, such as SNO, which guarantee optimal solutions. Through this research, it is shown that ADP can be beneficial for modeling problems with complex inter-facility dynamics, such as network flow and scheduling constraints. Future work in this context includes scaling up the network size so as to better assess the computational performance of ADP. Other problems relevant to the applicability of ADP in infrastructure management include capturing history dependence in the deterioration process, as observed in the case of corrosion in bridges.

References

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of American Mathematical Society 60*(6), 503–515.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Athena Scientific.

Chang, Y. Y., O. B. Sawaya, and A. K. Ziliaskopoulos (2001). A tabu search based approach for work zone scheduling. In *Proceeding of the Transportation Research Board 80th Annual Meeting, Washington, DC*.

Dekker, R., R. E. Wildeman, and F. A. van der Duyn Schouten (1997). A review of multicomponent maintenance models with economic dependence. *Mathematical Methods of Operations Research 45*(3), 411–435.

Durango-Cohen, P. L. (2004). Maintenance and repair decision making for infrastructure facilities without a deterioration model. *Journal of Infrastructure Systems 10*(1), 1–8.

Durango-Cohen, P. L. and P. Sarutipand (2007). Capturing interdependencies and heterogeneity in the management of multifacility transportation infrastructure systems. *Journal of Infrastructure Systems 13*(2), 115–123.

Fwa, T. F., R. L. Cheu, and A. Muntasir (1998). Scheduling of pavement maintenance to minimize traffic delays. *Transportation Research Record: Journal of the Transportation Research Board 1650*(1), 28–35.

Gao, L. and Z. Zhang (2009). Approximate dynamic programming approach to network-level budget planning and allocation for pavement infrastructure. In *Transportation Research Board 88th Annual Meeting*. Number 09-2344.

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

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

Hajdin, R. and H. P. Lindenmann (2007). Algorithm for the planning of optimum highway work zones. *Journal of Infrastructure Systems 13*(3), 202–214.

Kallenberg, L. C. M. (1994). Survey of linear programming for standard and nonstandard Markovian control problems. Part I: Theory. *Mathematical Methods of Operations Research 40*(1), 1–42.

Kuhn, K. and S. Madanat (2005). Model uncertainty and the management of a system of infrastructure facilities. *Transportation Research Part C: Emerging Technologies 13*(5-6), 391–404.

Kuhn, K. D. (2010). Network-level infrastructure management using approximate dynamic programming. *Journal of Infrastructure Systems 16*(2), 103–111.

Lee, E. B. and C. W. Ibbs (2005). Computer simulation model: Construction analysis for pavement rehabilitation strategies. *Journal of Construction Engineering and Management 131*(4), 449–458.

Madanat, S. (1993). Optimal infrastructure management decisions under uncertainty. *Transportation Research Part C: Emerging Technologies 1*(1), 77–88.

Madanat, S., S. Park, and K. Kuhn (2006). Adaptive optimization and systematic probing of infrastructure system maintenance policies under model uncertainty. *Journal of Infrastructure Systems 12*(3), 192–198.

Medury, A. and S. Madanat (2012). A simultaneous network optimization approach for pavement management systems. *Journal of Infrastructure Systems (under review).*

Ng, M., D. Y. Lin, and S. T. Waller (2009). Optimal long-term infrastructure maintenance planning accounting for traffic dynamics. *Journal of Computer-Aided Civil and Infrastructure Engineering 24*(7), 459–469.

Ohlmann, J. W. and J. C. Bean (2009). Resource-constrained management of heterogeneous assets with stochastic deterioration. *European Journal of Operational Research 199*(1), 198–208.

Ouyang, Y. (2007). Pavement resurfacing planning for highway networks: parametric policy iteration approach. *Journal of Infrastructure Systems 13*(1), 65–71.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality.* John Wiley & Sons, Inc.

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

Sinha, K., M. Saito, Y. Jiang, S. Murthy, A. Tee, and M. Bowman (1988). The Development of Optimal Strategies for Maintenance, Rehabilitation and Replacement of Highway Bridges, Vol. 1: The Elements of the Indiana Bridge Management System (IBMS). Technical Report FHWA/IN/JHRP-88/15-1. Joint Highway Research Project, Indiana Department of Transportation and Purdue University, West Lafayette, Indiana.

Smilowitz, K. and S. Madanat (2000). Optimal inspection and maintenance policies for infrastructure networks. *Journal of Computer-Aided Civil and Infrastructure Engineering 15*(1), 5–13.

Sousa, J. and L. Wolsey (1992). A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming 54*(1), 353–367.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: An introduction.* Cambridge University Press.

Tesauro, G. (1995). Temporal difference learning and TD-gammon. *Communications of the ACM 38*(3), 58–68.

Wunderlich, K. and D. Hardesty (2003). A snapshot of summer 2001 work zone activity. Technical Report EDL: 13793, National Transportation Library, FHWA, U.S. Department of Transportation.

Yeo, H., Y. Yoon, and S. Madanat (2012). Algorithms for bottom-up maintenance optimisation for heterogeneous infrastructure systems. *Structure and Infrastructure Engineering: Maintenance, Management, Life-Cycle Design and Performance*. , DOI: 10.1080/15732479.2012.657649.