Optimization of Joint Replacement Policies for Multipart Systems by a Rollout Framework

Tao Sun, *Student Member, IEEE*, Qianchuan Zhao, *Member, IEEE*, Peter B. Luh, *Fellow, IEEE*, and Robert N. Tomastik, *Senior Member, IEEE*

***Abstract*—Maintaining an asset with life-limited parts, e.g., a jet engine or an electric generator, may be costly. Certain costs, e.g., setup cost, can be shared if some parts of the asset are replaced jointly. Reducing the maintenance cost by good joint replacement policies is difficult in view of complicate asset dynamics, large problem sizes and the irregular optimal policy structures. This paper addresses these difficulties by using a rollout optimization framework. Based on a novel application of time-aggregated Markov decision processes, the "One-Stage Analysis" method is first developed. The policies obtained from the method are investigated and their effectiveness is demonstrated by examples. This method and the existing threshold method are then improved by the "rollout algorithm" for the total cost case and the average cost case. Based on ordinal optimization, it is shown that excessive simulations are not necessary for the rollout algorithm. Numerical testing demonstrates that the policies obtained by the rollout algorithms with either the "One-Stage Analysis" or the threshold method significantly outperform traditional threshold policies.**

***Note to Practitioners*—Maintaining an asset with life-limited parts, e.g., a jet engine or an electric generator, over its lifetime may be costly. Optimizing maintenance policies, however, is difficult because of complicated asset dynamics, large problem sizes, and irregular optimal policy structures. This paper addresses the problem by a rollout optimization framework. In this framework, the "One-Stage Analysis" method, which minimizes the expected average cost over one maintenance period, is first developed and investigated. The policies, obtained by either the "One-Stage Analysis" method or the existing threshold method, are used in simulation to evaluate and select good actions. Similar to those learning approaches, this simulation-based framework is flexible for variant performance criteria, e.g., total cost or average cost, and is applicable to problems without explicit mathematical models. Numerical results demonstrate that effective policies can be obtained by the rollout framework in a computationally efficient way and they significantly outperform traditional threshold policies.**

***Index Terms*—Joint replacement, Markov decision processes, multipart maintenance, rollout algorithm, time aggregation.**

Manuscript received January 23, 2007. First published April 24, 2008; current version published October 1, 2008. This paper was recommended for publication by Associate Editor Y. Narahari and Editor N. Viswanadham upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation of China under Grant 60574067, Grant 60721003, and Grant 60736027, in part by the NCET (NCET-04-0094) Program of China, and in part by the National 111 International Collaboration Project.

T. Sun and Q. Zhao are with the Center for Intelligent and Networked Systems, TNLIST, Tsinghua University, Beijing 100084, China (e-mail: suntao99@mails.thu.edu.cn; zhaoqc@tsinghua.edu.cn).

P. B. Luh is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157 USA (e-mail: peter.luh@uconn.edu) and also with the Center for Intelligent and Networked Systems, TNLIST, Tsinghua University, Beijing 100084, China.

R. N. Tomastik is with Pratt & Whitney, East Hartford, CT 06108 USA (e-mail: robert.tomastik@pw.utc.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2008.917173

I. Introduction

A COMPANY, e.g., an airline or a power company, needs to maintain its assets, e.g., jet engines or electric generators, during their lifetimes for reliable and sustained services. Such an asset is generally composed of several modules and each module in turn consists of many parts. A few of the parts are *safety-critical*, i.e., their failures or malfunctions may lead to disastrous consequences. For safety and reliability purposes, each of such parts is life-limited by "policy." Such a life limit is set at a point where exceeding the limit increases the conditional probability of failure, even though a part may still have actual remaining life. An asset has a maintenance shop visit when a life-limited part reaches its life limit. Moreover, accidents or failures of those nonsafety critical parts may warrant a maintenance visit. A shop visit leads to a fixed cost and variable costs. The fixed cost, i.e., setup cost, includes costs such as downtime, inspection, transportation of the asset, etc. Variable costs consist of module removal costs and new part costs. Since parts are installed in modules, replacing a part from an inner module may need to remove outer modules first. Thus the replacements of parts are coupled by module removal costs and the setup cost through structural dependence and economic dependence [8]. Maintaining an asset may be expensive, e.g., millions of dollars for maintaining a jet engine over several years [2]. If some parts are replaced jointly at the same shop visit, the setup cost and module removal costs can be shared and saved. In addition, an asset can operate longer until its next shop visit. However, jointly replacing the unexpired parts incurs more costs on new parts. Therefore, a good maintenance policy is to balance the setup cost, module removal costs and part costs.

Reducing the maintenance cost by good joint replacement policies is important for asset owners as well as for maintenance service providers. However, optimizing such policies is difficult because of complicate asset dynamics, large problem sizes and irregular optimal policy structures.

1) The asset dynamics are complicated and involve random failures of the asset. This makes it hard to evaluate a replacement decision or policy in a close form. Monte Carlo simulation may be the only way.
2) The number of part remaining life combinations increases exponentially with the number of parts and the part full lifetimes. For practical problems, an asset may have tens of safety-critical parts with full lifetimes ranging from tens to hundreds.
3) Similar to those joint replacement problems in the literature, the structures of optimal policies are quite irregular and have no simple form as the generally used threshold policies [15]. Therefore, it is difficult to describe and find an optimal policy.

The key features of our problem are life-limited parts and random asset-wise failures. As will be reviewed in Section II, most problems in the literature model failures of parts which are not life-limited. Our problem is formulated as a Markov decision process (MDP) in Section III with two widely used criteria: total cost and long-run average cost. In Section IV, the time aggregated approach for Markov decision processes is developed [4]. The application provides a novel view of maintenance problems and simplifies the analysis as well as the computation. Although time aggregation still cannot solve large problems, it helps to develop the "One-Stage Analysis" (OSA) method. The method approximates optimal policies by minimizing the expected average cost over one maintenance period. The policies obtained by this method are proved to preserve some properties of optimal policies. The performance bounds of OSA obtained for single-part problems show that the method performs well when the failure rate or the ratio of part cost to setup cost is low. Such properties are also demonstrated by examples for multipart problems.

To improve the policies obtained by OSA or the existing threshold method, the rollout algorithm is presented in Section V. The paper reveals that the idea of the rollout algorithm, i.e., evaluating feasible replacement actions via repeated simulations to select the best one, is consistent with ordinal optimization [11]: using crude model and performing ordinal comparison. Thus there is no need to perform excessive Monte Carlo simulations. Numerical results in Section VI demonstrate that OSA is an effective heuristic method compared with the widely used threshold method. The efficient OSA may be used if the problem is very large and CPU time is quite limited. Given more CPU time, the rollout algorithm improves OSA or the threshold method under both total cost and average cost criteria. This is demonstrated by a semi-realistic data set under variant parameter settings.

II. Literature Review

Modeling and optimizing the maintenance of multipart systems have attracted much interest over the past two decades as surveyed in [7], [8], and [21]. In this section, we will review traditional problems and methods, and then summarize related works on our problem. The problems in the literature concentrate on part-wise failures and the failure rates of parts increase with lifetimes [21]. A part must be replaced when it fails. The maintenance policies fall into several categories, e.g., corrective, preventive and opportunistic. Corrective maintenance is conducted only when a part breaks down. For preventive maintenance, inspections, replacements and part revisions are carried out to prevent failures. Corrective and preventive maintenance on a part may yield an opportunity for maintaining other parts. In this way, opportunistic maintenance saves costs by economy of scale.

