# InfraLib: enabling reinforcement learning and decision making for large scale infrastructure management

**Pranay Thangeda & Melkior Ornik**
Department of Aerospace Engineering
University of Illinois Urbana-Champaign
{pranayt2,mornik}@illinois.edu

**Trevor S. Betz & Michael N. Grussing**
Engineer Research and Development Center
US Army Corps of Engineers
{trevor.s.betz,michael.n.grussing}@erdc.dren.mil

## Abstract

Efficient management of infrastructure systems is crucial for economic stability, sustainability, and public safety. However, infrastructure management is challenging due to the vast scale of systems, stochastic deterioration of components, partial observability, and resource constraints. While data-driven approaches like reinforcement learning (RL) offer a promising avenue for optimizing management policies, their application to infrastructure has been limited by the lack of suitable simulation environments. We introduce InfraLib, a comprehensive framework for modeling and analyzing infrastructure management problems. InfraLib employs a hierarchical, stochastic approach to realistically model infrastructure systems and their deterioration. It supports key functionality such as modeling component unavailability, cyclical budgets, and catastrophic failures. To facilitate research, InfraLib provides tools for expert data collection, simulation-driven analysis, and visualization. We demonstrate InfraLib's capabilities through case studies on a real-world road network and a synthetic benchmark with 100,000 components. By providing a scalable, unified framework for simulating infrastructure systems, InfraLib aims to accelerate progress in data-driven infrastructure management.

## 1 Introduction

Infrastructure systems are the backbone of modern society, encompassing a wide array of essential services including transportation networks, utility systems, and public facilities. Efficient infrastructure management is crucial for modern society's functioning, influencing economic stability (1), (2), environmental sustainability (3), and public safety (4). Managing modern infrastructure systems is a complex and multifaceted task, involving the maintenance, repair, and replacement of numerous components distributed across facilities and networks (5), (6). The challenges of infrastructure management are further compounded by their vast scale, the stochastic nature of component deterioration (7), (8), stringent operational constraints (9), limited resources (6), and extreme weather events due to climate change (10), (11), (12).

Traditional approaches to infrastructure management typically involve rule-based methodologies that rely on deterministic models. These methods, while useful in controlled environments, often struggle to capture the inherent uncertainties and dynamic variations present in real-world scenarios (1). The complexity is further compounded by the need for strategic allocation of resources and budgetary considerations, which are critical yet challenging aspects of effective infrastructure management (13), (14).

In recent years, there has been a significant shift towards data-driven methodologies, particularly with the advent of machine learning techniques like reinforcement learning (RL) and imitation

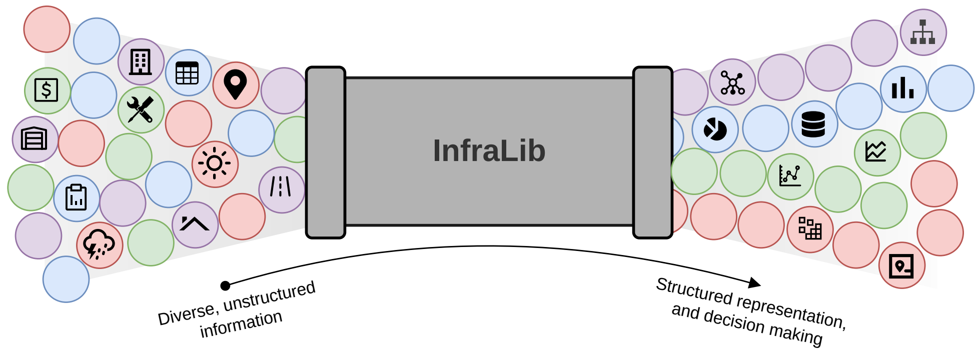



Figure 1: Illustration of the proposed InfraLib framework for infrastructure management.

learning (IL) (15), (16). These approaches offer a promising avenue for decision-making under uncertainty, allowing for adaptive and proactive infrastructure management strategies (17), (18). RL, in particular, has shown remarkable success in various domains, thanks to its ability to learn optimal policies through interaction with an environment (19) ,(20). However, the application of these techniques in infrastructure management is still in its infancy, primarily due to the lack of suitable simulation environments that can accurately model the complexity and scale of these systems (21), (18). There is a strong need for a unified framework modeling infrastructure management problems. Such a framework should provide a natural and intuitive way to represent the uncertainties and limitations in observability while ensuring scalability to handle large-scale, real-world problems. This tool would enable rapid progress in the field, providing a simulation environment for training and realistic benchmarks for comparing learning-based approaches, traditional optimization-based approaches, and rule-based methods.

To address these challenges, we introduce InfraLib, a comprehensive and versatile simulation framework designed for modeling and analyzing large-scale infrastructure management problems. InfraLib provides a realistic and granular representation of infrastructure systems by integrating a hierarchical model that captures the intricate relationships between different components and facilities (5). Moreover, InfraLib employs a stochastic approach to mimic the real-world uncertainties and partial observability inherent in infrastructure systems (7; 8), enabling the development and validation of infrastructure management strategies that are robust to the challenges faced in real-world scenarios.

Beyond serving as a simulation tool, InfraLib includes features for analysis, expert data collection, and modeling real-world budget schedules and failure modes. These features make it a valuable resource for both researchers looking to develop and test new management strategies and practitioners aiming to optimize operational efficiencies in real-world settings.

In this paper, we present a detailed overview of the architecture and capabilities of InfraLib, highlighting its capabilities and potential applications. We demonstrate the ability of InfraLib to create realistic scenarios for the deployment and evaluation of learning-based approaches, showcasing its ability to model the complexities and challenges encountered in real-world infrastructure systems. Furthermore, we provide a series of benchmarks and environments to illustrate the utility and scalability of InfraLib in facilitating the development and comparison of novel management strategies.

The rest of this paper is organized as follows. Section 2 introduces key concepts and background information on infrastructure management, Partially Observable Markov Decision Processes, and data-driven approaches for decision making. Section 3 formalizes the infrastructure management problem as a POMDP and discusses the various research challenges that arise in this context. We then present InfraLib in Section 4 and Section 5, detailing its structure, component dynamics, and key functionalities. Section 6 delves into the human interface aspects of InfraLib, including tools for expert data collection and analysis. Finally, in Section 7, we showcase example environments and benchmarks to demonstrate InfraLib's utility, versatility, and scalability.

