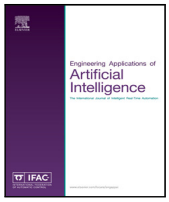
Engineering Applications of Artificial Intelligence 109 (2022) 104655

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Optimal maintenance scheduling under uncertainties using Linear Programming-enhanced Reinforcement Learning

Jueming Hu, Yuhao Wang, Yutian Pang, Yongming Liu *

Arizona State University, Tempe, AZ, 85281, USA

ARTICLE INFO

Keywords:
Maintenance scheduling
Rollout
Linear programming
Infinite horizon
Stochastic maintenance

ABSTRACT

Maintenance is of great importance for the safety and integrity of infrastructures. The expected optimal maintenance policy in this study should be able to minimize system maintenance cost while satisfying the system reliability requirements. Stochastic maintenance scheduling with an infinite horizon has not been investigated thoroughly in the literature. In this work, we formulate the maintenance optimization under uncertainties as a Markov Decision Process (MDP) problem and solve it using a modified Reinforcement Learning method. A Linear Programming-enhanced RollouT (LPRT) is proposed, which considers both constrained deterministic and stochastic maintenance scheduling with an infinite horizon. The novelty of the proposed approach is that it is suitable for online maintenance scheduling, which can include random unexpected maintenance performance and system degradation. The proposed method is demonstrated with numerical examples and compared with several existing methods. Results show that LPRT is able to determine the suitable optimal maintenance policy efficiently compared with existing methods with similar accuracy. Parametric studies are used to investigate the effect of uncertainty, subproblem size, and the number of stochastic stages on the final maintenance cost. Limitations and future work are given based on the proposed study.

1. Introduction

Maintenance is critical for the safety and integrity of engineering structures and infrastructures, such as aircraft, bridges, pipeline systems, railroads, and airports (Dekker and Scarf, 1998). Maintenance scheduling includes the determination of the execution time/interval of (major and minor) maintenance activities under other plans (e.g. planning shutdowns of major refinery units), the work preparation, and the determination of the required maintenance capacity (Dekker and Scarf, 1998). The objective of maintenance scheduling is to increase the profitability of the operation and to ensure the reliability of the system (Khan and Haddara, 2003). The expected optimal system maintenance policy in this study should be able to minimize the system maintenance cost while the system reliability requirements are satisfied.

Many mathematical models for maintenance scheduling have been proposed in the literature. Labib et al. (1998) developed a model of maintenance decision making using the analytic hierarchy process (AHP) to handle multiple criteria decision analysis, conflicting objectives, and subjective judgments. Bayesian updating is used in Nielsen and Sørensen (2011), Wang and Liu (2019) with the new information gained from the monitoring/inspections to reduce the model uncertainty. Genetic Algorithm (GA) (Mitchell, 1998) has received considerable attention in the maintenance optimization field due to the robust search capabilities that help reduce the computational complexity of large optimization problems (Ding and Kamaruddin, 2015). GA has been successfully implemented to determine the optimal maintenance policy for many systems, such as generators (Martorell et al., 2005), bridges (Miyamoto et al., 2000) and roads (Fwa et al., 1994). Monte Carlo simulation is used in Chen and Popova (2002) and Barata et al. (2002) to obtain the best maintenance policy. Markovian probabilistic models for optimizing maintenance policy have also been discussed by Bruns (2002), Marquez and Heguedas (2002), Chiang and Yuan (2001), and Lam (1999) in great detail (Garg and Deshmukh, 2006).

However, methods for stochastic maintenance scheduling and the infinite horizon maintenance scheduling have not been investigated thoroughly. Maintenance scheduling is stochastic in nature and many operations cause uncertainties, such as measurement errors, modeling errors, and human errors. These sources of uncertainties would affect maintenance and degradation performance. Van Noortwijk (1998) modeled stochastic deterioration processes through gamma processes and determined three cost-based criteria for comparing maintenance decisions over unbounded horizons. Uncertainties in corrosion initiation time and rate, as well as other relevant parameters are considered via Monte Carlo simulation in Akgül and Frangopol (2004). Liu and Frangopol (2006) conducted an exhaustive enumeration-based bridge network analysis to evaluate the adverse effects of probabilistic

* Corresponding author.
E-mail address: Yongming.liu@asu.edu (Y. Liu).

https://doi.org/10.1016/j.engappai.2021.104655
Received 24 May 2021; Received in revised form 13 September 2021; Accepted 30 December 2021
Available online xxxx
0952-1976/

bridge failures on the overall performance of the network. Furthermore, Maintenance scheduling over an infinite horizon is essential for infrastructures, typically built for long-term use, such as road maintenance (Gao and Zhang, 2013) and track maintenance (Meier-Hirmer et al., 2009). Grall et al. (2002) derived a computable expression for the expected maintenance cost rate on an infinite horizon depending on the two decision variables, the preventive replacement threshold, and the inspection scheduling function. Pan et al. (2010) proposed a maintenance policy with the aim of minimizing the long-term average maintenance cost rate. The above two methods utilized the concept of maintenance cost rate to solve the infinite horizon maintenance scheduling. Gao and Zhang (2013) focused on road maintenance over an infinite decision horizon and developed a multiobjective Markov-based model. Based on the above brief review of the stochastic maintenance scheduling and the infinite horizon maintenance scheduling, it is hard to find an optimal solution for stochastic optimization and an infinite horizon problem implies large computation, which makes stochastic maintenance scheduling with an infinite horizon a challenging problem. The motivation of this work is to develop an approach for both deterministic and stochastic maintenance scheduling with an infinite horizon. The proposed new approach is based on linear programming (LP) and a reinforcement learning algorithm called RollouT (RT).

LP has also received considerable attention due to its ability to get quickly into the proximity of a global optimum and is a promising tool for online real-time maintenance decision making. Puleo et al. (2014) presented a methodology based on LP for the optimal maintenance scheduling of pumps in real-time. Another study for pump maintenance scheduling problem was solved by a novel hybrid optimization method which makes use of LP and a Greedy Algorithm in Giacomello et al. (2013). Goel et al. (2003) proposed a mixed-integer linear programming (MILP) model to provide the designer with the opportunity to select the initial reliability of equipment at the design stage by balancing the associated costs with its impact on the design and the availability in the operational stage. Xenos et al. (2016) dealt with the optimal operation and maintenance of networks of compressors in chemical plants using a mixed-integer linear programming model.

Due to the fast development and wide applications of Reinforcement Learning (RL) in the past two decades, RL has also been used to solve a variety of realistic control and decision-making problems in the presence of uncertainty, including some applications to maintenance optimization. For example, Q learning is utilized for energy management in a hierarchical electricity market that considers both the service provider's (SP) profit and customers' (CUs) costs in Lu et al. (2018). Rocchetta et al. (2019) applied deep Q learning to do maintenance scheduling of power grid equipped with Prognostics and Health Management (PHM) devices. The advantage of RL is that domain-specific knowledge can be easily included in the RL agent by proper reward definitions but requires quite intensive computation or training. It should be noted that typical RL algorithms try to learn the MDP and solve it to find the optimal policy. Thus, the algorithm needs to solve the exploration and exploitation tradeoff. Other optimization methods proposed in Li et al. (2021), Liang et al. (2020), Chen et al. (2020) also discussed exploration and exploitation tradeoff. Another promising RL method is the rollout algorithm, which is first proposed for the approximate solution of discrete optimization problems by Bertsekas and Tsitsiklis (1996), by Bertsekas et al. (1997), and by Bertsekas and Castanon (1999). The general idea of a rollout algorithm is to start with a suboptimal solution method called base heuristic, and to aim at cost improvement. The rollout algorithm estimates action values by averaging the returns of many simulated trajectories that start with each possible action and then follow the base heuristic. Rollout is shown to be easy to implement and yield robust results (Bertsekas and Castanon, 1999). It does not maintain long-term memories of values or policies like classical learning algorithms, and guarantees a feasible solution (whose cost is no worse than the cost corresponding to the base heuristic) under suitable assumptions. Moreover, rollout solves an infinite horizon problem by solving an ℓ-step lookahead small version of the problem at each encountered state and only recording the first control.

Rollout has been utilized to improve other algorithms for solving optimization problems. Lee and Shayman (2005) applied the technique of rollout to systematically improve the performance of various heuristic algorithms for the logical topology design and traffic grooming problem in multihop wavelength division multiplexing (WDM) networks. Berger et al. (2008) improved known low complexity solutions based on Matching Pursuit using rollout concepts for the sparse estimation problem. Anuar et al. (2021) proposed a matheuristic approach based on a reduced two-stage Stochastic Integer Linear Programming (SILP) model to obtain a policy constructed dynamically on the go during the rollout algorithm. They applied the proposed approach to multi-depot dynamic vehicle routing problem with stochastic road capacity. To the best of our knowledge, this is the first paper to combine LP and rollout and apply the new method to both deterministic and stochastic maintenance scheduling problems.

In this paper, we propose an integrated method, which combines LP and rollout (called LPRT hereafter), to solve the optimal maintenance scheduling. The performance of LPRT is demonstrated using the example in Wang and Liu (2019), which deals with the maintenance scheduling of a pipeline system under fatigue degradation. The contributions of this work and the advantages of LPRT are:

- We formulate maintenance scheduling as an MDP problem and linear optimization problem.
- LPRT is able to provide continuous control for both deterministic and stochastic maintenance scheduling problems.
- LPRT can handle infinite time horizon issue of both deterministic and stochastic maintenance scheduling problems.
- LPRT has the capacity to consider maintenance uncertainties by using a stochastic model and can solve exactly any finite stochastic optimization problem.
- LPRT can be applied to online real-time planning.

The remainder of this paper is organized as follows. Section 2 describes the generic framework for maintenance scheduling and the mathematical definitions for the maintenance objective and constraints. Section 3 presents the details of LPRT for solving deterministic maintenance and stochastic maintenance problems. In Section 4, the numerical experiments are presented to show the capability of the proposed approach to solve constrained maintenance scheduling with and without uncertainties. Section 5 discusses the effect of uncertainty, subproblem size, and the number of stochastic stages. Section 6 concludes this paper and presents some future directions.

2. Problem formulation

The maintenance scheduling problem can be formulated as an optimization problem that aims to minimize the cumulative maintenance costs, while fulfilling reliability constraints over a given planning horizon. The optimization period is divided into a number of discrete control intervals. Thus, the maintenance scheduling problem can also be viewed as a sequential decision problem and a Markov Decision Process (MDP) problem. In an MDP, the learning agent chooses an action(control) and execute it. The control at each time leads the agent to a new state and results in a reward or penalty (cost) for the state transition. The objective of an MDP is to find the policy that maximize (or minimize) the accumulated reward (or cost) over a given horizon. The maintenance scheduling problem can be formulated as an MDP, taking the system condition as the state, maintenance operation at each time step as the action (control), and maintenance cost as the penalty. The output of the MDP is the maintenance operations to perform at each time step.

