Condition-based Maintenance for Multi-component Systems: Modeling, Structural Properties, and Algorithms

Zhicheng Zhu and Yisha Xiang

Texas Tech University, Lubbock, TX, 79409, USA

ABSTRACT

Condition-based maintenance (CBM) is an effective maintenance strategy to improve system performance while lowering operating and maintenance costs. Real-world systems typically consist of a large number of components with various interactions between components. However, existing studies on CBM focus on single-component systems. Multi-component condition-based maintenance, which joins the components' stochastic degradation processes and the combinatorial maintenance grouping problem, remains an open issue in the literature. In this paper, we study the CBM optimization problem for multi-component systems. We first develop a multi-stage stochastic integer model with the objective of minimizing the total maintenance cost over a finite planning horizon. We then investigate the structural properties of a two-stage model. Based on the structural properties, two efficient algorithms are designed to solve the two-stage model. Algorithm 1 solves the problem to its optimality and Algorithm 2 heuristically searches for high-quality solutions based on Algorithm 1. Our computational studies show that Algorithm 1 obtains optimal solutions in a reasonable amount of time and Algorithm 2 can find high-quality solutions quickly. The multi-stage problem is solved using a rolling horizon approach based on the algorithms for the two-stage problem.

KEYWORDS

Condition-based maintenance, multi-component systems, multi-stage stochastic integer programming, endogenous uncertainty

1. Introduction

Reliability is the central concern of many mission-critical systems, such as aerospace systems, electric power systems, and nuclear systems. Investigations show that many accidents are caused by equipment failures, which were attributed to the lack of effective maintenance methods. For example, the space shuttle Challenger accident [1] and the Deepwater Horizon drilling rig explosion [2] occur in part because of inadequate maintenance. As the complexity of modern engineering systems increases, it is imperative to develop cost-effective maintenance plans for complex systems.

Maintenance strategies can be generally classified into two categories: time-based maintenance (TBM) and condition-based maintenance (CBM). The literature on TBM and CBM for single-component systems is abundant [3, 4]. However, much less attention has been paid on multi-component systems. Existing studies on multi-component systems are mainly time-based [5–7]. Despite the fact that CBM can be more cost-effective compared to TBM [4, 8, 9], CBM for multi-component systems is underexplored.

A multi-component system is usually subject to various interactions among components, such as stochastic dependence, structural dependence, and economic dependence [5, 7]. Stochastic dependence means the state of one component influences the lifetime distributions of other components. Structural dependence applies if components structurally form a part, so that maintenance of a failed component implies maintenance of other components as well. Economic dependence occurs if any maintenance action incurs a fixed system-dependent cost, often referred to as setup cost, due to mobilizing repair crew, disassembling machines, and downtime loss [10]. This setup cost can be significant in many capital-intensive industries. For example, the production losses during the shutdown ranges from \$500 to \$100,000 per hour in a chemical plant and millions of dollars per day in offshore drilling refineries [11]. Therefore, significant cost savings can be achieved by maintaining multiple components jointly instead of separately.

In this paper, we study the CBM optimization problem for multi-component systems with economic dependence over a finite planning horizon using a stochastic programming approach. The objective is to minimize total maintenance cost by selecting components for maintenance at each decision period. This problem is challenging because it joins the components' stochastic degradation processes and the combinatorial maintenance grouping problem [5, 12, 13]. In addition, the component state transition probability depends on the maintenance decision, making the problem decision-dependent, which is different from the standard approach to formulating stochastic programs based on the assumption that the stochastic process is independent of the optimization decisions. This endogenous uncertainty can make the stochastic programs more computationally challenging. There is a lack of general methods to efficiently solve this type of problem. Existing studies on multi-component maintenance planning often use simplified assumptions [9, 14, 15] or resort to simulation methods [16,17] to reduce mathematical difficulties in modeling and solving this problem.

We develop a general multi-stage stochastic maintenance model and do not restrict any grouping opportunities. Due to the complexity of the multi-stage stochastic maintenance model with integer decision variables, we first consider a two-stage model and investigate its structural properties. Based on the structural properties, we design two efficient algorithms to solve the two-stage model. The multi-stage model is solved using a rolling horizon approach based on the algorithms for the two-stage model. The main contribution of this paper is threefold.

(1)Develop an analytical CBM model for multi-component systems using a stochastic programming approach. This model is among the very first efforts that provide analytical expressions for the cost function and maintenance decisions of multi-component CBM. The proposed model is general with no restrictions for grouping as opposed to exiting one that only allow grouping at PM or CM.

(2)Establish structural properties for the two-stage model. These theoretical properties provide the conditions and search directions of improving any feasible solution, and lead to significant reduction of the search space of the problem.

(3) Design efficient algorithms to find high-quality solutions. We develop algorithms for the two-stage problem based on its structural properties, which are then implemented on a rolling-horizon to solve the multi-stage problem. Computational

studies show that our algorithms can provide satisfactory solutions within a reasonable amount of time, particularly for large-scale problems.

The remainder of this paper is organized as follows. Section 2 reviews the related studies on multi-component maintenance and stochastic programming methods. In Section 3, we develop a CBM model for multi-component systems over multiple decision periods. Section 4 investigates structural properties for the two-stage model. In Section 5, we design two algorithms for the two-stage problem and use rolling horizon technique to approximate the multi-stage model. Computational studies are presented in Section 6. Section 7 concludes this research and discusses the future research directions.

2. Literature review

We model the CBM optimization problem for multi-component systems using stochastic programming. We first examine the existing literature on multi-component maintenance, and then review the solution techniques for stochastic programming.

2.1. Multi-component maintenance

Most studies on multi-component maintenance are focused on TBM, which can be further divided into direct-grouping [18, 19] and indirect-grouping approaches [20–23] . Direct-grouping approach partitions the components into several fixed groups and always maintains the components in a group jointly. By using this approach, the problem becomes a set-partitioning problem, which is NP-complete. Indirect-grouping groups preventive maintenance (PM) activities by making the PM interval a multiple of a basis interval, so that the maintenance of different components can coincide [20, 21], or performs major PM on all component jointly at the end of a common interval and allows minor or major PM within this interval [22, 23]. Unlike the fixed structure under direct-grouping, there is no fixed group structure under indirect-grouping. Some researchers formulate an indirect-grouping model as a mixed integer programming (MIP) problem [22–24]. Because of the simplified policy structure, the MIP model can be separated by components, which greatly reduces the computational complexity.

However, both direct- and indirect-grouping approaches only group PM activities and ignore the grouping opportunities provided by CM. Patriksson et al. [25] use stochastic programming to model a time-based multi-component replacement problem. However, CM and PM have the same cost and are not distinguished in their paper.

Much less attention has been paid to CBM for multi-component systems [4]. Opportunistic maintenance (OM) has been considered for multi-component CBM [14, 15, 26]. OM takes advantage of CM by performing PM on functioning components when any failure happens. Castanier et al. [15] consider both PM and CM as opportunities for maintaining other functioning components and formulate the problem as a semi-regenerative process. However, they only consider a two-component system because of the exponential growth of problem size. Some studies have used Markov decision process (MDP) to solve the multi-component maintenance optimization problem. However, due to the state space grows exponentially as the number of components and/or the number of states increase, this method is limited to small scale problems. For example, Jia [27] models the OM problem as an MDP and investigate the structural property of the optimal policy. A two-component system is studied in [27]. Several studies use simulation methods to find optimal opportunistic CBM policies [16, 17], which also suffers from curse of dimensionality.

Proportional hazard model (PHM) incorporates both event data and CM data by modeling the lifetime of a component as a hazard rate process [4, 28]. Tian et al. extends the PHM from single-component CBM to multi-component CBM [9]. They study two practical cases with systems of two components and three components.

2.2. Stochastic programming

Various methods and techniques have been developed to solve a stochastic programming problem. For a two-stage stochastic linear program, Benders decomposition [29, 30] and progressive hedging algorithm (PHA) [31, 32] are two major decomposition methods. Benders decomposition is a vertical decomposition approach that decomposes the problem into a master problem that consists of the first-stage decisions and the subproblems that consist of second-stage decisions of all scenarios. PHA is a horizontal decomposition approach that decomposes the problem by scenarios.

It first independently solves all subproblems at each iteration and then forces the non-anticipatively constraints converge.

Multi-stage stochastic programming extends two-stage stochastic programming by allowing revised decisions at each stage based on uncertainty realizations observed so far [33]. For a multi-stage stochastic linear program, nested Benders decomposition [34, 35] that extended from Benders decomposition and PHA are also two common solution approaches. However, because the size of the scenario tree grows exponentially as the number of stages increases, both approaches are computationally intractable. Stochastic dual dynamic programming (SDDP) [36, 37] overcomes the exploding scenario tree size problem in nested Benders decomposition by combining scenario tree nodes. The drawback of the SDDP approach is that it relies on special problem structure such as stage-wise independence [38]. Rolling horizon provides a heuristic approach to approximating a multi-stage stochastic program by solving the two-stage problem on a rolling basis and utilizing the first-stage solution [39–41]. This approximation approach requires the two-stage problem to be computationally tractable. Recently, rule-based method has attracted some interests in addressing the intractability issue in multi-stage stochastic programming [38, 42–44]. This method restricts the solution to have some specific function forms, such as linear [38], piecewise linear [44], and polynomial [43]. Because the optimal decision rules of arbitrary multi-stage stochastic programs do not have general forms, rule-based methods cannot guarantee the solution quality in general [42].

A stochastic integer program further combines the difficulty of stochastic programming and integer programming and is challenging to solve. Nested Benders decomposition and SDDP that utilize Benders cuts become prohibited to this problem because strong duality does not hold due to integrality constraints. Moreover, PHA does not perform well for this problem in general because the non-anciticipativity constraints may converge slowly due to the integrality constraints and the intractability of solving each integer subproblem. Integer L-shaped method is another approach in solving stochastic integer program by using integer L-shaped cuts within the Benders decomposition framework [29,45]. However, this method is typically inefficient because it needs to generate an integer L-shaped cut for every feasible solution in the worst case scenario.

Stochastic programming with endogenous uncertainty draws some attentions recently because this type of uncertainty presents in a large number of applications [46–48]. Endogenous uncertainty implies that the underlying stochastic process is influenced by the decisions. Therefore, the probabilities of scenarios are decision-dependent and usually nonlinear [47, 48]. There is a lack of efficient method to solve this type of problem.

Our review shows that there is no general method to solve the proposed multi-stage stochastic maintenance model with integer decision variables and endogenous uncertainty. Efficient algorithms are needed to find high-quality solutions.

3. Model development

Notation.