## 2 PRELIMINARIES AND BACKGROUND

In this section, we provide background on the infrastructure management domain, including the hierarchical nature of infrastructure systems, the metrics used to quantify component health, the dynamics of component deterioration, and the budget constraints that govern infrastructure management. We also introduce the concepts of Partially Observable Markov Decision Processes and data-driven approaches for decision-making. We start by defining the notation used in the paper.

Given a finite set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality and $\Delta(\mathcal{A})$ denotes the set of all probability distributions over the set $\mathcal{A}$. $\mathbb{N}_0$ denotes the set of natural numbers including 0 i.e. $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

### 2.1 INFRASTRUCTURE HIERARCHY AND MANAGEMENT

Infrastructure management is inherently hierarchical, comprising multiple layers of organization. At the base level, we have individual components which are the smallest units of individually managed infrastructure elements. These components are grouped into units, which are collections of components that are managed together. Multiple units collectively form a facility, which is the highest level of organization in the infrastructure hierarchy. This hierarchical structuring in inherent in real-world infrastructure systems and is crucial for systematic management and decision-making processes.

The condition of each component in the infrastructure is characterized by a *Condition Index (CI)* (22), a metric that reflects its health status. The CI of a component quantitatively represents the health of a component and it deteriorates over time due to environmental factors, wear-and-tear, and in some cases catastrophically due to a manufacturing defect or an external event. This deterioration is typically stochastic, arising from unpredictable environmental interactions and the complex nature of infrastructure materials. Moreover, the CI is not always directly observable, necessitating periodic inspections to estimate its current state. These inspections, while essential, incur additional costs. Even when the CI is observed by inspection, the observation is subjective depending on the inspector and the inspection method, and can be noisy.

Management of infrastructure systems at the component level include inspection, repair, and replacement. Inspection provides an estimate of the CI at a cost, replacement involves completely substituting the component at a higher expense, and repair, a more cost-effective option, aims to improve the CI. These actions are fundamental to maintaining the overall health of the infrastructure system.

### 2.2 PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

A discrete-time finite-horizon POMDP $M$ is specified by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, Z, R, H)$, where $\mathcal{S}$ denotes a finite set of states, $\mathcal{A}$ denotes a finite set of actions, and $\mathcal{O}$ denotes a finite set of observations. $T : \mathcal{S} \times A \rightarrow \Delta(\mathcal{S})$ denotes the transition probability function, where $\Delta(\mathcal{S})$ is the space of probability distributions over $\mathcal{S}$. Furthermore, and $Z : \mathcal{O} \times \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{O})$ denotes the observation probability function where $\Delta(\mathcal{O})$ is analogous to $\Delta(\mathcal{S})$. Finally, $R : \mathcal{S} \times \mathcal{A} \rightarrow [-R_{\text{min}}, R_{\text{max}}]$ denotes the reward function and $H \in \mathbb{N}_0$ denotes the finite planning horizon.

For the above POMDP, at each time step, the environment is in some state $s \in \mathcal{S}$ and the agent interacts with the environment by taking an action $a \in \mathcal{A}$. Doing so results in the environment transitioning to a new state $\bar{s} \in \mathcal{S}$ in the next time step with probability $T(s, a, \bar{s})$. Simultaneously, the agent receives an observation $o \in \mathcal{O}$ regarding the state of the environment with probability $Z(o|\bar{s}, a)$ which depends on the new state of the environment and the action taken by the agent. In a POMDP the agent doesn't have access to the true state of the environment. However, the agent can update it's belief about the true state of the environment using this observation. The agent also receives a reward $R(s, a)$.

The problem of optimal policy synthesis for a finite-horizon POMDP is that of choosing a sequence of actions which maximizes the expected total reward. $\mathbb{E}[\sum_{t=0}^{H} r_t]$ where $r_t$ is the reward earned at time instant $t$. Hence the optimal behavior may often include actions which are taken simply because they improve the agent's belief about the true state. After reaching the state $s'$, the agent receives observation $o \in \mathcal{O}$ with probably $Z(o|s', a)$. Let the belief $b$ be a probability distribution over $\mathcal{S}$. Then, $b(s)$ denotes the belief state and the agent updates the belief state according to Bayes' rule.

### 2.3 Data-Driven Approaches for Decision Making

Data-driven approaches, such as reinforcement learning (RL), inverse reinforcement learning (IRL) (23), and imitation learning (IL), have emerged as powerful tools for learning optimal decision-making policies in sequential decision-making problems. In RL, an agent learns to make decisions by interacting with an environment modeled as a (PO)MDP. The agent's goal is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which maps states to a probability distribution over actions, that maximizes the expected cumulative reward over a horizon $H$:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{H} R(s_t, a_t) \right]$$

where $R(s_t, a_t)$ is the reward obtained by taking action $a_t$ in state $s_t$ at time step $t$, and the expectation is taken over the trajectories generated by following policy $\pi$. RL algorithms can be broadly classified into value-based methods (24), which learn a value function that estimates the expected cumulative reward from each state or state-action pair, and policy-based methods (25), which directly learn a parametrized policy. RL's success in various domains can be attributed to its ability to adaptively learn optimal strategies through trial and error.

Inverse Reinforcement Learning (IRL) addresses the problem of learning a reward function that explains the behavior of an expert demonstrator. Given a set of expert demonstrations $\mathcal{D} = (s_0, a_0), (s_1, a_1), \ldots, (s_T, a_T)$, where $(s_t, a_t)$ represents the state-action pair at time step $t$, the goal of IRL is to find a reward function $R(s, a)$ that rationalizes the expert's behavior:

$$R^* = \arg\max_{R} \mathcal{L}(R \mid \mathcal{D})$$

where $\mathcal{L}(R \mid \mathcal{D})$ is a likelihood function that measures how well the reward function $R$ explains the expert demonstrations $\mathcal{D}$. Common approaches to IRL include maximum entropy IRL (26) and Bayesian IRL (27).