For multipart systems, joint replacement is a result of opportunistic maintenance to reduce the cost. It can be "group replacement" [21], for which certain parts are replaced together. The basic group replacement policy is the T-age group replacement policy. It calls for a group replacement if the age of the group is T, which is generally a predetermined constant. Another "base interval approach" [12] is to find a "base interval" such that the system is maintained at every base interval, and a part is replaced at some multiples of the base interval. To deal with the combinatorial difficulties caused by the couplings of parts, decomposition and coordination heuristics are explored by using Markov decision processes in [9] and [22]. In [9], the optimal replacement time of each part is first obtained without considering other parts. Then the replacement of multiple parts is coordinated based on the value functions of the MDP approach in a heuristic way.

Most of the above methods, when applied, result in threshold (or control limit) type policies. Such policies replace a part whose lifetime has exceeded a predefined threshold [21]. The optimal policies for problems with a single part are of the threshold type [20]. However, for general multipart problems, the optimal replacement policies may have irregular structures and are difficult to obtain [15]. Threshold policies are nevertheless still widely used in practice for their simplicity and easy implementation [20]. For example, the threshold method (or the fixed-threshold method [10]) employs a common threshold for all the parts. This common threshold can be optimized by exhaust search. As a generalization, the multithreshold method of [14] has different thresholds for different parts. However, these different thresholds may not be easily optimized because there may be too many threshold combinations for problems with many parts and large full lifetimes. For the models in the literature, minimizing the total cost ([9] and [19]) has not been widely investigated as compared with minimizing the long-run average cost because the nonstationary nature of the optimal policies makes it difficult to analyze the structures of the optimal policies. More information can be obtained in surveys [7], [8], and [21].

Our problem is motivated by maintaining a jet engine and is safety-critical part oriented. A safety-critical part must be replaced after its predefined lifetime runs out for high reliability purpose. Such parts are assumed to be reliable and do not fail, while there are random asset-wise failures. As will be pointed out in Section III-D, this may lead to multichain Markov decision processes, which are more complicated than the unichain cases in the literature. Despite of above differences, our model shares similar difficulties with existing models as summarized in Section I. Moreover, the generally used threshold method in the literature is also a heuristic method for our problem. The features of optimal policies are investigated in [15]. Lagrangian relaxation approach is presented based on a new separable problem formulation in [19].

III. Problem Formulation

This section formulates the multipart maintenance problem introduced in Section I as a Markov decision process. As assumed in the literature, the maintenance durations are negligible and not considered. For safety and reliability purposes, a safety-critical part is life limited and must be replaced after a predefined lifetime. New parts are assumed available whenever they are needed and the inventory is not an issue here. In the following, the system state is described by the remaining lives of the parts and a variable representing the random asset failure. The state evolves according to asset dynamics. After exploring the cost structures, two minimization criteria are considered: the

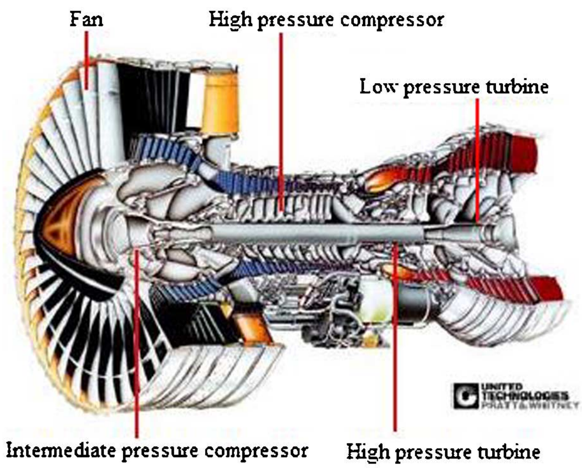


Fig. 1. Asset illustrated by a Pratt and Whitney PW4000 94-in jet engine. (Image available at http://www.pratt-whitney.com.)

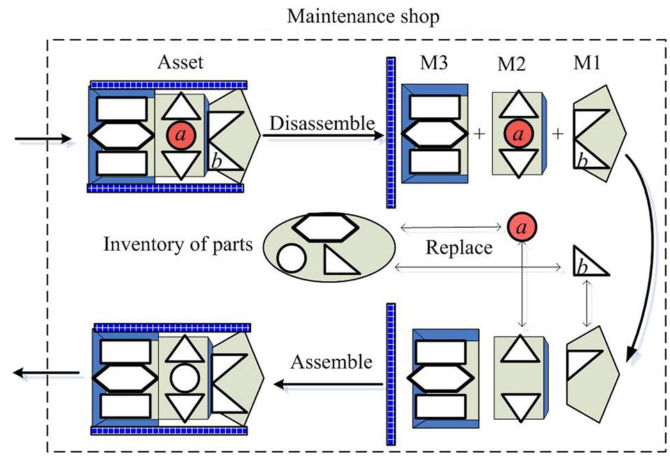


Fig. 2. In a maintenance shop, two parts are replaced jointly for an asset with three modules: M_1, M_2 and M_3. Part a is replaced because of expiration, while part b is close to expiration and replaced by opportunity to save cost. Replacing b only needs to remove M_3, while both M_3 and M_2 should be removed to replace a.

average cost in a long time and the total cost over a maintenance horizon.

A. Asset States and Uncertainties

An asset as illustrated by a jet engine in Fig. 1 is composed of M modules. Module m consists of many parts, and among them, N_m parts are safety-critical. The replacement policy of safety-critical parts, with total number $N = N_1 + \cdots N_M$, are the focus of our study. The state of part $n, n = 1, \ldots, N$, at time t is represented by its "remaining life" x_{nt} which is a nonnegative integer and degrades linearly from the "full lifetime" S_n when the asset is in service. Parts with different life reducing rates are normalized to have the same rate with adjusted full lifetimes. A part must be replaced immediately by a new one when its remaining life becomes zero.

In addition to part expirations, a maintenance shop visit may also be caused by an asset failure. Asset failures model the failures of those nonsafety critical parts and accidents such as bird strikes for a jet engine. Thus the failure is independent of the states of the safety-critical parts. For simplicity, the failure rate is assumed to be a constant f_r at each time unit. Let ξ_t be a random variable indicating whether the asset fails at t or not ($\xi_t = 1$ or 0, respectively). The asset failure has a following Bernoulli distribution:

$$P(\xi_t = 1) = f_r, \quad \text{and } P(\xi_t = 0) = 1 - f_r. \tag{1}$$

The asset state at time t is thus represented by a vector $\mathbf{x}_t \equiv (x_{1t}, \ldots, x_{Nt}; \xi_t)$, $0 \leqslant x_{nt} \leqslant S_n - 1$, and is within the state space χ. It can be seen that the number of asset states (i.e., combinations of part remaining lives and the asset failure status ξ_t) increases exponentially with the number of parts ($|\chi| = 2\prod_{n=1}^{N} S_n$). Here, $|\cdot|$ denotes the cardinality of the set argument. To reduce the number of states, ξ_t will be removed from the state vector when the problem is modeled by time aggregated Markov decision processes in Section IV.