$n:$ number of components

$\mathcal{N}:$ component set, $\mathcal{N} = \{1, 2, ..., n\}$

$T:$ number of decision stages

$\mathcal{T}:$ decision-stage set, $\mathcal{T} = \{1, 2, ..., T\}$

$\Omega_t:$ node set at stage $t \in \mathcal{T}$

$\omega_t:$ index of node at stage $t \in \mathcal{T}$, i.e., $\omega_t \in \Omega_t$

$a(\omega_t):$ ancestor node of $\omega_t \in \Omega_t$, $t \in \mathcal{T} \backslash \{1\}$

$\Omega(\omega_t):$ child nodes of $\omega_t \in \Omega_t$, $t \in \mathcal{T} \backslash \{T\}$

$g_{it}:$ state of component i at stage t

$g_{it}^{\omega_t}:$ state of component i in scenario $\omega_t \in \Omega_t$ at stage $t \in \mathcal{T}$

$Q_i(g, g'):$ state transition probability from state g to g' for component $i \in \mathcal{N}$

$c_{i,\text{pm}}:$ PM cost of component i

$c_{i,\text{cm}}:$ CM cost of component i

$c_{\text{s}}:$ setup cost

$\tilde{x}_{it}:$ maintenance decision of component i at stage t without considering economic dependence

$\tilde{x}_{it}^{*}:$ optimal maintenance decision of component $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$ without considering economic dependence

x_{it} : equals to 1 if any maintenance is performed on component $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$ and 0 otherwise

$x_{it}^{\omega_t}$: x_{it} in scenario $\omega_t \in \Omega_t$

x_t : vector of x_{it} for all $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$: $(x_{1,t}, x_{2,t}, ..., x_{n,t})$

$x_t^{\omega_t}$: vector of $x_{it}^{\omega_t}$ for all $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$ in scenario $\omega_t \in \Omega_t$: $(x_{1,t}^{\omega_t}, x_{2,t}^{\omega_t}, ..., x_{n,t}^{\omega_t})$

y_{it} : equals to 1 if CM is performed on component $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$ and 0 otherwise

$y_{it}^{\omega_t}$: y_{it} in scenario $\omega_t \in \Omega_t$

z_t : equals to 1 when any maintenance is performed at stage $t \in \mathcal{T}$ and 0 otherwise

$z_t^{\omega_t}$: z_t in scenario $\omega_t \in \Omega_t$

N_0 : do-nothing set at the first stage, $N_0 = \{i | x_{i,1} = 0, i \in \mathcal{N}\}$

N_1 : maintenanccе set at the first stage, $N_1 = \{i | x_{i,1} = 1, i \in \mathcal{N}\}$

We consider condition-based maintenance optimization for multi-component systems. The system consists of multiple components with economic dependence. Significant cost savings can be achieved by maintaining multiple components jointly rather than separately. We focus on systems with hidden failure which can only be revealed through inspection. For example, a production system may have failed but still operates, producing non-conforming products, and the failure can only be detected by inspection [49]. We assume components deteriorate independently. Such an assumption is common for systems where components are not subject to common cause failures or the deterioration dependence among components is weak [9, 17–19]. Each component has $\{1, 2, ..., m-1, m\}$ condition states, where a larger state represents a worse yet functioning condition and state m is the failure state. All components are subject to stochastic degradation. Without maintenance intervention, the condition of a component cannot return to a better state. Inspection is performed periodically on the system to reveal the states of all components and each inspection is a decision stage. In some real-world problems, an inspection schedule is already in place based on experiences or required by regulations. For example, many refinery and chemical plants conduct annual or biannual turnarounds during which they inspect their equipment. In cases where the interval length needs to be determined, an optimal interval length can be determined using decomposition methods [6, 19] or numerical search methods. At each decision stage, all failed components need to be correctively maintained and all functioning ones can be preventively maintained if desired. Both CM and PM restore a component to an as-good-as-new state, i.e., state 1.

This maintenance optimization problem is naturally a multi-stage stochastic integer program. At each stage $t \in \mathcal{T}$, we first observe all components' states g_{it}, $\forall i \in \mathcal{N}$. We then decide whether a component needs to be maintained $(x_{it}, i \in \mathcal{N}, t \in \mathcal{T})$. All failed components i are correctively maintained $(y_{it} = 1, i \in \mathcal{N}, t \in \mathcal{T})$. If there is any maintenance performed at stage $t \in \mathcal{T}$, the setup cost is incurred $(z_t = 1)$.

We illustrate the decision process using a scenario tree in Figure 1. In the scenario tree, we need to make maintenance decisions at each node $\omega_t \in \Omega_t, t \in \mathcal{T}$. Each node ω_t is characterized by a combination of all components' states, i.e., $(g_{1,t}^{\omega_t}, g_{2,t}^{\omega_t}, ..., g_{n,t}^{\omega_t})$, and Ω_t is the set of all nodes at stage $t \in \mathcal{T}$. For each node ω_t, $t \in \mathcal{T}\backslash\{T\}$, it has a set of child nodes $\Omega(\omega_t)$ at stage $t+1$, where $\Omega(\omega_t)$ collects all possible combinations of all components' states. For each node ω_t, $t \in \mathcal{T}\backslash\{1\}$, it has a unique ancestor node $a(\omega_t)$ at stage $t-1$. A node path from the root node (ω_1) to a last stage node $(\omega_T \in \Omega_T)$ is referred to as a scenario. The total number of scenarios is $|\Omega_T| = m^{n(T-1)}$, which grows exponentially as the number of components and/or stages increase.

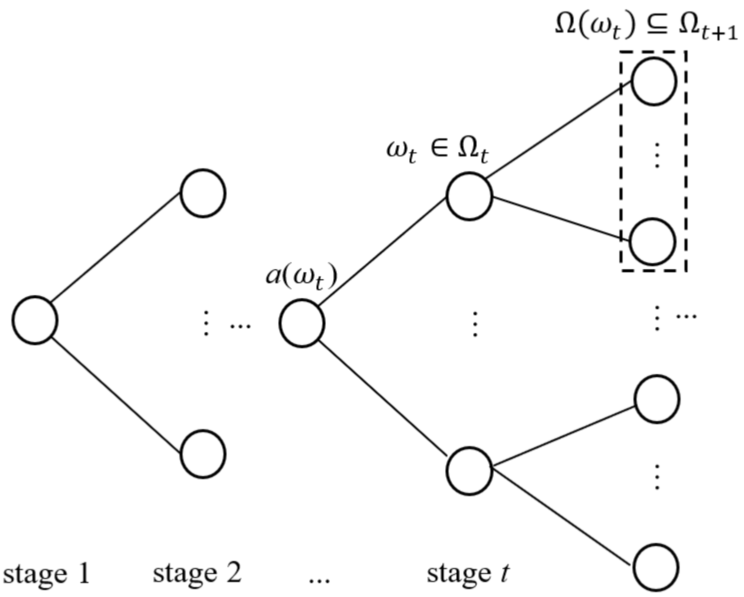

Figure 1.: Scenario tree

Our objective is to minimize the total cost over the planning horizon $\mathcal{T}$, where the total cost includes the first-stage cost and the expected second-stage cost of all nodes $\omega_2 \in \Omega_2$. For the cost at each node $\omega_t \in \Omega_t$ at stage $t \in \mathcal{T} \backslash \{T\}$, it consists of current-node cost and the expected cost of all child nodes $\Omega(\omega_t)$. For last stage nodes, i.e., $t = T$, the cost concerns the current-node cost only.

Given the component states $g_{it}^{\omega_t}$ for all components $i \in \mathcal{N}$ at node $\omega_t \in \Omega_t$ in stage $t \in \mathcal{T} \backslash \{T\}$, the probability from node ω_t to its child node ω_{t+1} depends on the maintenance decision $x_t^{\omega_t}$. For example, if the prior-maintenance state of a component in node ω_t is g, the post-maintenance transition probability is $Q(1, g')$ $(g' \in G)$ and $Q(g, g')$ $(g' \geq g)$ otherwise, and this leads to different node transition probabilities.

Denote $p(\omega_{t+1}|x_t^{\omega_t})$ as the probability from node $\omega_t, t \in \mathcal{T} \backslash \{T\}$ to its child node ω_{t+1} given decision $x_t^{\omega_t}$, and $Q_i(g, g')$ as the state transition probability from state g to g' for component $i \in \mathcal{N}$. Since components deteriorate independently, we have

$$p(\omega_{t+1}|x_t^{\omega_t}) = \prod_{i \in \mathcal{N}} [Q_i(g_{it}^{\omega_t}, g_{i,t+1}^{\omega_{t+1}})(1 - x_{it}^{\omega_t}) + Q_i(1, g_{i,t+1}^{\omega_{t+1}}) x_{it}^{\omega_t}].$$

Next, we develop the multi-stage stochastic model:

Decision variables $(i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T})$:

$x_{it}^{\omega_t}$: 1 if component i is maintained at node ω_t in stage t, and 0 otherwise.

$y_{it}^{\omega_t}$: 1 if component i is correctively maintained at node ω_t in stage t, and 0 otherwise.

$z_t^{\omega_t}$: 1 if there is any maintenance occurs at node ω_t in stage t, and 0 otherwise.

Multi-stage stochastic model (P1):

$$V_1 = \min_{x,y,z} \sum_{i \in \mathcal{N}} c_{i,\mathrm{pm}} x_{i,1} + \sum_{i \in \mathcal{N}} (c_{i,\mathrm{cm}} - c_{i,\mathrm{pm}}) y_{i,1} + c_{\mathrm{s}} z_1 + \sum_{\omega_2 \in \Omega_2} p(\omega_2|x_1) V_2(\omega_2) \tag{1}$$

s.t.

$$V_t(\omega_t) = \begin{cases} \begin{array}{l} \min\limits_{x,y,z} \sum\limits_{i \in \mathcal{N}} c_{i,\mathrm{pm}} x_{it}^{\omega_t} + \sum\limits_{i \in \mathcal{N}} (c_{i,\mathrm{cm}} - c_{i,\mathrm{pm}}) y_{it}^{\omega_t} \\ \quad + c_{\mathrm{s}} z_t^{\omega_t} + \sum\limits_{\omega_{t+1} \in \Omega(\omega_t)} p(\omega_{t+1}|x_t^{\omega_t}) V_{t+1}(\omega_{t+1}) \end{array} & , i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T} \backslash \{T\}, \\ \\ \begin{array}{l} \min\limits_{x,y,z} \sum\limits_{i \in \mathcal{N}} c_{i,\mathrm{pm}} x_{it}^{\omega_t} + \sum\limits_{i \in \mathcal{N}} (c_{i,\mathrm{cm}} - c_{i,\mathrm{pm}}) y_{it}^{\omega_t} \\ \quad + c_{\mathrm{s}} z_t^{\omega_t} \end{array} & , i \in \mathcal{N}, \omega_t \in \Omega_t, t = T \end{cases} \tag{2}$$

$$x_{it}^{\omega_t} \le z_t^{\omega_t}, i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T} \tag{3}$$

$$g_{it}^{\omega_t}(1 - y_{it}^{\omega_t}) \le m - 1, i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T} \tag{4}$$

$$y_{it}^{\omega_t} \le x_{it}^{\omega_t}, i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T} \tag{5}$$

$$x_{it}^{\omega_t}, y_{it}^{\omega_t} \in \{0, 1\}, i \in \mathcal{N}, \omega_t \in \Omega_t, t \in \mathcal{T} \tag{6}$$

$$z_t^{\omega_t} \in \{0, 1\}, \omega_t \in \Omega_t, t \in \mathcal{T} \tag{7}$$

Objective function (1) consists of the total cost at the first stage and the expected total cost at the second stage. The objective function $V_t(\omega_t)$ for node $\omega_t \in \Omega_t$ at stage $t \in \mathcal{T}$ is given by constraint (2). Constraints (3) ensure setup cost is incurred whenever a maintenance action is performed. Constraints (4) force CM actions on all failed components. Constraints (5) guarantee that the indicator of maintenance action ($x_{it}^{\omega_t}$) is set to 1 when CM is performed. Constraints (6) and (7) are integrality constraints for all decision variables.

As illustrated in Figure 1, the problem size of P1 grows exponentially as the number of components increases. As discussed in the literature review, there is no general method to solve this problem due to the lack of structural properties in multi-stage stochastic integer programs with endogenous uncertainty. Therefore, we first consider a two-stage problem.

The two-stage problem can be simplified by eliminating the second stage because the closed-form solutions for all second-stage subproblems can be obtained. Note that for the ease of notation, we drop the subscripts of ω_2, Ω_2 and V_2 in the two-stage model. First, for any subproblem $\omega \in \Omega$, the objective function $V(\omega)$ and all constraints are independent of the first-stage decisions and only depends on the components' states in scenario ω. Because the second stage is the last stage of the two-stage problem, to minimize any subproblem, it is obvious that we only need to correctively maintain all failed components to satisfy constraints (4) and do nothing on functioning components. Therefore, the optimal solutions in the second-stage subproblems are

$$(y_{i,2}^{\omega})^* = \lfloor \frac{g_{i,2}^{\omega}}{m} \rfloor, i \in \mathcal{N}, \omega \in \Omega, \tag{8}$$

$$(x_{i,2}^{\omega})^* = (y_{i,2}^{\omega})^*, i \in \mathcal{N}, \omega \in \Omega \tag{9}$$

and

$$(z_2^{\omega})^* = \min(1, \sum_{i\in\mathcal{N}}(x_{i,2}^{\omega})^*), \omega \in \Omega. \tag{10}$$

Based on Equations (8) to (10), the two-stage model (P2) is described as follows:

Two-stage stochastic model (P2):

$$\min \sum_{i\in\mathcal{N}} c_{i,\mathrm{pm}} x_{i,1} + \sum_{i\in\mathcal{N}}(c_{i,\mathrm{cm}} - c_{i,\mathrm{pm}})y_{i,1} + c_{\mathrm{s}} z_1 + V \tag{11}$$

s.t.

$$x_{i,1} \leq z_1, i \in \mathcal{N} \tag{12}$$

$$g_{i,1}(1 - y_{i,1}) \leq m - 1, i \in \mathcal{N} \tag{13}$$

$$y_{i,1} \leq x_{i,1}, i \in \mathcal{N} \tag{14}$$

$$x_{i,1}, y_{i,1} \in \{0, 1\}, i \in \mathcal{N} \tag{15}$$

$$z_1 \in \{0, 1\} \tag{16}$$

where

$$\begin{aligned} V = \sum_{\omega\in\Omega} p(\omega|x_1)V(\omega) = &\sum_{i\in\mathcal{N}}(Q_i(g_{i,1}, m)(1 - x_{i,1}) + Q_i(1, m)x_{i,1}))c_{i,\mathrm{cm}} \\ &+ (1 - \prod_{i\in\mathcal{N}}(1 - Q_i(g_{i,1}, m)(1 - x_{i,1}) - Q_i(1, m)x_{i,1}))c_{\mathrm{s}}. \end{aligned} \tag{17}$$

The two-stage model can be directly used in mission-critical applications where successfully completing a mission (one-period) is the primary concern.