In this section, we describe the generic framework for the system requiring maintenance scheduling and the optimization model as well. Additionally, we build connections between maintenance scheduling and MDP. In detail, we introduce how to define the four necessary components of an MDP problem, including state, control, transition model, and optimization objective, to translate the maintenance scheduling problem.

2.1. System formulation

Suppose we have a system that consists of Z elements which belong to one type of component and share the reliability behavior. These elements could be categorized into a total of n conditions according to the inspection or health monitoring. The planning horizon is divided into N stages/time intervals (N can be infinity).

2.1.1. State and control

For the whole system, we have a state (condition) vector $x_k = [x_k^1, x_k^2, \dots, x_k^n] \in \mathbb{R}^{1\times n}$ to represent the overall health of the system at time $k \in \{0, \dots, N\}$. x_k^i is the fraction of elements in condition i at time k. The terms in vector x_k should sum up to 1. Condition 1 indicates that the element is in excellent condition. A larger index of condition indicates the worse condition. The last term x_k^n is used to represent the failure category.

Let u_k denotes the control (maintenance decision) at time k, $u_k \in \{u_0, \dots, u_{N-1}\}$. u_k represents the maintenance method applied at time k, $u_k \in \mathbb{R}^{1\times nm}$, where m is the number of possible maintenance operations for each condition,

$$u_k = [u_k^{11}, u_k^{12}, \dots, u_k^{1m}, u_k^{21}, \dots, u_k^{2m}, \dots, u_k^{n1}, \dots, u_k^{nm}] \tag{1}$$

u_k^{is} represents the fraction of elements in condition i at time k that are subjected to maintenance operation s. Thus we have the constraint,

$$\begin{aligned} &\sum_{s=1}^{m} u_k^{is} = 1 \\ &u_k^{is} \geq 0 \quad \text{for } i \in \{0, \dots, n\} \text{ and } k \in \{0, \dots, N-1\} \end{aligned} \tag{2}$$

As an example, let maintenance operation 1 represent doing nothing (a natural degradation process) and operation 2 represent replacement. Then, ($u_k^{11}, u_k^{21}, \dots, u_k^{n1}$) gives the fractions of elements at condition $1, 2, \dots$, and n respectively, to keep a natural degradation process, and ($u_k^{12}, u_k^{22}, \dots, u_k^{n2}$) gives the fractions of elements at condition $1, 2, \dots$, and n, respectively, to take the replacement operation.

2.1.2. Transition model

Transition probabilities are necessary for predicting the overall health of the system at the next time step. Transition probabilities are obtained either from experiment data or by using domain-specific knowledge. Let p_{isj} denote the transition probability which is defined as the probability for an element to transit from the current condition i to the condition j at the next time step after implementing maintenance operation s.

The state $x_k = [x_k^1, \dots, x_k^n]$ evolves according to the system equation

$$x_{k+1}^j = \sum_{i=1}^{n}\sum_{s=1}^{m} p_{isj} u_k^{is} x_k^i \tag{3}$$

Eq. (3) is a generic transition model for a system. P is the transition probability vector with elements denoted by p_{isj}, and p_{isj} expresses the performance of the maintenance operation s and can also include the effect of other progresses such as degradation within one time interval. For example, in Wang and Liu (2019), the system equation is

$$\mathbf{D}_{new} = \sum_{m} \mathbf{D} \cdot \times \mathbf{X}(m, :) \times \mathbf{M}_m \times \mathbf{P} \tag{4}$$

where $\mathbf{D}$ is the condition vector, $\mathbf{X}$ is the maintenance decision matrix, $\mathbf{M}_m$ is the maintenance transition matrix corresponding to operation m, and $\mathbf{P}$ is the degradation transition matrix. Transferring Eq. (4) to the formulation of Eq. (3), p_{isj} will demonstrate transition probabilities corresponding to both $\mathbf{M}_m$ and $\mathbf{P}$. The transformation is introduced in detail in Section 4.

In this work, for simplicity P is static and does not change with time. But this work can be extended to dynamic P. $P \in \mathbb{R}^{n\times nm}$,

$$P = \begin{pmatrix} p_{111} & \cdots & p_{1m1} & p_{211} & \cdots & p_{2m1} & \cdots & p_{n11} & \cdots & p_{nm1} \\ p_{112} & \cdots & p_{1m2} & p_{212} & \cdots & p_{2m2} & \cdots & p_{n12} & \cdots & p_{nm2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{11n} & \cdots & p_{1mn} & p_{21n} & \cdots & p_{2mn} & \cdots & p_{n1n} & \cdots & p_{nmn} \end{pmatrix} \tag{5}$$

The jth row of P shows the probabilities of transferring from each possible condition and under each possible maintenance operation to the new condition j at next time step. The column $i \times m + \tilde{i}$ ($i \in \{0, \dots, n-1\}$, $\tilde{i} \in \{1, \dots, m\}$) of P indicates the probabilities of turning from the current state $i+1$ to each possible condition at next time step under maintenance operation $\tilde{i}$. We notice that the sum of each column should be 1.

2.2. Optimization model

The maintenance scheduling problem in this study aims to minimize the cumulative maintenance cost under the reliability constraint over a given planning horizon.

The cost at time k, g_k, for the system is given by

$$g_k(x_k, u_k) = Z \times \sum_{i=1}^{n}\sum_{s=1}^{m} c_{is} u_k^{is} x_k^i \tag{6}$$

where c_{is} is the cost of maintenance operation s per element at condition i. Let c_{is} forms a maintenance cost vector C,

$$C = [c_{11}, \dots, c_{1m}, c_{21}, \dots, c_{2m}, \dots, c_{n1}, \dots, c_{nm}] \tag{7}$$

Then, the cumulative cost over N stages, G, is

$$\begin{aligned} G &= \sum_{k=0}^{N-1} g_k(x_k, u_k) \\ &= \sum_{k=0}^{N-1} Z \times \sum_{i=1}^{n}\sum_{s=1}^{m} c_{is} u_k^{is} x_k^i \end{aligned} \tag{8}$$

The reliability requirement is related to the failure probability of the system, which is represented by x_k^n. When x_k^n reaches a reliability threshold value, the system fails. Mathematically, the reliability threshold is an upper bound, denoted by b, for x_k^n at all k,

$$x_k^n \leq b \quad \text{for } k \in \{0, \dots, N\} \tag{9}$$

Thus, the generic optimization model for the maintenance system is

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=0}^{N-1} Z \times \sum_{i=1}^{n}\sum_{s=1}^{m} c_{is} u_k^{is} x_k^i \\ \text{s.t.} \quad & x_k^n \leq b \quad \text{for } k \in \{0, \dots, N\} \end{aligned} \tag{10}$$

3. Methodology

This section presents the details of LPRT for solving deterministic maintenance and stochastic maintenance problems. Additionally, in the deterministic maintenance part, we provide a review of rollout and the linearization of the optimization model introduced above in Eq. (10). The linearization of the maintenance scheduling model is essential before applying LPRT. By integrating LP with rollout, LPRT is able to combine the benefits of both LP or rollout algorithms. Thus, LPRT is capable of delivering continuous control, while rollout needs to discretize control space before solving a problem. In addition, with the help of rollout, LPRT can solve both deterministic and stochastic maintenance problems with an infinite horizon, which cannot be done by pure LP. Another advantage of LPRT is the capability for online real-time planning, which is important and practical especially when new information for the system is available.

3.1. Deterministic maintenance planing

3.1.1. Review of rollout

Since the 1950s, MDP (Bellman, 1957) has been well studied and applied to a wide range of disciplines (Howard, 1964; White, 1993), including robotics (Bhattacharya et al., 2020), automatic control (Hu et al., 2020; Hu and Liu, 2020), economics, and manufacturing. In MDP, the agent may choose any control u that is available based on current state x at each time step. The process responds at the next time step by moving into a new state x' with certain transition probability and gives the agent a corresponding cost g. The cost at time k, g_k, is usually a function of the current state x_k and control u_k.

The goal of MDP is to find an optimal policy π^* and optimal control u^* at each time step,

$$\begin{aligned} &\pi^* = \{\mu_o^*(x_0), \dots, \mu_{N-1}^*(x_{N-1})\} \\ &u_k^* = \mu_k^*(x_k) \quad \text{for } k \in \{0, \dots, N-1\} \end{aligned} \tag{11}$$

The optimal control u_k^* is selected by minimizing the expected cumulative immediate costs,

$$\begin{aligned} u_k^* &\in \underset{u_k \in U_k(x_k)}{\arg\min}\, E\{J_k^*\} \\ &\in \underset{u_k \in U_k(x_k)}{\arg\min}\, E\{\sum_{i=k}^{N-1} g_i(x_i, u_i)\} \\ &\in \underset{u_k \in U_k(x_k)}{\arg\min}\, E\{g_k(x_k, u_k) + J_{k+1}^*(f_k(x_k, u_k))\} \end{aligned} \tag{12}$$

where $U_k(x_k)$ is the available control set at x_k, f_k is the transition function, and J_{k+1}^* is the optimal cost-to go function, which indicates the cumulative costs starting from the current state x_{k+1} to the terminal state x_N.

