INFORMS JOURNAL ON COMPUTING
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0899-1499 | EISSN 1526-5528 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Multi-component Maintenance Optimization: A Stochastic Programming Approach

Zhicheng Zhu
Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, Lubbock, TX 79409

Yisha Xiang
Department of Industrial, Manufacturing & Systems Engineering, Texas Tech University, Lubbock, TX 79409

Bo Zeng
Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261

Maintenance optimization has been extensively studied in the past decades. However, most of the existing maintenance models focus on single-component systems and are not applicable for complex systems consisting of multiple components, due to various interactions between the components. Multi-component maintenance optimization problem, which joins the stochastic processes regarding the failures of the components with the combinatorial problems regarding the grouping of maintenance activities, is challenging in both modeling and solution techniques, and has remained an open issue in the literature. In this paper, we study the multi-component maintenance problem over a finite planning horizon and formulate the problem as a multi-stage stochastic integer program with decision-dependent uncertainty. There is a lack of general efficient methods to solve this type of problem. To address this challenge, we use an alternative approach to model the underlying failure process and develop a novel two-stage model without decision-dependent uncertainty. Structural properties of the two-stage problem are investigated, and a progressive-hedging-based heuristic is developed based on the structural properties. Our heuristic algorithm demonstrates a significantly improved capacity in handling practically large-size two-stage problems comparing to three conventional methods for stochastic integer programming, and solving the two-stage model by our heuristic in a rolling horizon provides a good approximation of the multi-stage problem. The heuristic is further benchmarked with a dynamic programming approach and a structural policy, which are two commonly adopted approaches in the literature. Numerical results show that our heuristic can lead to significant cost savings compared with the benchmark approaches.

Key words: maintenance optimization, multi-component system, stochastic programming, progressive hedging algorithm, heuristic

1. Introduction

Effective maintenance plays an important role in maintaining high levels of productivity and safety in many capital-intensive industries, especially those that operate complex, hazardous systems, such as offshore oil and gas drilling systems, nuclear power plants, petrochemical plants, and space transport systems. A number of catastrophic failures, e.g., the space shuttle Challenger accident and the loss of Piper Alpha oil platform (Cowing et al. 2004), have occurred in part because of inadequate maintenance. Moreover, the downtime cost caused by either planned or unplanned maintenance shutdown in these industries is often significant. The production losses can range from \$5,000 to \$100,000 per hour during the shutdown in chemical plants and millions of dollars per day in offshore drilling/refineries (Amaran et al. 2016). As the demand for high reliability increases, it is more imperative to develop efficient maintenance schedules for complex systems.

Maintenance optimization has been extensively studied in the literature. However, most of the existing maintenance models focus on single-component systems, and are not applicable for complex systems consisting of multiple components, due to various interactions between the components. In general, there are three different types of interactions, economic, structural, and stochastic dependence (Thomas 1986). Economic dependence is the most common one among these three types of interactions. Systems with the economic dependence typically incur a common system-level cost, often referred to as setup cost, due to mobilizing repair crew, safety provisions, disassembling machines, special transportation, and the downtime loss. These costs are shared by all maintenance activities performed simultaneously. Considerable cost savings can be obtained by jointly maintaining several components instead of separately, especially when the setup cost is high.

Multi-component maintenance optimization problem, which joins the stochastic processes regarding the failures of the components with the combinatorial problems regarding the grouping of maintenance activities (Dekker et al. 1997, Scarf 1997, Dekker and Scarf 1998, Van Horenbeek and Pintelon 2013), is challenging in both modeling and solution techniques, and has remained an open issue in the literature. The problem can quickly end in complex models and explicit analytical expressions for optimal maintenance costs and the corresponding decisions are sometimes impossible to obtain. One often has to make special system assumptions (Tian and Liao 2011, Huynh et al. 2014), impose restrictions on maintenance grouping activities (Wildeman and Dekker 1997, Ding and Tian 2012, Van Horenbeek and Pintelon 2013), and/or resort to simulation tools (Barata et al. 2002, Laggoune et al. 2009) so that the decision problem can be formulated with less mathematical difficulty.

In this paper, we study the maintenance optimization problem for multi-component systems with economic dependence over a finite planning horizon. We aim to minimize the total maintenance cost by optimally determining maintenance decisions for all components at each decision period. We first formulate this problem as a multi-stage stochastic integer program. The system state transition probabilities at each stage are determined by not only the underlying failure processes but also maintenance decisions, and thus are decision-dependent. Such decision-dependent uncertainty is also referred to as endogenous uncertainty, and there is a lack of efficient methods to handle this type of problem (Peeta et al. 2010, Zhan et al. 2016, Apap and Grossmann 2017). We approximate the multi-stage model with a novel two-stage stochastic linear integer model in a rolling horizon. In both models, we do not restrict the types of maintenance activities that can be grouped or when the grouping can occur. In other words, joint execution of any combination of maintenance activities can occur anytime. A progressive-hedging-based heuristic is designed to solve practically large-size two-stage problems. Computational studies are performed to assess the performance of the heuristic. We further compare the results of the two-stage model with those of a direct-grouping model using a dynamic-programming approach (Van Horenbeek and Pintelon 2013) and a structural (t_i, T_i) policy (Wijnmalen and Hontelez 1997) over the rolling horizon. The main contributions of this paper are as follows.

(1) From a modeling perspective, the proposed multi-stage model and the two-stage model are sufficiently general, permitting grouping of any maintenance activities at any time and allowing general failure distributions. This work extends the multi-component maintenance literature by using a stochastic programming approach. Stochastic programming is a powerful modeling technique and facilitates the derivation of analytical expressions of the total cost function and maintenance decisions, which are difficult to obtain using commonly adopted approaches for the multi-component problem, such as dynamic programming.

(2) Formulating the multi-component maintenance problem as a stochastic program further enables the use of stochastic optimization tools. We design an efficient heuristic algorithm in the progressive hedging framework based on the problem structural properties of the two-stage model. Our heuristic algorithm demonstrates a significantly improved capacity in handling practically large-size two-stage problems comparing to three conventional methods for stochastic integer programming, and solving the two-stage model by

our heuristic in a rolling horizon provides a good approximation of the multi-stage problem. Using a rolling horizon scheme, we further assess the performance of our heuristic by comparing it with a dynamic programming approach and a structural (t_i, T_i) policy widely adopted in the literature. Numerical results show that the two-stage maintenance model and the designed heuristic can lead to substantial cost savings.

The remainder of this paper is organized as follows. Section 2 reviews multi-component maintenance and important solution methods in stochastic programming. In Section 3, we develop a multi-stage model stochastic maintenance model and approximate it with a two-stage model. Section 4 describes the progressive-hedging-based heuristic and three conventional algorithms in detail. Computational studies are presented in Section 5. We conclude this study and discuss future research directions in Section 6.

2. Literature review

We review two streams of literature that are relevant to our work: literature on multi-component maintenance and literature on solution methods in stochastic programming.

2.1. Multi-component maintenance

Common approaches to coordinating maintenance activities of multi-components include direct-grouping, indirect-grouping and opportunistic maintenance. Direct-grouping partitions the components into a number of fixed groups and then always maintain the components in a group jointly (van Dijkhuizen and van Harten 1996). The problem formulated with this approach is an NP-complete set-partitioning problem. The optimal grouping decision can be found for only a small number of components due to the computational complexity. There are some efforts that reduce the set-partitioning problem for multi-component maintenance to a dynamic-programming problem with a quadratic time complexity under some special assumptions (Dekker et al. 1996, Wildeman et al. 1997). Van Horenbeek and Pintelon (2013) and Vu et al. (2014) extend Dekker et al. (1996) and Wildeman et al. (1997) by considering dynamic information, e.g., usage of components and environmental conditions. A major deficiency of this approach is that grouping activity iteratively takes place within a time window that is often determined by the maximum individual maintenance interval among all components. Within this window, each component is preventively maintained only one time. This assumption is not relevant since a system may be composed of different components with different lifetime cycles, and maintenance intervals of components can be significantly different (Laggoune et al. 2009).

Unlike the direct-grouping that yields a fixed group structure, indirect grouping usually groups preventive maintenance (PM) activities by making the PM interval a multiple of a basis interval, so the maintenance of different components can coincide (Goyal and Gunasekaran 1992, Schouten et al. 1998). An alternative indirect grouping strategy performs major PM on all components jointly at the end of a common interval and allows minor or major PM within this interval. Indirect grouping model of this type is sometimes formulated as a mixed integer programming (MIP) problem (Epstein and Wilamowsky 1985, Hariga 1994). Because of the simplified policy structure, the MIP model can be separated by components, which greatly reduces the computational complexity.

Both direct- and indirect-grouping focus on grouping PM activities, and ignore maintenance opportunities generated by corrective maintenance (CM) at failures. To take advantage of the time window of CM and use it as opportunities for PM of other functioning components, many opportunistic maintenance (OM) models have been proposed. Ding and Tian (2012) and Koochaki et al. (2012) use a simulation-based optimization method to find optimal OM policies. Shafiee and Finkelstein (2015) consider a simplified OM policy that preventively replaces all non-failed components when there is a failure. More recently, Patriksson et al. (2015) use a stochastic programming approach in OM. However, the integer L-shaped method proposed in Patriksson et al. (2015) cannot solve large-scale problems. Castanier et al. (2005) consider a condition-based OM policy and formulate it as a semi-regenerative process. This approach also suffers the computational intractability, because the problem size grows exponentially as the number of components increases. As a result, their analysis is limited to a two-component system. For more details regarding the multi-component maintenance problem, the readers are referred to review papers by Dekker et al. (1997), and Nicolai and Dekker (2008).