4. Structural properties of the two-stage model

In this section, we establish three structural properties for P2. The first property provides an optimal solution to P2 based on the optimal solution without considering economic dependence. Because the optimal solution without considering economic dependence can be obtained easily, we can quickly identify the optimal solution to P2 when the condition in Proposition 1 is satisfied. The second property establishes the condition when changing the decision(s) of certain component(s) from do-nothing to PM reduces the total maintenance cost. The third property establishes the condition when changing the decision(s) of certain component(s) from PM to do-nothing reduces the total maintenance cost. Propositions 2 and 3 are the theoretical foundation of Algorithm 1 that solves P2 optimally.

Proposition 1. *If $\tilde{x}^*_{i,1} = 1\ \forall i \in \mathcal{N}$, then $x^*_{i,1} = 1\ \forall i \in \mathcal{N}$.* □

Proof. See Appendix A.1. □

Proposition 1 shows that it is optimal to maintain all components (i.e. $x^*_{i,1} = 1$) if all components need to be maintained when ignoring economic dependence (i.e. $\tilde{x}^*_{i,1} = 1$). The optimal maintenance decision without considering economic dependence $\tilde{x}^*_{i,1}$ of component $i \in \mathcal{N}$ can be obtained easily as follows:

$$\tilde{x}^*_{i,1} = \begin{cases} 1, & \text{when } Q_i(g_{i,1}, m) > \frac{c_{i,\mathrm{pm}+c_\mathrm{s}}}{c_{i,\mathrm{cm}+c_\mathrm{s}}} + Q_i(1, m) \text{ or } g_{i,1} = m, \\ \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

Proposition 1 leads to an optimal solution to P2 when $\tilde{x}^*_{i,1} = 1$ for all components $i \in \mathcal{N}$. However, the condition of $\tilde{x}^*_{i,1} = 1$ for all $i \in \mathcal{N}$ is a special scenario. Next, we explore more general structural properties of the two-stage model.

Definition 1. A partition (N_0, N_1) of set $\mathcal{N}$, i.e., $N_0 \cup N_1 = \mathcal{N}$ and $N_0 \cap N_1 = \emptyset$, is a solution to P2, where N_0 is the do-nothing set that collects all components that are not maintained at the first stage, i.e., $N_0 = \{i | x_{i,1} = 0, i \in \mathcal{N}\}$ and N_1 is the maintenancce set that includes all components that are maintained at the first stage, i.e., $N_1 = \{i | x_{i,1} = 1, i \in \mathcal{N}\}$. □

A partition (N_0, N_1) of $\mathcal{N}$ is *feasible* if every failed component at the first stage belongs to N_1. Therefore, determining the optimal x^*_1 is now equivalent to find out a feasible and optimal partition (N^*_0, N^*_1) of $\mathcal{N}$ that minimizes the total cost. Next, we give two propositions regarding how to improve a feasible partition (N_0, N_1).

Proposition 2. *Consider two feasible partitions (N_0, N_1) and (N'_0, N'_1) of $\mathcal{N}$. Let C and C' be their respective total costs. If $N'_1 \backslash N_1 = N$ and $N \neq \emptyset$, we have $C' < C$ if and only if $\Delta_r(N_0, N_1, N) < 1$, where*

$$\Delta_r(N_0, N_1, N) = \begin{cases} \frac{\sum_{k \in N} \rho_k}{r_N} \times \frac{1}{p(N_0, N_1)}, & \text{when } N_1 \neq \emptyset, \\ \frac{1 + \sum_{k \in N} \rho_k}{r_N} \times \frac{1}{p(N_0, N_1)}, & \text{otherwise.} \end{cases}$$

$$r_N = \prod_{i \in N} \frac{1 - Q_i(1, m)}{1 - Q_i(g_{i,1}, m)} - 1,$$

$$\rho_k = \frac{c_{k,pm} - (Q_k(g_{k,1}, m) - Q_k(1, m))c_{k,cm}}{c_s},$$

and

$$p(N_0, N_1) = \prod_{i \in N_0} (1 - Q_i(g_{i,1}, m)) \prod_{i \in N_1} (1 - Q_i(1, m)) \quad \square$$

Proof. See Appendix A.2. □

Proposition 2 helps to quickly identify a set $N \subseteq N_0$ to improve the current partition (N_0, N_1) by moving set N from the do-nothing set to the maintenance set. $\Delta_{\mathrm{r}}(N_0, N_1, N)$ consists of two parts: $\dfrac{\sum_{k \in N} \rho_k}{r_N}$ and $p(N_0, N_1)$. The first part is determined by the components in set N and the second part is the probability that all components will survive in the second stage given the current decision partition, i.e., (N_0, N_1). The probability $p(N_0, N_1)$ increases as more components are maintained.

Let us first examine the condition in Proposition 2 when $|N| = 1$. Suppose $N = \{k\}$, $k \in N_0$, Proposition 2 provides the condition of improving the current partition by maintaining component k. When $N_1 \neq \emptyset$, we have

$$\begin{aligned}\Delta_{\mathrm{r}}(N_0, N_1, \{k\}) =& \frac{(c_{k,\mathrm{pm}} - (Q_k(g_{k,1}, m) - Q_k(1, m))c_{k,\mathrm{cm}})(1 - Q_k(g_{k,1}, m))}{c_{\mathrm{s}}(Q_k(g_{k,1}, m) - Q_k(1, m))p(N_0, N_1)} \\ \approx& \frac{(c_{k,\mathrm{pm}} - Q_k(g_{k,1}, m)c_{k,\mathrm{cm}})(1 - Q_k(g_{k,1}, m))}{c_{\mathrm{s}}Q_k(g_{k,1}, m)p(N_0, N_1)}\end{aligned} \tag{19}$$

because $Q_k(1, m)$ is the state transition probability from the perfect state to the failed state and therefore $Q_k(1, m) \approx 0$ holds in many scenarios (e.g., when inspection interval is not long). Based on Equation (19), we have several important observations: (1) $\Delta_{\mathrm{r}}(N_0, N_1, \{k\})$ increases as $c_{k,\mathrm{pm}}$ increases. The increase in Δ_{r} indicates that it is less likely that we change the decision on k from no maintenance to PM. This is because a higher PM cost makes it less cost-effective to perform PM at the first stage. (2) $\Delta_{\mathrm{r}}(N_0, N_1, \{k\})$ increases as $c_{k,\mathrm{cm}}$ decreases. It is less incentive to perform PM on component k with other components at the first stage when CM cost of component k is lower. (3) $\Delta_{\mathrm{r}}(N_0, N_1, \{k\})$ increases as c_{s} decreases, because the decrease of setup cost makes sharing setup cost at the first stage less cost-effective. And (4) $\Delta_{\mathrm{r}}(N_0, N_1, \{k\})$

increases as $g_{k,1}$ decreases, meaning a better component condition makes it less worthy to maintain the component at the first stage. Similar patterns can be observed when $|N| \geq 2$. Proposition 2 also provides the maintenance action for a new component. Specifically, when $g_{k,1} = 1$, $k \in N_0$, we have $\Delta_{\mathrm{r}}(N_0, N_1, \{k\}) = +\infty > 1$, which implies we should never maintain a new component.

Proposition 3. *Consider two feasible partitions* (N_0, N_1) *and* (N_0', N_1') *of* $\mathcal{N}$*. Let* C *and* C' *be their respective total costs. If* $N_0' \backslash N_0 = N$ *and* $N \neq \emptyset$*, we have* $C' < C$ *if and only if* $\Delta_s(N_0, N_1, N) > 1$*, where*

$$\Delta_s(N_0, N_1, N) = \begin{cases} \dfrac{\sum_{k \in N} \rho_k}{s_N} \times \dfrac{1}{p(N_0, N_1)}, & \text{when } N_1 \neq \emptyset, \\ \dfrac{1 + \sum_{k \in N} \rho_k}{s_N} \times \dfrac{1}{p(N_0, N_1)}, & \text{otherwise.} \end{cases}$$

$$s_N = 1 - \prod_{i \in N} \frac{1 - Q_i(g_{i,1}, m)}{1 - Q_i(1, m)},$$

and values ρ_k *and* $p(N_0, N_1)$ *are the defined in Proposition 2.* □

Proof. See Appendix A.3. □

Proposition 3 helps to quickly identify a set $N \subseteq N_1$ to improve the current partition (N_0, N_1) by moving set N from the maintenance set to the do-nothing set. Note that in contrast to considering $N \subseteq N_0$ in Proposition 2, Proposition 3 considers $N \subseteq N_1$.

We similarly first investigate the condition in Proposition 3 when $|N| = 1$. Suppose $|N| = \{k\}$, $k \in N_1$, Proposition 3 establishes the condition of improving the current partition by not maintaining component k. When $N_1' \neq \emptyset$, we have

$$\begin{aligned} \Delta_{\mathrm{s}}(N_0, N_1, \{k\}) = & \frac{(c_{k,\mathrm{pm}} - (Q_k(g_{k,1}, m) - Q_k(1, m))c_{k,\mathrm{cm}})(1 - Q_k(1, m))}{c_{\mathrm{s}}(Q_k(g_{k,1}, m) - Q_k(1, m))p(N_0, N_1)} \\ \approx & \frac{c_{k,\mathrm{pm}} - Q_k(g_{k,1}, m)c_{k,\mathrm{cm}}}{c_{\mathrm{s}}Q_k(g_{k,1}, m)p(N_0, N_1)} \end{aligned} \tag{20}$$

because $Q_k(1, m) \approx 0$ in many scenarios. Examining Equation (20), we observe similar patterns regarding whether changing a component from PM to do-nothing reduces the total maintenance costs as the ones we see from Equation (19).

Corollary 1. *Let $(N_0 \cup N_\mathrm{u}, N_1)$ and $(N_0, N_1 \cup N_\mathrm{u})$ be two feasible partitions of $\mathcal{N}$. For any set $N \subseteq N_\mathrm{u}$ and $N \neq \emptyset$, we have $\Delta_r(N_0 \cup N_\mathrm{u}, N_1, N) \geq \Delta_s(N_0, N_1 \cup N_\mathrm{u}, N)$, where equality holds when $N = N_\mathrm{u}$.* □

Proof. See Appendix B.1. □

Corollary 1 shows that any set $N \subset N_\mathrm{u}$ satisfies either Proposition 2 or Proposition 3 or none. When $N = N_\mathrm{u}$, set N satisfies either Proposition 2 or Proposition 3. This corollary is needed to prove Proposition 4 in the next section.

5. Solution algorithms

Based on Propositions 2 and 3, we design Algorithm 1 that finds the optimal partition (N_0^*, N_1^*) for P2. Although the computational studies in the next section show that Algorithm 1 is fast for most test cases, the time complexity of Algorithm 1 is $O(2^n)$ in the worst case scenario. Therefore, we develop Algorithm 2 to heuristically search a better solution based on the results from the early termination of Algorithm 1. We further use the two-stage model and rolling horizon technique to approximate the multi-stage problem P1.

5.1. Algorithm 1

Propositions 2 and 3 help to find a better solution given any feasible solution. However, they do not necessarily lead any feasible solution to an optimal one. Next we show that if Propositions 2 and 3 are applied following a certain procedure, an optimal solution can be obtained.