The cost-to-go J_{k+1}^* is hard to obtain due to large computation complexity and memory requirements. The key idea of rollout is to compute a cost-to-go approximation, $\tilde{J}_{k+1}$, according to a predefined heuristic base policy $\pi = \{\mu_1, \dots, \mu_{N-1}\}$. $\tilde{J}_{k+1}$ is the summation of immediate costs from x_{k+1} to the terminal state x_N and the control at each time step follows the base policy π,

$$\begin{aligned} u_k &= \mu_k(x_k) \\ \tilde{J}_{k+1} &= \sum_{i=k+1}^{N-1} g_i(x_i, \mu_i(x_i)) \end{aligned} \tag{13}$$

Then Eq. (12) is turned to,

$$\tilde{u}_k \in \underset{u_k \in U_k(x_k)}{\arg\min}\, E\{g_k(x_k, u_k) + \tilde{J}_{k+1}(f_k(x_k, u_k))\} \tag{14}$$

and the minimizing $\tilde{u}_k$ at the states x_k define a policy, called rollout policy. It is the suboptimal solution obtained through rollout. $\tilde{J}_{k+1}$ is calculated by running the base policy, starting from each possible x_{k+1}. Fig. 1 shows a schematic illustration of rollout with one-step lookahead for a deterministic problem.

$$\tilde{Q}_k(x_k, u_k) = g_k(x_k, u_k) + g_{k+1}(x_{k+1}, u_{k+1}) + \cdots + g_{N-1}(x_{N-1}, u_{N-1}) + g_N(x_N) \tag{15}$$

The base policy may be obtained in a variety of ways, including sophisticated off-line methods that are based on approximation in value or policy space. What is important in the context of rollout is that the cost-to-go values of the base policy starting from any state can be calculated in some way, possibly including simulation. Approximation-based alternatives for the evaluation of costs-to-go of the base policy are also provided in Bertsekas (2020), including the off-line approximation of the Q-factors of the base policy and off-line approximation of the rollout policy. A variation of off-line approximation of the Q-factors of the base policy is a truncated rollout scheme, where a parametric approximation of the cost function of the base policy is used as terminal cost function approximation. Fig. 2 shows an example of the truncated rollout scheme for a stochastic system. w_k represents the random disturbance (e.g., physical noise, market uncertainties, unpredictable breakdowns, etc.) at time k. Besides (x_k, u_k), g_k and f_k in a stochastic system also depend on w_k. It involves three steps, (1) multistep lookahead over ℓ stages (possibly $\ell = (1)$, (2)running a heuristic algorithm/base policy from state $x_{k+\ell}$ for a number of steps, say m, (3)adding a cost function approximation at the end of the $\ell + m$ steps. The proposed method for the stochastic maintenance problem uses this multistep lookahead approach.

3.1.2. Linearization

The linear formulation in this study is suitable for the maintenance scheduling problems, in which the system transition probability and the cost of each maintenance operation are predetermined and are not related to the history of the system condition or maintenance operations, such as maintenance scheduling for the wind turbine (Froger et al., 2018) and overhead lines in power distribution systems (Abiri-Jahromi et al., 2009).

To linearize the system transition model in Eq. (3) and objective function in Eq. (8), we introduce a new variable $y_k \in \mathbb{R}^{1\times nm}$:

$$\begin{aligned} y_k &= (y_k^{11}, \dots, y_k^{1m}, y_k^{21}, \dots, y_k^{2m}, \dots, y_k^{n1}, \dots, y_k^{nm}) \\ y_k^{is} &= u_k^{is} x_k^i \end{aligned} \tag{16}$$

Then the system transition model (Eq. (3)) and objective function (Eq. (8)) are reformulated as,

$$x_{k+1}^j = \sum_{i=1}^{n}\sum_{s=1}^{m} p_{isj} y_k^{is} \tag{17}$$

$$G = \sum_{k=0}^{N-1} Z \times \sum_{i=1}^{n}\sum_{s=1}^{m} c_{is} y_k^{is} \tag{18}$$

Under this formulation, the system transition equation as well as the maintenance cumulative cost are a linear function of y_k^{is}. Furthermore, we set constraints on u_k and y_k to represent Eq. (16). The optimization model in Eq. (10) is turned into a multistage linear optimization problem, which is solvable by linear optimization software:

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=0}^{N-1} Z \times \sum_{i=1}^{n}\sum_{s=1}^{m} c_{is} y_k^{is} \\ \text{s.t.} \quad & x_{k+1}^j = \sum_{i=1}^{n}\sum_{s=1}^{m} p_{isj} y_k^{is}, \\ & x_k^n \le b, \\ & \sum_{s=1}^{m} u_k^{is} = 1, \\ & u_k^{is} \ge 0, \\ & \sum_{s=1}^{m} y_k^{is} = x_k^i \end{aligned} \tag{19}$$

3.1.3. LPRT method

By integrating LP with rollout, LPRT is able to combine the benefits of both LP or rollout algorithms. Thus, LPRT is capable of delivering continuous control, which compensates for the disadvantage of rollout. In addition, with the help of rollout, LPRT can solve both deterministic and stochastic maintenance problems with an infinite horizon.

For the deterministic case, the major benefit of LPRT is the capability to solve the maintenance problem with an infinite horizon.

Suppose we have a differentiable closed-form expression for the heuristic cost $\tilde{J}_{k+1}$ in the infinite horizon problem, the minimization of Eq. (14) can be solved with one of the many gradient-based methods that are available for differentiable unconstrained and constrained optimization problems. However, the preceding approach requires the heuristic cost $\tilde{J}_{k+1}$ to be differentiated. It should either be available in closed form, which is highly restrictive, or can be differentiated numerically, which can be inconvenient and/or unreliable. These difficulties can be circumvented by introducing a base heuristic based on multistep

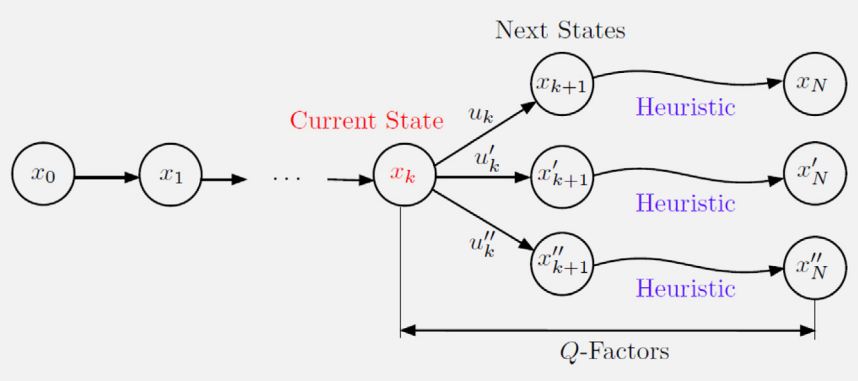


Fig. 1. Schematic illustration of rollout with one-step lookahead for a deterministic problem. At state x_k, for every pair (x_k, u_k), $u_k \in U_k(x_k)$, the base heuristic generates a Q-factor $\tilde{Q}_k(x_k, u_k)$ (see Eq. (15)), and selects the control $\tilde{\mu}_k(x_k)$ with minimal Q-factor (Bertsekas, 2020).

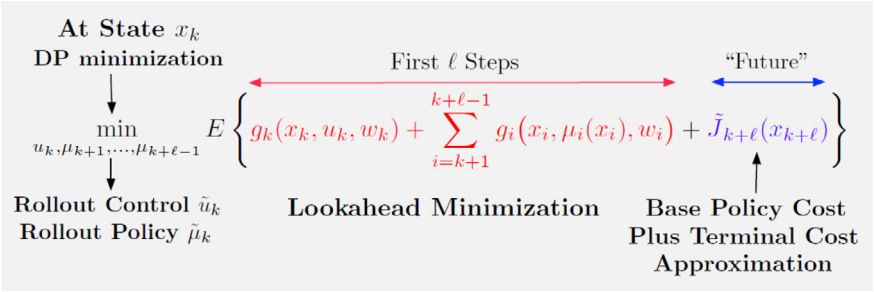


Fig. 2. The structure of truncated rollout with ℓ-step lookahead at state x_k (Bertsekas, 2020).

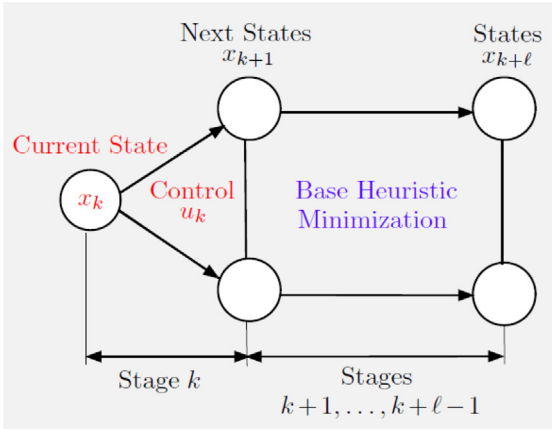


Fig. 3. Schematic illustration of rollout for a deterministic problem with infinite horizon (Bertsekas, 2020).

optimization. This has also been discussed in Bertsekas (2020) (see Section 2.5.1).

The schematic illustration of rollout for a deterministic problem with an infinite horizon is shown as follows (see Fig. 3). The base heuristic is obtained through solving an $(\ell - 1)$-stage deterministic optimal control subproblem which is related to the original optimization problem, and conducting the kth stage minimization over $u_k \in U_k(x_k)$. This process is the same as an ℓ-stage continuous-space optimal control problem that starts at state x_k.

Regarding the specific maintenance scheduling problem in this study, the ℓ-stage problem is derived in Eq. (19) when replacing N with ℓ. Fig. 4 shows the procedures of LPRT. LPRT integrates rollout with LP. This integrated LPRT method implies that at time k, we solve a shorter horizon subproblem of ℓ stages starting at the current state x_k by linear programming, and use the first control as the result of u_k. The next state x_{k+1} is predictable using x_k, u_k and P. This process is repeated N times.

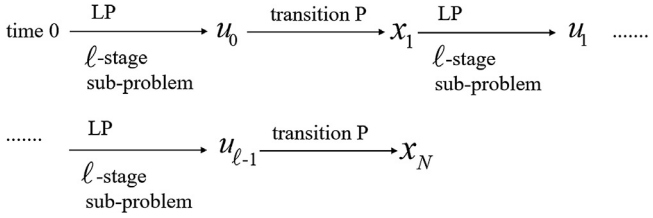


Fig. 4. Schematic illustration of LPRT.

Another benefit of LPRT is the capability for online real-time planning. This is important and practical especially when new information for the system is available from the non-destructive evaluation (NDE) or structural health monitoring (SHM). The optimal maintenance scheduling needs to be continuously updated during the operation. Real-time decision support will be very desirable.

3.2. Stochastic maintenance scheduling

Uncertainty quantification and risk assessment are essential to engineering problems (Liu and Goebel, 2018; Pang and Liu, 2020; Pang et al., 2021). The performance of the maintenance operation or the degradation process is obtained either from experiment data or through

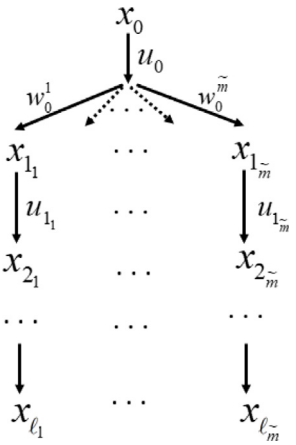


Fig. 5. Schematic illustration of the number of control variables where $\tilde{\ell}s$ is 1 and starting from x_0.

specific domain knowledge. However, many sources of uncertainties exist, such as measurement errors, modeling errors, and human errors. These sources of uncertainties would affect the maintenance and degradation performance. As a result, it is necessary to consider uncertainty in the maintenance scheduling problem.