2.2. Solution methods in stochastic programming

In this paper, we formulate the multi-component maintenance optimization problem as a stochastic integer program. Various decomposition methods have been developed to solve stochastic integer programs. Benders decomposition (Birge and Louveaux 2011, Bodur et al. 2017) and progressive hedging algorithm (PHA) (Rockafellar and Wets 1991, Watson and Woodruff 2011, Gade et al. 2016) are two important decomposition methods for solving stochastic integer programming problems. Benders decomposition vertically decomposes a problem into a master problem that only concerns first-stage decisions and subproblems that include second-stage decisions of all scenarios. Benders cut and integer L-shaped cut are two major types of cuts that are added within Benders decomposition framework. However, Benders cut may become useless since strong duality does not hold in an integer program, and integer L-shaped cut is typically inefficient because every feasible solution may need an integer L-shaped cut in the worst-case scenario. The PHA decomposes the extensive form according to scenario, and iteratively solving penalized versions of the subproblems to gradually enforce non-anticipativity. The performance of PHA, to a great extent, depends on how efficient each subproblem is solved in a stochastic integer program. In our problem, each scenario subproblem with deterministic individuals' lifetimes is essentially an NP-complete set-partitioning problem. Efficient heuristic algorithm for each subproblem is needed.

Our review of the literature shows that few research has considered grouping at both preventive and corrective maintenance occasions under practical assumptions, which significantly affects the optimality of the solutions because of the simplified models and reduced solution space. There is also a lack of efficient algorithms that can provide satisfactory results for practically large-scale multi-component maintenance problems.

3. Model development

3.1. Problem statement

In this paper, we consider maintenance optimization for a multi-component system that consists of $\mathcal{N} = \{1, 2, \ldots, n\}$ components with economic dependence. The objective is to minimize total expected maintenance cost in a finite planning horizon. We consider two types of maintenance, preventive replacement (PR) and corrective replacement (CR). At each decision period, maintenance decisions need to be made for all components. Any maintenance activities of any components can be performed together to save the setup cost and improve the system performance. Note that any failed component is correctively replaced. Both CR and PR use a new component and are therefore perfect. Components' failure models can be any failure distribution.

3.2. Multi-stage stochastic maintenance model

We consider a discretized finite planning horizon $\mathcal{T} = \{0, 1, \ldots, T\}$, where the length of a decision period is δ. Each component in the system is considered as a different type of component regardless of its physical type. To distinguish the component type and the component itself, we use component only when referring to its type and refer to physical components as individuals. For example, individual I_{ir} is the individual used for the r^{th} maintenance replacement of type i component. We denote the set of individuals by $R = \{1, 2, \ldots, q\}$, where q is the maximum number of individuals replaced for all components. Denote the PR cost and CR cost for each component by $c_{i,\text{pr}}$ and $c_{i,\text{cr}}$ respectively, and assume $c_{i,\text{pr}} < c_{i,\text{cr}}$ for all $i \in \mathcal{N}$. The system setup cost is d at any maintenance occasion. If n individuals are replaced at the same time, the total savings from executing these n maintenance activities jointly is $d(n–1)$.

This problem is naturally a multi-stage stochastic programming problem. Denote the state for the operating individual of component $i \in \mathcal{N}$ at stage $t \in \mathcal{T}$ by ξ_{it}, i.e., $\xi_{it} = 1$ if the operating individual fails and 0 otherwise. Let a_{it} be the age of the operating individual of component i prior to the maintenance decision at stage t, and $a_t = (a_{1t}, a_{2t}, \ldots, a_{nt})$ be the age vector of all operating individuals. At each decision stage t, after observing the states $\xi_t = (\xi_{1t}, \xi_{2t}, \ldots, \xi_{nt})$ of all operating individuals, we first correctively replace all failed individuals and then select a group of individuals for PR if desired.

Denote the decision variables by

$$x_{it} = \begin{cases} 1 \text{ ,if an individual of component } i \text{ is replaced at stage } t \\ 0 \text{ ,otherwise} \end{cases}, i \in \mathcal{N}, t \in \mathcal{T}$$

and

$$z_t = \begin{cases} 1 \text{ ,if any maintenance occurs at stage } t \\ 0 \text{ ,otherwise} \end{cases}, t \in \mathcal{T}$$

The multi-stage stochastic model (P1) is defined as follows:

$$f_0(a_0, \xi_0) = \min \sum_{i \in \mathcal{N}} c_{i,\text{pr}} x_{i,0} + \sum_{i \in \mathcal{N}} (c_{i,\text{cr}} - c_{i,\text{pr}}) \xi_{i,0} + d z_0 + V_0(x_0, a_0) \tag{1}$$

subject to

$$V_t(x_t, a_t) = \begin{cases} \mathbb{E}_{\xi_{t+1}}[f_{t+1}(a_{t+1}, \xi_{t+1})] & , t \in \mathcal{T} \backslash \{T\} \\ 0 & , t = T \end{cases} \tag{2}$$

$$a_{t+1} = a_t(1 - x_t) + \delta, t \in \mathcal{T} \backslash \{T\} \tag{3}$$

$$x_{it} \geq \xi_{it}, i \in \mathcal{N}, t \in \mathcal{T} \tag{4}$$

$$z_t \geq x_{it}, i \in \mathcal{N}, t \in \mathcal{T} \tag{5}$$

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

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

Objective function (1) includes maintenance cost at the first stage and the expected minimum cost at the second stage. The expected minimum second-stage cost is given by Constraints (2). Constraints (3) provide the age of each work individual at each decision stage. Constraints (4) ensure that the indicator of replacement x_{it} is 1 when an individual failed. Constraints (5) force that setup cost is incurred whenever a replacement is performed. Constraints (6) and (7) are integrality constraints.

The maintenance decision at each decision stage influences the system state transition probability. Scenario tree in Figure 1 illustrates the interactions between maintenance decision and the underlying stochastic failure process of a two-component system. Model P1 is therefore a stochastic integer program with decision-dependent uncertainty (i.e., endogenous uncertainty). This type of problem is difficult to solve. First, general efficient method for stochastic integer programs is lacking. Second, the problem size grows exponentially as the number of components and/or decision stages increases. Third, there is a lack of general efficient algorithms to solve a stochastic program with endogenous uncertainty. Next, we use a novel two-stage model to approximate P1 and design an efficient heuristic to find high-quality solutions.

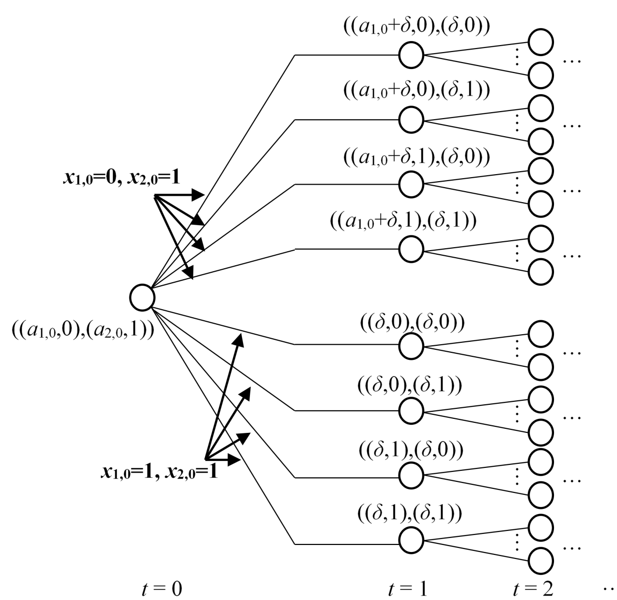


Figure 1 Decision-dependent scenario tree

3.3. Two-stage stochastic maintenance model

A common approach to approximating a multi-stage stochastic program is to utilize a two-stage model in a rolling horizon. The two-stage approximation model is usually obtained by combining all future periods together as a second stage problem with all future's non-anticipativity constraints removed (Beraldi et al. 2011).

In this two-stage model, instead of using failure probability as the underlying stochastic element to generate scenarios, we use the random lifetimes as the equivalent stochastic element to capture the uncertainty. Specifically, let T_{ir}^{ω} be the lifetime of individual I_{ir} that is drawn from its failure distribution. A scenario $\omega \in \tilde{\Omega}$ is a lifetime combination of all individuals of all components, i.e., $(T_{1,1}^{\omega}, \ldots, T_{1,q}^{\omega}, \ldots, T_{n,1}^{\omega}, \ldots, T_{nq}^{\omega})$, where $\tilde{\Omega}$ is the collection of all possible scenarios. Note that the endogenous uncertainty is removed by using this alternative approach to describe the uncertainty.

The two-stage model can be represented by

$$\min_{x \in X} \sum_{\omega \in \tilde{\Omega}} p(\omega) F(x, \omega),$$

where x is the vector of decision variables and X is the feasible region. Function $F(x,\omega)$ is the total maintenance cost given the lifetime combination in scenario $\omega \in \tilde{\Omega}$, and $p(\omega)$is the probability of scenario ω. Since the realization of lifetime T_{ir}^{ω} is infinite for the majority of lifetime distributions, e.g., Weibull distribution, the total number of scenarios $|\tilde{\Omega}|$ is infinite. We therefore use the sample average approximation (SAA) method (Kleywegt et al. 2002) to approximate the two-stage model. Specifically, we have

$$\sum_{\omega \in \tilde{\Omega}} p(\omega) F(x, \omega) \approx \sum_{\omega \in \Omega} p(\omega) F(x, \omega),$$

where $\Omega \subset \tilde{\Omega}$ and $p(\omega) = \dfrac{1}{|\Omega|}$.

Before presenting function $F(x,\omega)$, we first introduce decision variables in the two-stage model. Let

$$x_i = \begin{cases} 1, \text{if an individual of component } i \text{ is replaced at the first stage} \\ 0, \text{otherwise} \end{cases}, i \in \mathcal{N},$$

$$\tilde{x}_{it}^{r\omega} = \begin{cases} 1, \text{if } I_{ir} \text{ is replaced at stage } t \text{ in scenario } \omega \\ 0, \text{otherwise} \end{cases}, i \in \mathcal{N}, t \in \mathcal{T}, r \in R, \omega \in \Omega,$$

and

$$z_t^{\omega} = \begin{cases} 1, \text{if maintenance occurs at stage} t \text{ in scenario } \omega \\ 0, \text{otherwise} \end{cases}, t \in \mathcal{T}, \omega \in \Omega.$$

To facilitate the model development, we introduce two auxiliary binary variables $Y_i^{r\omega}$ and $w_{it}^{r\omega}$ based on $\tilde{x}_{it}^{r\omega}$. The variable $Y_i^{r\omega}$ is an indicator of the maintenance type. More details regarding these auxiliary variables will be discussed in Section 3.3.1. The deterministic extensive form (DEF) of the two-stage model, which explicitly describes the second-stage decision variables for all scenarios (Birge and Louveaux 2011), is formulated as follows.