B. Asset Dynamics

The remaining lives of the parts degrade linearly with time when the asset is in service. Suppose that the asset is in service at t with state $\mathbf{x}_t = (x_{1t}, \ldots, x_{Nt}; 0)$, then all parts' remaining lives decrease by 1 after one time unit as

$$\begin{aligned} &\mathbf{x}_{t+1} = (x_{1t} - 1, \ldots, x_{Nt} - 1; \quad \xi_{t+1}); \\ &\text{for } t = 1, \ldots, T; n = 1, \ldots, N; \end{aligned}$$

and

$$1 \leqslant x_{nt} \leqslant S_n - 1. \tag{2}$$

Part expirations ($x_{nt} = 0$ for some n) or asset failures ($\xi_t = 1$) lead to shop visits. Let $\mathcal{A}$ be the space of replacement actions, and $\mathcal{A}(\mathbf{x}_t)$, a subset of $\mathcal{A}$, represents the set of feasible actions for a shop visit at t. A feasible action $a(\mathbf{x}_t) \in \mathcal{A}(\mathbf{x}_t)$ is a Boolean vector with the nth element denoting whether part n is to be replaced or not ($a_{nt} = 1$ or 0, respectively). Those expired parts must be replaced, and other parts may or may not be replaced, i.e.,

$$a(\mathbf{x}_t) \equiv (a_{1t}, \ldots, a_{Nt}); \quad \text{and} \quad a_{nt} = 1 \text{ if} x_{nt} = 0; \quad \text{else } a_{nt} \in \{1, 0\}. \tag{3}$$

After maintenance, a replaced part n will resume a full life S_n, and those not replaced evolve from their current remaining lives, i.e.,

$$\mathbf{x}_{t+1} = (x_{1t}^+ - 1, \ldots, x_{Nt}^+ - 1; \xi_{t+1}) \tag{4}$$

with $x_{nt}^+ = a_{nt} S_n + (1 - a_{nt}) x_{nt}$, for $t = 1, \ldots, T$; $n = 1, \ldots, N$; and $1 \leqslant x_{nt}^+ \leqslant S_n$.

C. Cost Structures

The maintenance costs have three aspects: the setup cost C_S for having a maintenance visits, module removal costs C_{Mm} for removing module $m, m = 1, \ldots, M$, and new part costs C_{Pn} for replacing part $n, n = 1, \ldots, N$. The setup cost occurs at every shop visit and includes costs for asset downtime, transportation, inspection, etc. The module removal costs depend on the asset structure. For simplicity, an asset is assumed to have a linear structure, and replacing parts of inner modules requires the removal of outer modules. Fig. 2 illustrates that to replace the expired part a in M_2, both M_3 and M_2 should be removed. In this case, a part (as part b in Fig. 2) with nonzero remaining life but close to expiration may be replaced to share model removal costs and the setup cost. To mathematically model module removal costs under this linear structure, modules from inner to outer of the asset are ordered as $1,\ldots,M$; and parts in an inner module have smaller indexes than parts belonging to outer modules. Let $r_m[a(\mathbf{x}_t)](\in \{1,0\})$ represent whether module m is removed or not when $a(\mathbf{x}_t)$ is taken. Then

$$r_m[a(\mathbf{x}_t)] \equiv \max(a_{nt}), \text{ with } n \in 1,\ldots,\sum_{m'=1}^{m} N_{m'}. \quad (5)$$

Therefore, the maintenance cost for a shop visit at time t can be calculated as

$$C[a(\mathbf{x}_t)] = C_S + \sum_{m=1}^{M} r_m[a(\mathbf{x}_t)]C_{Mm} + \sum_{n=1}^{N} a_{nt}C_{Pn}. \quad (6)$$

D. The Total Cost and Average Cost Performance Criteria

The above formulation is Markovian, i.e., at time t, the state transitions, transition probabilities and action costs are determined by the current state and action and are independent of historical information. Thus the problem is a Markov decision process (MDP). For the MDP, a policy φ is a mapping $\varphi : \chi\mathcal{A}$, where φ specifies an action $a^{\varphi}(\mathbf{x}_t) \in \mathcal{A}(\mathbf{x}_t)$ for any state $\mathbf{x}_t$ at a shop visit. Two important performance criteria will be investigated: the finite horizon total cost (total cost for short) and infinite horizon average cost (average cost for short).

The total cost criterion models the cases where an asset is maintained by the asset owner over a predefined lifetime or by a maintenance service provider over the term of a contract [2]. The problem is to minimize the expected total cost over a maintenance horizon T

$$\min E\left\{\sum_{t=1}^{T} C[a^{\varphi}(x_t)]\right\}, \ s.t.\ a^{\varphi}(\mathbf{x}_t) \in A(\mathbf{x}_t)\mathbf{x}_1 = \mathbf{x}. \quad (7)$$

The optimal policies for (7) are generally nonstationary. Namely, an optimal action depends not only on the asset state but also on the time of the shop visit.

The average cost criterion models the cases of indefinite future of asset lifetime or contract term. This performance criterion has been widely investigated in the literature, e.g., [8], [12], and [18]. The problem is to minimize the flowing long-run average cost as

$$\begin{aligned} &\min \eta^{\varphi}, \eta^{\varphi} \equiv \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} C[a^{\varphi}(\mathbf{x}_t)], \\ &s.t.\ a^{\varphi}(\mathbf{x}_t) \in A(\mathbf{x}_t). \end{aligned} \quad (8)$$

In contrast to (7), the optimal policies for the average cost problems in (8) are stationary, i.e., the policies are independent of time. It should be noticed that this average cost problem, as compared with those problems in the literature, may be a multichain MDP [16]. Namely, there exists a feasible policy leading to more than one recurrent class within the state space. Consider a case where that an asset with two identical parts is maintained by a policy which only replaces the expired part. There are at least two recurrent classes: two parts with same remaining lives and two parts with different remaining lives. Multichain MDPs are more difficult to address than unichain MDPs.

IV. OSA Based on Time Aggregated Markov Decision Processes

A. Overview

This paper addresses those difficulties of the problem presented in Section I by using a rollout optimization framework. In this framework, Q-factors, which identify the goodness of feasible actions for a state, are estimated to select an action for the state at a shop visit. The estimates are achieved through Monte Carlo simulations, which run a certain base policy on a simulation model. Thus the optimization framework is an online approach, and is in essence a single step of policy iteration over the base policy. Such a base policy is obtained from certain efficient heuristic method, e.g., OSA developed in this section or the existing threshold method.

The average cost problem is analyzed in this section to develop a computationally efficient and cost effective heuristic method within the rollout framework. To achieve this goal, traditional MDPs are briefly reviewed first. To reduce the complexity, time aggregated MDPs [4] are introduced and applied to our problem. The application is innovative for maintenance problems and simplifies the analysis and computation as compared with traditional MDPs. Although the time aggregated MDP is still intractable for solving large problems, it helps derive a new formula to compute the performance of a policy. Based on this formula, OSA is developed to approximate optimal policies through minimizing the expected average cost over one maintenance period. The performance bounds derived for single-part problems reveal the situations where OSA achieves near-optimal performance. In addition, the policies obtained by OSA are proved to share some common features with optimal policies.