Imitation Learning (IL) focuses on learning a policy that mimics the behavior of an expert demonstrator. Given a set of expert demonstrations $\mathcal{D}$, the goal of IL is to learn a policy $\pi$ that generates behavior similar to the expert. IL can be approached through behavioral cloning (28), which treats IL as a supervised learning problem and learns a mapping from states to actions by minimizing a loss function between the predicted actions and the expert actions:

$$\pi^* = \arg\min_{\pi} \sum_{(s,a) \in \mathcal{D}} \ell(\pi(s), a)$$

where $\ell(\cdot, \cdot)$ is the chosen loss function. Alternatively, apprenticeship learning seeks to learn a policy that achieves a similar expected cumulative reward as the expert policy under some unknown reward function, often by iteratively solving an RL problem with a reward function learned via IRL based on the expert demonstrations.

The effectiveness of RL and IL is heavily contingent on the availability of accurate and comprehensive simulation environments and expert demonstrations, which presents unique challenges in the infrastructure domain. Benchmarks and baselines are essential in the realm of RL and IL as they provide standard metrics and methods for comparing different approaches. Benchmarks offer predefined problems with set parameters and goals, allowing for a consistent and fair evaluation of various strategies. Baselines, typically consisting of established methods or algorithms, serve as a reference point to gauge the performance of new approaches.

## 3 Problem Formulation

Infrastructure management is a complex problem influenced by environmental, manufacturing, and operational factors. In real-world infrastructure systems, control over the environment and manufacturing is limited. Therefore, the space of possible decisions spans over the operational factors. We focus on optimizing management decisions under budget constraints, while ensuring that our model captures the stochastic nature of component deterioration and the partial observability of infrastructure condition.

In this section, we formalize the infrastructure management problem as a Partially Observable Markov Decision Process (POMDP) closely following the multi-component POMDP with shared budget formulation in (21) and proceed to discuss the challenges and research questions arising when optimizing infrastructure management decisions.

### 3.1 Modeling Infrastructure Systems as POMDPs

We model a large-scale, hierarchical infrastructure system as a single collection of components. Let $n$ denote the total number of components in the infrastructure system. We model the condition index (CI) dynamics of each component as an independent POMDP with the assumption that the detoriation dynamics of individual components are not related. Specifically, let $M^i$ denote a POMDP representing the dynamics of component $i$ for $i \in \{1, 2, \ldots, n\}$. For each component, the state space $\mathcal{S} \subset \mathbb{N}_0$ is given by $\mathcal{S} = \{0, 1, 2, \ldots, s_{max}\}$, where $s_{max} \in \mathbb{N}_0$. The state at any time step $k$ denotes the CI of the component at that time step. The observation space is given by $\mathcal{O} = \mathcal{S} \cup \{e\}$, where $e \in \mathbb{N}_0$ is a null observation that does not provide any information regarding the true state of the system.

The action space for each component is given by $\mathcal{A} = \{d, q, r, m\}$ where (i) action $d$, the *do nothing action*, lets the component's CI transition to a new state following the deterioration dynamics, (ii) action $q$, the *inspection action*, follows similar state transition dynamics as action $d$ and provides the true state as the observation, (iii) action $r$, the *repair action*, improves the CI of the component to a new state $s'$ where $s < s' \leq s_{max}$ and provides the true state as the observation, and (iv) action $m$, the *replace action*, drives the component state to $s_{max}$, and also provides the true state as the observation.

The transition probability function for each component, governed by its deterioration dynamics $D$, is defined as

$$T(s, a, \bar{s}) = \begin{cases} 1, & \text{if } \bar{s} = s_{max} \text{ and } a = m, \\ 1, & \text{if } \bar{s} = s' \text{ and } a = r, \\ D(s, \bar{s}), & \text{if } \bar{s} \leq s \text{ and } a \in \{d, q\}, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, the observation probability function for each component is defined as

$$Z(\bar{s}, a, o) = \begin{cases} 1, & \text{if } o = \bar{s} \text{ and } a \in \{q, m, r\} \\ 1, & \text{if } o = e \text{ and } a = d \\ 0, & \text{otherwise.} \end{cases}$$

The reward function for each component depends on the objective and constraints of the research problem considered. We discuss the reward formulation in detail in later sections.

In addition to the POMDP model $M^i$, each component is also associated with additional parameters and meta-data that capture the component's importance, maintenance costs, hierarchy, among other attributes. These parameters are essential for modeling the infrastructure system as a whole and are used to define the budget constraints, resource availability, and other operational considerations. Let $\Omega^i$ denote the set of additional parameters associated with component $i$ including $\lambda^i \in [0, 1]$, the relative importance of component $i$ in the infrastructure system, $\delta^i \in [0, s_{max}]$, the failure threshold of component $i$, and $c^i_d, c^i_q, c^i_r, c^i_m$, the costs associated with taking actions $d, q, r$ and $m$ respectively for component $i$.

We manage the collection of $n$ components $\{(M^1, \Omega^i), (M^2, \Omega^2), \ldots, (M^n, \Omega^n)\}$ with a shared budget $B$. The budget $B$ is allocated across the components to perform maintenance, repair, and replacement actions. Assume that the number of $d^i, q^i, m^i$ and $r^i$ actions taken for component $i$ for a horizon $H$ are $n^i_d, n^i_q, n^i_m$ and $n^i_r$ respectively. Then, the total cost incurred for the all the components for the horizon $H$ is given by:

$$C_H = \sum_{i=1}^{|\mathcal{A}|} (n^i_d c^i_d + n^i_q c^i_i + n^i_r c^i_r + n^i_m c^i_m).$$

### 3.2 Problem Statement

For an infrastructure system $\{(M^1, \Omega^1), (M^2, \Omega^2), \ldots, (M^n, \Omega^n)\}$ with a shared budget $B$, we study a series of research problems that aim to find an optimal policy$\pi^*$ that maximizes the time before the components reach their failure thresholds while operating under the budget and other operational constraints. Formally, our goal is to find an optimal policy $\pi^*$ under objective functions of the form

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H}\sum_{i=1}^{n} \lambda^i \cdot \mathbb{I}(s_t^i > \delta^i)\right]$$

while ensuring that, at minimum, the total cost incurred over the time horizon $H$ does not exceed the total budget $B$ i.e. $C_H = \sum_{i=1}^{n}(n_d^i c_d^i + n_q^i c_q^i + n_m^i c_m^i + n_r^i c_r^i) \leq B$.

### 3.3 Research Problems in Infrastructure Management

In this section we present some of the interesting research questions that arise while solving the infrastructure management formulation presented in the previous section.