Model DEF:

$$\min \sum_{\omega\in\Omega}\left(p(\omega)\left(\sum_{i\in\mathcal{N}}\left(\underbrace{\sum_{r=1}^{q}(c_{i,\text{pr}}Y_i^{r\omega})}_{C_{i,\text{pr}}}+\underbrace{\sum_{r=1}^{q}(c_{i,\text{cr}}(1-Y_i^{r\omega}))-c_{i,\text{cr}}(1-\tilde{x}_{iT}^{r\omega})}_{C_{i,\text{cr}}}\right)\right)+\underbrace{\sum_{t=0}^{T}dz_t^{\omega}}_{C_\text{s}}\right) \quad (8)$$

subject to

$$\tilde{x}_{it}^{r\omega}\le\tilde{x}_{i,t+1}^{r\omega}, i\in\mathcal{N}, t\in\mathcal{T}\backslash\{T\}, r\in R, \omega\in\Omega \quad (9)$$

$$\tilde{x}_{i,t+1}^{r+1,\omega}\le\tilde{x}_{it}^{r\omega}, i\in\mathcal{N}, t\in\mathcal{T}\backslash\{T\}, r\in R\backslash\{q\}, \omega\in\Omega \quad (10)$$

$$\sum_{r\in R}(\tilde{x}_{it}^{r\omega}-\tilde{x}_{i,t-1}^{r\omega})\le z_t^{\omega}, i\in\mathcal{N}, t\in\mathcal{T}\backslash\{0\}, \omega\in\Omega \quad (11)$$

$$\tilde{x}_{i,0}^{1,\omega}\le z_0^{\omega}, i\in\mathcal{N}, \omega\in\Omega \quad (12)$$

$$\tilde{x}_{it}^{r\omega}\le\tilde{x}_{i,t+T_{i,r+1}^{\omega}}^{r+1,\omega}, i\in\mathcal{N}, t\in\{0,\dots,T-T_{i,r+1}^{\omega}\}, r\in R\backslash\{q\}, \omega\in\Omega \quad (13)$$

$$\tilde{x}_{i,T_{i,1}^{\omega}}^{1,\omega}=1, i\in\{j\in\mathcal{N}|T_{j,1}^{\omega}\le T\}, \omega\in\Omega \quad (14)$$

$$\tilde{x}_{i,0}^{r\omega}=0, i\in\mathcal{N}, r\in R\backslash\{1\}, \omega\in\Omega \quad (15)$$

$$x_i=\tilde{x}_{i,0}^{1,\omega}, i\in\mathcal{N}, \omega\in\Omega \quad (16)$$

$$x_i\ge\xi_i, i\in\mathcal{N} \quad (17)$$

$$Y_i^{1,\omega}=1-w_{i,T_{i,1}^{\omega}}^{1,\omega}, i\in\mathcal{N}, \omega\in\Omega \quad (18)$$

$$Y_i^{r\omega}=\frac{(\sum_{t=T_{ir}^{\omega}}^{T+T_{ir}^{\omega}}|y_{it}^{r\omega}|+\sum_{t=0}^{T_{ir}^{\omega}-1}w_{it}^{r\omega})}{2}, i\in\mathcal{N}, r\in R\backslash\{1\}, \omega\in\Omega \quad (19)$$

$$y_{it}^{r\omega}=w_{it}^{r\omega}-w_{i,t-T_{ir}^{\omega}}^{r-1,\omega}, i\in\mathcal{N}, r\in R\backslash\{1\}, t\in\{T_{ir}^{\omega},\dots,T'\}, \omega\in\Omega \quad (20)$$

$$w_{it}^{r\omega}=\tilde{x}_{it}^{r\omega}-\tilde{x}_{i,t-1}^{r,\omega}, i\in\mathcal{N}, r\in R, t\in\mathcal{T}\backslash\{0\}, \omega\in\Omega \quad (21)$$

$$w_{i,0}^{r\omega}=\tilde{x}_{i,0}^{r\omega}, i\in\mathcal{N}, r\in R, \omega\in\Omega \quad (22)$$

$$w_{it}^{r\omega}=0, i\in\mathcal{N}, r\in R, t\in\{T+1,\dots,T'\}, \omega\in\Omega \quad (23)$$

$$\tilde{x}_{it}^{r\omega}\in\{0,1\}, i\in\mathcal{N}, r\in R, t\in\mathcal{T}, \omega\in\Omega \quad (24)$$

$$x_i\in\{0,1\}, i\in\mathcal{N} \quad (25)$$

$$z_t^{\omega}\in\{0,1\}, t\in\mathcal{T}, \omega\in\Omega \quad (26)$$

$$w_{it}^{r\omega}\in\{0,1\}, i\in\mathcal{N}, r\in R, t\in\{0,\dots,T'\}, \omega\in\Omega \quad (27)$$

$$Y_i^{r\omega}\in\{0,1\}, i\in\mathcal{N}, r\in R, \omega\in\Omega \quad (28)$$

Function (8) is the objective function. Decision variables x_i and z concern maintenance decisions at the first stage, and $\tilde{x}_{it}^{r\omega}$ and z_t^{ω} are the second-stage decisions for scenario ω. We provide detailed derivation of the objective function in Section 3.3.1 and explain the constraints in Section 3.3.2.

3.3.1. Derivation of the objective function

In objective function (8), the total cost includes: (1) sum of the PR and CR costs incurred by individuals of component i in the planning horizon, denoted by $C_{i,\text{pr}}$ and $C_{i,\text{cr}}$ respectively, and (2) total system setup cost C_s. We break the derivation of the total cost function into the calculations of these cost elements.

- **Derivation of $C_{i,\text{pr}}$**

For component i, the total cost of individuals preventively replaced over the planning horizon is given by $C_{i,\text{pr}} = \sum_{r=1}^{q} c_{i,\text{pr}} Y_i^{r\omega}$, where $Y_i^{r\omega}$ is defined in constraints (18) and (19).

Next, we explain why $Y_i^{r\omega}$ can be used to identify the replacement type and this determination is a key element of the model development. We drop the superscript ω in the following discussions for notational convenience. It is obvious that the decision variables $\tilde{x}_{it}^{r}$ and w_{it}^{r} only concern when a placement is performed and have no indication on the type of replacement. For an individual I_{ir}, one way to determine its replacement type is to examine the time interval between the replacements of individuals $I_{i,r-1}$ and I_{ir}. Suppose that individuals $I_{i,r-1}$ and I_{ir} are replaced at times t_1 and t_2 (i.e., $w_{i,t_1}^{r-1} = 1$ and $w_{i,t_2}^{r-1} = 1$), respectively. If the difference between t_2 and t_1 equals to the lifetime of I_{ir}, namely T_{ir}, then I_{ir} is replaced at the end of its lifetime and the replacement type is CR. The replacement is PR otherwise. Therefore, if CR is performed on this individual, we have $w_{it}^{r} - w_{i,t-T_{ir}}^{r-1} = 0$ $\forall t \in \mathcal{T}$, which leads to $\sum_{t=0}^{T} |y_{it}^{r}| = \sum_{t=0}^{T} |w_{it}^{r} - w_{i,t-T_{ir}}^{r-1}| = 0$ (Figure 2(a)). If PR is performed on this individual, then $w_{i,t_2}^{r} - w_{i,t_2-T_{ir}}^{r-1} = 1$, $w_{i,t_1+T_{ir}}^{r} - w_{i,t_1}^{r-1} = -1$, and $w_{i,t+T_{ir}}^{r} - w_{it}^{r-1} = 0$ for all $t \in \{t | t \in \mathcal{T}, t \neq t_1, t \neq t_2\}$ (Figure 2(b)), and consequently, $\sum_{t=0}^{T} |y_{it}^{r}| = 2$. This makes the value of $\dfrac{\sum_{t=0}^{T} |y_{it}^{r}|}{2}$ a good indicator for determining the replacement type, and $\sum_{t=0}^{T} |y_{it}^{r}|$ is calculated as follows:

$$\sum_{t=0}^{T} |y_{it}^{r}| = \sum_{t=0}^{T} |w_{it}^{r} - w_{i,t-T_{ir}}^{r-1}| \tag{29}$$

However, there are two boundary issues in Equation (29).

(1) Decision times of w_{it}^r need to be extended beyond $\mathcal{T}$. This is because Equation (29) does not count the maintenance decisions made for individual $I_{i,r-1}$ at times $\{T–T_{ir},\ldots,T\}$. To include these decisions, the planning horizon for w_{it}^r needs to extend to $T' = T + \max\{T_{ir}, i \in \mathcal{N}, r \in R\}$, and let $w_{it}^r = 0$ for $t > T$. See the region labeled "Not Defined" in Figure 3 for an illustration.