Mathematically, we consider the transition model is stochastic and p_{isj} contains uncertainty. For example, in Wang and Liu (2019), either the maintenance operation transition probability matrix **M** or the degradation matrix **P** has uncertainty. Then an uncertain event at stage k ($k \in \{0, \dots, N-1\}$) represented by a random variable w_k will occur, whereby w_k will take one of the values in $w_k^1, \dots, w_k^{\tilde{m}}$ with corresponding probabilities $p(w_k^1), \dots, p(w_k^{\tilde{m}})$. $\tilde{m}$ is the number of values that a disturbance can take at each stage. In this study, vector P in Eq. (5) would have $\tilde{m}$ different versions and each element p_{isj} would have $\tilde{m}$ value.

Rollout for stochastic programming has also been discussed in Bertsekas (2020) (see Section 2.5.2). The key step of the proposed LPRT method is that at $x_k (k \in \{0, \dots, N-1\})$, we solve an optimal control problem of ℓ stages ($\ell = \tilde{\ell}s + \tilde{\ell}$) by LP, where the uncertainty is taken into account in the first $\tilde{\ell}s$ stages ($\tilde{\ell}s$ possibly equals to 1) similar to stochastic programming. In the remaining $\tilde{\ell}$ stages, the uncertainty is dealt with by fixing the disturbances $w_{k+\tilde{\ell}s}, \dots, w_{k+\ell-1}$ at some nominal values. Only the result of the first control u_k from LP is saved and then we apply u_k online, and generate the next state x_{k+1}. Afterwards, at time $k+1$ and state x_{k+1}, we perform a new optimization with ℓ stages again. This process is repeated for N times.

We first introduce the mathematical formulas when $\tilde{\ell}s$ is 1 and then extend the formulation to more complex cases. In detail, at the first stage of each shorter horizon subproblem of ℓ stages, we take the first minimization where we select u_0, and assume that the first stage uncertainty takes $\tilde{m}$ values $w_0^1, \dots, w_0^{\tilde{m}}$. Then u_0 and $w_0^{\tilde{i}}$ would result in $x_{1_{\tilde{i}}}$ and $u_{1_{\tilde{i}}}$, with the cost of $g_0(x_0, u_0, w_0^{\tilde{i}})$. w at other stages $(w_1, \dots, w_{N-1})$ are deterministic, so $x_{1_{\tilde{i}}}$ and $u_{1_{\tilde{i}}}$ would result in corresponding $u_{2_{\tilde{i}}}, \dots, u_{\ell-1_{\tilde{i}}}$. The linear programming for the subproblem will return the control solution **U**, with a total of $1 + \tilde{m}(\ell - 1)$ control vectors,

$$\mathbf{U} = (u_0, u_{1_1}, \dots, u_{1_{\tilde{m}}}, u_{2_1}, \dots, u_{2_{\tilde{m}}}, \dots, u_{\ell-1_1}, \dots, u_{\ell-1_{\tilde{m}}}) \tag{20}$$

The schematic illustration of the number of control variables where $\tilde{\ell}s$ is 1 and starting from x_0 is shown in Fig. 5. The expected cost for minimization of a general subproblem are reformulated as follows.

$$\begin{aligned}
&\min_{u_k} \sum_{\tilde{i}=1}^{\tilde{m}} p(w_0^{\tilde{i}})(g_0(x_0, u_0, w_0^{\tilde{i}}) + \tilde{J}_1(x_{1_{\tilde{i}}})) \\
&= \min_{u_k} \quad p(w_0^1) * (g_0(x_0, u_0, w_0^1) + g_1(x_{1_1}, u_{1_1}, w_1) + \cdots \\
&\qquad + g_{\ell-1}(x_{\ell-1_1}, u_{\ell-1_1}, w_{\ell-1})) \\
&\qquad + p(w_0^2) * (g_0(x_0, u_0, w_0^2) + g_1(x_{1_2}, u_{1_2}, w_1) + \cdots \\
&\qquad + g_{\ell-1}(x_{\ell-1_2}, u_{\ell-1_2}, w_{\ell-1})) \\
&\qquad \cdots \\
&\qquad + p(w_0^{\tilde{m}}) * (g_0(x_0, u_0, w_0^{\tilde{m}}) + g_1(x_{1_{\tilde{m}}}, u_{1_{\tilde{m}}}, w_1) + \cdots \\
&\qquad + g_{\ell-1}(x_{\ell-1_{\tilde{m}}}, u_{\ell-1_{\tilde{m}}}, w_{\ell-1})) \\
&= \min_{u_k} \quad \sum_{\tilde{i}=1}^{\tilde{m}} p(w_0^{\tilde{i}}) * (g_0(x_0, u_0, w_0^{\tilde{i}}) + \sum_{j=1}^{\ell-1} g_j(x_{j_{\tilde{i}}}, u_{j_{\tilde{i}}}, w_j)) \\
&= \min_{u_k} \quad \sum_{\tilde{i}=1}^{\tilde{m}} p(w_0^{\tilde{i}}) * g_0(x_0, u_0, w_0^{\tilde{i}}) + \sum_{\tilde{i}=1}^{\tilde{m}} p(w_0^{\tilde{i}}) * g_1(x_{1_{\tilde{i}}}, u_{1_{\tilde{i}}}, w_1) + \cdots \\
&\qquad + \sum_{\tilde{i}=1}^{\tilde{m}} p(w_0^{\tilde{i}}) * g_{\ell-1}(x_{\ell-1_{\tilde{i}}}, u_{\ell-1_{\tilde{i}}}, w_{\ell-1})
\end{aligned} \tag{21}$$

In this study, the objective and constraints for a ℓ-stage maintenance scheduling subproblem with uncertainties on the first stage are turned to,

$$\begin{aligned}
\min_{u_k} \quad & Z \times (\sum_{i=1}^{n} \sum_{s=1}^{m} c_{is} y_0^{is} + \sum_{k=1}^{\ell-1} \sum_{\tilde{i}=1}^{\tilde{m}} (p(w_0^{\tilde{i}}) \times \sum_{i=1}^{n} \sum_{s=1}^{m} c_{is} y_{k_{\tilde{i}}}^{is})) \\
\text{s.t.} \quad & x_{1_{\tilde{i}}} = f(x_0, u_0, P^{\tilde{i}}), \\
& x_{k+1_{\tilde{i}}} = f(x_k, u_{k_{\tilde{i}}}, P) \quad \text{for} \quad k \geq 1, \\
& x_{k_{\tilde{i}}}^n \leq b, \\
& \sum_{s=1}^{m} u_{k_{\tilde{i}}}^{is} = 1, \\
& u_{k_{\tilde{i}}}^{is} \geq 0, \\
& \sum_{s=1}^{m} y_{k_{\tilde{i}}}^{is} = x_{k_{\tilde{i}}}^{i}
\end{aligned} \tag{22}$$

LPRT with larger $\tilde{\ell}s$, called multistep lookahead LPRT, aims to better represent the stochastic process that the system experiences over ℓ time intervals. By increasing the number of stages where stochastic uncertainty is explicitly accounted for, any finite horizon stochastic optimization problem is solved exactly. For the short horizon optimization with ℓ stages, involving $\tilde{\ell}s$ stochastic stages and $\tilde{\ell}$ subsequent deterministic stages, the total number of control variables are,

$$d(1 + \tilde{m} + \tilde{m}^2 + \cdots + \tilde{m}^{\tilde{\ell}s-1} + \tilde{m}^{\tilde{\ell}s} \times (\ell - \tilde{\ell}s)) \tag{23}$$

where d is the dimension of the control vector.

We also present an example of control variables where $\tilde{\ell}s$ is 3 and starting from x_0, which is shown in Fig. 6. In this example, action u_0 has one value which results in $\tilde{m}$ values of x_1. Each x_1 corresponds to a unique u_1, so u_1 has $\tilde{m}$ values. Afterwards, each u_1 results in $\tilde{m}$ values of x_2 at the second stage and therefore x_2 has $\tilde{m}^2$ values. At the third stage, there are totally $\tilde{m}^2$ values of u_2 and as a consequence, x_3 has $\tilde{m}^3$ values. Then, subsequent stages are deterministic, indicating one action generates one deterministic state at the next time step.

A summary of the advantages, disadvantages, and suggested applications of the three methods are listed in Table 1.

4. Numerical experiments

LPRT presented in Section 3 has been implemented using GLPK solver (Makhorin, 2001) and Python Optimization Modeling Objects (Pyomo) package (Hart et al., 2017). Two demonstrations are presented below. The first experiment tests the performance of LPRT for

Table 1
Summary of LP, Rollout and LPRT.

	LP	Rollout	LPRT
Characteristic	Fast solving; Handle uncertainties for short horizon; Continuous control; Demand linear objective and constraint functions	Handle infinite horizon; Discrete control	Solve both deterministic and stochastic optimization with infinite horizon; Continuous control
Suggested application	Linear optimization problem; Online real-time planning	Infinite horizon	Online planning; Stochastic optimization; Infinite horizon

Table 2
The five condition stages for gas pipeline according to the crack length.

Condition	Excellent	Good	Fair	Poor	Failure
Crack length (in)	$1\times10^{-3}\sim2\times10^{-3}$	$2\times10^{-3}\sim6\times10^{-3}$	$6\times10^{-3}\sim2\times10^{-2}$	$2\times10^{-2}\sim0.1$	>0.1

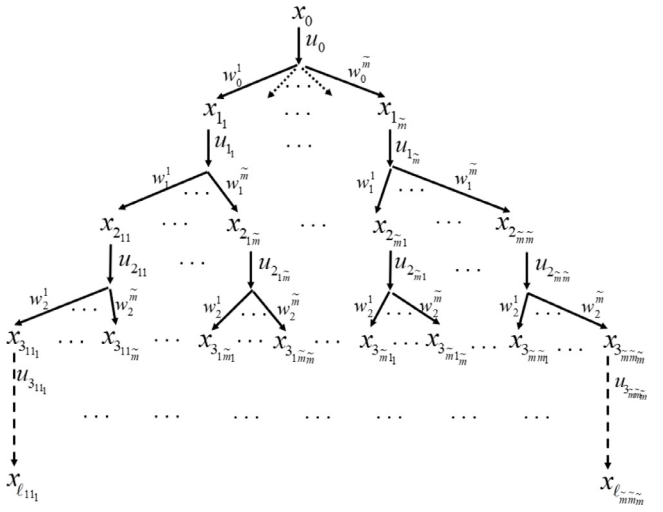


Fig. 6. Schematic illustration of the number of control variables where $\tilde{\ell}s$ is 3 and starting from x_0.

deterministic maintenance scheduling and the results are compared to the results obtained from LP, multiagent rollout, genetic algorithm (GA), and Proximal Policy Optimization (PPO). The second experiment examines LPRT for stochastic maintenance scheduling and is followed by sensitivity analysis.