B. Markov Decision Processes

MDPs are used to characterize sequential decision problems with Markovian properties. A classical algorithm for solving MDPs is policy iteration which is based on the Bellman optimality equation [16]. The core of this algorithm is a set of simultaneous linear equations, the so called Poisson equation [4] (the index t of $\mathbf{x}$ is removed in view of the stationarity of average cost MDPs)

$$\begin{aligned} \eta^{\varphi} = C[a^{\varphi}(\mathbf{x})] &+ \sum_{\mathbf{x}'\in X} p[\mathbf{x}'|\mathbf{x}, a^{\varphi}(\mathbf{x}).]g^{\varphi}(\mathbf{x}') \\ &- g^{\varphi}(\mathbf{x}), \text{ for all } \mathbf{x} \in X. \end{aligned} \quad (9)$$

In the above, $p[\mathbf{x}\prime|\mathbf{x}, a^{\phi}(\mathbf{x})]$ is the transition probability from $\mathbf{x}$ to $\mathbf{x}\prime$ when $a^{\phi}(\mathbf{x})$ is taken according to the policy ϕ. The term g^{ϕ} is the performance potential ([3], or relative value in [16]) which signifies the goodness of a state under φ. After the variable g^{ϕ}

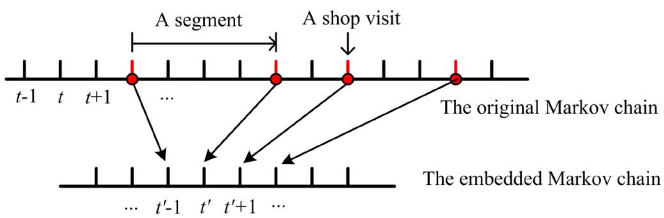


Fig. 3. Controllable states marked with circles are picked from the original Markov chain to form an embedded Markov chain. For our problem, the asset states are controllable at shop visits. A segment is generated by the states between two successive shop visits.

is obtained by solving $|\chi|$ simultaneous linear (9) ([4] and [16]), the policy ϕ is improved through the following step:

$$a(x) \in \arg\min_{a\in A(\mathbf{x})}\left\{C(a)+\sum_{\mathbf{x}'\in X} p[\mathbf{x}'|\mathbf{x},a]g^{\varphi}(\mathbf{x}')\right\} \\ = \arg\min_{a\in A(\mathbf{x})}\{C(a)+E[g^{\varphi}(\mathbf{x}')|\mathbf{x},a]\}. \quad (10)$$

The policy iteration algorithm calculates g^{ϕ} and carries out the policy improvement step (10) iteratively until an optimal policy is obtained. This is intractable for practical problems with a large number of states. Nevertheless, MDPs play important roles on modeling the problem, characterizing optimal policies and analyzing the performance of certain heuristic methods for maintenance problems [15], [20], [22].

C. Time Aggregated Markov Decision Processes

To make MDPs applicable for large problems, recent studies focus on exploring the problem structures by using e.g., state aggregation or time aggregation. The time aggregation approach applies to problems where some states are controllable and some are not. The controllable states divide the original Markov chain into segments. These states are picked out to form an embedded Markov chain as illustrated in Fig. 3. The expected cost and the length of a segment are aggregated to define a new cost function for the embedded chain [4]. This is called time aggregated MDP (TAMDP) which can be solved by using the corresponding policy iteration algorithm.

The above time aggregated approach is well suited for our problem in view that there is no need to make replacement decisions when the asset is in service [23]. Thus considering state transitions directly between shop visits instead of time units is a more efficient and elegant way. Those states at shop visits, denoted by $\tilde{\chi}(\tilde{\chi} \subseteq \chi)$, are selected to construct an embedded Markov chain. For any state $\mathbf{x} \in \tilde{\chi}$ without part expirations, the Boolean variable ξ is restricted to 1, i.e., the shop visit must be caused by an asset failure. For those states $\mathbf{x} \in \tilde{\chi}$ with part expirations, the maintenance costs and state transitions are independent of ξ. Therefore, the symbol ξ can be omitted for those states by time aggregation. This reduces the size of the state space to about a half, i.e., $|\tilde{\chi}| \approx 0.5|\chi|$.

To obtain an aggregated MDP, the cost function of the original MDP will be converted as follows. The expected length of a segment, i.e., $E_T[a^{\varphi}(\mathbf{x})]$, is the expected time units from one shop visit to the next when action $a^{\varphi}(\mathbf{x})$ is taken for $\mathbf{x}$. The expected total cost of a segment is simply the maintenance cost $C[a^{\varphi}(\mathbf{x})]$ at the shop visit. If the chain has a unique recurrent class under ϕ, the performance of ϕ can be calculated by either a long-run sample path defined in (8) or by the steady-state probabilities of the states at shop visits [4]

$$\eta^{\varphi} = \frac{\sum_{\mathbf{x}\in\tilde{\chi}} \pi^{\varphi}(\mathbf{x})C[a^{\varphi}(\mathbf{x})]}{\sum_{\mathbf{x}\in\tilde{\chi}} \pi^{\varphi}(\mathbf{x})E_T[a^{\varphi}(\mathbf{x})]}. \quad (11)$$

In (11), $\pi^{\varphi}(\mathbf{x})$ is the steady-state probability of $\mathbf{x}$ for the embedded chain under ϕ. Following [4], the cost function for a state in the aggregated Markov chain $r[a^{\varphi}(\mathbf{x})]$ involves the expected cost and length of a segment

$$r[a^{\varphi}(\mathbf{x})] \equiv C[a^{\varphi}(\mathbf{x})] - \eta^{\varphi}E_T[a^{\varphi}(\mathbf{x})]. \quad (12)$$

Based on above definitions, iteratively solving the Poisson (9) and carrying out the policy improvement (10) results in a policy optimal for both the aggregated and the original MDP [4, Algorithm 1].

D. OSA Method

Although the number of states is reduced, TAMDP still suffers from the combinatorial difficulty when an asset has many parts and parts have large new lifetimes. In addition, either MDP or TAMDP is complicated because the problem may generate a chain with multiple recurrent classes as analyzed in Section III-D.