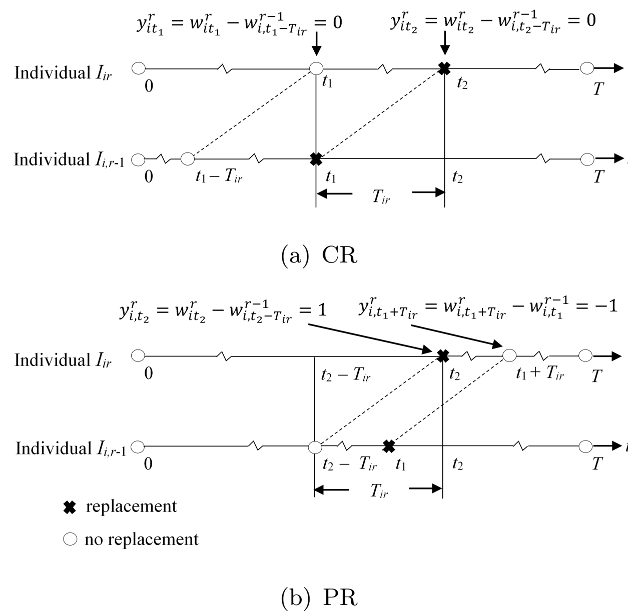


Figure 2 Illustration of distinguishing CR and PR

(2) In Equation (29), the decision times considered for individual I_{ir} implicitly start from T_{ir}, and all decisions made before T_{ir} are excluded (illustrated in the region labeled "Excluded"). To recover decisions made for individual I_{ir} at times $\{0,\ldots,T_{ir}-1\}$, we add $\sum_{t=0}^{T_{ir}-1} w_{it}^r$ to Equation (29). Equation (29) is now rewritten as follows,

$$Y_i^{r\omega} = \sum_{t=T_{ir}^{\omega}}^{T+T_{ir}} |y_{it}^{r\omega}| + \sum_{t=0}^{T_{ir}^{\omega}-1} w_{it}^{r\omega}, i \in \mathcal{N}, r \in R, \omega \in \Omega.$$

The absolute function, $|y_{it}^{r\omega}|$, can be linearized by a pair of deviation variables (Rardin 1998).

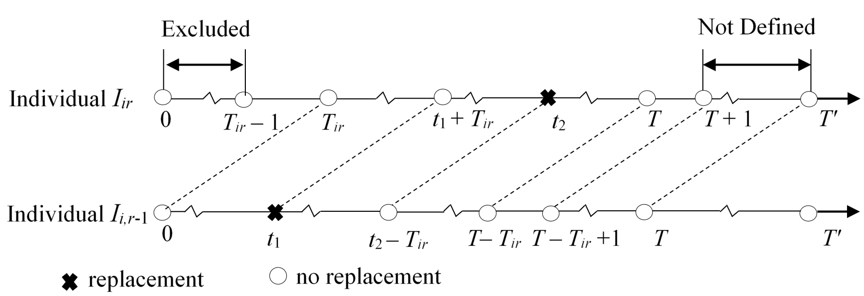


Figure 3 Illustration of the boundary issue of Equation (29)

- **Derivation of $C_{i,\mathbf{cr}}$ and $C_{\mathbf{s}}$**

For component i, the total cost of individuals correctively replaced over the planning horizon is given by $C_{i,\mathrm{cr}} = \sum_{r=1}^{q}(c_{i,\mathrm{cr}}(1-Y_i^{r\omega}) - c_{i,\mathrm{cr}}(1-\tilde{x}_{iT}^{r\omega}))$. As explained in the derivation of $C_{i,\mathrm{pr}}$, the expression $Y_i^{r\omega}=0$ implies a CR for individual I_{ir}. However, for any component type, the number of individuals used for replacement is unknown due to the unknown maintenance decisions, and therefore the maximum number of individuals needed for any component is considered in the optimization model. It is likely that some individuals are not used in the planning horizon. If neither individual $I_{i,r-1}$ nor I_{ir} is used for replacement during the planning horizon, the value of $Y_i^{r\omega}$ is also zero. We need to distinguish these two scenarios that both have $Y_i^{r\omega}=0$. This can be done by examining the value of $\tilde{x}_{iT}^{r\omega}$. If an individual is not used, we have $\tilde{x}_{iT}^{r\omega}=0$ and $\tilde{x}_{iT}^{r\omega}=1$ otherwise. The false corrective cost caused by an individual that is not used is $c_{i,\mathrm{cr}}(1-\tilde{x}_{iT}^{r\omega})$, and needs to be subtracted from the total cost, $C_{i,\mathrm{cr}}$. Lastly, the total setup cost over the planning horizon is $C_{\mathrm{s}} = \sum_{t\in\mathcal{T}} dz_t^{\omega}$.

3.3.2. Constraints

Constraints (9) are the definition of $\tilde{x}_{it}^{r\omega}$, which ensure that individual I_{ir} is replaced at or before $t+1$ when it is replaced at or before t. Constraints (10) imply that individual $I_{i,r+1}$ can only be replaced after I_{ir} is replaced. Constraints (11) and (12) ensure that maintenance cost d incurs when any component is replaced at time t. Constraints (13) and (14) ensure that individual I_{ir} is replaced at the latest when it has been inside the system for T_{ir}^{ω} time units. In other words, I_{ir} has to be replaced before or at the end of its lifetime. Constraints (15) imply that only the first individual can be replaced at time 1. Constraints (16) impose the non-anticipativity constraint, forcing the decisions at the first stage to be the same. The constraints (17) force all failed components at the first stage to be replaced. Constraints (18) and (19) define the auxiliary variable $Y_i^{r\omega}$. Constraints (20) provide the definition of variable $y_{it}^{r\omega}$. Constraints (21) – (23) are the definition of variable $w_{it}^{r\omega}$. The remaining constraints (24) – (28) are binary constraints.

4. Optimization algorithms

All decision variables in DEF are binary. Properties of stochastic integer programs are scarce, and general efficient methods are lacking. We therefore design a heuristic algorithm under the framework of PHA to solve practical-size problems within moderate CPU time. To assess the performance of the proposed heuristic algorithm in solving DEF, we compare the performance of the proposed algorithms with three conventional algorithms, namely, basic Benders decomposition (Algorithm 1), integer L-shaped method with Benders cuts (Algorithm 2) and standard PHA (Algorithm 3). The basic Benders decomposition and integer L-shaped method with Benders cuts are considered because of LP relaxation and branch-and-cut are common methods for solving integer programs. Standard PHA, which decomposes a problem by scenarios, provides a flexible framework for stochastic integer problem, and is also considered for comparison.

4.1. The benchmark algorithms

4.1.1. Basic Benders decomposition algorithm

The basic Benders algorithm first solves the Benders master integer problem, and then solves the LP relaxation of subproblems to generate cuts which are added back to Benders master problem (Birge and Louveaux 2011). The procedure is repeated until no cuts found. We first define the initial master problem (MP) as

$$\text{MP:} \min dz + \sum_{i \in \mathcal{N}} c_{i,\text{pr}} x_i + \sum_{i \in \mathcal{N}} (c_{i,\text{cr}} - c_{i,\text{pr}}) \xi_i + \sum_{\omega \in \Omega} p(\omega) \theta_\omega$$

subject to

$$\theta_\omega \geq Q(x, \omega), x_i \geq \xi_i, z \geq x_i, i \in \mathcal{N}, \omega \in \Omega$$
$$x_i, z \in \{0, 1\}, i \in \mathcal{N},$$

where $Q(x, \omega)$ is the objective value of the second-stage problem with respect to scenario ω, given by

$$Q(x, \omega) = \min \sum_{i \in \mathcal{N}} (C_{i,\text{pr}} + C_{i,\text{pr}} + C_{i,\text{cr}} - c_{i,\text{pr}} x_i - (c_{i,\text{cr}} - c_{i,\text{pr}}) \xi_i) + C_\text{s} - dz$$

and subject to constraints (9) to (28) except (12), (17), and (25). Note that all decision variables should be LP-relaxed. Constraints (12), (17) and (25) are excluded from the subproblem since they only concern the decision variables in the first stage. For each scenario ω, Benders cut can be written as

$$\theta_\omega \geq \tilde{e}_{m\omega}^\text{T} x - e_{m\omega}, m \in \{1, \ldots, M\}, \omega \in \Omega, \tag{30}$$

where $\tilde{e}_{m\omega}^\text{T} x - e_{m\omega}$ can be viewed as the dual objective function of problem $Q(x, \omega)$ with dual solutions obtained and unknown first-stage decisions x at iteration m for scenario ω, and M denotes the maximum iterations.

Algorithm 1 Basic Benders decomposition

1: **Initialization:** $\theta_\omega \leftarrow -\infty$, $\forall \omega \in \Omega$, $\tilde{\epsilon} \leftarrow 10^{-2}$, and assign an integer feasible x to the subproblem.
2: Solve the LP relaxation of the subproblem, $Q(x,\omega)$, for each $\omega \in \Omega$.
3: **if** $\theta_\omega - Q(x,\omega) \le \tilde{\epsilon}\ \forall \omega \in \Omega$ **then return** Optimal solution:$(x^*, \theta^*_\omega) \leftarrow (x, \theta_\omega)$. **end if**
4: Add Benders cuts using Equation (30) into the MP.
5: Solve the MP to get new $(x, \theta_\omega)\ \forall \omega \in \Omega$. Go to step 2.

4.1.2. Integer L-shaped method with Benders cuts

In Algorithm 2, we initialize Benders master problem with Benders cuts. More specifically, the root node is obtained by solving the LP relaxation of the master problem via Benders decomposition and keeping the cuts. In the branch-and-cut process, at each node, if the solution is integer feasible, the subproblem is solved to generate integer optimality cuts which are defined as follows (Laporte and Louveaux 1993):

$$\theta \ge (Q(x^*) - L)(\sum_{i \in S(x^*)} x_i - \sum_{i \notin S(x^*)} x_i - |S(x^*)|) + Q(x^*) \tag{31}$$

where $Q(x) = \sum_{\omega \in \Omega} p(\omega) Q(x,\omega)$, L is a lower bound of $Q(x)$, and $S(x^*) := \{i | x^*_i = 1\}$ given a first-stage solution $x^* \in \{0,1\}^n$.

In addition to the integer optimality cuts, Benders cuts are also generated and added into the MP if violated by the candidate solution, in order to improve the performance of the Integer L-shaped method. Therefore, for each node in the branch-and-cut search tree, if the candidate solution is integer feasible, both Benders cuts and integer optimality cuts are added via lazy constraint callback routine, otherwise only Benders cuts are added by using user-cut callback routine.

4.1.3. Standard progressive hedging algorithm

We also examine the performance of the standard PHA on our problem. The PHA mitigates the computational difficulty associated with large problem instances by decomposing the extensive form according to scenario, and iteratively solving penalized versions of the subproblems to gradually enforce non-anticipativity (Aydin 2012). Solving individual scenario subproblems separately is generally much less computationally challenging and may allow a solver to exploit any special combinatorial structure that may be present. Moreover, the time expended for each iteration can be dramatically reduced by a very straightforward parallelization.