#### Hierarchical Decision Making

Infrastructure systems exhibit an inherent hierarchical structure, including components, units, and facilities. This hierarchy adds complexity to decision-making processes, as in many real-world applications the decisions are often made at different levels of the hierarchy. Further, the decisions made at lower levels can have cascading effects on higher levels.

#### Stochastic Component Deterioration

The deterioration of infrastructure components is inherently stochastic, influenced by environmental factors, wear-and-tear, and unexpected events. Accurately modeling this stochastic deterioration and integrating these models into decision-making processes is crucial, especially to ensure that the data-driven approaches trained in simulation environments work robustly in real-world deployments. Some additional challenges include how RL can be adapted to operate in environments with high levels of uncertainty and variability.

#### Budget Constraints

Management of infrastructure systems often involves operating under strict budget constraints. Key research questions involve optimal resource allocation for maintenance, repair, and replacement actions, and balancing short-term costs against long-term infrastructure health and functionality. Further, the budget is often not fixed and can vary over time, requiring adaptive policies and policies that plan over a long horizon to ensure optimal resource utilization.

#### Partial Observability

Most learning-based approaches including state-of-the-article reinforcement learning algorithms often assume a full observability of the environment. However, in infrastructure management, the condition of components is often only partially observable, requiring costly inspections to estimate the true state. Research is needed to develop algorithms that can effectively handle partial observability and make informed decisions based on uncertain or incomplete information.

#### Interpretability

Unlike other applications of learning-based decision making, interpretability and explainability are crucial for the adoption of intelligent approaches in real-world infrastructure management. Infrastructure management often involves critical decisions that impact public safety and economic stability and therefore it is essential to understand why learned-policies make certain decisions and how to ensure that these decisions align with domain-specific perceptions, requirements and constraints.

### Sim2Real Gap

Simulation environments are essential for training and evaluating RL models. However, the gap between simulated environments and real-world dynamics can lead to ineffective policies when deployed in the real world. Research is required to develop simulation environments that accurately reflect the complexity and stochasticity of the real world, as well as algorithms that can bridge the Sim2Real gap.

### Sparsity and Time Scales

Unlike typical environments modeled as (PO)MDPs, infrastructure management decisions are made over long time horizons and often involve sparse actions where the agent often has to stay idle and wait for the environment to evolve before taking an action. Further, the decisions often have long-term impacts, with rewards or consequences of actions not immediately observable. Research into approaches and reward shaping techniques capable of handling sparse rewards is crucial for effective infrastructure management.

### Scalability and Computational Efficiency

Real-world infrastructure systems are large-scale, often involving millions of individual components distributed across facilities. The simulation environment and the decision-making algorithms need to be scalable, capable of handling large-scale problems efficiently. Further, policies trained using transfer-learning and meta-learning methods should be able to generalize across different infrastructure systems and scenarios while maintaining computational efficiency. Research into scalable solution methods and efficient computation techniques is vital for practical applicability.

# 4 InfraLib

InfraLib is a comprehensive modeling, simulation, and analysis framework designed to enable research into data-driven, learning-based decision making for infrastructure management under uncertainty. It provides predefined, structured environments while also allowing users to flexibly define custom scenarios and constraints. The code, documentation, example environments, and tutorials are available at `https://infralib.github.io/`.

## 4.1 InfraLib Structure

InfraLib framework adopts a modular architecture, which enables separation of concerns and easy extensibility. The core infrastructure model is designed to be highly configurable, allowing users to define custom components, deterioration models, objectives, constraints, and management actions. The hierarchical structure of infrastructure systems is also configurable, enabling users to group components into units and facilities in domain-specific ways.

InfraLib is implemented as a Python library, leveraging popular scientific computing packages like NumPy and Numba for efficient computation. The framework is designed to be user-friendly, with a simple and intuitive API that abstracts the underlying complexity. This makes InfraLib accessible to a wide range of users, from researchers and practitioners to students and educators. The functionalities of InfraLib library are organized into different modules, with the Core module providing the foundational capabilities of modeling and simulating large-scale infrastructure systems. Additional modules, including the analysis module, visualization module, and expert data collection module, offer advanced tools for understanding infrastructure dynamics and assessing policy performance. The input-output module ensures that all data is stored and retrieved in a standardized format, facilitating seamless integration with external tools and libraries and enabling reproducibility and collaboration.

A key emphasis in InfraLib's design is scalability and computational efficiency. Through a scalable software architecture and efficient algorithms, the framework can simulate infrastructure systems comprising millions of components and spanning long time horizons. This massive scale is crucial for bridging the gap between research and the complexity of real-world infrastructure networks.

### 4.2 Component Condition and Cost Dynamics

In InfraLib, the Condition Index of each component, used to quantitatively represent the component's current state of degradation or functionality, takes values in the range $[0, 100]$. The CI evolves stochastically over time, and the dynamics of component deterioration are modeled as a Markov chain with transition probability function $D(s, s')$. Following the literature (22), we model the CI dynamics as a Weibull distribution tailored to each component's deterioration pattern. The Weibull distribution is a flexible model that can capture a wide range of real-world deterioration behaviors, from early-life failures to wear-out failures. The Weibull distribution is parameterized by shape parameter $k$ and scale parameter $\lambda$, with the CDF given as:

$$F(x; k, \lambda) = 1 - e^{-\left(\frac{x}{\lambda}\right)^k}.$$

For every component, based on real-world data, we assume access to the mean and variance of the shape and scale parameters its Weibull distribution. To generate the transition function $D^i(s, s')$ for component $i$, we collect multiple samples of $k$ and $\lambda$ from their respective distributions:

$$k \sim N(\mu_k, \sigma_k^2)$$

$$\lambda \sim N(\mu_\lambda, \sigma_\lambda^2)$$

and then compute the transition probabilities by estimating them from scaled Weibull CDF values where in each sample trajectory, the CI at time step $k$ is given by:

$$\text{CI}(t) = \lfloor 100 \times (1 - F(t; k, \lambda)) \rfloor.$$

This equation ensures that when $t$ is $0$ (representing a new or fully functional component), CI$(t)$ is 100 (least degraded), and as $t$ increases towards the end of the component's expected lifecycle, CI$(t)$ approaches 0 (most degraded).