Apply the Poisson equation (9) to the aggregated chain with the cost function (12) and the transition probabilities among the aggregated states calculated based on the failure rate and state dynamics. Then the average cost the aggregated MDP will be 0 [4, Eq. 13], i.e.,

$$0 = r[a^{\varphi}(\mathbf{x})] + \sum_{\mathbf{x}'\in\chi} p[\mathbf{x}'|\mathbf{x}, a^{\varphi}(\mathbf{x})]g^{\varphi}(\mathbf{x}') - g^{\varphi}(\mathbf{x}). \quad (13)$$

Therefore, a new formula for calculating the performance of ϕ can be derived as

$$\eta^{\varphi} = \frac{C[a^{\varphi}(\mathbf{x})]}{E_T[a^{\varphi}(\mathbf{x})]} + \frac{\sum_{\mathbf{x}'\in\tilde{\chi}} p(\mathbf{x}'|\mathbf{x}, a^{\varphi}(\mathbf{x}))g^{\varphi}(\mathbf{x}') - g^{\varphi}(\mathbf{x})}{E_T[a^{\varphi}(\mathbf{x})]}. \quad (14)$$

Instead of minimizing the average cost to obtain an optimal policy, OSA, is developed to minimize the first term on the right side of (14) as a heuristic method for large problems. OSA selects an action $a^{\circ}(\mathbf{x})$ for any $\mathbf{x} \in \tilde{\chi}$ as

$$a^{\circ}(\mathbf{x}) \in \mathrm{argmin}_{a(\mathbf{x})\in\mathcal{A}(\mathbf{x})}\{\Theta^{\circ}[a(\mathbf{x})]\} \quad (15)$$

where

$$\Theta^{\circ}[a(\mathbf{x})] \equiv \frac{C[a(\mathbf{x})]}{E_T[a(\mathbf{x})]},\ a(\mathbf{x}) \in \mathcal{A}(\mathbf{x}). \quad (16)$$

It can be seen that OSA minimizes the ratio of the cost to the expected length between two subsequent shop visits. Similar idea can be found for other maintenance problems in the literature, e.g., minimize the "cost rate" in [18] for preventive maintenance.

E. Characteristics of OSA

From (14)–(16), it can be seen that OSA approximates the optimality equation of TAMDPs through dropping the effect of potentials g^{φ}. The approximation works well when optimal policies result in similar potential values for different states so that the second term on the right of (14) has little effect to the performance. To illustrate this, consider an extreme case where the optimal policy replaces all the parts at every shop visit (this may be true if the setup cost is very high). This results in the same action cost and state transitions for all states. Therefore, potentials are identical for all the states in TAMDP according to (13). Then the second term on the right of (14) is zero and OSA obtains optimal policies. For general cases, the performance and the policy structures obtained by OSA are analyzed below.

1) Single-Part Problems: To obtain insights about the situations where OSA performs well, single-part problems will be investigated first. Since the asset has only one part, an optimal policy for such a problem replaces the part at shop visits when its remaining life is less than a threshold [20]. Based on this simple structure of optimal policies, the performance bounds of OSA are derived as follows.

2) Proposition 1: For a single-part problem, let η° be the average cost obtained by OSA, and η^* be the optimal performance, then

$$\eta^* \leqslant \eta^{\circ} \leqslant \eta^* \left(1 + \frac{1-(1-f_r)^{S-1}}{\beta + (1-f_r)^{S-1}}\right) \text{ where } \beta \equiv \frac{C_S}{C_P}. \tag{17}$$

In the above, S is the full lifetime and C_P is the part cost. The detailed proof is provided in Appendix A.

Proposition 1 shows that OSA achieves near optimal performance especially when the failure rate or the ratio of part cost to setup cost is low. This will also be demonstrated in Section VI on multipart problems for which the performance bounds cannot be analytically obtained.

3) Multipart Problems: The optimal policies for multipart problems have irregular structures ([15], [17]). Nevertheless, two necessary conditions on optimal policies have been known. The first is "*Shortest Remaining Life First Rule* [23]," which states that optimal policies always first replace those parts who have less remaining lives than other parts, i.e., the following.

4) Necessary Condition 1: If n and $n\prime$ belong to the same module and $x_n \leqslant x_{n\prime}$, then $a_n \geqslant a_{n\prime}$.

This condition presents a feature of an optimal action for each state. The second condition, modified from item b of [15, Theorem 5] for our model, deals with the relations among actions for different states. The intuition of this condition is that if a part is replaced according to an optimal action for a state, then the part should also be replaced in a "worse" state. Let $N_R\{a(\mathbf{x})\}$ be the set of parts that are replaced according to $a(\mathbf{x})$, i.e., $N_R\{a(\mathbf{x})\} \equiv \{n | n \in 1,\ldots,N;\text{ and } a_n = 1\}$. Then the formal description of the condition is as follows.

Necessary Condition 2: Suppose that $a^*(\mathbf{x})$ and $a^*(\mathbf{x}\prime)$ are actions for $\mathbf{x}$ and $\mathbf{x}\prime$ according to optimal policies. If $x\prime_n \leqslant x_n$ for $n \in N_R\{a^*(\mathbf{x})\}$, $x\prime_n = x_n$ for $n \notin N_R\{a^*(\mathbf{x})\}$; then $a^*(\mathbf{x}\prime) = a^*(\mathbf{x})$.

Both conditions listed above are intuitively plausible. We have the following proposition, and the proof is provided in Appendix B.

Proposition 2: The actions obtained by OSA satisfy Necessary Conditions 1 and 2.

V. Improve Heuristic Methods by the Rollout Algorithm

In the second phase of our optimization framework, the rollout algorithm [1] is used to improve the policies obtained by OSA or the threshold method. The improvement is achieved by a single step of policy iteration that minimizes Q-factors. Since the rollout algorithm is originally developed for total cost problems, this section will develop Q-factors based on the "potential [3]" to address average cost problems. Since Q-factors are random variables, they are estimated by Monte Carlo simulations. Several methods will be used to alleviate the computationally intensive burden.

A. Definition of Q-Factors

The policy improvement procedure in (10) is based on the potential obtained by solving the Poisson (9). As the asymptotic relative difference in total cost between starting the process from a particular state and from the steady-state distribution, the potential may also be calculated based on sample paths ([3] and [16])

$$g^{\varphi}(\mathbf{x}) = \lim_{T\to\infty} E\left\{\sum_{t=0}^{T-1} [C[a^{\varphi}(\mathbf{x}_t)] - \eta^{\varphi}] | \mathbf{x}_0 = \mathbf{x}\right\}. \tag{18}$$

Employing the potential, a Q-factor is defined for a state to identify the goodness of a feasible action

$$Q^{\varphi}(\mathbf{x}, a) \equiv C(a) - \eta^{\varphi} + E[\mathbf{g}^{\varphi}(\mathbf{x}')|x, a], a \in \mathcal{A}(\mathbf{x}). \tag{19}$$

By (18) and (19), the following equation holds according to the strong law of large numbers:

$$\begin{aligned} Q^{\varphi}(\mathbf{x}, a) &= C(a) - \eta^{\varphi} \\ &\quad + \sum_{\mathbf{x}' \in \chi} p(\mathbf{x}'|\mathbf{x}, a) \left\{ \lim_{K\to\infty} \lim_{W\to\infty} \frac{1}{W} \sum_{w=1}^{W} \right. \\ &\quad \left. \times \left\{ \sum_{k=1}^{K} \{ C\left[a^{\varphi}\left(\mathbf{x}_k^{\xi_w}\right)\right] - \eta^{\varphi} | \mathbf{x}_1^{\xi_w} = \mathbf{x}' \} \right\} \right\} \\ &= \lim_{K\to\infty} \lim_{W\to\infty} \frac{1}{W} \sum_{w=1}^{W} \left\{ \sum_{k=0}^{K} \left\{ C\left[a^{\varphi}\left(\mathbf{x}_k^{\xi_w}\right)\right] \right. \right. \\ &\quad \left. \left. - \eta^{\varphi} | \mathbf{x}_0^{\xi_w} = \mathbf{x}, a^{\varphi}\left(\mathbf{x}_0^{\xi_w}\right) = a \right\} \right\}. \end{aligned} \tag{20}$$

In the above, W is the number of replications and K is the time horizon of sample paths. The symbol ξ_w represents the realization of uncertainties, i.e., random failures for our problem on the wth sample path (or scenario) at each time unit.