Let N_u be the undetermined set in which all components' first-stage decisions are not determined. Constructing an optimal partition (N_0, N_1) implies optimally moving all subsets $N \subseteq N_\mathrm{u}$ to N_0 or N_1. The proposed procedure starts from searching all subsets with $|N| = 1$ and increases the cardinality of N by 1 until some N is moved to N_1 based on Proposition 2 or N_0 based on Proposition 3. Because the conditions of Propositions 2 and 3 change after moving N, the search restarts from $|N| = 1$. This process is repeated until $N_\mathrm{u} = \emptyset$. It can be easily verified that 2^n sets need to be examined in the worst case scenario.

Algorithm 1 Determining an optimal partition (N_0^*, N_1^*) for P2

```
Input: Component set $\mathcal{N}$;
       PM cost $c_{i,\mathrm{pm}}$ and CM cost $c_{i,\mathrm{cm}}$ $\forall i \in \mathcal{N}$ and setup cost $c_\mathrm{s}$;
       Component state $g_{i,1}$ $\forall i \in \mathcal{N}$
       State transition probability $Q_i(1, m)$ and $Q_i(g_{i,1}, m)$ $\forall i \in \mathcal{N}$.
Output: Optimal partition $(N_0^*, N_1^*)$ of $\mathcal{N}$
Initial $N_0^* \leftarrow \emptyset$, $N_1^* \leftarrow \emptyset$, $j \leftarrow 1$, undetermined set $N_\mathrm{u} \leftarrow \mathcal{N}$;
for $i \in \mathcal{N}$ do// maintain all failed components
    if $g_{i,1} = m$ then
        $N_1^* = N_1^* \cup \{i\}$,$N_\mathrm{u} = N_\mathrm{u} \backslash \{i\}$;
    end if
end for
while $N_\mathrm{u} \neq \emptyset$ do // Keep moving $N \subseteq N_\mathrm{u}$ to $N_0^*$ and $N_1^*$ until $N_\mathrm{u} = \emptyset$
    $N_0 \leftarrow \emptyset$, $N_1 \leftarrow \emptyset$, $u \leftarrow |N_\mathrm{u}|$;
    $N^j \leftarrow$ all subsets of $N_\mathrm{u}$ with cardinality $j$;
    for each set $N \in N^j$ do
        if $\Delta_\mathrm{r}(N_0^* \cup N_\mathrm{u}, N_1^*, N) < 1$ then
            $N_1 \leftarrow N_1 \cup N$; //$N$ satisfies Proposition 2, move $N$ to $N_1$
        else if $\Delta_\mathrm{s}(N_0^*, N_1^* \cup N_\mathrm{u}, N) \geq 1$ then
            $N_0 \leftarrow N_0 \cup N$; //$N$ satisfies Proposition 3, move $N$ to $N_0$
        end if
    end for
    $N_\mathrm{u} \leftarrow N_\mathrm{u} \backslash (N_1 \cup N_0)$, $N_0^* \leftarrow N_0^* \cup N_0$, $N_1^* \leftarrow N_1^* \cup N_1$;
    if $u > |N_\mathrm{u}|$ then // If $N_\mathrm{u}$ is reduced
        $j \leftarrow 1$; //Search $N$ from $|N| = 1$.
    else
        $j \leftarrow j + 1$;//Search $N$ at a higher cardinality.
    end if
end while
return $(N_0^*, N_1^*)$;
```

Specifically, we initialize $N_0 = \emptyset$, N_1 to include all failed components to ensure the feasibility, and $N_\mathrm{u} = \mathcal{N} \backslash N_1$. If there is any subset $N \subseteq N_\mathrm{u}$ with $|N| = 1$ satisfies Proposition 2 (Proposition 3), we move N from N_u to N_1 (N_0) after all subsets with $|N| = 1$ are searched. If there is no subset N with $|N| = 1$ satisfies Propositions 2 or 3, we search the subsets N with $|N| = 2, 3, ..., |N_\mathrm{u}|$ in an ascending order until some N satisfying the condition in Propositions 2 or 3 is obtained and moved out of N_u. We then update N_u and restart to search the subset N from $|N| = 1$. The construction of the optimal partition (N_0, N_1) terminates when $N_\mathrm{u} = \emptyset$. The optimality of partition

(N_0, N_1) given by Algorithm 1 is proved in Proposition 4.

Proposition 4. *The partition (N_0^*, N_1^*) given by Algorithm 1 is an optimal partition.* □

Proof. See Appendix A.4. □

5.2. Algorithm 2

As stated previously, Algorithm 1 requires to examine 2^n sets in the worst case scenario. To ensure that we obtain a high-quality solution in a reasonable amount of time, we develop Algorithm 2 that heuristically finds a sub-optimal solution based on Algorithm 1.

```
Algorithm 2 Heuristic algorithm for P2
Input: $N_0^*$ and $N_1^*$ ← results from Algorithm 1 by setting $J$ as the maximum set cardinality to search;
        Maximum number of partitions $M$;
Output: Sub-optimal partition $(N_0', N_1')$ of $\mathcal{N}$.
  $N_\mathrm{u} \leftarrow \mathcal{N} \backslash (N_0^* \cup N_1^*)$;
  Randomly generate $M$ partitions $(N_0^* \cup \bar{N}_0, N_1^* \cup \bar{N}_1)$, including $(N_0^* \cup N_\mathrm{u}, N_1^*)$ and $(N_0^*, N_1^* \cup N_\mathrm{u})$, where $\bar{N}_0 \cup \bar{N}_1 = N_\mathrm{u}$;
  Select the best partition $(N_0', N_1')$ that has the minimum cost among $M$ partitions;
  return $(N_0', N_1')$;
```

Specifically, we first terminate Algorithm 1 after the cardinality of $|N|$ exceeds the maximum cardinality J specified, which means we only search the component set that has no more than J components. Based on N_0^* and N_1^* obtained from early termination of Algorithm 1, we randomly generate M partitions $(\bar{N}_0, \bar{N}_1)$ of undetermined set $N_\mathrm{u} = \mathcal{N} \backslash (N_0^* \cup N_1^*)$, and select the best partition $(N_0^* \cup \bar{N}_0, N_1^* \cup \bar{N}_1)$ of $\mathcal{N}$. Note that it is suggested to include the options of maintaining all and none of undetermined components as candidate solutions, because many of our experiments show that it is likely that the optimal partition is either $(N_0^* \cup N_\mathrm{u}, N_1^*)$ or $(N_0^*, N_1^* \cup N_\mathrm{u})$.

5.3. Algorithm 3

We further use P2 to approximate the multi-stage model (P1) by utilizing the rolling horizon technique. At each decision period t, we solve P2 which consists of periods t and $t + 1$ using Algorithm 2 and employ the first-stage solutions as the decisions for period t. The states components transition to in the next period (i.e., period $t+1$) are determined by the decision made in the previous period (i.e., period t) and the transition probabilities. We then solve a new P2 consisting of decision periods $t+1$ and $t+2$ and use the first-stage solution as the decisions for period $t+1$. This process is repeated until the last decision period is reached. This procedure is summarized in Algorithm 3.

Algorithm 3 Solving multi-stage model using a rolling-horizon approach

```
Output: A solution over planning horizon $\mathcal{T}$
for $t \in \mathcal{T} \backslash \{T\}$ do
    Solve P2 using Algorithm 2, and obtain the first-stage solutions $x_{it}, y_{it}$ and $z_t$ for all components $i \in \mathcal{N}$;
end for
The last stage solutions $x_{iT}, y_{iT}$ and $z_T$ for all components $i \in \mathcal{N}$ are given by Equations (8) to (10);
return $x_{it}, y_{it}$ and $z_t$ for all components $i \in \mathcal{N}$ at all stages $t \in \mathcal{T}$.
```

In many real applications, when more degradation information becomes available upon inspection at each decision period, Bayesian updating can be easily performed to obtain more accurate degradation distributions and better-informed maintenance decisions can be made.

6. Computational study

In this section, we first linearize P2 so that small-scale problems of P2 can be solved by commercial solvers such as CPLEX for comparison purposes. We then conduct computational studies to examine the performance of Algorithms 1 and 2. The proposed models and algorithms are then illustrated by two real-world cases.

6.1. Linearization of P2

In Equation (17), the term $\prod_{i \in \mathcal{N}}(1 - Q_i(g_{i,1}, m)(1 - x_{i,1}) - Q_i(1, m)x_{i,1})$ is non-linear, which is linearized first. The term $\prod_{i \in \mathcal{N}}(1 - Q_i(g_{i,1}, m)(1 - x_{i,1}) - Q_i(1, m)x_{i,1})$ can be expanded to a polynomial function of $x_{i,1}, i \in \mathcal{N}$, with degree of n. After the expansion, we observe that all non-linear terms are the products of multiple (from 2 to n) binary decision variables $x_{i,1}, i \in \mathcal{N}$. Standard linearization method for the multiplication of multiple binary variables are applied here [50]. After linearization, we replace Equation (17) by Equation (21) in model P2,

$$
\begin{aligned}
V = & \sum_{i \in \mathcal{N}} (Q_i(g_{i,1}, m)(1 - x_{i,1}) + Q_i(1, m) x_{i,1}) c_{i,\mathrm{cm}} \\
& + (1 - \sum_{j=0}^{1} \sum_{k=1}^{|N^j|} \prod_{i \in N_k^j} \prod_{r \in \mathcal{N} \setminus N_k^j} a_i x_{i,1} b_r - \sum_{j=2}^{n} \sum_{k=1}^{|N^j|} u_k^j \prod_{i \in N_k^j} \prod_{r \in \mathcal{N} \setminus N_k^j} a_i b_r) c_{\mathrm{s}}
\end{aligned} \tag{21}
$$

where

$$b_i = 1 - Q_i(g_{i,1}, m), i \in \mathcal{N} \tag{22}$$

$$a_i = Q_i(g_{i,1}, m) - Q_i(1, m), i \in \mathcal{N} \tag{23}$$

$$u_j^k \leq x_{i,1}, j \in \{2, ..., n\}, k \in \{1, ..., |N^j|\}, i \in N_k^j \tag{24}$$

$$u_j^k \geq \sum_{i \in N_k^j} x_{i,1} - (j - 1), j \in \{2, ..., n\}, k \in \{1, ..., |N^j|\} \tag{25}$$

$$u_j^k \in \{0, 1\}, j \in \{2, ..., n\}, k \in \{1, ..., |N^j|\}. \tag{26}$$

Note that set N^j collects all subsets of $\mathcal{N}$ that have cardinality j and therefore $|N^j| = \binom{n}{j}$, $j \in \{0, 1, 2, ..., n\}$. For each set $N_k^j \in N^j, k \in \{1, ..., |N^j|\}$, we have $N_k^j \subseteq \mathcal{N}$ and $|N_k^j| = j$.

6.2. Computational studies

We first compare the computational time of Algorithm 1 with CPLEX for small-scale problems. We then examine the computational time and cost error of Algorithms 1 and 2 for large-scale problems.

We assume the degradation of all components can be described by gamma processes with shape parameter αt and rate parameter γ. The continuous degradation levels are divided into several intervals to represent different states, and the transition probabilities can be computed accordingly. Without loss of generality, we assume inspection interval is 1. We arbitrarily set $M = 100$, which is the maximum partitions generated in Algorithm 2. We consider systems with different number of components $n \in \{10, 11, ..., 19\}$. For each n, we consider 100 instances with different combinations of degradation processes, and costs of PM and CM. The degradation parameters, and the costs of PM and CM are drawn from uniform distributions $U(\cdot,\cdot)$. Therefore, a total of 10,000 experiments are run. Table 1 summarizes the baseline parameters.

Table 1. Baseline parameters for numerical example

α	γ	failure threshold	m	PM cost	CM cost	c_{s}	instance	M
$U(1,5)$	$U(0.2,1)$	20	11	$U(1,5)$	$U(10,30)$	20	100	100

For each n, we examine the average performance of 100 problem instances. Table 2 presents the computational times of solving P2 by CPLEX and Algorithm 1 for different numbers of components. NA is reported when the computational time is either longer than 1 day or out of memory. From Table 2, we can see that the computational time of using CPLEX grows exponentially as the number of components increases. In contrast, Algorithm 1 finds the optimal solutions in a short amount of time.