The cost of repairing a component in InfraLib is designed to reflect the degree of degradation and the urgency of intervention, based on the condition index. The cost is dynamically calculated based on the state of the component at the time of repair and the effectiveness of the repair action, and is given as:

$$c_r^i = \left(\frac{100 - s^i}{100 - \delta^i}\right)^{\alpha^i} \times c_m^i$$

where $s^i$ is the current CI of component $i$, $\delta^i$ is the failure threshold, $\alpha^i$ is a parameter that adjusts the sensitivity of repair costs to damage, and $c_m^i$ is replacement cost of component $i$. This formulation ensures that repairing severely damaged components is proportionally more expensive, aligning repair costs with the component's condition and the urgency of repairs.

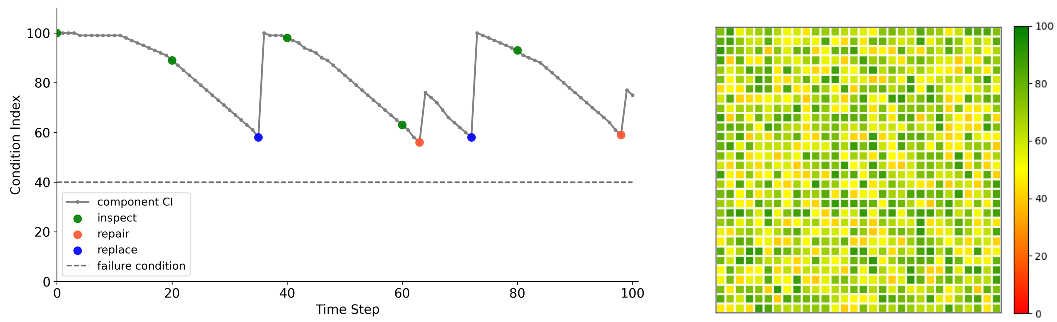
## 5 InfraLib Functionality

InfraLib supports resource allocation problems under several constraints and scenarios that are common in real-world infrastructure management. In this section, we highlight some of the key functionalities of InfraLib and discuss how they can be used to address critical research questions in infrastructure management.

### Optimal Budget Allocation

InfraLib is fundamentally designed to enable optimal budget allocation for large-scale infrastructure systems comprising numerous components. As discussed in 3.1, the modeling framework allows users to optimize actions while considering budget constraints, importance scores, and component deterioration dynamics. Figure 2 illustrates a simulation of this model through the visualization of condition indices for different components.

In a given instance, InfraLib can simulate the evolution of millions of component instances of tens of thousands of component types over a long time horizon. The simulation accepts the actions taken on each component at each time step as input, and transitions the components to new states after verifying that the actions are feasible under the budget constraints.




(a) (b)

Figure 2: Visualization of component condition indices in InfraLib during the rollout of a polcy. (a) Condition index of a single component over time, (b) condition indices of 900 components in a single simulation at a particular time step.

### INTERMITTENT COMPONENT AVAILABILITY

All the components in an infrastructure systems may not always be available for management actions. This is especially relevant in the case of real-world infrastructure systems that are in remote or inaccessible locations, or critical infrastructure components that cannot be taken offline for maintanance. InfraLib supports modeling intermittent component availability, where components can be marked as unavailable for certain time periods. Figure 3 illustrates the condition index of a component that is intermittently unavailable for inspect, repair, and replacement actions.

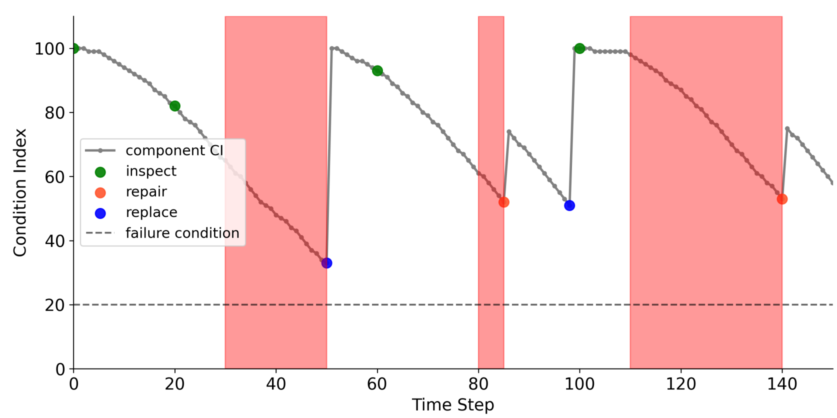



Figure 3: Illustration condition index of a component over time with intermittent unavailability periods highlighted in red.

In addition to enabling users to simulate and analyze scenarios where components are only available for inspection, repair, or replacement during specific time windows, InfraLib also allows users to evaluate the impact of these constraints on their management policies.

### CYCLIC BUDGET

InfraLib can model scenarios with a cyclic budget, where the total budget allotted for infrastructure management is reset to a fixed amount periodically. For instance, the budget could be replenished annually to a predetermined value. Under a cyclic budget schedule, the user can specify either a fixed budget and cycle length or a budget profile with replenished Budget and cycle length that varies over time. Any resources that are not utilized within the current cycle are forfeited and do not carry over.

In addition to the cyclic budget schedules that are directly supported, users can also define custom budget profiles that reflect the budget allocation patterns in their system. Modeling such budget schedules allows testing management policies under real-world resource constraints where budget allocations tend to be more complex than simple cyclic schedules.

### Catastrophic Failures

InfraLib enables modeling of unexpected catastrophic failure events that severely impact infrastructure components instantly. Users can specify failure events to occur at predefined time steps during a simulation or optionally let the library generate random failure events based on a built-in, predetermined distribution. The catastrophic failures can affect one or more components, and can be configured based on the component metadata such as the component type, location, or facility to introduce spatial and temporal dependencies. Figure 4 illustrates how the condition indices of components change in the event of a catastrophic failure.

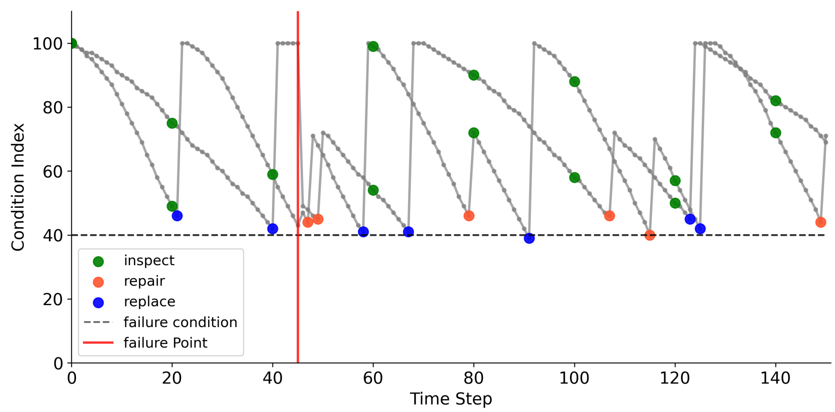



Figure 4: Illustration of the CI of two components over time with a catastrophic failure event occurring at time step 45.