B. Rollout Algorithms for Total Cost or Average Cost MDPs

Policy iteration aims at achieving an optimal policy, and minimizes Q-factors for all the states to improve a policy in each iteration. This is intractable for problems with large state spaces. To address large problems, the rollout algorithm relaxes the goal by improving an existing policy for the states encountered. The improvement is conducted by estimating Q-factors through simulations. The simulation takes the action to be evaluated for $\mathbf{x}_t$ at t, and then follows the existing policy thereafter, i.e.,

$$Q(\mathbf{x}_t, a, \xi_w) = C(a) + \sum_{k=1}^{K} C\left[a^{\varphi}\left(\mathbf{x}_{t+k}^{\xi_w}\right)\right], a \in \mathcal{A}(\mathbf{x}_t). \tag{21}$$

The look-ahead horizon in simulations is truncated, and K is a parameter instead of approaching infinity as in (20). The sample mean of a Q-factor is calculated by repeated simulations

$$\bar{Q}(\mathbf{x}_t, a) = \frac{1}{W}\sum_{w=1}^{W} Q(\mathbf{x}_t, a, \xi_w), a \in \mathcal{A}(\mathbf{x}_t). \tag{22}$$

The rollout algorithm selects the action that attains the minimal sample mean of Q-factors

$$a^r(\mathbf{x}_t) \in \arg\min_{a \in \mathcal{A}(\mathbf{x}_t)}\{\bar{Q}(\mathbf{x}_t, a)\}. \tag{23}$$

Note that η^{φ} is not calculated in (22) as in (20) because it is a constant and does not affect the comparisons among actions in (23).

The rollout algorithm is in essence a single step of policy iteration as (10). From (23), it can be seen that only costs at each time unit are calculated based on sample paths, while the exact information of the underling model, e.g., the transition probabilities among states, is not necessarily known. This is the advantage of the rollout algorithm. To reduce the variances of the comparisons of Q-factors, the simulations are run with "common random numbers" [13], i.e., using the same ξ_w for different actions. In our problem, Necessary Condition 1 is applied to the algorithm to cut the number of actions without loss of performance for the policies obtained. The base policy φ may be obtained by either OSA or the threshold method.

For total cost MDPs, the Q-factor is defined as the immediate cost plus the cost-to-go [1], as opposed to the immediate cost plus the expected potential as in (19) for average cost MDPs. Although based on different concepts, the algorithms are similar in implementation. Since the state dynamics and cost structures are the same, the only difference is the look-ahead horizon K in (21) for simulations. For total cost MDPs, simulations look ahead to the end of the maintenance horizon T, i.e., K is computed according to current time t as $K = T - t$. In contrast, for average cost MDPs, K is a truncated parameter which is used to approximate potentials defined by infinite-long sample paths as in (18). Although there is no guideline, our numerical examples demonstrate that K can be readily chosen for average cost MDPs to achieve good results.

C. Improve the Efficiency of the Rollout Algorithm

The rollout algorithm estimates Q-factors through repeated Monte Carlo simulations running a base policy. The estimation accuracy (e.g., the confidence interval) improves slowly and no faster than $O(1/\sqrt{W})$ [6], [11]. However, it can be seen from (23) that the key of the algorithm is to correctly determine the orders of Q-factors. From another point of view, the rollout algorithm approximates the true ranks of Q-factors by a crude model. This is because the base policy is obtained by a heuristic method, while a "precise model" employs an optimal policy as the base policy. Therefore, both the order comparison and the use of crude model for the rollout algorithm are consistent with the tenets of ordinal optimization [11]. By ordinal optimization, it is much easier to determine "order" than "value." The order converges exponentially as compared with the slow convergence of value [11]. By incorporating the concept of ordinal optimization, our idea for the rollout algorithm is that there is no need to estimate Q-factors precisely by excessive simulation runs.

Besides the idea of ordinal optimization, the simulation can be made more efficient by using the "optimal computing budge allocation" (OCBA) [6] technique. Rather than directly performing the same number of replications W, more replications may be conducted for those promising actions with small sample means of Q-factors. This improves the correctness of the final selection. In addition, the simulations to estimate Q-factors have intrinsic parallelism. Except for actions, other ingredients are the same, e.g., the current state, simulation model and the uncertainties realized based on common random numbers. Thus the program can be easily implemented by using parallel simulation in the *Single Program Multiple Data* fashion.

VI. Numerical Results

OSA and the rollout algorithm developed in Sections IV and V have been implemented using the programming language C++ and tested on a PC with Pentium IV 2.0 GHz CPU, 512 Mb RAM, and Windows XP OS. For comparison purpose, the generally used fixed-threshold method has also been implemented. Extensive numerical testing has been conducted. Three examples are presented below and each examines several problem variations. The first example tests *Proposition* 1 of Section IV for OSA by two-part small problems for which optimal policies can be obtained by TAMDP. The second example examines the performance of the rollout algorithm under several parameter settings for both total cost and average cost criteria with medium-sized problems. The effectiveness of OSA and the rollout algorithm is demonstrated. The third example demonstrates the effectiveness and the computational efficiency of the overall rollout optimization framework for large problems by using a semi-realistic data set. Remarks on the two types of criteria are provided based on the results obtained.

Example 1: Consider a simple example where an asset has two modules and each module has only one part. Since the performance bounds of OSA are obtained for average cost MDPs, this example tests OSA with the objective to minimize the average cost. Although the problem is small, it has irregular structured optimal policies as large problems [15] and [17].

The average costs obtained by OSA for problems with different setup costs and failure rates are compared with the optimal costs η^* obtained by using TAMDP as presented in Table I.

TABLE I
COMPARE THE PERFORMANCE OF OSA WITH OPTIMAL POLICIES

$f_r = 0.1$			$C_S = 30$		
C_S	η^*	OSA	f_r	η^*	OSA
1	4.934	5.355	0.5	21.16	21.78
10	7.227	7.624	0.2	13.64	13.98
20	9.454	9.828	0.15	12.52	12.58
30	11.48	11.56	0.1	11.48	11.56
40	13.49	13.52	0.05	10.54	10.54
70	19.39	19.40	0.01	9.868	9.869
90	23.29	23.29	0	9.714	9.714

$C_{M1} = C_{M2} = 4$, $C_{P1} = 20$, $C_{P2} = 10$

TABLE II
TOTAL COSTS FOR PROBLEMS WITH MAINTENANCE HORIZON $T = 300$ AND 500

Rollout Base Policies		Without Rollout	Replication Number: W				
			5	20	100	250	500
Th	M/300	0.696	0.668	0.636	0.626	0.624	0.623
	STD	12.3	10.2	7.5	6.6	6.6	6.8
	M/500	0.690	0.659	0.628	0.623	0.621	0.620
	STD	20.2	13.5	8.5	7.9	7.7	8.2
OSA	M/300	0.643	0.624	0.621	0.620	0.619	0.620
	STD	6.8	7.2	6.5	6.6	6.7	6.7
	M/500	0.627	0.630	0.624	0.621	0.621	0.621
	STD	8.7	10.2	9.6	8.8	9.0	8.9

Th: the fixed-threshold method.