The numerical experiments in this study consider the maintenance scheduling for a distribution pipeline system. The distribution pipes are made of plastic and are working under constant pressure, which makes them subject to creep damage under room temperature. The damage accumulation can lead to abrupt failure and can lead to extreme events. A proper maintenance scheduling decision can help maintain the safety and integrity of the pipeline system.

In the pipeline industry, the pipeline system can be discretized into pipe sections and each section of pipe can be categorized into different conditions according to their crack length. Following the experiment in Wang and Liu (2019), we categorize the pipe sections into 5 different condition stages according to the crack length in Table 2. The objective of the maintenance scheduling is to minimize the overall maintenance cost over 10 stages (with the stage interval of 1500 h) under the reliability constraint that the failure probability, x_k^5 (for $k\in\{0,\ldots,10\}$), should not exceed 5%. The initial state of the system is assumed as,

$$x_0=[0.10,0.20,0.50,0.15,0.05] \tag{24}$$

We also used the data of pipe property in Wang and Liu (2019) to conduct the experiment. After a stage interval of 1500 h, the probability of transition for the pipe from one condition to another, the degradation matrix **P**, is:

$$\mathbf{P}=\begin{pmatrix}0.7945 & 0.2051 & 0.0003 & 0.0001 & 0\\ 0 & 0.8121 & 0.1875 & 0.0003 & 0\\ 0 & 0 & 0.8085 & 0.1914 & 9.34e-05\\ 0 & 0 & 0 & 0.8999 & 0.1001\\ 0 & 0 & 0 & 0 & 1\end{pmatrix} \tag{25}$$

Three maintenance operations are available: doing nothing(a natural degradation process), replacement, and repair. The probability transition matrix for doing nothing (Wang and Liu, 2019), $\mathbf{M_1}$, is the identity matrix. Replacement can be considered the most robust since it replaces the damaged pipes with new ones. Considering the manufacture defect for the new pipes, the probability transition matrix for replacement (Wang and Liu, 2019), $\mathbf{M_2}$, is:

$$\mathbf{M_2}=\begin{pmatrix}0.99 & 0.01 & 0 & 0 & 0\\ 0.99 & 0.01 & 0 & 0 & 0\\ 0.99 & 0.01 & 0 & 0 & 0\\ 0.99 & 0.01 & 0 & 0 & 0\\ 0.99 & 0.01 & 0 & 0 & 0\end{pmatrix} \tag{26}$$

Typical repair methods in pipeline industry include heating, clamping, and recoating etc. A repair can cure the creep cracks and would partially recover the condition of a pipe. Its effect depends on the current stage of a pipe. The probability transition matrix for doing repair (Wang and Liu, 2019), $\mathbf{M_3}$, is:

$$\mathbf{M_3}=\begin{pmatrix}1 & 0 & 0 & 0 & 0\\ 0.5000 & 0.5000 & 0 & 0 & 0\\ 0.1000 & 0.3000 & 0.6000 & 0 & 0\\ 0.0100 & 0.0400 & 0.2500 & 0.7000 & 0\\ 0 & 0.0100 & 0.0400 & 0.1500 & 0.8000\end{pmatrix} \tag{27}$$

The transition probability, p_{isj} is computed based on maintenance performance matrix, $\mathbf{M_1},\mathbf{M_2},\mathbf{M_3}$, and degradation matrix, **P**,

$$\begin{aligned}p_{isj}&=\tilde{P}(j,s,i)\\ \tilde{P}(i,:,:)&=\begin{pmatrix}\mathbf{M_1}(i,:)\\ \mathbf{M_2}(i,:)\\ \mathbf{M_3}(i,:)\end{pmatrix}\times\mathbf{P},i\in\{1,\ldots,n\}\end{aligned} \tag{28}$$

The maintenance cost, C, also follows (Wang and Liu, 2019),

$$C=\begin{pmatrix}0 & 0 & 0 & 0 & 0\\ 1600 & 1600 & 1600 & 2500 & 3000\\ 30 & 30 & 50 & 100 & 200\end{pmatrix} \tag{29}$$

4.1. Deterministic maintenance

At each stage, we solve a subproblem of 6 stages, indicating ℓ is 6. Table 3 shows the plan of replacement operation for each condition at

Table 3
The fraction of the elements subjected to replacement.

Stage	0	1	2	3	4	5	6	7	8	9
Excellent	0	0	0	0	0	0	0	0	0	0
Good	0	0	0	0	0	0	0	0	0	0
Fair	0	0	0	0	0	0	0	0	0	0
Poor	0	0	0	0	0	0	0	0	0	0
Failure	0.032	0.059	0.042	0.014	0	0	0	0	0	0

Table 4
The fraction of the elements subjected to repair.

Stage	0	1	2	3	4	5	6	7	8	9
Excellent	0	0	0	0	0	0	0	0	0	0
Good	1	1	1	1	1	0.841	0.195	0.287	0.327	0.345
Fair	1	1	1	1	1	1	1	1	1	1
Poor	1	1	1	1	1	1	1	1	1	1
Failure	0.968	0.941	0.958	0.986	1	0.773	0.722	0.641	0.590	0.562

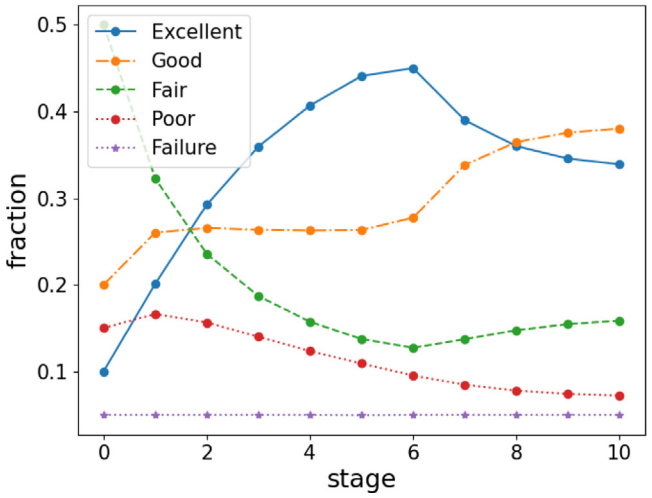


Fig. 7. The system condition at each stage.

each stage. Only the elements in failure condition need to be replaced at the first 4 stages and the amount is less than 10%. Table 4 demonstrates the fraction of elements in each condition subjected to repair operation at each stage. The entire elements in fair and poor conditions demand repair at all the 10 stages. All the elements in good condition require repair for the first 5 stages, and afterwards, the amount decreases to about 20% at stage 6, along with a moderate growth within the succeeding 3 stages. It is not necessary for both the entire elements in excellent condition to apply any maintenance operation. Fig. 7 presents the system condition at each stage. It can be seen that the failure probability maintains 5% over the whole maintenance period, which implies that LPRT has the capability to solve the constrained maintenance scheduling problem. Also, more elements transit to the conditions of excellent and good after maintenance operations.

For comparison purposes, we also obtain the results from pure LP (Chvatal et al., 1983) implemented using GLPK solver (Makhorin, 2001) and Pyomo package (Hart et al., 2017), multiagent rollout (Bertsekas, 2019), GA (Mitchell, 1998) implemented using MATLAB global optimization toolbox (The MathWorks, 2018), and PPO (Schulman et al., 2017) implemented using OpenAI Baselines (Dhariwal et al., 2017). LP method provides the global optimum result which is the optimal solution to this experiment. Comparing with LP result helps evaluate the performance of the proposed LPRT. Multiagent rollout is a recently developed reinforcement learning method in Bertsekas (2019) aiming to reduce computational cost. Since the proposed LPRT is based on LP and rollout, we compare the result of LPRT with pure rollout to see any advantage after the combination of the two methods. GA is another popular method for optimization problems (Tabassum et al., 2014) and is also adopted in Wang and Liu (2019). PPO (Schulman et al., 2017) is an advanced reinforcement learning algorithm for solving continuous problems and has achieved success in automation (Hu et al., 2020). We study the potential of PPO to solve maintenance scheduling and compare it with LPRT.

Following is the first work to implement multiagent rollout to a real-world engineering problem. Compared to the standard rollout approach, the idea of multiagent rollout is to split the control u_k into several parts, and actions in each part are chosen by one agent. Section 2.1.1 mentioned u_k is a $1\times nm$ vector in this study. Let available action set U_k has $\tilde{n}$ values. The computation required by the rollout algorithm is of order $O(\tilde{n}^{nm})$ per stage. We break down the control u_k into the sequence of n (the same as the number of conditions) control sequences, or we call it agents.

$$u_k = (u_{k1}, u_{k2}, \dots, u_{kn}) \tag{30}$$

Agent $\tilde{i}$ has the control $u_{k\tilde{i}} = (u_k^{\tilde{i}1}, \dots, u_k^{\tilde{i}m})$. The physical meaning for $u_{k\tilde{i}}$ is the maintenance operation plan for the elements in condition $\tilde{i}$. For each stage k, the algorithm requires a sequence of n minimizations, once over each of the agent controls, with the past controls determined by the rollout policy $\tilde{\pi}$, and the future controls determined by the base policy μ. The computation reduces to $O(\tilde{n}\times nm)$ per stage. The schematic illustrative of multiagent rollout algorithm is shown in Fig. 8.

Mathematically,

$$\tilde{\mu}_{k\tilde{i}}(x_k) \in \underset{u_{k1}\in U_{k\tilde{i}}(x_k)}{\arg\min} E\{g_k(x_k, \tilde{\mu}_{k1}(x_k), \dots, \tilde{\mu}_{k(\tilde{i}-1)}(x_k), u_{k\tilde{i}}, \mu_{k(\tilde{i}+1)}, \dots, \mu_{kn}, w_k) + J_{k+1}(f_k(x_k, \tilde{\mu}_{k1}(x_k), \dots, \tilde{\mu}_{k(\tilde{i}-1)}(x_k), u_{k\tilde{i}}, \mu_{k(\tilde{i}+1)}, \dots, \mu_{kn}, w_k))\} \tag{31}$$

The discrete available action set $U_k(x_k)$ for multiagent rollout is set to,

$$U_K(x_k) = (0, 0.05, 0.1, \dots, 0.95, 1) \tag{32}$$

The base policy is required to meet the constraints. Thus, we set the base policy as the replacement for all the components at each stage.