Table 2. Computational time of solving P2 (in seconds)

n	solver	Algorithm 1	n	solver	Algorithm 1
10	0.422	0.0002	15	186.171	0.0004
11	1.029	0.0003	16	832.674	0.0004
12	3.064	0.0003	17	5093.700	0.0005
13	11.943	0.0003	18	NA	0.0005
14	46.014	0.0004	19	NA	0.0005

Next, we investigate the performances of Algorithms 1 and 2 for large-scale problems. For each n, we similarly examine 100 problem instances. Note that CPLEX cannot solve any large-scale cases tested. Table 3 summarizes the performance of Algorithms 1 and 2 for large-scale problems. For each n in Algorithm 1, we are interested in the average computational time of the 100 problem instances (avg. time), the maximum computational time (max time), the average j_{m} (avg. j_{m}) and the maximum j_{m} (max j_{m}), where j_{m} is the maximum set cardinality j that Algorithm 1 searched. From Table 3, we can see that the average time in general increases as the number of components increases. It is also noted that the maximum search time of Algorithm 1 increases substantially as the number of components increases. This is because the solution space increases significantly as the number of components increases. As a result, Algorithm 1 may have to search more sets at higher cardinalities of $|N|$ before reaching the optimality criterion (i.e., undetermined set is empty). This is evidenced by the increase of j_{m}. A higher cardinality generates more sets to be examined in Propositions 2 and 3, and this consumes more computational time. For Algorithm 2, we examine the computational time and cost error for different stopping criteria J, which is the maximum set cardinality that Algorithm 1 allowed to search. We note that cost errors are all zero compared with the true objective value obtained by Algorithm 1, which shows Algorithm 2 can find high-quality solutions within a reasonable amount of time. We similarly observe that computational time increases as J increases.

Table 3. Performance of Algorithm 1&2 in large-scale problems (in seconds)

	Algorithm 1				Algorithm 2											
n	avg.	max	avg.	max	avg. time						max time					
	time	time	j_{m}	j_{m}	J=1	J=2	J=3	J=4	J=5	J=6	J=1	J=2	J=3	J=4	J=5	J=6
20	0.001	0.002	1.19	4	0.001	0.001	0.001	0.001	0.001	0.001	0.003	0.003	0.004	0.001	0.001	0.001
40	0.002	0.006	1.01	2	0.002	0.002	0.002	0.002	0.002	0.002	0.008	0.004	0.004	0.003	0.006	0.006
60	0.004	0.012	1.00	1	0.004	0.004	0.004	0.004	0.004	0.004	0.012	0.012	0.010	0.011	0.012	0.011
80	0.007	0.009	1.00	1	0.007	0.007	0.007	0.007	0.007	0.007	0.012	0.021	0.022	0.018	0.021	0.017
100	0.015	0.236	1.08	3	0.013	0.013	0.015	0.014	0.014	0.015	0.036	0.062	0.251	0.239	0.239	0.240
120	0.023	0.247	1.11	3	0.019	0.020	0.022	0.022	0.023	0.022	0.073	0.073	0.278	0.246	0.253	0.248
140	11.4	939	1.65	7	0.034	0.044	0.119	0.405	1.284	4.307	0.078	0.146	0.948	7.750	45.33	230.5
160	7.07	381	1.86	6	0.052	0.066	0.160	0.589	1.779	7.006	0.123	0.218	1.307	10.53	71.90	380.9
180	9.85	348	2.19	6	0.072	0.102	0.355	1.447	4.602	9.940	0.174	0.396	3.022	18.51	137.9	349.1
200	106	6379	2.65	8	0.091	0.134	0.515	2.559	9.622	24.75	0.173	0.317	2.494	23.79	185.4	1205

We further examine the performance the two-stage rolling horizon approach (Algorithm 3) in approximating the multi-stage model by comparing results from our approach with optimal results on small-scale problems where exact solutions can be obtained. We arbitrarily use one problem setting by taking a sample of parameters based on Table 1 and replicate the problem 1000 times to obtain the average cost using Algorithm 3. Enumeration approach is used to obtain the exact solutions, since the problem size is small. Table 4 summarizes the computational results of different instances that can be solved within one day. From Table 4, we can see that cost percentage errors are below 20% for all cases considered, which shows that the two-stage rolling horizon approach provides an acceptable approximation to the multi-stage problem.

Table 4. Performance of two-stage rolling horizon in approximating multi-stage model

		multi-stage	two-stage rolling horizon	
n	T	cost	avg. cost	error %
2	3	24.49	24.85	1.47%
	4	31.37	34.83	11.03%
	5	37.09	40.33	8.74%
3	3	30.34	35.72	17.73%
	4	40.67	47.92	17.83%
	5	53.78	63.50	18.07%

6.3. Case 1: degradation of wind turbine blades

Offshore wind farms are rapidly [26] developing in recent years to provide the renewable energy for sustainable development. An offshore wind farm is usually built thousand meters away from the coastline and typically has hundreds of wind turbines. A wind turbine consists of multiple components, such as blade, main bearing, gearbox, and generator. If a maintenance team is sent to maintain a wind turbine, it is economically beneficial to jointly maintain other wind turbines [9].

Due to the tensile mechanical loading and corrosive marine environment, stress corrosion cracking (SCC) is one of the major contributors to blades' degradation. Shafiee et al. [26] model the monthly propagation of SCC as a stationary gamma process with an estimated shape parameter $\hat{\alpha} = 0.542$ and rate parameter $\hat{\gamma} = 1.147$.

We consider a three-blade wind turbine system in this case study. Consider a planning horizon $T = 10$ and an inspection interval of 12 months. Based on some pilot studies, we discretize the condition of a blade into 11 states, because this number of states provides us an acceptable decision accuracy while ensuring that the discretized states are robust to measurement errors. The PM cost is 200,000 Monetary Unit (MU). We consider two levels of CM costs: 600,000 MU and 1,000,000 MU.The setup cost is 130,000 MU and the failure threshold is 20 cm [26].

Table 5. Results of multi-stage approximation (CM cost = 600,000)

$t\backslash i$	1		2		3		$\tilde{\xi}^*$
	state	decision	state	decision	state	decision	
1	6	no action	5	no action	**8**	**no action**	8
2	10	PM	7	no action	11	CM	
3	4	no action	10	PM	4	no action	
4	5	no action	4	no action	**8**	**no action**	
5	6	no action	9	PM	11	CM	
6	**8**	**no action**	2	no action	2	no action	
7	10	PM	5	no action	5	no action	
8	6	no action	**8**	**no action**	10	PM	
9	8	PM	10	PM	4	no action	
10	4	no action	4	no action	6	no action	

Note: the decisions that are different from the decisions without economic dependence are shown in boldface

Table 6. Results of multi-stage approximation (CM cost = 1,000,000)

$t\backslash i$	1		2		3		$\tilde{\xi}^*$
	state	decision	state	decision	state	decision	
1	6	no action	5	no action	8	PM	8
2	10	PM	7	no action	5	no action	
3	4	no action	10	PM	8	PM	
4	5	no action	4	no action	5	no action	
5	6	no action	9	PM	8	PM	
6	8	PM	2	no action	2	no action	
7	3	no action	5	no action	5	no action	
8	8	PM	**7**	**PM**	9	PM	
9	3	no action	3	no action	4	no action	
10	6	no action	5	no action	6	no action	

Note: the decisions that are different from the decisions without economic dependence are shown in boldface

We use Algorithm 3 to solve this maintenance planning problem. We compare the decisions with and without considering economic dependence. Denote the PM threshold for each component in the two-stage model without considering economic dependence by $\tilde{\xi}^*$: If the component state is below $\tilde{\xi}^*$, no maintenance is performed, and if the component is functioning and the state exceeds or equals to $\tilde{\xi}^*$, PM is performed.

Tables 5 and 6 summarize the results when CM cost is 600,000 MU and 1,000,000 MU respectively. The threshold $\tilde{\xi}^*$ is 8 in both cases. From Tables 5 and 6, we can observe that maintenance decisions with and without consideration of economic dependence are different. For example, at the first decision stage ($t = 1$) in Table 5, we can see that component 3 is not preventively maintained as it would be without considering economic dependence, so it can share the setup cost with component 1 at decision stage 2. From Table 6, we can see that component 2 is preventively maintained at decision period 8 when it is in state 7, which is below the optimal PM threshold when ignoring economic dependence.

6.4. Case 2: degradation of crude-oil pipelines

The reliability of crude-oil pipelines are critical to the safety of liquid energy supply in modern industries. Due to the corrosion, crack and mechanical damage, pipelines gradually deteriorate, which result in the decrease of pipeline wall thickness.

Based on the degradation data of six pipelines provided by a local chemical plant, we model the degradation process as a gamma process with random effects, where the shape parameter is αt and the rate parameter is γ. Random effects is used to capture the heterogeneities among all pipelines by assuming the rate parameter γ follows a gamma distribution with shape parameter κ and rate parameter λ. We regard the γ as unknown for all pipelines, and use the expectation-maximization algorithm [51] to estimate the parameters of α, κ and λ. Based on the data, we obtain the estimated parameters $\hat{\alpha} = 1.0824$, $\hat{\kappa} = 8.556$ and $\hat{\lambda} = 7.654$.

We consider 17 pipelines located in a small region and all pipes are shutdown when any pipe is maintained. For each pipeline, the wall thickness is $10mm$ when it is new, and the retirement thickness (failure threshold) $8mm$. The number of decision stages is 5. Suppose the costs of PM and CM are 5 and 20, and setup cost is 200.

We solve this multi-stage pipeline maintenance problem by Algorithm 3. We similarly compare the decisions with and without economic dependence. In this case,

Table 7. Components' states and maintenance decisions

$i \backslash t$	1		2		3		4		5	
1	4	no action	6	no action	**9**	**PM**	5	no action	9	no action
2	**8**	**PM**	4	no action	5	no action	**10**	**no action**	11	CM
3	10	PM	4	no action	**7**	**PM**	5	no action	9	no action
4	5	no action	**10**	**no action**	11	CM	3	no action	5	no action
5	4	no action	5	no action	**9**	**PM**	5	no action	7	no action
6	**9**	**PM**	3	no action	**6**	**PM**	4	no action	6	no action
7	2	no action	3	no action	5	no action	7	no action	9	no action
8	10	PM	4	no action	**7**	**PM**	4	no action	6	no action
9	**9**	**PM**	3	no action	**6**	**PM**	3	no action	6	no action
10	**9**	**PM**	3	no action	4	no action	7	no action	8	no action
11	10	PM	3	no action	5	no action	6	no action	9	no action
12	**7**	**PM**	5	no action	**8**	**PM**	3	no action	4	no action
13	**9**	**PM**	2	no action	5	no action	7	no action	10	no action
14	1	no action	5	no action	**9**	**PM**	3	no action	4	no action
15	5	no action	6	no action	**8**	**PM**	4	no action	6	no action
16	10	PM	3	no action	**8**	**PM**	2	no action	5	no action
17	2	no action	3	no action	4	no action	7	no action	10	no action

the optimal PM threshold without considering economic dependence is $\tilde{\xi}^* = 10$. Table 7 presents the state and maintenance action for each component i at stage t. The decisions different from those without considering economic dependence are shown in boldface. Because setup cost is much higher than the CM cost, from Table 7, we can see that there is a large number of different decisions, which shows the necessity of considering economic dependence when it exists.

We further investigate the impacts of parameter estimation uncertainty by considering estimated parameters as random variables. Let $\Theta = (\alpha, \kappa, \lambda)$ be the vector of parameters to be estimated. Given a parameter estimation $\hat{\Theta}$ the conditional cost is denoted by $C(\hat{\Theta})$ and the PDF value of $\hat{\Theta}$ is denoted by $f(\hat{\Theta})$. Based on the estimations, the unconditional cost is $C = \int C(\hat{\Theta}) f(\hat{\Theta}) d(\hat{\Theta})$. The form of PDF $f(\hat{\Theta})$ is typically complicated, and therefore it is difficult to derive the closed-form of C. We use the method in [52] to approximate the unconditional cost. Specifically, we first use the bootstrap method to generate 500 samples of parameters. We then use the average conditional costs based on these samples to approximate the unconditional cost C. Our result shows that the mean and standard deviation of conditional costs are 770.57 and 260.5. Compared with the total maintenance cost 740 when parameter uncertainty is not considered, the impact of parameter estimation uncertainty is acceptable.