Simulating catastrophic failures provides a mechanism to stress-test infrastructure systems and management policies. It is particularly crucial for evaluating the resiliency of management policies to extreme weather events, natural disasters, or other unforeseen circumstances.

## 5.1 RL Environments

InfraLib generates standardized reinforcement learning environments that encapsulate the complexities of infrastructure management problems while maintaining compatibility with popular RL libraries. Given an infrastructure system modeled in InfraLib, we generate an RL environment $E = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The state space $\mathcal{S} = \prod_{i=1}^{n} \mathcal{S}_i \times \mathbb{R}^+$ incorporates the condition indices of all components and the remaining budget. The action space $\mathcal{A} = \prod_{i=1}^{n} \mathcal{A}_i$ represents all possible combinations of actions across components. The transition probability function $\mathcal{P}(s'|s,a) = \prod_{i=1}^{n} T_i(s'_i|s_i,a_i) \cdot \mathbb{I}(b' = b - c(a))$ is derived from component-level transition probabilities, where $T_i$ is the transition function for component $i$, $c(a)$ is the total action cost, and $\mathbb{I}$ is the indicator function. The generic reward function $\mathcal{R}(s,a,s') = \sum_{i=1}^{n} w_i \cdot f_i(s_i, s'_i) - \lambda \cdot c(a)$ balances the change in component conditions with action costs based on user specification.

To model partial observability, InfraLib can generate POMDP environments $E' = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{O}, Z, \gamma)$, where the observation space $\mathcal{O} = \prod_{i=1}^{n} (\mathcal{S}_i \cup u) \times \mathbb{R}^+$ includes an unknown state $u$ for uninspected components. The observation function $Z(o|s,a) = \prod i = 1^n Z_i(o_i|s_i,a_i)$ reflects the inspection history and recent actions. These environments provide standardized interfaces (`reset()`, `step(action)`, `render()`) compatible with popular RL libraries, facilitating the application and evaluation of RL algorithms to infrastructure management problems.

So far, we have discussed InfraLib from the perspective of modeling and simulating infrastructure systems for learning decision-making policies and evaluating them under various constraints. In the following section, we will delve into the tools provided by InfraLib for analysis and data collection from experts.

## 6 INFRALIB HUMAN INTERFACE

Analysis of existing infrastructure management policies in a unified framework is crucial for enabling decision-makers to compare and evaluate different policies and their impact on different aspects of the infrastructure system. In addition, the ability to collect expert data from human decision-makers is essential for training imitation learning approaches that can leverage these demonstrations without specific reward functions or extensive exploration. In this section, we discuss the tools provided by InfraLib for expert data collection and analysis.

At the core of InfraLib's human interface for analysis and data collection is an intuitive web-based dashboard interface. The interface is powered by a simulation process running in the background and provides experts with detailed information about the current state of a simulated infrastructure system, including component condition indices, recent observations, and historical management actions. The experts can inspect components and allocate resources for maintenance, repair, or replacement based on their domain knowledge. A snapshot of the dashboard with the options available for the user for analysis is shown in Figure 5.

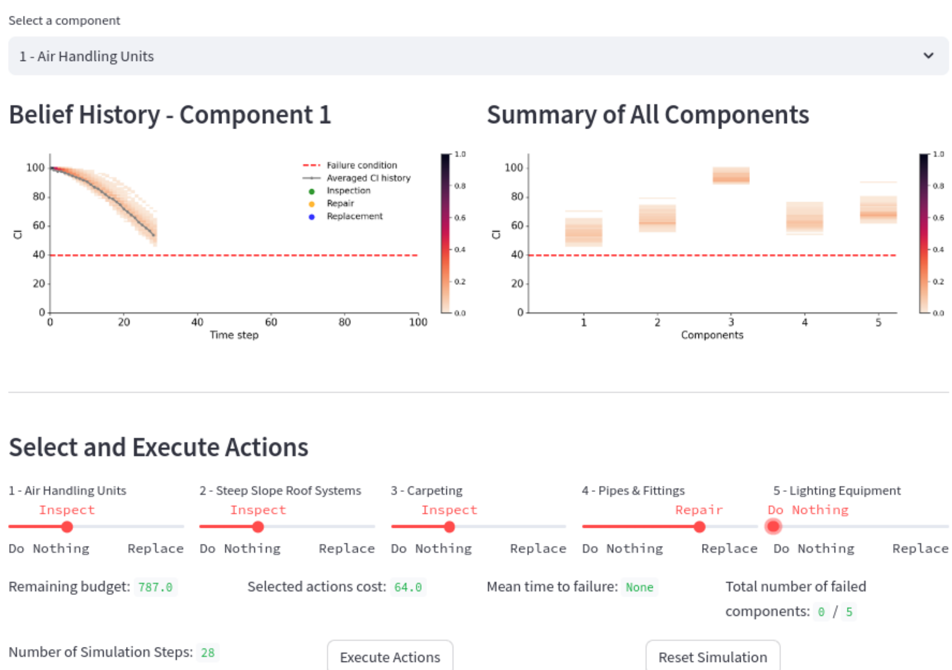



Figure 5: Snapshot from InfraLib's analysis and data collection dashboard interface.

Behind the scenes, InfraLib logs the full trajectory of expert actions, observations, and environment states, and the metadata of the scenario analyzed by the expert. These demonstrations can be used to train imitation learning algorithms to mimic expert behavior or to infer expert's preferences and priorities using inverse reinforcement learning. The expert data collection process is designed to be seamless and user-friendly, allowing experts to focus on demonstrating their management strategies without worrying about the technical details of the simulation environment. In addition to the expert demonstrations collected using the dashboard, InfraLib also supports batch uploads of expert demonstrations generated externally.