TABLE III
AVERAGE COSTS OF THE ROLLOUT ALGORITHM BASED ON THE THRESHOLD METHOD

Look ahead: K	Replication Number: W					CPU Time: W=500
	5	20	100	250	500	
20	0.688	0.691	0.693	0.693	0.693	0.15 s
50	0.677	0.665	0.661	0.661	0.661	0.37 s
150	0.670	0.641	0.633	0.632	0.632	1.10 s
300	0.670	0.644	0.630	0.628	0.627	2.16 s
500	0.684	0.647	0.632	0.629	0.628	3.60 s

For this problem, the threshold method achieved average cost 0.690 and OSA achieved 0.627.

TABLE IV
TESTING METHODS BY SEMI-REALISTIC DATA SET FOR TOTAL COST

		Threshold	OSA	Rollout-T	Rollout-O
$f_r = 0$	Mean	633.3	617.0	605.7	597.7
	STD	9.2	11.3	6.0	5.3
	CPU Time	30 sec *	0.001 s	0.61sec	31 sec
$f_r =$ 0.015	Mean	668.5	637.6	618.7	618.4
	STD	13.2	11.7	8.3	8.4
	CPU Time	40 sec *	0.002 sec	1 min	50 min

Rollout-T: the rollout algorithm based on the threshold method, Rollout-O: the rollout algorithm based on OSA.

The results demonstrate that OSA obtains near or even true optimal costs (e.g., $C_S = 90, f_r = 0.1$ and $C_S = 30, f_r = 0$) especially for situations where the failure rate or the ratio of part costs to setup cost is low. Therefore, this multipart example also exhibits the property of *Proposition* 1 for OSA.

Example 2: An asset has three modules and each module consists of two parts. The full lifetimes of these six parts range from 30 to 70 time units. It is hard to obtain optimal policies by TAMDP because the number of states $\prod_{n=1}^{6} S_n$ is large. OSA and the rollout algorithm are tested under different replication number W for total cost problems, and tested under different W and K for average cost problems.

Testing Total Cost Problems: Two maintenance horizons of 300 and 500 are considered. Since total costs depend on initial states, Table II summarizes the means and standard deviations of the total costs for 100 samples with randomly generated initial asset states. For easy comparison with average cost problems, the means of total costs are divided by T and shown as average costs over the maintenance horizons. The results show that OSA performs much better than the threshold method. Since OSA itself has achieved near optimal performance, the improvement of the rollout algorithm for OSA is not as significant as that for the threshold method. Take the case of $T = 300$ for example. OSA achieved 0.643, which is significantly less than 0.696 obtained by using the threshold method. These two costs decreased to 0.619 and 0.623, respectively, by rolling out the corresponding policies. As pointed out in Section V-C, Table II reveals that there is no need to estimate Q-factors precisely by excessive simulation replications: good enough results can be obtained with only 20 replications for all the four cases tested.

Testing Average Cost Problems: Since the improvement for the threshold method by the rollout algorithm is significant as observed in Table II, the threshold method is used by the rollout algorithm to investigate the effects of parameters K and W. The maintenance horizon is set to be 500 000 time units, which is long enough to approximate infinite horizon for this problem. The average costs obtained are summarized in Table III. It can be seen that the costs may not decrease by increasing K or W alone. The reason is that a short look-ahead horizon K is short sighted. However, long look-ahead horizon with small replication number W may lead to large estimation variances of Q-factors. Nevertheless, the rollout algorithm drastically improves the base policy under most settings of K and W.

Example 3: Consider a semi-realistic data set provided by the United Technologies Research Center in conjunction with Pratt & Whitney. A jet engine has three modules, which consist of 3, 7, and 10 safety-critical parts from inside to outside. The full lifetimes of these 20 parts range from 50 to 140 time units. The problem has been tested under both total cost and average cost criteria. Two failure rates of 0 and 0.015 are tested. When $f_r = 0$, the problem is deterministic and a Q-factor can be computed perfectly by only one sample path $(W = 1)$. When $f_r = 0.015$, each Q-factor is estimated by 100 sample runs.

Testing Results for the Semi-Realistic Data Set: For the total cost criterion, the sample means and standard deviations of the total costs over 1000 maintenance horizon are summarized in Table IV based on 50 simulation runs starting from randomly generated initial states. Under both failure rates, the results in Table IV show that OSA significantly outperforms the threshold

TABLE V
TESTING METHODS BY SEMI-REALISTIC DATA SET FOR AVERAGE COST

		Threshold	OSA	Rollout-T
f_r= 0.015	Average cost	0.6658	0.6351	0.6219
	CPU Time	400 sec *	0.002 sec	40 sec

method. The rollout algorithm drastically improved the performance of the base policies obtained by using either OSA or the threshold method.

For the average cost criterion, the performance is evaluated by a sample path with a long enough maintenance horizon, which consists of thousands of shop visits. In the interest of the computational efficiency, the rollout algorithm employs a base policy obtained by the threshold method. Similar to the results in Example 2 for middle-scale problems, the long-run average costs in Table V obtained by the methods are close to the time unit costs in Table IV for total cost problems.

CPU Time of the Methods: The CPU time required to obtain an action for a state is provided in Table IV. For the threshold method, the time is spent on the exhaust search for an optimal threshold by a number of sample paths (100). For the total cost problem with $f_r = 0.015$, it takes 40 seconds to obtain the threshold. Given an optimal threshold, the action for a state is obtained by comparing the remaining lives of the parts with the threshold. In contrast, OSA chooses an action for a state according to (15) at every time unit. This takes about 0.0002 seconds in this example. Accordingly, the CPU times required by the rollout algorithm are different under variant base policies. When $W = 100$ and $K = 1000$, the CPU times for the rollout algorithm based on the threshold method and OSA are 1 and 50 min, respectively. The CPU times for large problems demonstrate that the effective rollout framework is computationally acceptable.

Remarks: For total cost problems, the "parallel rollout algorithm" [5] employs several base policies to estimate Q-factors separately and select the smallest Q-factor. For some cases, the parallel rollout algorithm may achieve a better policy than any policy obtained by rolling out one single base policy as shown in [5]. We have implemented the parallel algorithm with base policies obtained by OSA and the threshold method. In our experiments, almost all the smallest Q-factors in the parallel rollout algorithm are obtained under base policies obtained by OSA. Thus the parallel rollout algorithm achieves the same results as the rollout algorithm based on OSA.

For average cost problems, the rollout algorithm is developed under the assumption that only one recurrent class exists under all feasible policies for the MDP. However, our problem is essentially a MDP involving multiple recurrent classes, as has been analyzed. Although the application is flawed, the algorithm results in performance improvement for the rollout algorithm as demonstrated in Examples 2 and 3.