7. Conclusion and future research

In this paper, we study CBM optimization problem for multi-component systems over a finite planning horizon. We formulate the problem as a multi-stage stochastic integer program, providing analytical expressions for total cost and maintenance decisions. The proposed multi-stage stochastic maintenance optimization model has integer decision variables and non-linear transition probability due to the endogenous uncertainty, and is computationally intractable. We first investigate structural properties of the two-stage problem and design efficient algorithms to obtain high-quality solutions based on the structural properties. The multi-stage model is then approximated by the two-stage model using a rolling horizon approach. Computational studies show that Algorithm 1 can solve many cases to optimality quickly and Algorithm 2 can find high-quality solutions within a very short amount of time.

This work provides a new modeling approach in modeling multi-component condition-based maintenance. Future research will consider other practical assumptions, such as the limit of maintenance budget, the requirement of system's reliability and availability, state-dependent PM cost, and state-dependent operational cost. In this paper, we mainly consider economic dependence, it is worth to further consider stochastic and structure dependences. It will also be interesting to address situations when we do not know the exact transition probabilities. A robust optimization approach may be applicable.

Appendix

A.1. Proof of Proposition 1

Proof. (1) We first consider the case where there is no failed component in the first stage.

We need to compare the total costs among three cases for partition (N_0, N_1): (a) $N_0 = \emptyset$, (b) $N_0 \neq \emptyset$ and $N_1 \neq \emptyset$ and (c) $N_0 = \mathcal{N}$. Denote C_1, C_2 and C_3 by the total costs for the three cases respectively, we show that C_1 is minimum.

Denote the total cost for component $i \in \mathcal{N}$ *without* considering economic dependence by

$$\begin{cases} TC_i^1 = c_{i,\mathrm{pm}} + c_{\mathrm{s}} + Q_i(1,m)(c_{i,\mathrm{cm}} + c_{\mathrm{s}}), & \tilde{x}_{i,1} = 1, \\ TC_i^0 = Q_i(g_{i,1},m)(c_{i,\mathrm{cm}} + c_{\mathrm{s}}), & \tilde{x}_{i,1} = 0, \end{cases}$$

Because $\tilde{x}^*_{i,1} = 1$, we have $TC_i^1 < TC_i^0$, $\forall i \in \mathcal{N}$.

Thus, we have

$$\begin{aligned} C_1 =& \sum_{i\in\mathcal{N}} TC_i^1 - (n-1)c_{\mathrm{s}} - c_{\mathrm{s}} \sum_{i\in\mathcal{N}} Q_i(1,m) + c_{\mathrm{s}}(1 - \prod_{i\in\mathcal{N}}(1 - Q_i(1,m)), \\ C_2 =& \sum_{i\in N_0} TC_i^0 + \sum_{i\in N_1} TC_i^1 - (|N_1|-1)c_{\mathrm{s}} - c_{\mathrm{s}}(\sum_{i\in N_0} Q_i(g_{i,1},m) + \sum_{i\in N_1} Q_i(1,m)) \\ &+ c_{\mathrm{s}}(1 - \prod_{i\in N_0}(1 - Q_i(g_{i,1},m)) \prod_{i\in N_1}(1 - Q_i(1,m))) \text{ and} \\ C_3 =& \sum_{i\in\mathcal{N}} TC_i^0 - c_{\mathrm{s}} \sum_{i\in\mathcal{N}} Q_i(g_{i,1},m) + c_{\mathrm{s}}(1 - \prod_{i\in\mathcal{N}}(1 - Q_i(g_{i,1},m)). \end{aligned}$$

(1a) Prove $C_1 < C_2$.

Because

$$\begin{cases} TC_i^0 > TC_i^1 \\ (|N_1| - 1)\, c_{\mathrm{s}} + c_{\mathrm{s}} \left(\sum_{i\in N_0} Q_i(g_{i,1},m) + \sum_{i\in N_1} Q_i(1,m)\right) < (n-1)c_{\mathrm{s}} + c_{\mathrm{s}} \sum_{i\in\mathcal{N}} Q_i(1,m) \\ c_{\mathrm{s}}(1 - \prod_{i\in N_0}(1 - Q_i(g_{i,1},m)) \prod_{i\in N_1}(1 - Q_i(1,m))) > c_{\mathrm{s}}(1 - \prod_{i\in\mathcal{N}}(1 - Q_i(1,m)) \end{cases}$$

we have $C_1 < C_2$.

(1b) Prove $C_1 < C_3$

It is easy to show that function $f(v_1, v_2, ..., v_n) = \sum_{i\in\mathcal{N}} v_i + \prod_{i\in\mathcal{N}}(1 - v_i)$ has $\frac{\partial f}{\partial v_i} \geq 0$ for all $0 \leq v_i \leq 1$, $i \in \mathcal{N}$. Therefore, we have

$$\max(C_1) = C_1|_{Q_i(1,m)=0,\forall i\in\mathcal{N}} = \sum_{i\in\mathcal{N}} TC_i^1 - (n-1)c_{\text{s}}$$

and

$$\min(C_3) = C_3|_{Q_i(g_{i,1},m)=1,\forall i\in\mathcal{N}} = \sum_{i\in\mathcal{N}} TC_i^0 - (n-1)c_{\text{s}}.$$

Because $TC_i^0 > TC_i^1$ for all $i \in \mathcal{N}$, we have $C_1 \leq \max(C_1) < \min(C_3) \leq C_3$.

Therefore, C_1 is minimum.

(2) Consider the case where there exists at least one component failed at the first stage.

Let set $N \subseteq \mathcal{N}$ collect all failed components and $N \neq \emptyset$. Following proof (1), we only need to compare case (a) and feasible case (b) because case (c) is not feasible.

The cost of case (a) and feasible case (b) are denoted by C_1' and C_2' respectively, where

$$C_1' = C_1 + \sum_{i\in N}(c_{i,\text{cm}} - c_{i,\text{pm}})$$

and

$$C_2' = C_2 + \sum_{i\in N}(c_{i,\text{cm}} - c_{i,\text{pm}}).$$

From $C_1 < C_2$ in proof (1a), we have $C_1' < C_2'$.

□

A.2. Proof of Proposition 2

Proof. Denote the total cost for component $i \in \mathcal{N}$ *without* considering economic dependence by

$$\begin{cases} TC_i^1 = c_{i,\mathrm{pm}} + c_\mathrm{s} + Q_i(1,m)(c_{i,\mathrm{cm}} + c_\mathrm{s}), & \tilde{x}_{i,1} = 1, \\ TC_i^0 = Q_i(g_{i,1},m)(c_{i,\mathrm{cm}} + c_\mathrm{s}). & \tilde{x}_{i,1} = 0, \end{cases}$$

and let $Q_i(1,m) = Q_i(1)$ and $Q_i(g_{i,1},m) = Q_i(g)$ $\forall i \in \mathcal{N}$, then we have

$$\begin{aligned} C = &\sum_{i\in N_0} TC_i^0 + \sum_{i\in N_1} TC_i^1 - (\max(|N_1|-1,0))c_\mathrm{s} - c_\mathrm{s}(\sum_{i\in N_0} Q_i(g) + \sum_{i\in N_1} Q_i(1)) \\ &+ c_\mathrm{s}(1 - \prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))) \end{aligned}$$

$$\begin{aligned} C' = &\sum_{i\in N_0'} TC_i^0 + \sum_{i\in N_1'} TC_i^1 - (\max(|N_1'|-1,0))c_\mathrm{s} \\ &- c_\mathrm{s}(\sum_{i\in N_0'} Q_i(g) + \sum_{i\in N_1'} Q_i(1)) + c_\mathrm{s}(1 - \prod_{i\in N_0'}(1-Q_i(g))\prod_{i\in N_1'}(1-Q_i(1))) \end{aligned}$$

If $N_1 = \emptyset$, we have

$$\begin{aligned} C' - C = &\sum_{k\in N}(TC_k^1 - TC_k^0) + c_\mathrm{s}\sum_{k\in N}(Q_k(g) - Q_k(1)) \\ &+ c_\mathrm{s}(\prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1)) - \prod_{i\in N_0'}(1-Q_i(g))\prod_{i\in N_1'}(1-Q_i(1))) \\ = &\sum_{k\in N}\underbrace{(c_{k,\mathrm{pm}} - (Q_k(g) - Q_k(1))\,c_{k,\mathrm{cm}})}_{\rho_k c_\mathrm{s}} + c_\mathrm{s} \\ &- c_\mathrm{s}\underbrace{\prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))}_{p(N_0,N_1)}\underbrace{\left(\prod_{k\in N}\frac{1-Q_k(1)}{1-Q_k(g)} - 1\right)}_{r_N} \\ = &\sum_{k\in N}\rho_k c_\mathrm{s} + c_\mathrm{s} - c_\mathrm{s} p(N_0,N_1) r_N \end{aligned}$$

Therefore, from $C' < C$, we have

$$\frac{\sum_{k\in N}\rho_k c_\mathrm{s} + c_\mathrm{s}}{c_\mathrm{s} r_N p(N_0,N_1)} = \frac{1 + \sum_{k\in N}\rho_k}{r_N p(N_0,N_1)} = \Delta_\mathrm{r}(N_0,N_1,N) < 1.$$

From $\Delta_\mathrm{r}(N_0,N_1,N) < 1$, we have $C' < C$.

Similarly, if $N_1 \neq \emptyset$,

$$\begin{aligned}
C' - C &= \sum_{k \in N} (TC_k^1 - TC_k^0) - |N| c_{\mathrm{s}} + c_{\mathrm{s}} \sum_{k \in N} (Q_k(g) - Q_k(1)) \\
&\quad + c_{\mathrm{s}} \Big(\prod_{i \in N_0} (1 - Q_i(g)) \prod_{i \in N_1} (1 - Q_i(1)) - \prod_{i \in N_0'} (1 - Q_i(g)) \prod_{i \in N_1'} (1 - Q_i(1)) \Big) \\
&= \sum_{k \in N} \underbrace{(c_{k,\mathrm{pm}} - (Q_k(g) - Q_k(1))\, c_{k,\mathrm{cm}})}_{\rho_k c_{\mathrm{s}}} \\
&\quad - c_{\mathrm{s}} \underbrace{\prod_{i \in N_0} (1 - Q_i(g)) \prod_{i \in N_1} (1 - Q_i(1))}_{p(N_0, N_1)} \underbrace{\left(\prod_{k \in N} \frac{1 - Q_k(1)}{1 - Q_k(g)} - 1 \right)}_{r_N} \\
&= \sum_{k \in N} \rho_k c_{\mathrm{s}} - c_{\mathrm{s}} p(N_0, N_1) r_N
\end{aligned}$$

Therefore, from $C' < C$, we have

$$\frac{\sum_{k \in N} \rho_k c_{\mathrm{s}}}{c_{\mathrm{s}} r_N p(N_0, N_1)} = \frac{\sum_{k \in N} \rho_k}{r_N p(N_0, N_1)} = \Delta_{\mathrm{r}}(N_0, N_1, N) < 1.$$

From $\Delta_{\mathrm{r}}(N_0, N_1, N) < 1$, we have $C' < C$. □

A.3. Proof of Proposition 3

Proof. Denote the total cost for component $i \in \mathcal{N}$ *without* considering economic dependence by

$$\begin{cases}
TC_i^1 = c_{i,\mathrm{pm}} + c_{\mathrm{s}} + Q_i(1, m)(c_{i,\mathrm{cm}} + c_{\mathrm{s}}), & \tilde{x}_{i,1} = 1, \\
TC_i^0 = Q_i(g_{i,1}, m)(c_{i,\mathrm{cm}} + c_{\mathrm{s}}). & \tilde{x}_{i,1} = 0,
\end{cases}$$

and let $Q_i(1, m) = Q_i(1)$ and $Q_i(g_{i,1}, m) = Q_i(g)$ $\forall i \in \mathcal{N}$, then we have

$$\begin{aligned}
C = &\sum_{i \in N_0} TC_i^0 + \sum_{i \in N_1} TC_i^1 - (\max(|N_1| - 1, 0)) c_{\mathrm{s}} - c_{\mathrm{s}} \Big(\sum_{i \in N_0} Q_i(g) + \sum_{i \in N_1} Q_i(1) \Big) \\
&+ c_{\mathrm{s}} \Big(1 - \prod_{i \in N_0} (1 - Q_i(g)) \prod_{i \in N_1} (1 - Q_i(1)) \Big)
\end{aligned}$$

$$
\begin{aligned}
C' = &\sum_{i\in N_0'} TC_i^0 + \sum_{i\in N_1'} TC_i^1 - (\max(|N_1'|-1,0))c_\mathrm{s} - c_\mathrm{s}(\sum_{i\in N_0'} Q_i(g) + \sum_{i\in N_1'} Q_i(1)) \\
&+ c_\mathrm{s}(1 - \prod_{i\in N_0'}(1-Q_i(g))\prod_{i\in N_1'}(1-Q_i(1)))
\end{aligned}
$$