The implementation of PPO algorithm is conducted by OpenAI Baselines (Dhariwal et al., 2017) and the default settings in OpenAI Baselines are used. The state s is defined to include the system conditions and the reliability threshold, $s = (x_k^1, x_k^2, x_k^3, x_k^4, x_k^5, b)$. The action a indicates the maintenance decisions, and is defined to be $a \in \mathbb{R}^{n\times(m-1)}$ to ensure the sum of u_k^{is} over each condition i is 1. The reward function is set as follows,

$$r = \begin{cases} G - (x_k^5 - 0.05) - 3\times 10^8 & \text{if} \quad x_k^5 > b \\ G - (x_k^5 - 0.05) + 5000 & \text{otherwise} \end{cases} \tag{33}$$

The results of overall maintenance cost are shown in Fig. 9. The values in Fig. 9 are obtained by one single run respectively. Table 5

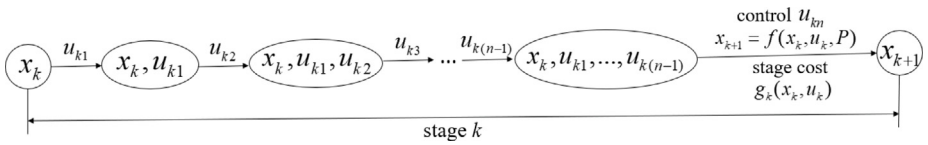


Fig. 8. Equivalent formulation of the optimization problem for the case where the control u_k consists of n parts (Bertsekas, 2019).

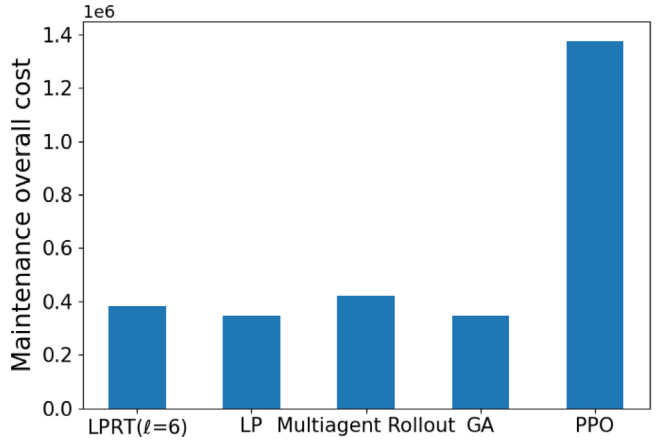


Fig. 9. Overall maintenance cost comparison.

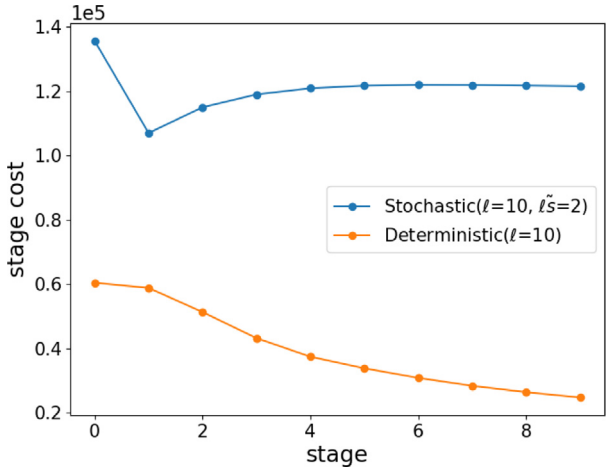


Fig. 10. An example of the stochastic maintenance scheduling.

Table 5
Comparison of computation time (s).

LPRT($\ell = 6$)	LP	Multiagent Rollout	GA	PPO
0.4	0.1	6.7×10^{-5}	179.4	537.0

Table 6
Comparison of relative optimality gap.

LPRT ($\ell = 6$)	LP	Multiagent Rollout	GA	PPO
11.02%	0	22.12%	0.64%	298.98%

shows the relative optimality gap, which is the absolute gap divided by the optimal result(obtained by LP). Additionally, the computation time of different method is shown in Table 6. The result by LPRT is close to the global optimum, which implies that LPRT is able to return a good result for maintenance scheduling. LP has the optimal solution and is more computationally efficient than LPRT, but LP cannot solve infinite horizon maintenance scheduling which LPRT can solve. The multiagent rollout method is the most computationally efficient method, but has a larger relative optimality gap than LPRT. The GA method has a better result than LPRT but needs a much longer computation time. PPO is a learning algorithm and therefore, needs a long training time. The result of PPO also relies on the user-defined reward function. Thus, PPO is less efficient to solve maintenance scheduling compared with LPRT, multiagent rollout or GA. Based on the above discussion, LPRT is the method that can balance the required maintenance cost and computational efficiency. Further, LPRT has the power to solve infinite horizon maintenance scheduling.

4.2. Stochastic maintenance

As an example of Section 3.2, we assume that element, p_{isj}, in the transition probability vector P ($j < n$ (except the elements in the last row)) follows a normal distribution $\tilde{P}$, ($P' \in \tilde{P}$, $\tilde{P} = N(P, \sigma)$). We sample three values of p_{isj}, denoted by (P_1, P_2, P_3), at one stage to demonstrate the uncertainties in maintenance operations. The elements in the last row are excluded to meet the requirement that the sum of each column should be 1. The mean of the normal distribution is set as the value in the deterministic case. Different p_{isj} are obtained from different normal distributions but under the same cumulative distribution function value. The cumulative distribution function values for choosing (P_1, P_2, P_3) are set to be $(0.45, 0.4, 0.3)$ in this numerical example for better comparison with deterministic results,

$$\begin{aligned} p(P' \le P_1) &= 0.45 \\ p(P' \le P_2) &= 0.4 \\ p(P' \le P_3) &= 0.3 \end{aligned} \tag{34}$$

The probability for p_{isj} taking P_i is calculated as the normalized cumulative distribution function values. For simplification, stochastic P in this numerical example is static and does not change with time. For further application, stochastic P at each stage can also be explicitly given, such as based on experimental data. As for predicting the next state x_{k+1}, we use the information of x_k, u_k, and P. In future applications, x_{k+1} can also be determined by a system inspection and health monitoring.

In this example, we set the subproblem to 10 stages involving 2 stochastic stages, and the coefficient of variation (COV) of the transition probability distribution is set to 0.02. The result is shown in Fig. 10. The information in blue is the result of the stochastic maintenance scheduling; the information in orange is the result of the deterministic maintenance scheduling with the same subproblem size for comparison. The costs when considering uncertainties are larger at all stages. The reason is that the cumulative distribution function values are all set to be smaller than 0.5, $p_{isj}(j \neq n)$ is smaller than the value in the deterministic case. Such setting results in a smaller probability of turning from condition i to $j(j < n)$. While the probability of turning to the worst condition n is larger. So maintenance performances in stochastic cases are worse than those set in the deterministic case, which leads to more maintenance operations and more budget.

Moreover, we study the effect of standard deviation, σ, on the overall maintenance cost, which is presented in Fig. 11. The coefficient of variation (COV) is chosen between 0 and 0.1. The subproblem size is set to 6-stage and 2 stochastic stages within each subproblem. Fig. 11 shows that the overall maintenance cost experiences continuous growth with the increase of standard deviation. The smallest cost occurs when

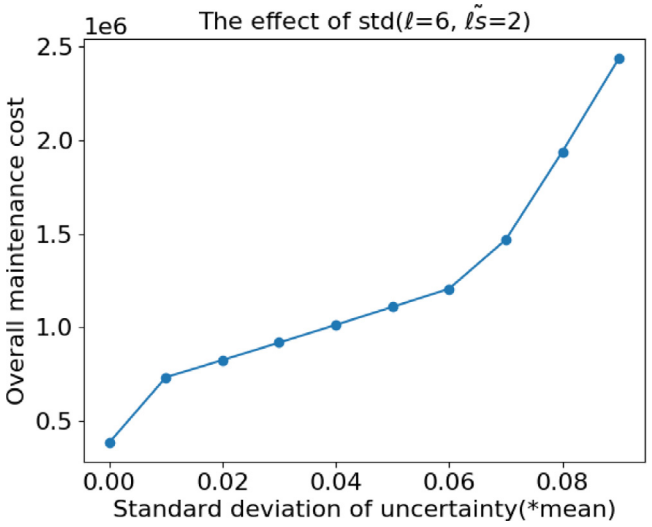


Fig. 11. The effect of standard deviation on overall maintenance cost.

σ is 0, which is the result of the deterministic case. The reason is the worse maintenance performance in stochastic maintenance scheduling. Furthermore, the growth rate of cost increases when σ is 0.06.

5. Discussion

This section shows the benefit of LPRT in infinite stochastic maintenance: reducing the dimension of control variables. The sensitivity analyses of the subproblem size effect for the deterministic maintenance scheduling and stochastic maintenance scheduling are conducted. In addition, the effect of the stochastic stage size is also analyzed.

5.1. Benefit of LPRT in the infinite stochastic maintenance

A standard stochastic programming model, such as the nonlinear programming formulation, converts the N-stage stochastic problem into a deterministic optimization problem of dimension that grows exponentially with the number of stages. In particular, for an N-stage problem, the number of control variables expands by a factor $\tilde{m}$ with each additional stage (Bertsekas, 2020). The total number of variables is bounded by

$$d(1+\tilde{m}+\tilde{m}^2+\cdots+\tilde{m}^{N-1}) \tag{35}$$

The dimension of the preceding nonlinear programming formulation can be very large, even for moderate values of N.

The benefit of LPRT in the stochastic maintenance problem is reducing the dimension of control variables. As mentioned in Section 3.2, the total number of control variables is reduced to,

$$d(1+\tilde{m}+\tilde{m}^2+\cdots+\tilde{m}^{\tilde{\ell}s-1}+\tilde{m}^{\tilde{\ell}s}\times(\ell-\tilde{\ell}s)) \tag{36}$$

The benefit of reducing the problem dimension comes from two aspects. Firstly, LPRT forms a related subproblem at each stage, which significantly reduces the problem size and makes the infinite maintenance scheduling solvable. Secondly, within the subproblem, the size of stochastic stages is flexible and can be further determined to balance the computational efficiency and accuracy.