Algorithm 2 Integer L-shaped method with Benders cuts

```
Initialization: θ* ← +∞;
    Initialize the MP by solving the LP relaxation via Benders, and keep cuts ⇒ (x, θ).
Branch and Cut:
At each node in the search tree:
  Solve LP relaxation ⇒ (x, θ).
  if LP bound exceeds known incumbent θ* then Prune. end if
  if x is integer feasible then
    Solve subproblem Q(x) to generate integer optimality cuts using Equation (31).
    Solve LP relaxation of the subproblem Q(x) to generate Benders cuts.
    if (x, θ) violates any Benders cut or integer optimality cut then
      Add cut to LP relaxation of the MP and resolve.
    else Update the incumbent, θ* ← θ. end if
  end if
  if x is not integer feasible then
    Solve LP relaxation of the subproblem Q(x) to generate Benders cuts.
    if (x, θ) violates any Benders cut then Add cut to LP relaxation and resolve.
    else Branch to create new nodes. end if
  end if
```

Specifically, the PHA proceeds by relaxing the non-anticipativity constraints using augmented Lagrangian relaxation and the problem becomes separable by each scenario. The scenario subproblems have augmented objective functions which include Lagrangian penalty functions corresponding to the relaxed non-anticipativity constraints. At each iteration of the PHA algorithm, these scenario subproblems are solved as deterministic problems. Solutions from all scenario subproblems are then collected and averaged according to their non-anticipativity constraints and scenario probabilities. The deviation of each scenario subproblem solution from these averages is used to update the Lagrangian multipliers. Next, the scenario subproblems are re-solved with the updated augmented Lagrangian objective function. This iterative process continues until the Lagrangian dual problem converges to a solution satisfying the non-anticipativity constraints.

Details of the PHA are described in Algorithm 3. A different form of the objective function, $cx + \mathbb{E}(Q(x,\omega))$ is used for a concise presentation of the algorithm (Gade et al. 2016).

Algorithm 3 The standard PHA

1: **Initialization:** Let $v \leftarrow 0, \tilde{\epsilon} \leftarrow 10^{-2}$, $x_\omega^v \leftarrow \arg\min_x(cx + Q(x,\omega))$, $\bar{x}^v \leftarrow \sum_{\omega\in\Omega} p(\omega)x_\omega^v$, and $W_\omega^v \leftarrow \rho(x_\omega^v - \bar{x}^v)$, $\forall \omega \in \Omega$.
2: **Update the iteration counter:**$v \leftarrow v+1$.
3: **Decomposition:** $x_\omega^v \leftarrow \arg\min_x(cx + W_\omega^{v-1}x + \frac{\rho}{2}||x - \bar{x}^{v-1}||^2) + Q(x,\omega), \forall \omega \in \Omega$.
4: **Aggregation**: $\bar{x}^v \leftarrow \sum_{\omega\in\Omega} p(\omega)x_\omega^v$.
5: **Update price**: $W_\omega^v \leftarrow W_\omega^{v-1} + \rho(x_\omega^v - \bar{x}^v), \forall \omega \in \Omega$.
6: **Calculate converge distance**: $g^v \leftarrow \sum_{\omega\in\Omega} p(\omega)||x_\omega^v - \bar{x}^v||, \forall \omega \in \Omega$.
7: **Termination**: **if** $g^v < \tilde{\epsilon}$ **then return** Optimal solution $\bar{x}^v$. **else** Go to step 2. **end if**

4.2. Progressive-hedging-based heuristic algorithm

Algorithm 1 cannot provide meaningful results due to the LP relaxation employed, and becomes more difficult and time-consuming as more cuts are added. Algorithm 2 is also computationally intensive as the number of binary variables and constraints increases. Standard PHA similarly suffers the computational intractability, since even for a small-scale multi-component maintenance problem, the scenario subproblem in the DEF can have a large number of decision variables and constraints, beyond what commercial solvers can handle. However, PHA provides a flexible framework for solving stochastic integer problems. To address the bottleneck in solving the scenario subproblem using the standard PHA, i.e., step 3 in Algorithm 3, we develop an efficient heuristic algorithm based on the problem structure for the scenario subproblems, and use the PHA framework as a "wrapper" to force non-anticaptivity constraints.

The basic idea of our heuristic algorithm is as follows. Given a scenario subproblem, the heuristic iteratively groups maintenance activities of *operating* individuals to reduce the setup costs and ultimately reduce the total maintenance costs. At each iteration it first obtains tentative replacement schedules for all operating individuals without considering economic dependence, and then considers a shifting window and groups maintenance activities within the shifting window. The tentative replacement schedule and the shift window are optimized to find the lowest total maintenance cost. The time complexity of the heuristic algorithm is polynomial (see Online Supplement OS2 for proof).

Before describing the details of the heuristic, we first present two properties regarding the optimal solution.

Theorem 1. *For each scenario subproblem, there exists an optimal solution such that at each decision period $t \in \mathcal{T}$, if there is any group of individuals (including one-individual* *group) that is maintained, there is at least one individual in the group that is replaced at one time unit before its failure or at its failure, except for all types of components' last individuals that are replaced in the planning horizon. (Proof is in Online Supplement OS3).*

Theorem 2. *Given a set of operating individuals sorted according to their failure times, there exists an optimal solution for this set such that maintenance activities are executed following the same order. (Proof is shown in Online Supplement OS4).*

Theorem 1 helps determine tentative replacement schedules for each individual. Based on Theorem 1, we only need to consider two tentative replacement schedules for individuals, i.e., replacing onetime unit before a failure or at the failure. Theorem 2 further ensures that it is optimal to execute the replacement activities for all operating individuals in the order they are tentatively planned according to Theorem 1. Theorems 1 and 2 significantly decrease the number of possible feasible solutions needed to be considered in the heuristic, and thus substantially reduce the algorithm complexity.

The details of the heuristic are described as follows. Let K denote the set of all operating individuals at the current iteration of the heuristic. Let β_{ij} and β'_{ij} denote the tentative and actual replacement times of individual I_{ij}, respectively. Let K' represent the sorted set of K according to β_{ij}, and $K'[i]$ represent the individual in the i^{th} position in set K'. For example, consider a four-component system, and the tentative replacement times of four individuals at one iteration are provided in Figure 4. In this example, we have $K = \{I_{1,5}, I_{2,3}, I_{3,2}, I_{4,4}\}$ and $K' = \{I_{2,3}, I_{1,5}, I_{4,4}, I_{3,2}\}$.

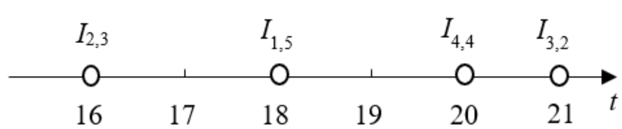
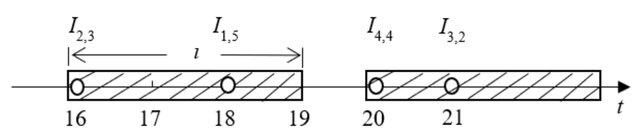


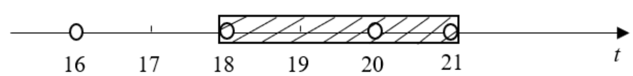
Figure 4 Operating individuals at one iteration

Given a shifting window ι and an operating individual set K, the **Grouping Rule** constructs candidate grouping options and selects the one with the minimum weighted PHA replacement cost cumulated till the current iteration. Specifically, the first candidate grouping option starts from $K'[1]$ and groups all current operating individuals with a tentative replacement time between $\beta_{K'[1]}$ and $\beta_{K'[1]} + \iota$. Let $K'[\nu]$ represent the last individual grouped with $K'[1]$. The next shifting window starts from $K'[\nu+1]$ and ends at $K'[\nu+1]+\iota$. operating individuals with a tentative replacement time in this window are grouped together. When moving an individual of a component to an earlier time for maintenance so that it can be grouped with an individual(s) of other component(s), it is possible that one or more new individuals of that component are needed to cover the planning horizon. Therefore, we consider a penalty cost: If individual I_{ir} is grouped with I_{jr} ($\beta_{ir} > \beta_{jr}$), then the penalty cost is $r_i(c_{i,\text{pr}} + d)$, where r_i is the number of new individuals needed to cover the planning horizon due to this shift. The grouping process of the first option continues until $\nu = |K'|$, implying no more individual can be grouped. The second grouping option starts from $K'[2]$ and the same grouping process is repeated. The total number of group options is $|K'|$–1. The replacement time of any group in the optimal grouping option for set K becomes actual replacement time of individuals within this group. These individuals are replaced with their immediate successors. Note that if the new individual has a tentative replacement schedule beyond the planning horizon, it is removed from set K. Now we have a new set of operating individuals and the **Grouping Rule** is applied for the new set. The heuristic stops when set K is empty.

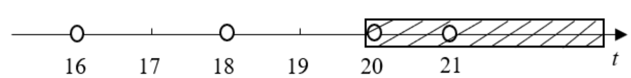
We use the same four-component system considered earlier to illustrate the grouping process using the Grouping Rule. Suppose the shifting window $\iota = 3$. The three candidate group options are illustrated in Figure 5. Among all three options, group option 2 has the minimum weighted PHA cost cumulated. Therefore, the actual replacement time for $I_{2,3}$ is 16 and 18 for the other individuals. The heuristic is summarized in Algorithm 4 and the Grouping Rule.



(a) Group option 1



(b) Group option 2



(c) Group option 3

Figure 5 Group options at one iteration

Our modeling and solution approach can be easily extended to incorporate several other modeling aspects, such as cost discounts depending on the number of components maintained, restriction on the number of components replaced together, and nonlinear operation costs. It can also be easily adapted to solve opportunistic maintenance (OM). We can find the optimal OM policy that groups

Algorithm 4 Heuristic algorithm for one scenario

1: **Initialization:**

Not-used-residual lifetime $\Delta \leftarrow \{0, 1\}$, and determine a set of values for ι, $\iota = \{\iota_1, \iota_2, \dots\}$.