If $N_1' = \emptyset$, we have

$$
\begin{aligned}
C - C' =& \sum_{k\in N}(TC_k^1 - TC_k^0) + c_\mathrm{s}\sum_{k\in N}(Q_k(g) - Q_k(1)) \\
&+ c_\mathrm{s}(\prod_{i\in N_0'}(1-Q_i(g))\prod_{i\in N_1'}(1-Q_i(1)) - \prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))) \\
=& \sum_{k\in N}\underbrace{(c_{k,\mathrm{pm}} - (Q_k(g)-Q_k(1))\,c_{k,\mathrm{cm}})}_{\rho_k c_\mathrm{s}} + c_\mathrm{s} \\
&- c_\mathrm{s}\underbrace{\prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))}_{p(N_0,N_1)}\underbrace{\left(1 - \prod_{k\in N}\frac{1-Q_k(g)}{1-Q_k(1)}\right)}_{s_N} \\
=& \sum_{k\in N}\rho_k c_\mathrm{s} + c_\mathrm{s} - c_\mathrm{s}p(N_0,N_1)s_N
\end{aligned}
$$

From $C > C'$, we have $\sum_{k\in N}\rho_k c_\mathrm{s} + c_\mathrm{s} - c_\mathrm{s}p(N_0,N_1)s_N > 0$. Therefore,

$$
\frac{\sum_{k\in N}\rho_k c_\mathrm{s} + c_\mathrm{s}}{c_\mathrm{s}s_N p(N_0,N_1)} = \frac{1+\sum_{k\in N}\rho_k}{s_N p(N_0,N_1)} = \Delta_\mathrm{s}(N_0,N_1,N) > 1,
$$

From $\Delta_\mathrm{s}(N_0,N_1,N) > 1$, we have $C > C'$.

Similarly, if $N_1' \neq \emptyset$,

$$
\begin{aligned}
C - C' =& \sum_{k\in N}(TC_k^1 - TC_k^0) - |N|c_\mathrm{s} + c_\mathrm{s}\sum_{k\in N}(Q_k(g) - Q_k(1)) \\
&+ c_\mathrm{s}(\prod_{i\in N_0'}(1-Q_i(g))\prod_{i\in N_1'}(1-Q_i(1)) - \prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))) \\
=& \sum_{k\in N}\underbrace{(c_{k,\mathrm{pm}} - (Q_k(g)-Q_k(1))\,c_{k,\mathrm{cm}})}_{\rho_k c_\mathrm{s}} \\
&- c_\mathrm{s}\underbrace{\prod_{i\in N_0}(1-Q_i(g))\prod_{i\in N_1}(1-Q_i(1))}_{p(N_0,N_1)}\underbrace{\left(1 - \prod_{k\in N}\frac{1-Q_k(g)}{1-Q_k(1)}\right)}_{s_N} \\
=& \sum_{k\in N}\rho_k c_\mathrm{s} - c_\mathrm{s}p(N_0,N_1)s_N
\end{aligned}
$$

From $C > C'$, we have

$$\frac{\sum_{k\in N} \rho_k c_{\mathrm{s}}}{c_{\mathrm{s}} s_N p(N_0, N_1)} = \frac{\sum_{k\in N} \rho_k}{s_N p(N_0, N_1)} = \Delta_{\mathrm{s}}(N_0, N_1, N) > 1,$$

From $\Delta_{\mathrm{s}}(N_0, N_1, N) > 1$, we have $C > C'$ □

A.4. Proof of Proposition 4

Proof. We prove this proposition by showing that the cost of partition (N_0^*, N_1^*) is no worse than that of any other feasible partitions.

For any other feasible partition (N_0', N_1') and the partition (N_0^*, N_1^*) that is obtained by Algorithm 1, we always rewrite $(N_0', N_1') = (N_0 \cup N_b, N_1 \cup N_a)$ and $(N_0^*, N_1^*) = (N_0 \cup N_a, N_1 \cup N_b)$ respectively, where set $N_0 = N_0' \cap N_0^*$, $N_1 = N_1' \cap N_1^*$, $N_b = N_0' \backslash N_0 = N_1^* \backslash N_1$ and $N_a = N_1' \backslash N_1 = N_0^* \backslash N_0$. We now show that the cost of partition (N_0^*, N_1^*) is no worse than that of (N_0', N_1') by the following three parts: (1) When $N_b \neq \emptyset$, we have cost relationship $(N_0^*, N_1^*) = (N_0 \cup N_a, N_1 \cup N_b) < (N_0 \cup N_a \cup N_b, N_1)$, (2) when $N_b \neq \emptyset$, we have cost relationship $(N_0 \cup N_a \cup N_b, N_1) < (N_0 \cup N_b, N_1 \cup N_a) = (N_0', N_1')$, and (3) we have cost $(N_0', N_1') = (N_0^*, N_1^*)$ if and only if $N_b = \emptyset$ and $\Delta_{\mathrm{r}}(N_0 \cup N_a, N_1, N_a) = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_a, N_a) = 1$.

(1) When $N_b \neq \emptyset$, we have cost relationship $(N_0^*, N_1^*) = (N_0 \cup N_a, N_1 \cup N_b) < (N_0 \cup N_a \cup N_b, N_1)$.

This is equivalent to show that given current partition $(N_0 \cup N_a \cup N_b, N_1)$, moving N_b from the do-nothing set to the maintenance set can reduce cost. We next show that if we keep moving the component that arrives first in N_b in Algorithm 1 to the maintenance set, the cost keeps reducing until $N_b = \emptyset$, which implies moving the whole set N_b to the maintenance set reduces cost.

Denote the costs of $(N_0 \cup N_a, N_1 \cup N_b)$ and $(N_0 \cup N_a \cup N_b, N_1)$ by C and C_0 respectively, and initialize $C' = C_0$. We prove $C < C_0$ by the following steps:

Step 1: If all components in N_b are moved into N_1^* after set N_1 does in Algorithm 1, then $C < C_0$ because the cost reduces if we repeat how Algorithm 1 moves N_b to N_1^*.

Step 2: In this step, there exists at least one component $i \in N_b$ that joins N_1^* no later than some component in N_1. Suppose component $k \in N_b$ is the earliest one in N_b that joins N_1^* and suppose k joins N_1^* along with set S^j, i.e., $k \in S^j$, where

$|S^j| = j$ and $S^j \subseteq N_1^*$. Therefore, when $S^j \subseteq N_1^*$ joins N_1^*, the current partition is $(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S)$, where set $S^j \backslash \{k\} \subseteq S$, and hence from Proposition 2, we have

$$\Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S, S^j) < 1. \tag{27}$$

Step 3: If $j = 1$, then $S^j = \{k\}$. Denote the costs for partition $(N_0 \cup N_a \cup N_b \backslash \{k\}, N_1 \cup \{k\})$ by C_1. From Inequation (27), we have $\Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b, N_1, \{k\}) < \Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S, \{k\}) < 1$ and therefore $C_1 < C'$. We then update $N_b = N_b \backslash \{k\}$ and $C' = C_1$ and go to Step 1.

Step 4: In this step, we have $j > 1$. From Algorithm 1, we know that any subset $N \subset S^j$ cannot join N_1^* given current partition $(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S)$. From Proposition 2, we have

$$\Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S, N) > 1. \tag{28}$$

Let $N + \{k\} = S^j$. We have

$$r_N p(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S) + \rho_k < \sum_{i \in N} \rho_i + \rho_k < r_{S^j} p(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S),$$

where the first inequality is from Inequation (28) and the second inequality is from Inequation (27). Therefore, we have $\rho_k < (r_{S^j} - r_N) p(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S)$ and hence

$$\begin{aligned} \Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b, N_1, \{k\}) &= \frac{\rho_k}{r_{\{k\}} p(N_0 \cup N_a \cup N_b, N_1)} \\ &< \frac{(r_{S^j} - r_N) p(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S)}{r_{\{k\}} p(N_0 \cup N_a \cup N_b, N_1)} \\ &= \frac{\prod_{i \in N} \frac{1 - Q_i(1,m)}{1 - Q_i(g_{i,1}, m)} p(N_0 \cup N_a \cup N_b \cup S, N_1 \backslash S)}{p(N_0 \cup N_a \cup N_b, N_1)} \\ &= \frac{p(N_0 \cup N_a \cup N_b \cup S \backslash N, (N_1 \cup N) \backslash S)}{p(N_0 \cup N_a \cup N_b, N_1)} \leq 1, \end{aligned}$$

where the last inequality is from $N = S^j \backslash \{k\} \subseteq S$. From Proposition 2, by denoting the cost of partition $(N_0 \cup N_a \cup N_b \backslash \{k\}, N_1 \cup \{k\})$ by C_1, we have $C_1 < C'$ since

$\Delta_{\mathrm{r}}(N_0 \cup N_a \cup N_b, N_1, \{k\}) < 1$. We then update $N_b = N_b \backslash \{k\}$ and $C' = C_1$ and go to Step 1.

Therefore, we can always lower the cost C' by moving one component from N_b to the maintenance set. When $N_b = \emptyset$, we have $C' = C < C_0$.

(2) When $N_b \neq \emptyset$, we have cost relationship $(N_0 \cup N_a \cup N_b, N_1) < (N_0 \cup N_b, N_1 \cup N_a) = (N_0', N_1')$.

This is equivalent to show that moving set N_a from the maintenance set to the do-nothing set can lower cost. From Proposition 3, we need to prove $\Delta_{\mathrm{s}}(N_0 \cup N_b, N_1 \cup N_a, N_a) > 1$.

By using the same method as proof (1), we can also prove the cost relationship $(N_0 \cup N_a, N_1 \cup N_b) < (N_0, N_1 \cup N_a \cup N_b)$ when $N_b \neq \emptyset$. From Proposition 3, we have $\Delta_{\mathrm{s}}(N_0, N_1 \cup N_a \cup N_b, N_a) > 1$. Therefore,

$$\Delta_{\mathrm{s}}(N_0 \cup N_b, N_1 \cup N_a, N_a) > \Delta_{\mathrm{s}}(N_0, N_1 \cup N_a \cup N_b, N_a) > 1$$

(3) We have cost $(N_0', N_1') = (N_0^*, N_1^*)$ if and only if $N_b = \emptyset$ and $\Delta_{\mathrm{r}}(N_0 \cup N_a, N_1, N_a) = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_a, N_a) = 1$.

When $(N_0^*, N_1^*) = (N_0', N_1')$, we have $(N_0 \cup N_a, N_1 \cup N_b) = (N_0 \cup N_a \cup N_b, N_1) = (N_0 \cup N_b, N_1 \cup N_a)$.

The first equality $(N_0 \cup N_a, N_1 \cup N_b) = (N_0 \cup N_a \cup N_b, N_1)$ holds if and only if $N_b = \emptyset$. Otherwise, following the steps of proof (1), we can always have $(N_0 \cup N_a, N_1 \cup N_b) < (N_0 \cup N_a \cup N_b, N_1)$.

Given $N_b = \emptyset$, the second equality is equivalent to $(N_0 \cup N_a, N_1) = (N_0, N_1 \cup N_a)$, which happens if and only if $\Delta_{\mathrm{r}}(N_0 \cup N_a, N_1, N_a) = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_a, N_a) = 1$ based on Corollary 1.

□

B.1. Proof of Corollary 1:

Proof. We first show $p(N_0 \cup N_{\mathrm{u}}, N_1)r_N \leq p(N_0, N_1 \cup N_{\mathrm{u}})s_N$.

$$\begin{aligned}
&p(N_0 \cup N_{\mathrm{u}}, N_1)r_N \\
=&\prod_{i\in N_0\cup N_{\mathrm{u}}}(1-Q_i(g_{i,1},m))\prod_{i\in N_1}(1-Q_i(1,m))(\frac{\prod_{i\in N}(1-Q_i(1,m))-\prod_{i\in N}(1-Q_i(g_{i,1},m))}{\prod_{i\in N}(1-Q_i(g_{i,1},m))}) \\
=&\prod_{i\in N_0\cup N_{\mathrm{u}}-N}(1-Q_i(g_{i,1},m))\prod_{i\in N_1\cup N}(1-Q_i(1,m))(\frac{\prod_{i\in N}(1-Q_i(1,m))-\prod_{i\in N}(1-Q_i(g_{i,1},m))}{\prod_{i\in N}(1-Q_i(1,m))}) \\
=&p(N_0 \cup N_{\mathrm{u}}\backslash N, N_1 \cup N)s_N \leq p(N_0, N_1 \cup N_{\mathrm{u}})s_N,
\end{aligned}$$

where equality holds when $N = N_{\mathrm{u}}$.