5.2. Sensitivity analysis to the subproblem size ℓ

5.2.1. Effect of ℓ on the deterministic maintenance scheduling

As for the deterministic maintenance scheduling, the effect of ℓ on the overall maintenance cost is shown in Fig. 12. The problem setup is the same as the example in Section 4.1. We test ℓ from 1 to 10. The

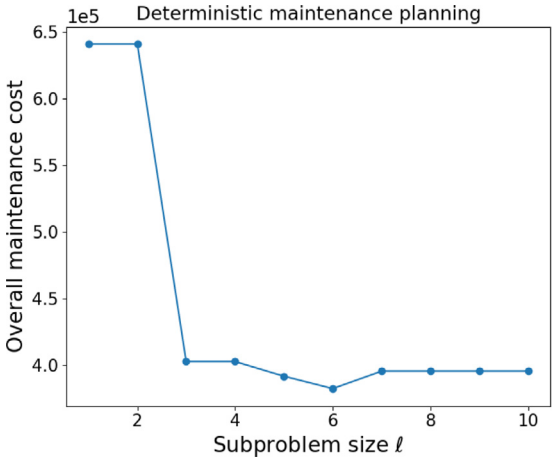


Fig. 12. The effect of ℓ on overall maintenance cost in the deterministic maintenance scheduling.

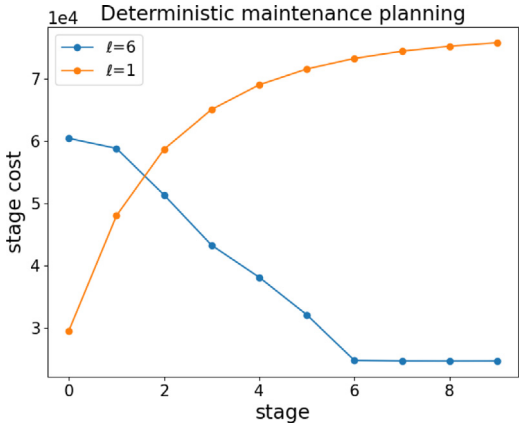


Fig. 13. The maintenance cost at each stage when ℓ is 1 and 6.

minimum overall cost is achieved when ℓ is 6. With a small ℓ, the overall cost is much higher. The overall cost converges as ℓ is larger than 6.

To investigate the reason, we plot the maintenance cost at each stage with ℓ of 1 and 6 in Fig. 13. The model with a large ℓ spends more at stage 0, but the system needs less maintenance cost in the later stage. On the contrary, the model with small ℓ needs less maintenance cost at the beginning, but demands more maintenance operations in the later stage. The phenomenon in Fig. 13 represents two maintenance strategies. Strategy envisioned with longer time windows (e.g., with large ℓ) will sacrifice maintenance cost at the beginning in order to convert the system to a better condition. Then the system needs much fewer maintenance operations in the future. Strategy focusing on the current and next stage only tends to save the money at the beginning, but may need very expensive replacement in the future. This example illustrates the benefit of long-term planning to save the entire maintenance cost. It should be noted that the conclusion also depends on the parameters for system degradation, maintenance method and associated cost, and transitional matrix. The proposed study provides a method to evaluate the maintenance strategies.

5.2.2. Effect of ℓ on the stochastic maintenance scheduling

As for the stochastic maintenance scheduling, Fig. 14 shows the effect of the subproblem size ℓ on the overall maintenance cost. The problem setup is the same as the example in Section 4.2. There is one stochastic stage within each subproblem and COV is set to 0.02. We test the subproblem size ℓ from 1 to 10. The cost keeps unchanged when

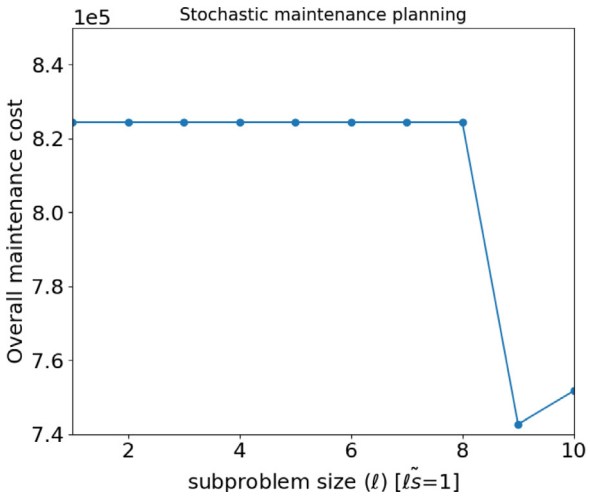


Fig. 14. The effect of ℓ on overall maintenance cost in the stochastic maintenance scheduling.

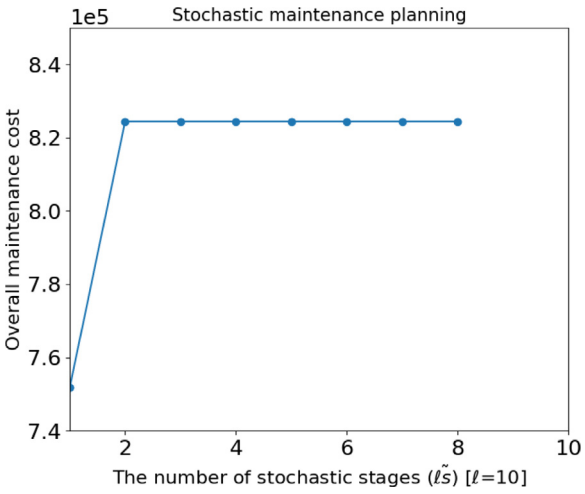


Fig. 16. The effect of $\tilde{\ell} s$ on overall maintenance cost.

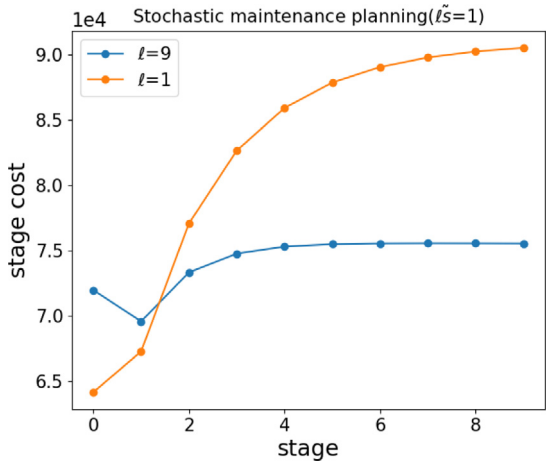


Fig. 15. The maintenance cost at each stage when ℓ is 1 and 9.

ℓ is from 1 to 8, followed by a decrease of about 10% at ℓ of 9 and a slow increase at ℓ of 10.

We plot the maintenance cost at each stage with ℓ of 1 and 9 in Fig. 15. The reason for the less cost with ℓ of 9 is similar to the analysis in Section 5.2.1. With a longer horizon, the system prefers to take costly maintenance operations in the first stage to get a reliable system. However, comparing Fig. 14 to Fig. 12, the uncertainties did show impact compared to the conclusion from the deterministic planning results. First, the uncertainties make the benefits of long-term planning delayed. For example, if only 7–8 stages are considered, no benefits of the cost-saving can be observed. The time window needs to be extended to 9–10 to see the cost-saving. This observation suggests that the decision-making for maintenance strategies needs to consider the impact of uncertainties. Patience is needed; when uncertainties exist, the expected investment return time will be longer. Another important observation is that the relative cost saving between long-term planning and short-term planning is reduced (i.e., from about 7e4 in deterministic planning (Fig. 13) to about 1.5e4 in stochastic planning (Fig. 15)). This observation suggests that the uncertainties require more maintenance actions to ensure the safety level and reduction of uncertainties will be beneficial, especially for long-term planning. A Certain amount of investment for techniques to reduce the uncertainties, such as health monitoring systems or non-destructive testing, can be justified from the maintenance cost savings.

5.3. Effect of the stochastic stage size $\tilde{\ell} s$ on the stochastic maintenance scheduling

Fig. 16 shows the effect of the size of the stochastic stages on the overall maintenance cost. The problem setup is the same as the example in Section 4.2. The subproblem has 10 stages and COV is set to 0.02. We test the number of stochastic stages from 1 to 8. The overall cost increases as $\tilde{\ell} s$ increases from 1 to 2 and then gets converged. The growth is because multistep rollout experiences more stages with worse maintenance performance. When $\tilde{\ell} s$ equals ℓ, the problem is solved exactly. Thus, we recommend $\tilde{\ell} s$ be carefully chosen since insufficient stochastic stages would lose essential uncertainties while too many stochastic stages would cause non-necessary computation.

6. Conclusion

In this work, a generic model for maintenance scheduling is introduced, in which the maintenance planning is formulated as an MDP problem and linear optimization problem. A novel LPRT method is proposed, which is based on LP and rollout. LPRT combines the advantages of two methods and is well-suited to deal with the maintenance scheduling problem with an infinite horizon and stochastic uncertainties. The proposed method is applied to both the deterministic case and the stochastic case. Results in the deterministic case show that LPRT has the capability to solve the constrained maintenance scheduling problem and deliver a good result compared to the results from several existing methods, such as LP, multiagent rollout, GA, and PPO. Sensitivity analysis is conducted in both the deterministic and stochastic cases, including the effects of uncertainty, subproblem size, and the number of stochastic stages.

The current formulation assumes the linear cost function or takes the linear approximation to use the LPRT algorithm. The performance for highly nonlinear functions is expected to be low and further investigation is needed for such cases. The comparison with many other reinforcement learning algorithms has not been done, especially for the case of stochastic maintenance scheduling and will need additional evaluation. The current maintenance scheduling problem uses the static transitional matrix and cost matrix. In practice, both of them can be functions of time and needs to be updated with the most recent information about the system and market. Extension to incorporate time-dependent maintenance information is another direction to be explored. Besides maintenance scheduling problems, LPRT also has the capacity to solve other engineering problems that can be formulated as an MDP and linear optimization problem. The application of LPRT to other problems is also worth further investigation.

CRediT authorship contribution statement

Jueming Hu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Visualization, Original writing. **Yuhao Wang:** Resources, Data curation, Software. **Yutian Pang:** Software, Visualization, Writing - review & editing. **Yongming Liu:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research reported in this paper was supported by funds from NASA University Leadership Initiative program (Contract No. NNX17AJ86 A, PI: Yongming Liu, Technical Officer: Anupa Bajwa). The support is gratefully acknowledged. The authors would like to thank Prof. Dimitri Bertsekas for helpful discussions.