2: For all combinations of Δ and ι, select the one that has the minimum total weighted PHA replacement cost and return the corresponding optimal solution.

2.1 **Initialization:** Assign tentative replacement times for the first individual of each component $K \leftarrow \{I_{1,1}, I_{2,1}, \dots, I_{n,1}\}$, and $\beta_{i,1} \leftarrow T_{i,1} - \Delta$, $\forall i \in \mathcal{N}$.

2.2 Apply **Grouping Rule** to obtain the optimal group option W.

2.3 Update set K.

$\forall I_{ij} \in W$:

Replace I_{ij} in set K with $I_{i,j+1}$;

Assign tentative replacement schedule to $I_{i,j+1}$, $\beta_{i,j+1} \leftarrow \beta'_{ij} + (T_{i,j+1} - \Delta)$;

$\forall I_{ij} \in K$, **if** $\beta_{ij} > T$, **then** Remove I_{ij} from set K. **end if**

if K is empty, **then Stop**. **else** Go to step 2.2. **end if**

The Grouping Rule

1: Sort K in ascending order based on $\beta_{ij} \Rightarrow$ sorted set K'.

2: Select the option m that has the lowest cost of weighted PHA replacement cost cumulated and penalty cost $\Rightarrow W$.

Group option m: m from 1 to $|K'| - 1$.

2.1: Initialize the last individual grouped as $K'[\nu] : \nu \leftarrow m$.

2.2: Let $t \leftarrow \beta(K'[\nu])$. Group individuals in set K' if the actual replacement times of their predecessors are before t until $\beta(K'[\nu']) > t + \iota, \nu' = \nu + 1, \nu + 2, \dots$.

2.3: Let ϑ denote the position of the last individual grouped in step 2.2, and update actual replacement times: $\tau'(K'[\nu']) \leftarrow t, \nu' = \nu + 1, \nu + 2, \dots, \vartheta$, $\nu \leftarrow \vartheta + 1$.

2.4: **if** $\nu \geq |K'|$, **then** Compute the weighted PHA replacement cost cumulated.

else Go to step 2.2. **end if**

3: **if** $K'[1] \notin W$ **then** $W \leftarrow W \cup \{(K'[1])\}$. **end if**

4: **return** set W.

maintenance only at CM by restricting the not-used-residual lifetime in our heuristic to $\Delta = 0$, and find the optimal OM policy that only groups maintenance at PM by only allowing $\Delta = 1$. More structured policies can be derived based on the patterns of results from our general model. For example, fixed groups in direct-grouping can be obtained based on how frequent some components are grouped for maintenance in our results. Indirect grouping policy can be derived based on the individual PM replacement intervals.

5. Computational study

In this computational study, we first introduce an analytical approach to determining the minimum number of scenarios needed for a given accuracy, then examine the performance of all four algorithms, and assess the approximation performance of the multi-stage model by using proposed heuristic algorithm (Algorithm 4) in a rolling horizon. We perform our computational study on a computer with a CPU of Intel i7-6700, 3.4G Hz and a RAM of 16G. A python based package Pyomo (Hart et al. 2011, 2017) is used to implement the algorithms with the solver of CPLEX v12.7.1. All input parameters, data sets and the source code used for the computational study can be found in our online repository [1].

5.1. Scenario sampling

Consider a stochastic programming problem $V^* = \min_{x\in X}\{f(x) := \mathbb{E}[F(x,\xi)]\}$ and its sample average approximation (SAA) problem $\hat{V}_{|\Omega|} = \min_{x\in X}\{\hat{f}_{|\omega|}(x) := \frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|} F(x,\xi^i)\}$, where X is the feasible region of decision variable x, and Ω is a sample of the random vector ξ . Denote the ϵ-optimal solution sets of the stochastic program and SAA problem by $S^\epsilon := \{x \in X : f(x) \le V^* + \epsilon\}$ and $\hat{S}^\epsilon_{|\Omega|} := \{x \in X : \hat{f}_{|\Omega|}(x) \le V_{|\Omega|} + \epsilon\}$ respectively. To guarantee that the optimal solution of SAA problem is the ϵ-optimal solution of the true stochastic programming problem with probability 1–α, Shapiro et al. (2009) provide an analytical expression of the minimum sample size required. Theorem 3 summarizes the main result in their study.

THEOREM 3. *(Theorem 5.17 (Shapiro et al. 2009)) Suppose there exists a constant $\sigma > 0$ such that for any $x \in X\backslash S^\epsilon$ the moment generating function $M_x(t)$ of the random variable $H(x,\xi) - \mathbb{E}[H(x,\xi)]$ satisfies $M_x(t) \le e^{(\sigma^2 t^2/2)}$ for all $t \in \mathbb{R}$, then for $\epsilon > 0$, $0 \le \tau < \epsilon$ and $\alpha \in (0,1)$, and sample size $|\Omega|$ satisfying*

$$|\Omega| \ge \frac{2\sigma^2}{(\epsilon-\tau)^2}\ln\left(\frac{|X|}{\alpha}\right) \tag{32}$$

it follows that $\Pr(\hat{\mathrm{S}}^\tau_{|\Omega|} \subset \mathrm{S}^\epsilon) \ge 1-\alpha$, where $H(x,\xi) := F(u(x),\xi) - F(x,\xi)$, $x \in X\backslash S^\epsilon$ and mapping $u : x \in X\backslash S^\epsilon \to X$ satisfies $f(u(x)) \le f(x) - \epsilon^*$ for some $\epsilon^* \ge \epsilon$.

To determine constant σ, Shapiro et al. (2009) show that if $H(x,\xi) - \mathbb{E}[H(x,\xi)] < b$ is satisfied with some $b > 0$ for all $x \in X$, then $\sigma^2 := b^2$. If ϵ^* is small compared to $\max_x F(x,\xi)$, then in DEF, an upper bound b can be estimated by considering CR for all individuals over the planning horizon, i.e.,

$$\begin{aligned} H(x,\xi) - \mathbb{E}[H(x,\xi)] &\le |H(x,\xi)| + |\mathbb{E}[H(x,\xi)]| \approx |H(x,\xi)| \\ &\le |F(u(x),\xi)| + |F(x,\xi)| \le 2F(x,\xi) \le 2T(\sum_{i\in\mathcal{N}} c_{i,\mathrm{cr}} + d) = b \end{aligned} \tag{33}$$

5.2. Performance of algorithms in solving DEF

In this section, we compare the computational times and cost errors of the four algorithms in solving DEF. For standard PHA (Algorithm 3), we run our experiments in a stochastic programming package PySP inside Pyomo. Assume that all components' lifetimes follow Weibull distributions. For each component, we draw the shape and scale parameters from uniform distributions $U(4,7)$ and $U(1,8)$, respectively. The cost of CR ($c_{i,\mathrm{cr}}$) is drawn from a uniform distribution $U(6,16)$. Parameter values that are randomly drawn and used in Section 5.2 are provided in Table OS5 in Online Supplement. Given the number of components in a test case, we use the parameters of the first n components in Table OS5. Without loss of generality, the cost of PR ($c_{i,\mathrm{pr}}$) is assumed to be 1. Suppose that setup cost d is 5, the initial ages of all first individuals are 2, and the individual of component 1 is assumed to be failed at the first stage.

The number of scenarios needed for each test case in Table 1 is determined using Equations (32) and (33) by choosing $\epsilon = 0.1\sigma$, $\tau = 0.1\epsilon$ and $\alpha = 0.1$. This parameter setting guarantees the optimal solution of SAA problem is an 0.1σ-optimal solution of the true stochastic programming problem with probability 0.9. We use the same parameter setting to determine the number of scenarios throughout the paper.

We compare the performance of Algorithms 1 – 4 for 18 cases. Table 1 summarizes the performance of the four algorithms. NA is reported if the computational time is longer than one day or out of memory, or if the true objective of DEF cannot be obtained for computing objective percentage error. From Table 1, we can see that Algorithms 1 – 3 can only solve small problems (e.g., $n \le 4$) and Algorithm 4 is the only algorithm that can solve all test cases efficiently. We further examine the performance of Algorithm 4. We compute the percentage error between the objective values obtained from using Algorithm 4 and solving DEF exactly by CPLEX. We can see that the performance of Algorithm 4 is very good for small-scale problems with a maximum percentage error of 9.89%. Based on our computational studies, the proposed heuristic algorithm (Algorithm 4) performs well and is capable of solving practically large-scale problems.

Table 1 **Algorithm performance in solving DEF**

			solver		Algorithm 1				Algorithm 2			Algorithm 3			Algorithm 4			
case	n	T	CPU time (sec.)	Obj.	Itera tions	Cuts	CPU time (sec.)	Obj.	Itera tions	CPU time (sec.)	Obj.	Itera tions	CPU time (sec.)	Obj.	Itera tions	CPU time (sec.)	Obj.	Obj. error %
1		5	103	25.80	3	1478	466	-99.65	4	1156	25.80	3	1361	25.80	1	17	26.41	2.36%
2	2	7	202	28.36	3	1474	546	-111.53	4	1556	28.63	2	1315	28.63	1	14	30.87	7.82%
3		9	503	32.82	3	1380	760	-167.22	4	1670	32.82	3	1775	32.82	2	25	34.76	5.91%
4		5	237	26.92	3	1818	884	-143.87	9	2505	26.92	5	2502	26.92	2	51	27.78	3.19%
5	3	7	561	30.14	4	1830	1613	-187.43	13	4070	30.14		NA		6	103	33.12	9.89%
6		9	2777	34.50	5	3070	2575	-239.20		NA			NA		4	92	37.71	9.30%
7		5	472	30.02	5	3263	2198	-176.37							4	145	32.31	7.63%
8	4	7	NA		6	4764	3895	-227.89		NA			NA		4	181	38.86	NA
9		9	NA			NA									4	269	44.5	NA
10		5			5	2848	3052	-221.73							4	324	37.35	
11	5	7	NA			NA				NA			NA		4	422	45.59	NA
12		9				NA									4	551	53.66	
13		5													4	623	41.4	
14	6	7	NA			NA				NA			NA		4	837	51.28	NA
15		9													4	1117	60.94	
16		5													4	1385	48.77	
17	7	7	NA			NA				NA			NA		5	2812	62.13	NA
18		9													5	3398	74.55	

We further investigate the performance of the grouping heuristic by comparing it with the optimal solution (i.e., solving DEF of one scenario subproblem using the solver CPLEX). We use the same parameter settings used for computational studies summarized in Table 1. For each test case, we generate 100 scenarios. Each scenario is solved using both the solver and the grouping heuristic. The comparison of average costs obtained from the solver and the heuristic is summarized in Table 2.