InfraLib allows injection of expert knowledge directly into the simulations. Experts can specify replacement thresholds, priority rules, or repair strategies for different components. This expert knowledge can help make the simulations more realistic and guide the agent's exploration in reinforcement learning approaches revealing areas that require tighter mimicking of experts versus allowances for more agent creativity. The modular design also enables the use of expert sub-policies for managing specific components alongside learning-based controllers.

## 7 Example Environments and Benchmarks

This section provides some sample problems and scenarios that can be modeled using InfraLib. The goal of this section is two fold: (i) to provide a set of ready-to-start scenarios where other researchers can directly test their approaches and use similar templates to design their own custom environments, (ii) to provide baseline benchmarks that other researchers can use to compare the performance of their approach.

### Champaign-Urbana Road Network Management

We model the road network in Champaign-Urbana, a metropolitan area in Illinois, United States, using InfraLib and simulate its deterioration without intervention. The road network data is sourced from OpenStreetMap (OSM), which provides detailed attributes for each road segment. These attributes are used to parameterize the deterioration dynamics in the simulation.

The key OSM attributes utilized in the model include the road type (specified by the highway tag), number of lanes, maximum speed limit, and surface material. The highway tag is particularly important as it classifies the road into categories such as motorway, trunk, primary, secondary, tertiary, residential, and service, which have distinct deterioration characteristics.

In the InfraLib model, each road segment is treated as a separate component with its own Weibull deterioration dynamics. The shape and scale parameters of the Weibull distribution for each segment are determined based on a combination of the OSM attributes. For instance, segments with a higher highway classification (e.g., motorway, trunk) and higher speed limits are assumed to have slower deterioration rates compared to lower-class roads (29). Similarly, the surface material affects the deterioration, with asphalt and concrete roads having slower deterioration compared to gravel roads (30). The number of lanes and road width also influence the deterioration dynamics, as wider and multi-lane roads typically have higher construction standards and are more resilient to wear and tear.

Figure 6 illustrates the simulated deterioration of the Champaign-Urbana road network over a 50-year period using InfraLib. The condition index of each road segment is visualized on a color scale, with blue indicating good condition and red indicating poor condition. As seen in the figure, the road network progressively deteriorates over time, with different segments deteriorating at different rates based on their attributes.

This realistic simulation of the Champaign-Urbana road network showcases InfraLib's ability to model large-scale infrastructure systems with heterogeneous components having unique deterioration characteristics.

### LargeSys-100K - Large-Scale infrastructure System Management

To demonstrate InfraLib's scalability and ability to handle large-scale infrastructure systems, we introduce the LargeSys-100K benchmark. This synthetic dataset consists of a massive network with 100,000 component instances spanning 1000 different component types. Each component type has 100 instances, resulting in a total of 100,000 components.

The deterioration dynamics and cost parameters for each component type in LargeSys-100K are synthesized based on realistic ranges observed in real-world infrastructure data. The Weibull distribution shape and scale parameters, inspection costs, repair parameters, and replacement costs for each component type are randomly generated while ensuring they fall within these practicable ranges.

LargeSys-100K serves as a standardized benchmark for comparing the performance and scalability of different infrastructure management approaches. By utilizing a large number of components and component types, LargeSys-100K aims to test the scalability and computational efficiency of infrastructure management algorithms. The vast scale of this benchmark poses challenges in terms of memory usage and computation time, pushing the boundaries of optimization and learning-based approaches.

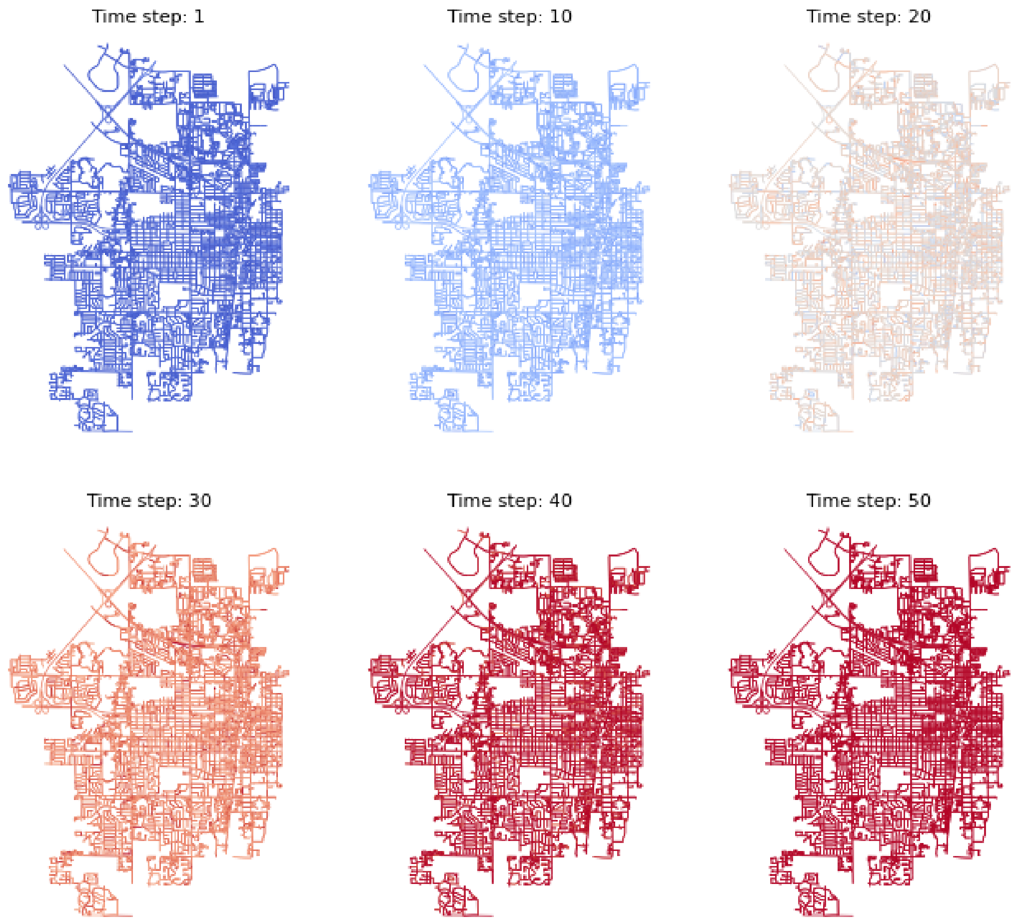



Figure 6: Deterioration of road network in Champaign-Urbana over time. Blue indicates good condition, red indicates poor condition.