References

Abiri-Jahromi, A., Fotuhi-Firuzabad, M., Abbasi, E., 2009. An efficient mixed-integer linear formulation for long-term overhead lines maintenance scheduling in power distribution systems. IEEE Trans. Power Deliv. 24 (4), 2043–2053.

Akgül, F., Frangopol, D.M., 2004. Computational platform for predicting lifetime system reliability profiles for different structure types in a network. J. Comput Civil Eng. 18 (2), 92–104.

Anuar, W.K., Lee, L.S., Seow, H.-V., Pickl, S., 2021. A multi-depot vehicle routing problem with stochastic road capacity and reduced two-stage stochastic integer linear programming models for rollout algorithm. Mathematics 9 (13), 1572.

Barata, J., Soares, C.G., Marseguerra, M., Zio, E., 2002. Simulation modelling of repairable multi-component deteriorating systems for 'on condition'maintenance optimisation. Reliab. Eng. Syst. Saf. 76 (3), 255–264.

Bellman, R., 1957. A Markovian decision process. Indiana Univ. Math. J. 6, 679–684.

Berger, C.R., Areta, J., Pattipati, K., Willett, P., 2008. Compressed sensing-a look beyond linear programming. In: 2008 IEEE International Conference On Acoustics, Speech And Signal Processing. IEEE, pp. 3857–3860.

Bertsekas, D., 2019. Multiagent rollout algorithms and reinforcement learning. arXiv preprint arXiv:1910.00120.

Bertsekas, D.P., 2020. Rollout, Policy Iteration, And Distributed Reinforcement Learning. Athena Scientific.

Bertsekas, D.P., Castanon, D.A., 1999. Rollout algorithms for stochastic scheduling problems. J. Heuristics 5 (1), 89–108.

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-Dynamic Programming. Athena Scientific.

Bertsekas, D.P., Tsitsiklis, J.N., Wu, C., 1997. Rollout algorithms for combinatorial optimization. J. Heuristics 3 (3), 245–262.

Bhattacharya, S., Badyal, S., Wheeler, T., Gil, S., Bertsekas, D., 2020. Reinforcement learning for POMDP: Partitioned rollout and policy iteration with application to autonomous sequential repair problems. IEEE Robot. Automa. Lett. 5 (3), 3967–3974.

Bruns, P., 2002. Optimal maintenance strategies for systems with partial repair options and without assuming bounded costs. Eur. J. Oper. Res. 139 (1), 146–165.

Chen, Y., He, F., Li, H., Zhang, D., Wu, Y., 2020. A full migration BBO algorithm with enhanced population quality bounds for multimodal biomedical image registration. Appl. Soft Comput. 93, 106335.

Chen, T., Popova, E., 2002. Maintenance policies with two-dimensional warranty. Reliab. Eng. Syst. Saf. 77 (1), 61–69.

Chiang, J.-H., Yuan, J., 2001. Optimal maintenance policy for a Markovian system under periodic inspection. Reliab. Eng. Syst. Saf. 71 (2), 165–172.

Chvatal, V., Chvatal, V., et al., 1983. Linear Programming. Macmillan.

Dekker, R., Scarf, P.A., 1998. On the impact of optimisation models in maintenance decision making: the state of the art. Reliab. Eng. Syst. Saf. 60 (2), 111–119.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., Zhokhov, P., 2017. Openai baselines. GitHub Repository, GitHub https://github.com/openai/baselines.

Ding, S.-H., Kamaruddin, S., 2015. Maintenance policy optimization—literature review and directions. Int. J. Adv. Manuf. Technol. 76 (5–8), 1263–1283.

Froger, A., Gendreau, M., Mendoza, J.E., Pinson, E., Rousseau, L.-M., 2018. Solving a wind turbine maintenance scheduling problem. J. Schedul. 21 (1), 53–76.

Fwa, T., Tan, C., Chan, W., 1994. Road-maintenance planning using genetic algorithms. II: Analysis. J. Trans. Eng. 120 (5), 710–722.

Gao, H., Zhang, X., 2013. A Markov-based road maintenance optimization model considering user costs. Comput.-Aided Civil Infrast. Eng. 28 (6), 451–464.

Garg, A., Deshmukh, S., 2006. Maintenance management: literature review and directions. J. Qual. Maint. Eng..

Giacomello, C., Kapelan, Z., Nicolini, M., 2013. Fast hybrid optimization method for effective pump scheduling. J. Water Res. Plann. Manag. 139 (2), 175–183.

Goel, H.D., Grievink, J., Weijnen, M.P., 2003. Integrated optimal reliable design, production, and maintenance planning for multipurpose process plants. Comput. Chem. Eng. 27 (11), 1543–1555.

Grall, A., Dieulle, L., Bérenguer, C., Roussignol, M., 2002. Continuous-time predictive-maintenance scheduling for a deteriorating system. IEEE Trans. Reliabil. 51 (2), 141–150.

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D., 2017. Pyomo-optimization modeling in python. vol. 67, Springer.

Howard, R.A., 1964. Dynamic programming and Markov processes. NEW YORK: JOHN-WILEY, http://dx.doi.org/10.1126/science.132.3428.667.

Hu, J., Liu, Y., 2020. UAS conflict resolution integrating a risk-based operational safety bound as airspace reservation with reinforcement learning. In: AIAA Scitech 2020 Forum. p. 1372.

Hu, J., Yang, X., Wang, W., Wei, P., Ying, L., Liu, Y., 2020. UAS conflict resolution in continuous action space using deep reinforcement learning. In: AIAA AVIATION 2020 FORUM. p. 2909.

Khan, F.I., Haddara, M.M., 2003. Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning. J. Loss Prev. Process Ind. 16 (6), 561–573.

Labib, A.W., O'Connor, R.F., Williams, G.B., 1998. An effective maintenance system using the analytic hierarchy process. Integ. Manuf. Syst..

Lam, Y., 1999. An optimal maintenance model for a combination of secondhand–new or outdated–updated system. Eur. J. Operat. Res. 119 (3), 739–752.

Lee, K., Shayman, M.A., 2005. Rollout algorithms for logical topology design and traffic grooming in multihop WDM networks. In: GLOBECOM'05. IEEE Global Telecommunications Conference, 2005. 4, IEEE, pp. 5–pp.

Li, H., He, F., Chen, Y., Pan, Y., 2021. MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. Memet. Comput. 13 (1), 1–18.

Liang, Y., He, F., Zeng, X., 2020. 3D mesh simplification with feature preservation based on whale optimization algorithm and differential evolution. Integr. Comput.-Aided Eng. (Preprint), 1–19.

Liu, M., Frangopol, D.M., 2006. Optimizing bridge network maintenance management under uncertainty with conflicting criteria: Life-cycle maintenance, failure, and user costs. J. Struct. Eng. 132 (11), 1835–1845.

Liu, Y., Goebel, K., 2018. Information fusion for national airspace system prognostics. In: PHM Society Conference. 10, (1).

Lu, R., Hong, S.H., Zhang, X., 2018. A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach. Appl. Energy 220, 220–230.

Makhorin, A., 2001. GLPK linear programming kit: Implementation of the revised simplex method. Moscow Aviation Institute, Moscow, Russia.

Marquez, A.C., Heguedas, A.S., 2002. Models for maintenance optimization: a study for repairable systems and finite time periods. Reliab. Eng. Syst. Saf. 75 (3), 367–377.

Martorell, S., Villanueva, J.F., Carlos, S., Nebot, Y., Sánchez, A., Pitarch, J.L., Serradell, V., 2005. RAMS+ C informed decision-making with application to multi-objective optimization of technical specifications and maintenance using genetic algorithms. Reliab. Eng. Syst. Saf. 87 (1), 65–75.

Meier-Hirmer, C., Riboulet, G., Sourget, F., Roussignol, M., 2009. Maintenance optimization for a system with a gamma deterioration process and intervention delay: application to track maintenance. Proc. Instit. Mech. Eng. Part O:J. Risk Reliabil. 223 (3), 189–198.

Mitchell, M., 1998. An Introduction to Genetic Algorithms. MIT Press.

Miyamoto, A., Kawamura, K., Nakamura, H., 2000. Bridge management system and maintenance optimization for existing bridges. Comput.-Aided Civil Infrast. Eng. 15 (1), 45–55.

Nielsen, J.J., Sørensen, J.D., 2011. On risk-based operation and maintenance of offshore wind turbine components. Reliab. Eng. Syst. Saf. 96 (1), 218–229.

Pan, E.-s., Liao, W.-z., Zhuo, M.-l., 2010. Periodic preventive maintenance policy with infinite time and limit of reliability based on health index. J. Shanghai Jiaotong Univ. (Science) 15 (2), 231–235.

Pang, Y., Liu, Y., 2020. Probabilistic aircraft trajectory prediction considering weather uncertainties using dropout as Bayesian approximate variational inference. In: AIAA Scitech 2020 Forum. p. 1413.

Pang, Y., Zhao, X., Yan, H., Liu, Y., 2021. Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach. Transp. Res. C 130, 103326. http://dx.doi.org/10.1016/j.trc.2021.103326, URL https://www.sciencedirect.com/science/article/pii/S0968090X21003314.

Puleo, V., Morley, M., Freni, G., Savić, D., 2014. Multi-stage linear programming optimization for pump scheduling. Procedia Eng. 70, 1378–1385.

Rocchetta, R., Bellani, L., Compare, M., Zio, E., Patelli, E., 2019. A reinforcement learning framework for optimal operation and maintenance of power grids. Appl. Energy 241, 291–301.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Tabassum, M., Mathew, K., et al., 2014. A genetic algorithm analysis towards optimization solutions. Int. J. Digital Inform. Wirel. Commun. (IJDIWC) 4 (1), 124–142.

The MathWorks, I., 2018. Global optimization toolbox: User's guide (r2018b).

Van Noortwijk, J.M., 1998. Optimal replacement decisions for structures under stochastic deterioration. In: Proceedings Of The Eighth IFIP WG. 7, (5), pp. 273–280.

Wang, Y., Liu, Y., 2019. An adaptive probabilistic maintenance framework for decision planning optimization. In: AIAA Scitech 2019 Forum. p. 0442.

White, D.J., 1993. A survey of applications of Markov decision processes. J. Opera. Res. Soc. 44 (11), 1073–1096. http://dx.doi.org/10.1057/jors.1993.181.

Xenos, D.P., Kopanos, G.M., Cicciotti, M., Thornhill, N.F., 2016. Operational optimization of networks of compressors considering condition-based maintenance. Comput. Chem. Eng. 84, 117–131.