Table 2 Average performance of Algorithm 4 over 100 scenarios

n	T	optimal avg. cost	avg. cost of Alg. 4	avg. percentage gap
	9	31.91	32.04	0.38%
2	29	67.48	69.95	4.80%
	49	104.23	107.17	2.80%
	9	33.35	34.06	2.18%
3	29	73.94	78.18	5.67%
	49	114.46	122.19	6.67%
	9	38.12	39.54	3.44%
4	29	90.13	99.11	9.58%
	49	142.21	158.64	11.43%

From Table 2, we can see that the average percentage gap is 5.2%. We believe that the grouping heuristic performs well for the scenario subproblem.

5.3. Performance in approximating the multi-stage model using a rolling horizon approach

In this section, we compare our heuristic in a rolling horizon two benchmark policies. We also compare the performances using our approach in a rolling horizon with the results from solving the multi-stage model using an exact method, and assess the performance of the proposed approximation method.

Benchmark 1 is a widely used direct-grouping model (Van Horenbeek and Pintelon 2013) which uses a dynamic-programming algorithm that first finds the optimal replacement schedule for each component without considering economic dependence and then sort the components based on that. At iteration j, the algorithm identifies two groups that cover all maintenance activities of components 1 to j and provide the best savings for these components. The best grouping structure can be found by backtracking. This algorithm has in the worst case a time complexity of $o(n^2)$. However, the limitation of this algorithm is that it only considers the group structure of two groups at each iteration and ignores all other options (e.g., partition all maintenance activities into three or more groups). Benchmark 2 is a structural policy known as (t_i, T_i) policy which works as follows: At each decision period, if the age of component i exceeds T_i, preventive maintenance is performed; if the age of component i is between t_i and T_i ($t_i \leq T_i$) and there is one or more other component's age exceed their respective T_i, preventive maintenance is also performed on component i. We adopt the method in (Wijnmalen and Hontelez 1997) which extends the decomposition and aggregation approach used in multi-unit inventory control problem Federgruen et al. (1984) to our multi-component maintenance optimization problem.

We conduct a sensitivity analysis to assess the performance of Algorithm 4. Suppose the length of a decision period δ equals to 1. At each decision period, we consider a two-stage stochastic maintenance optimization problem (DEF) where the second stage combines decisions of the remaining periods. We repeat this procedure five times to obtain the average total maintenance cost because five replicates can provide the acceptable precision and computational efficiency. The PR cost is assumed to be 1 for each type of component, and the CR costs are drawn from two different uniform distributions, $U(6,16)$ and $U(17,27)$. The lifetime of each individual is assumed to follow a Weibull distribution. To introduce more heterogeneity to the system, the shape parameter (η) of the Weibull distribution is drawn from two uniform distributions $U(1,3)$ and $U(4,7)$, and the scale parameter (λ) of the Weibull distributions is drawn from two different distributions, $U(1,5)$ and $U(5,10)$. Two levels of setup costs are considered. Assume all operating individuals are functioning and have an age of 0 at the first decision period. Different levels of parameters are provided in Table 3. Parameter values that are randomly drawn and used for the comparison are summarized in Table OS6 in Online Supplement.

Table 3 Different levels of parameters

Level	shape parameter η	scale parameter λ	d	CR cost
High	$U(4,7)$	$U(5,10)$	100	$U(17,27)$
Low	$U(1,3)$	$U(1,5)$	5	$U(6,16)$

Table 4 summarizes the comparison results under different parameter settings. The optimal expected costs (in column multi-stage (P1)) are obtained by enumeration. We enumerate all possible combinations of the maintenance decisions and node states at all decision periods. Expected cost for each decision policy can be computed accordingly. The optimal cost is the minimum of the expected costs of all decision policies. Note that the optimal expected costs using benchmark algorithms and the proposed algorithm are obtained using a rolling horizon approach. In each rolling horizon step, we simulate the (remaining) lifetimes to determine the failure status of all individuals at the next decision period. From Table 4, we can see that all percentage gaps of Algorithm 4 are below 10%, compared to the optimal costs obtained by enumeration, and Algorithm 4 outperforms the two benchmarks in the majority of the test cases.

We further compare the performance of Algorithm 4 and the benchmark algorithms on large-scale problems. The results are summarized in Tables 5 and 6. Note that exact method cannot solve the problem instances in Tables 5 and 6 and are therefore not included. From Tables 5 and 6, we can see that Algorithm 4 provides significant better overall results than the two benchmark algorithms for large-scale problems.

Table 4 **Performance of Algorithm 4 for small-scale multi-stage problems** ($n=2, T=9, |\Omega|=910$)

#	η	λ	d	CR cost	multi-stage (P1)	Benchmark 1			Benchmark 2			Algorithm 4				
						mean	var.	% gap	mean	var.	% gap	mean	var.	% gap	% saving v.s. Benchmark 1	% saving v.s. Benchmark 2
1	H	H	H	H	105.39	106.2	465.3	0.73%	124.9	387.7	18.46%	112.7	127.3	6.93%	-5.79%	10.79%
2	H	H	H	L	104.51	105.1	370.5	0.53%	112.5	279.7	7.67%	109.8	293	5.09%	-4.34%	2.46%
3	H	H	L	H	8.88	9.92	64.31	11.71%	32.38	69.62	264.64%	9.32	25.15	4.95%	6.44%	247.42%
4	H	H	L	L	8	8.61	20.4	7.62%	20.52	27.9	156.50%	8.34	17.96	4.25%	3.24%	146.04%
5	H	L	H	H	268.3	410.1	41.22	52.87%	278.1	4469	3.67%	271.5	1949	1.17%	51.09%	2.46%
6	H	L	H	L	244.6	409	9.73	67.23%	262.3	3369	7.24%	245	1547	0.18%	66.93%	7.04%
7	H	L	L	H	31.53	33.68	106.3	6.82%	49.29	627	56.33%	32.01	68.47	1.52%	5.22%	53.98%
8	H	L	L	L	28.03	30.82	26.36	9.95%	32.81	170.1	17.05%	30.04	21.01	7.17%	2.60%	9.22%
9	L	H	H	H	136.13	413.4	133.8	203.67%	146.1	4124	7.29%	142.7	6493	4.79%	189.79%	2.38%
10	L	H	H	L	123.43	215.9	913.3	74.92%	132.6	3347	7.45%	127.4	5118	3.18%	69.52%	4.14%
11	L	H	L	H	22.48	30.53	77.66	35.81%	37.38	310.8	66.28%	22.87	528.6	1.73%	33.49%	63.45%
12	L	H	L	L	16.79	20.22	65.45	20.43%	22.27	132.2	32.64%	17.72	164.8	5.54%	14.11%	25.68%
13	L	L	H	H	340.36	842.4	1693	147.49%	351.9	7021	3.39%	345.2	8584	1.41%	144.06%	1.96%
14	L	L	H	L	306.54	835.3	1335	172.50%	318.2	5532	3.82%	310.8	7009	1.40%	168.74%	2.38%
15	L	L	L	H	60.12	70.1	357.1	16.60%	90.53	616.4	50.58%	63.7	458	5.95%	10.05%	42.12%
16	L	L	L	L	44.7	61.92	66.56	38.52%	54.14	236.8	21.12%	48.63	148.9	8.79%	27.33%	11.33%
average								54.21%			45.26%			4.00%	48.90%	39.55%

Table 5 Performance of Algorithm 4 for large-scale multi-stage problems ($n=4, T=19, |\Omega|=1250$)

#	η	λ	d	CR cost	Benchmark 1		Benchmark 2		Algorithm 4			
					mean	var.	mean	var.	mean	var.	% saving v.s. Benchmark 1	% saving v.s. Benchmark 2
1	H	H	H	H	434.5	617	349.3	5002	306	3362.2	41.98%	14.16%
2	H	H	H	L	329.3	1261	331.1	4145	289.9	4207.2	13.60%	14.23%
3	H	H	L	H	41.61	129.9	86.63	429.3	36.08	123.33	15.33%	140.11%
4	H	H	L	L	34.8	83.44	51.43	132.1	31.88	21.77	9.16%	61.32%
5	H	L	H	H	961	592.7	805.3	4543	722.9	2173.2	32.93%	11.39%
6	H	L	H	L	947	212.3	726.4	3088	634.5	2103	49.25%	14.48%
7	H	L	L	H	136.3	506.6	199.2	1403	99.68	131.45	36.75%	99.81%
8	H	L	L	L	111.8	103.3	120.3	338.1	88.68	26.81	26.04%	35.62%
9	L	H	H	H	1002	4434	565.6	23024	485.8	3089.4	106.25%	16.43%
10	L	H	H	L	689	5240	517.2	20262	440.8	3016.7	58.33%	17.32%
11	L	H	L	H	120.4	870.6	141.9	1471	100.4	173.21	19.89%	41.34%
12	L	H	L	L	84.26	391.4	85.82	632.1	59	565.8	42.81%	45.46%
13	L	L	H	H	2005	5618	1179	28140	906.8	12645	121.06%	30.03%
14	L	L	H	L	1945	3059	1045	23242	749	4364.6	159.64%	39.53%
15	L	L	L	H	287	2240	324.1	2119	228.2	587.12	25.78%	42.04%
16	L	L	L	L	237.2	407.1	229.1	689.8	181.4	340.75	30.75%	26.29%
									average		49.35%	40.60%