Moreover, the diversity of component types in LargeSys-100K adds an additional layer of complexity. With varying deterioration dynamics and costs across component types, algorithms must be able to effectively prioritize and allocate resources considering the heterogeneity of the infrastructure system.

## 8 Conclusion and Future Work

This paper introduced InfraLib, a comprehensive and versatile simulation framework designed to model and analyze large-scale infrastructure management problems. By providing a realistic and granular representation of infrastructure systems, InfraLib enables the application of reinforcement learning and other learning-based decision-making techniques to the complex domain of infrastructure management. The framework's hierarchical and stochastic approach accurately captures the nuances of real-world systems, including budget constraints, resource availability, and the geographical distribution of components. Through a variety of realistic scenarios and benchmarks, InfraLib demonstrates its potential to significantly impact the field, offering researchers and practitioners the means to develop, test, and refine strategies for efficient and effective infrastructure maintenance and allocation.

Looking ahead, there are several promising avenues for future work and enhancements to InfraLib. One direction is to expand the framework's capabilities to model finite crew allocation and scheduling problems, which go hand-in-hand with infrastructure management. Another direction is to integrate transfer learning and meta-learning algorithms to enable the rapid adaptation of learned policies to new infrastructure systems or changing environmental conditions. Furthermore, integrating InfraLib with other tools and platforms commonly used in infrastructure management, such as geographic information systems (GIS) and asset management software, would streamline the data exchange process and facilitate the adoption of learning-based approaches in practice. Finally, building a vibrant community around InfraLib is crucial for its long-term success and impact. Encouraging researchers and practitioners to contribute new components, deterioration models, and management strategies will ensure that the framework remains up-to-date and relevant.

## REFERENCES


[1] Bathula Srinivasu, Srinivasa Rao, et al. Infrastructure development and economic growth: prospects and perspective. *Journal of Business Management and Social Sciences Research*, 2 (1):81–91, 2013.

[2] Balazs Egert, Tomasz J. Kozluk, and Douglas Sutherland. Infrastructure and growth: empirical evidence. Working Paper 2700, CESifo Working Paper, July 2009.

[3] Yifan Yang, Thomas Ng, Frank Xu, and Martin Skitmore. Towards sustainable and resilient high density cities through better integration of infrastructure networks. *Sustainable Cities and Society*, 42:407–422, 2018.

[4] Rae Zimmerman. Social implications of infrastructure network interactions. *Journal of Urban Technology*, 8(3):97–119, 2001.

[5] Steven M Rinaldi, James P Peerenboom, and Terrence K Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine*, 21 (6):11–25, 2001.

[6] Golam Kabir, Rehan Sadiq, and Solomon Tesfamariam. A review of multi-criteria decision-making methods for infrastructure management. *Structure and Infrastructure Engineering*, 10 (9):1176–1210, 2014.

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

[8] Dan Frangopol and Paolo Bocchini. Bridge network performance, maintenance and optimisation under uncertainty: accomplishments and challenges. *Structures and Infrastructure Systems*, pages 30–45, 2019.

[9] Dejan Makovšek, Vincent Benezech, and Stephen Perkins. Efficiency in railway operations and infrastructure management. OECD/ITF Roundtable Reports 177, OECD Publishing, 2015.

[10] Timothy McDaniels, Stephanie Chang, Darren Cole, Joseph Mikawoz, and Holly Longstaff. Fostering resilience to extreme events within infrastructure systems: characterizing decision contexts for mitigation and adaptation. *Global Environmental Change*, 18(2):310–318, 2008.

[11] Thomas Wilbanks, Steven Fernandez, George Backus, Pablo Garcia, Karl Jonietz, Paul Kirshen, Michael Savonis, Bill Solecki, and Leah Toole. Climate change and infrastructure, urban systems, and vulnerabilities. Technical report, U.S. Department of Energy, 2013.

[12] Igor Linkov, Todd Bridges, Felix Creutzig, Jennifer Decker, Cate Fox-Lent, Wolfgang Kröger, James H Lambert, Anders Levermann, Benoit Montreuil, Jatin Nathwani, et al. Changing the resilience paradigm. *Nature Climate Change*, 4(6):407–409, 2014.

[13] Khaled A Abaza. Optimum flexible pavement life-cycle analysis model. *Journal of Transportation Engineering*, 128(6):542–549, 2002.

[14] Roman Denysiuk, André V Moreira, José C Matos, Joel RM Oliveira, and Adriana Santos. Two-stage multiobjective optimization of maintenance scheduling for pavements. *Journal of Infrastructure Systems*, 23(3):04017001, 2017.

[15] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[16] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: a survey of learning methods. *ACM Computing Surveys*, 50(2):1–35, 2017.

[17] Linyi Yao, Qiao Dong, Jiwang Jiang, and Fujian Ni. Deep reinforcement learning for long-term pavement maintenance planning. *Computer-Aided Civil and Infrastructure Engineering*, 35(11):1230–1245, 2020.

[18] Charalampos P Andriotis and Konstantinos G Papakonstantinou. Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering and System Safety*, 191:106483, 2019.

[19] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529 (7587):484–489, 2016.

[20] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[21] Manav Vora, Pranay Thangeda, Michael N Grussing, and Melkior Ornik. Welfare maximization algorithm for solving budget-constrained multi-component POMDPs. *IEEE Control Systems Letters*, 2023.

[22] Michael N Grussing, Donald R Uzarski, and Lance R Marrano. Condition and reliability prediction models using the weibull probability distribution. In *Applications of Advanced Technology in Transportation*, pages 19–24. 2006.

[23] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pages 663–670, 2000.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, and Georg Ostrovski. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017.

[26] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[27] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence*, volume 7, pages 2586–2591, 2007.

[28] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[29] Khaled A Abaza. Empirical approach for estimating the pavement transition probabilities used in non-homogenous Markov chains. *International Journal of Pavement Engineering*, 18(2): 128–137, 2017.

[30] Jidong Yang, Jian John Lu, Manjriker Gunaratne, and Qiaojun Xiang. Forecasting overall pavement condition with neural networks: application on florida highway network. *Transportation Research Record*, 1853(1):3–12, 2003.