(1) When $N_1 \neq \emptyset$, we have

$$\Delta_{\mathrm{r}}(N_0 \cup N_{\mathrm{u}}, N_1, N) = \frac{\sum_{k\in N}\rho_k}{r_N p(N_0 \cup N_{\mathrm{u}}, N_1)} \geq \frac{\sum_{k\in N}\rho_k}{s_N p(N_0, N_1 \cup N_{\mathrm{u}})} = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_{\mathrm{u}}, N),$$

where equality holds when $N = N_{\mathrm{u}}$.

(2) When $N_1 = \emptyset$ and $N \neq N_{\mathrm{u}}$, we have

$$\Delta_{\mathrm{r}}(N_0 \cup N_{\mathrm{u}}, N_1, N) = \frac{1+\sum_{k\in N}\rho_k}{r_N p(N_0 \cup N_{\mathrm{u}}, N_1)} > \frac{\sum_{k\in N}\rho_k}{s_N p(N_0, N_1 \cup N_{\mathrm{u}})} = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_{\mathrm{u}}, N).$$

(3)When $N_1 = \emptyset$ and $N = N_{\mathrm{u}}$, we have

$$\Delta_{\mathrm{r}}(N_0 \cup N_{\mathrm{u}}, N_1, N) = \frac{1+\sum_{k\in N}\rho_k}{r_N p(N_0 \cup N_{\mathrm{u}}, N_1)} = \frac{1+\sum_{k\in N}\rho_k}{s_N p(N_0, N_1 \cup N_{\mathrm{u}})} = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_{\mathrm{u}}, N).$$

Therefore, $\Delta_{\mathrm{r}}(N_0 \cup N_{\mathrm{u}}, N_1, N) > \Delta_{\mathrm{s}}(N_0, N_1 \cup N_{\mathrm{u}}, N)$ when $N \subset N_{\mathrm{u}}$ and $\Delta_{\mathrm{r}}(N_0 \cup N_{\mathrm{u}}, N_1, N) = \Delta_{\mathrm{s}}(N_0, N_1 \cup N_{\mathrm{u}}, N)$ when $N = N_{\mathrm{u}}$. □

References

[1] R. Feynman, "Report of the presidential commission on the space shuttle challenger accident," *Appendix F*, 1986.

[2] M. E. Paté-Cornell, "Learning from the piper alpha accident: A postmortem analysis of technical and organizational factors," *Risk Analysis*, vol. 13, no. 2, pp. 215–232, 1993.

[3] S.-H. Ding and S. Kamaruddin, "Maintenance policy optimization—literature review and directions," *The International Journal of Advanced Manufacturing Technology*, vol. 76, no. 5-8, pp. 1263–1283, 2015.

[4] S. Alaswad and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system," *Reliability Engineering & System Safety*, vol. 157, pp. 54–63, 2017.

[5] R. Dekker, R. E. Wildeman, and F. A. Van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence," *Mathematical methods of operations research*, vol. 45, no. 3, pp. 411–435, 1997.

[6] R. P. Nicolai and R. Dekker, "Optimal maintenance of multi-component systems: a review," in *Complex system maintenance handbook*, pp. 263–286, Springer, 2008.

[7] L. Thomas, "A survey of maintenance and replacement models for maintainability and reliability of multi-item systems," *Reliability Engineering*, vol. 16, no. 4, pp. 297–309, 1986.

[8] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 135–149, 2012.

[9] Z. Tian and H. Liao, "Condition based maintenance optimization for multi-component systems using proportional hazards model," *Reliability Engineering & System Safety*, vol. 96, no. 5, pp. 581–589, 2011.

[10] R. Laggoune, A. Chateauneuf, and D. Aissani, "Impact of few failure data on the opportunistic replacement policy for multi-component systems," *Reliability Engineering & System Safety*, vol. 95, no. 2, pp. 108–119, 2010.

[11] S. Amaran, T. Zhang, N. V. Sahinidis, B. Sharda, and S. J. Bury, "Medium-term maintenance turnaround planning under uncertainty for integrated chemical sites," *Computers & Chemical Engineering*, vol. 84, pp. 422–433, 2016.

[12] P. A. Scarf, "On the application of mathematical models in maintenance," *Euro-*

pean Journal of operational research, vol. 99, no. 3, pp. 493–506, 1997.

[13] R. Dekker and P. A. Scarf, "On the impact of optimisation models in maintenance decision making: the state of the art," *Reliability Engineering & System Safety*, vol. 60, no. 2, pp. 111–119, 1998.

[14] A. Grall, C. Bérenguer, and L. Dieulle, "A condition-based maintenance policy for stochastically deteriorating systems," *Reliability Engineering & System Safety*, vol. 76, no. 2, pp. 167–180, 2002.

[15] B. Castanier, A. Grall, and C. Bérenguer, "A condition-based maintenance policy with non-periodic inspections for a two-unit series system," *Reliability Engineering & System Safety*, vol. 87, no. 1, pp. 109–120, 2005.

[16] J. Barata, C. G. Soares, M. Marseguerra, and E. Zio, "Simulation modelling of repairable multi-component deteriorating systems for 'on condition'maintenance optimisation," *Reliability Engineering & System Safety*, vol. 76, no. 3, pp. 255–264, 2002.

[17] R. Laggoune, A. Chateauneuf, and D. Aissani, "Opportunistic policy for optimal preventive maintenance of a multi-component system in continuous operating units," *Computers & Chemical Engineering*, vol. 33, no. 9, pp. 1499–1510, 2009.

[18] R. Dekker, R. E. Wildeman, and R. Van Egmond, "Joint replacement in an operational planning phase," *European Journal of Operational Research*, vol. 91, no. 1, pp. 74–88, 1996.

[19] R. E. Wildeman, R. Dekker, and A. Smit, "A dynamic policy for grouping maintenance activities," *European Journal of Operational Research*, vol. 99, no. 3, pp. 530–551, 1997.

[20] S. Goyal and M. Kusy, "Determining economic maintenance frequency for a family of machines," *Journal of the Operational Research Society*, vol. 36, no. 12, pp. 1125–1128, 1985.

[21] S. K. Goyal and A. Gunasekaran, "Determining economic maintenance frequency of a transport fleet," *International Journal of Systems Science*, vol. 23, no. 4, pp. 655–659, 1992.

[22] S. Epstein and Y. Wilamowsky, "Opportunistic replacement in a deterministic environment," *Computers & operations research*, vol. 12, no. 3, pp. 311–322, 1985.

[23] M. Hariga, "A deterministic maintenance-scheduling problem for a group of non-identical machines," *International Journal of Operations & Production Manage-*

ment, vol. 14, no. 7, pp. 27–36, 1994.

[24] D. R. Sule and B. Harmon, "Determination of coordinated maintenance scheduling frequencies for a group of machines," *AIIE Transactions*, vol. 11, no. 1, pp. 48–53, 1979.

[25] M. Patriksson, A.-B. Strömberg, and A. Wojciechowski, "The stochastic opportunistic replacement problem, part ii: a two-stage solution approach," *Annals of Operations Research*, vol. 224, no. 1, pp. 51–75, 2015.

[26] M. Shafiee, M. Finkelstein, and C. Bérenguer, "An opportunistic condition-based maintenance policy for offshore wind turbine blades subjected to degradation and environmental shocks," *Reliability Engineering & System Safety*, vol. 142, pp. 463–471, 2015.

[27] Q.-S. Jia, "A structural property of optimal policies for multi-component maintenance problems," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 677–680, 2010.

[28] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and health management: A review on data driven approaches," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[29] J. R. Birge and F. Louveaux, *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[30] M. Bodur, S. Dash, O. Günlük, and J. Luedtke, "Strengthened benders cuts for stochastic integer programs with continuous recourse," *INFORMS Journal on Computing*, vol. 29, no. 1, pp. 77–91, 2016.

[31] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, vol. 16, no. 1, pp. 119–147, 1991.

[32] J.-P. Watson and D. L. Woodruff, "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems," *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011.

[33] S. Ahmed, A. J. King, and G. Parija, "A multi-stage stochastic integer programming approach for capacity expansion under uncertainty," *Journal of Global Optimization*, vol. 26, no. 1, pp. 3–24, 2003.

[34] J. Naoum-Sawaya and S. Elhedhli, "A nested benders decomposition approach for telecommunication network planning," *Naval Research Logistics (NRL)*, vol. 57,

no. 6, pp. 519–539, 2010.

[35] P. Parpas and B. Rustem, "Computational assessment of nested benders and augmented lagrangian decomposition for mean-variance multistage stochastic problems," *INFORMS Journal on Computing*, vol. 19, no. 2, pp. 239–247, 2007.

[36] A. Shapiro, "Analysis of stochastic dual dynamic programming method," *European Journal of Operational Research*, vol. 209, no. 1, pp. 63–72, 2011.

[37] M. V. Pereira and L. M. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical programming*, vol. 52, no. 1-3, pp. 359–375, 1991.

[38] M. Bodur and J. R. Luedtke, "Two-stage linear decision rules for multi-stage stochastic programming," *Mathematical Programming*, pp. 1–34, 2017.

[39] R. Kouwenberg, "Scenario generation and stochastic programming models for asset liability management," *European Journal of Operational Research*, vol. 134, no. 2, pp. 279–292, 2001.

[40] P. Beraldi, A. Violi, N. Scordino, and N. Sorrentino, "Short-term electricity procurement: A rolling horizon stochastic programming approach," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 3980–3990, 2011.

[41] T. Tolio and M. Urgo, "A rolling horizon approach to plan outsourcing in manufacturing-to-order environments affected by uncertainty," *CIRP annals*, vol. 56, no. 1, pp. 487–490, 2007.

[42] A. Shapiro and A. Nemirovski, "On complexity of stochastic programming problems," in *Continuous optimization*, pp. 111–146, Springer, 2005.

[43] D. Bampou and D. Kuhn, "Scenario-free stochastic programming with polynomial decision rules," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7806–7812, IEEE, 2011.

[44] X. Chen, M. Sim, P. Sun, and J. Zhang, "A linear decision-based approximation approach to stochastic programming," *Operations Research*, vol. 56, no. 2, pp. 344–357, 2008.

[45] G. Laporte and F. V. Louveaux, "The integer l-shaped method for stochastic integer programs with complete recourse," *Operations research letters*, vol. 13, no. 3, pp. 133–142, 1993.

[46] V. Goel and I. E. Grossmann, "A class of stochastic programs with decision dependent uncertainty," *Mathematical programming*, vol. 108, no. 2-3, pp. 355–394, 2006.

[47] Y. Zhan, Q. P. Zheng, J. Wang, and P. Pinson, "Generation expansion planning with large amounts of wind power via decision-dependent stochastic programming," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3015–3026, 2017.

[48] S. Peeta, F. S. Salman, D. Gunnec, and K. Viswanath, "Pre-disaster investment decisions for strengthening a highway network," *Computers & Operations Research*, vol. 37, no. 10, pp. 1708–1719, 2010.

[49] C. R. Cassady, R. O. Bowden, L. Liew, and E. A. Pohl, "Combining preventive maintenance and statistical process control: a preliminary investigation," *Iie Transactions*, vol. 32, no. 6, pp. 471–478, 2000.

[50] R. L. Rardin and R. L. Rardin, *Optimization in operations research*, vol. 166. Prentice Hall Upper Saddle River, NJ, 1998.

[51] Z.-S. Ye, M. Xie, L.-C. Tang, and N. Chen, "Semiparametric estimation of gamma processes for deteriorating products," *Technometrics*, vol. 56, no. 4, pp. 504–513, 2014.

[52] Z.-S. Ye, M. Xie, L.-C. Tang, and Y. Shen, "Degradation-based burn-in planning under competing risks," *Technometrics*, vol. 54, no. 2, pp. 159–168, 2012.