Table 6 **Performance of Algorithm 4 for large-scale multi-stage problems** ($n = 8, T = 19, |\Omega| = 1940$)

#	η	λ	d	CR cost	Benchmark 1		Benchmark 2		Algorithm 4			
					mean	var.	mean	var.	mean	var.	% saving v.s. Benchmark 1	% saving v.s. Benchmark 2
1	H	H	H	H	481.3	3993	508.04	1972	389.12	192.45	23.69%	30.56%
2	H	H	H	L	453.0	1338	455.24	1878	383.6	287.1	18.08%	18.68%
3	H	H	L	H	78.47	691.1	147.04	267.6	67.4	466.9	16.42%	118.16%
4	H	H	L	L	72.6	180.3	94.24	90.14	57.5	72.4	26.26%	63.90%
5	H	L	H	H	1996	450.2	1276.28	2498	1137.7	304.4	75.47%	12.18%
6	H	L	H	L	1952	101.3	1150.7	1551	975.64	1615	100.10%	17.94%
7	H	L	L	H	240.6	579.2	345.28	2000	208.5	244.5	15.41%	65.60%
8	H	L	L	L	220.2	247.5	219.68	393.2	185.3	200.1	18.84%	18.55%
9	L	H	H	H	1265	4709	1100.5	1691	936.43	4795.7	35.12%	17.52%
10	L	H	H	L	1018	5146	1034.2	11094	865.32	9781.3	17.59%	19.52%
11	L	H	L	H	296.3	535.1	321.5	338.6	180.25	406.5	64.39%	78.36%
12	L	H	L	L	202.8	641.9	198.2	166.1	150.18	197.33	35.00%	31.97%
13	L	L	H	H	2443	3831	2166.4	1033	1962.6	5380.3	24.49%	10.38%
14	L	L	H	L	2212	1928	1992.3	2844	1794.4	3413.8	23.26%	11.03%
15	L	L	L	H	666.7	2750	741.38	1911	599.1	944.4	11.29%	23.75%
16	L	L	L	L	448.7	406	469.2	177.3	406.9	300.2	10.27%	15.31%
									average		32.23%	34.59%

6. Conclusion and future research

In this paper, we consider the problem of multi-component maintenance optimization over the finite planning horizon. We formulate the problem as a multi-stage decision-dependent stochastic integer program, and approximate it with a novel two-stage stochastic linear integer model in a rolling horizon. The proposed models are general with no restrictions on maintenance grouping. A progressive-hedging-based heuristic is designed to solve practically large-size two-stage problems. To assess the performance of the heuristic, we compare it with three conventional algorithms and our computational studies show that the proposed heuristic provides satisfying results and is capable of solving practically large-scale problems. We also evaluate the performance of the heuristic in a rolling horizon relative to the true global optimal for small problems. Results show that solving our two-stage model by the proposed heuristic in a rolling horizon provides a good approximation of the multi-stage problem. The proposed heuristic in a rolling horizon is further benchmarked with a widely studied dynamic-programming-based algorithm and a commonly adopted structural policy. Our heuristic significantly outperforms the benchmark algorithms based on our numerical experiments. Our work has extended the available literature in multi-component maintenance by using stochastic programming approach. The modeling and solution techniques developed in this paper opens new research and implementation opportunities. Future research will consider a different widely used maintenance policy, condition-based maintenance (CBM). CBM leverages sensor information on components' health status through inspection or real-time monitoring and aims to perform maintenance just in time by setting optimal control thresholds. Capturing these complexities requires a different problem formulation and different optimization algorithms. Moreover, maintenance activities are often subject to a pre-determined budget with a requirement on a system's reliability or availability. Future work will incorporate these constraints into the decision model. Lastly, it is worth extending the problem for more complex systems with stochastic and structural dependences, in addition to the economic dependence.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation under Award 1855408.

Notes

[1]https://github.com/yishaxiang/IJOC01.git.

References

Amaran S, Zhang T, Sahinidis NV, Sharda B, Bury SJ (2016) Medium-term maintenance turnaround planning under uncertainty for integrated chemical sites. *Computers & Chemical Engineering* 84:422–433.

Apap RM, Grossmann IE (2017) Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties. *Computers & Chemical Engineering* 103:233–274.

Aydin N (2012) Sampling based progressive hedging algorithms for stochastic programming problems .

Barata J, Soares CG, Marseguerra M, Zio E (2002) Simulation modelling of repairable multi-component deteriorating systems for 'on condition'maintenance optimisation. *Reliability Engineering & System Safety* 76(3):255–264.

Beraldi P, Violi A, Scordino N, Sorrentino N (2011) Short-term electricity procurement: A rolling horizon stochastic programming approach. *Applied Mathematical Modelling* 35(8):3980–3990.

Birge JR, Louveaux F (2011) *Introduction to stochastic programming* (Springer Science & Business Media).

Bodur M, Dash S, Günlük O, Luedtke J (2017) Strengthened benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing* 29(1):77–91.

Castanier B, Grall A, Bérenguer C (2005) A condition-based maintenance policy with non-periodic inspections for a two-unit series system. *Reliability Engineering & System Safety* 87(1):109–120.

Cowing MM, Paté-Cornell ME, Glynn PW (2004) Dynamic modeling of the tradeoff between productivity and safety in critical engineering systems. *Reliability Engineering & System Safety* 86(3):269–284.

Dekker R, Scarf PA (1998) On the impact of optimisation models in maintenance decision making: the state of the art. *Reliability Engineering & System Safety* 60(2):111–119.

Dekker R, Wildeman R, Van Egmond R (1996) Joint replacement in an operational planning phase. *European Journal of Operational Research* 91(1):74–88.

Dekker R, Wildeman RE, Van der Duyn Schouten FA (1997) A review of multi-component maintenance models with economic dependence. *Mathematical methods of operations research* 45(3):411–435.

Ding F, Tian Z (2012) Opportunistic maintenance for wind farms considering multi-level imperfect maintenance thresholds. *Renewable Energy* 45:175–182.

Epstein S, Wilamowsky Y (1985) Opportunistic replacement in a deterministic environment. *Computers & operations research* 12(3):311–322.

Federgruen A, Groenevelt H, Tijms HC (1984) Coordinated replenishments in a multi-item inventory system with compound poisson demands. *Management Science* 30(3):344–357.

Gade D, Hackebeil G, Ryan SM, Watson JP, Wets RJB, Woodruff DL (2016) Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Mathematical Programming* 157(1):47–67.

Goyal SK, Gunasekaran A (1992) Determining economic maintenance frequency of a transport fleet. *International Journal of Systems Science* 23(4):655–659.

Hariga M (1994) A deterministic maintenance-scheduling problem for a group of non-identical machines. *International Journal of Operations & Production Management* 14(7):27–36.

Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) *Pyomo-optimization modeling in python*, volume 67 (Springer).

Hart WE, Watson JP, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation* 3(3):219.

Huynh KT, Barros A, Bérenguer C (2014) Multi-level decision-making for the predictive maintenance of k-out-of-n: F deteriorating systems. *IEEE transactions on Reliability* 64(1):94–117.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Koochaki J, Bokhorst JA, Wortmann H, Klingenberg W (2012) Condition based maintenance in the context of opportunistic maintenance. *International Journal of Production Research* 50(23):6918–6929.

Laggoune R, Chateauneuf A, Aissani D (2009) Opportunistic policy for optimal preventive maintenance of a multi-component system in continuous operating units. *Computers & Chemical Engineering* 33(9):1499–1510.

Laporte G, Louveaux FV (1993) The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters* 13(3):133–142.

Nicolai RP, Dekker R (2008) Optimal maintenance of multi-component systems: a review. *Complex system maintenance handbook*, 263–286 (Springer).

Patriksson M, Strömberg AB, Wojciechowski A (2015) The stochastic opportunistic replacement problem, part ii: a two-stage solution approach. *Annals of Operations Research* 224(1):51–75.

Peeta S, Salman FS, Gunnec D, Viswanath K (2010) Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research* 37(10):1708–1719.

Rardin RL (1998) *Optimization in operations research*, volume 166 (Prentice Hall Upper Saddle River, NJ).

Rockafellar RT, Wets RJB (1991) Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research* 16(1):119–147.

Scarf PA (1997) On the application of mathematical models in maintenance. *European Journal of operational research* 99(3):493–506.

Schouten FVDD, van Vlijmen B, de Wael SV (1998) Replacement policies fro traffic control signals. *IMA Journal of Management Mathematics* 9(4):325–346.

Shafiee M, Finkelstein M (2015) An optimal age-based group maintenance policy for multi-unit degrading systems. *Reliability Engineering & System Safety* 134:230–238.

Shapiro A, Dentcheva D, Ruszczynski A (2009) Lectures on stochastic programming: modeling and theory (siam). *MPS, Philadelphia* .

Thomas L (1986) A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering* 16(4):297–309.

Tian Z, Liao H (2011) Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering & System Safety* 96(5):581–589.

van Dijkhuizen G, van Harten A (1996) Two-stage maintenance: a generalized age maintenance policy. *T&M working paper* (UT-TBK. OMST. WP. 96.05).

Van Horenbeek A, Pintelon L (2013) A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety* 120:39–50.

Vu HC, Do P, Barros A, Bérenguer C (2014) Maintenance grouping strategy for multi-component systems with dynamic contexts. *Reliability Engineering & System Safety* 132:233–249.

Watson JP, Woodruff DL (2011) Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science* 8(4):355–370.

Wijnmalen DJ, Hontelez JA (1997) Coordinated condition-based repair strategies for components of a multi-component maintenance system with discounts. *European Journal of Operational Research* 98(1):52–63.

Wildeman R, Dekker R, Smit A (1997) A dynamic policy for grouping maintenance activities. *European Journal of Operational Research* 99(3):530–551.

Wildeman RE, Dekker R (1997) Dynamic influences in multi-component maintenance. *Quality and reliability engineering international* 13(4):199–207.

Zhan Y, Zheng QP, Wang J, Pinson P (2016) Generation expansion planning with large amounts of wind power via decision-dependent stochastic programming. *IEEE Transactions on Power Systems* 32(4):3015–3026.