VII. CONCLUSION

The main contribution of this paper is to model and address a new joint replacement problem with life-limited parts. Our OSA and the overall optimization framework are cost effective and computationally efficient as demonstrated by numerical testing results. The results also reveal that the optimization should use the rollout algorithm to improve heuristic methods if CPU time is not a critical issue. An interesting result is that the policies obtained by the threshold method, although may be much worse than those policies obtained OSA, can be significantly improved by the rollout algorithm. Thus the cost savings may be significant in view of the wide application of the threshold method in practice.

The simulation based rollout algorithm is flexible to address some variations of our problem through incorporating them into the simulation model. These variations, for example, may be nonconstant failure rate of the asset or have constraints on the asset conditions at the end of a contract, e.g., the remaining lives should be above some levels. The framework can be extended to those maintenance problems in the literature. In fact, threshold type policies are generally used for those problems and can be similarly improved by the rollout algorithm as done in our paper.

APPENDIX A

Proposition 1: For a single-part problem, let η° be the average cost obtained by OSA and η^* be the optimal performance, then

$$\eta^* \leqslant \eta^\circ \leqslant \eta^* \left(1 + \frac{1-(1-f_r)^{S-1}}{\beta + (1-f_r)^{S-1}}\right), \quad \beta \equiv \frac{C_S}{C_P}. \tag{24}$$

In the above, S is the full lifetime and C_P is the new part cost.

Proof: To get the performance bounds, the effect of the second item on the right side of (14) is examined first. For a single-part problem, there are only two actions a_0 (not replace) and a_1 (replace). The asset state $\mathbf{x}$ at a maintenance decision time may be $0, \ldots, S-1$. For state 0, the only feasible action is a_1. When (12) is used as the cost function of the aggregated chain, by the optimality of the policy [16]

$$g^*(0) = c(a_1) - \eta^* E_T(a_1) + \sum_{j=0}^{S-1} p(j|S) g^*(j),$$

$$g^*(\mathbf{x}) = \min \begin{cases} c(a_0) - \eta^* E_T(a_0) + \sum_{j=0}^{\mathbf{x}-1} p(j|\mathbf{x}) g^*(j) \\ c(a_1) - \eta^* E_T(a_1) + \sum_{j=0}^{S-1} p(j|S) g^*(j). \end{cases} \qquad g^*(\mathbf{x}) - g^*(0)$$

$$= \min \Bigg\{ - C_P + \eta^* [E_T(a_1) - E_T(a_0)] + \sum_{j=0}^{\mathbf{x}-1} p(j|\mathbf{x}) g^*(j) - \sum_{j=0}^{S-1} p(j|S) g^*(j); 0 \Bigg\}. \tag{25}$$

The expected length of the next shop visit $E_T(a_1)$ is larger than $E_T(a_0)$ for any $\mathbf{x}$. In addition, it is easy to prove that $g^*(\mathbf{x}) \leqslant g^*(\mathbf{x}')$ if $\mathbf{x} \geqslant \mathbf{x}'$ based on (14) and the optimality. Thus by the definition of transition probabilities and (25)

$$-C_p \leqslant g^*(\mathbf{x}) - g^*(0) \leqslant 0. \tag{26}$$

OSA chooses action as

$$a^{\circ}(x) = \arg\min_{a(x)\in\{a_1,a_0\}} \left\{ \frac{C(a(x))}{E_T[a(x)]} \right\}. \tag{27}$$

Let π° be a row vector of the steady-state probabilities of the aggregated Markov chain under the policy obtained by OSA. By (11) and (27), the performance of OSA is

$$\eta^{\circ} = \frac{\sum_{x=0}^{S-1} \pi^{\circ} C[a^{\circ}(x)]}{\sum_{x=0}^{S-1} \pi^{\circ} E_T[a^{\circ}(x)]} \leqslant \frac{C(a_1)}{E_T(a_1)}. \tag{28}$$

Obtain $E_T(a_1)$ from (14), then

$$\frac{C(a_1)}{E_T(a_1)} = \frac{\eta^* C(a_1)}{C(a_1) + \sum_{j=0}^{S-1} p(j|S) g^*(j) - g^*(0)}. \tag{29}$$

Therefore, by (26) and (28)

$$\begin{aligned}\eta^{\circ} &\leqslant \frac{C(a_1)}{E_T(a_1)} \leqslant \frac{\eta^* C(a_1)}{C(a_1) - C_P[1-(1-f_r)^{S-1}]}\\ &= \eta^* \frac{C_S + C_P}{C_S + C_P(1-f_r)^{S-1}}\\ &= \eta^* \left(1 + \frac{1-(1-f_r)^{S-1}}{\beta + (1-f_r)^{S-1}}\right);\ \beta = \frac{C_S}{C_P}.\end{aligned}$$

□

Appendix B

1) Proposition 2: The actions obtained by OSA satisfy Necessary Conditions 1 and 2.

Proof: The actions obtained by OSA satisfy Necessary Conditions 1 can be proved by contradiction as follows. Let a_i° be the action for part i according to $a^{\circ}(\mathbf{x})$ obtained by OSA. If there exist i and j satisfying that $a_j^{\circ} = 1, a_i^{\circ} = 0$ and $x_i < x_j$. Then choose an action $a\prime$ as: $a\prime_n = a_n^{\circ}$ for $n \neq j$, and $a\prime_j = 0$. Thus $E_T[a^{\circ}(\mathbf{x})] = E_T[a\prime(\mathbf{x})]$, while $C[a\prime(\mathbf{x})] < C[a^{\circ}(\mathbf{x})]$, thus $\Theta^{\circ}[a\prime(\mathbf{x})] < \Theta^{\circ}[a^{\circ}(\mathbf{x})]$. This contradicts with (15).

For $\mathbf{x}$ and $\mathbf{x}\prime$, if the assumption of Necessary Condition 2 hold, then $\mathbf{x}\prime \leqslant \mathbf{x}$ and $\mathcal{A}(\mathbf{x}\prime) \subseteq \mathcal{A}(\mathbf{x})$ [by definition of $a(\mathbf{x})$ in (3)]. Let $\Theta^{\circ}[\mathbf{x}, a]$ denote taking action a for $\mathbf{x}$ in (16). For any $a(\mathbf{x}\prime) \in \mathcal{A}(\mathbf{x}\prime)$

$$\begin{aligned}\Theta^{\circ}[\mathbf{x}\prime, a^{\circ}(\mathbf{x})] = \Theta^{\circ}[\mathbf{x}, a^{\circ}(\mathbf{x})] &\leqslant \Theta^{\circ}[\mathbf{x}, a(\mathbf{x}\prime)]\\ &\leqslant \Theta^{\circ}[\mathbf{x}\prime, a(\mathbf{x}\prime)]. \end{aligned} \tag{29}$$

The first equality is obtained by the assumption in Necessary Condition 2. The first inequality follows the definition of OSA in (15). The second inequality is obtained by (16) noticing that $E_T[a(\mathbf{x}\prime)] \leqslant E_T[\mathbf{a}(\mathbf{x})]$ for the same a. Thus the actions obtained by OSA satisfy Necessary Condition 2. □

References

[1] D. P. Bertsekas and D. A. Castañon, "Rollout algorithms for stochastic scheduling problems," *J. Heuristics*, vol. 5, pp. 89–108, 1999.

[2] R. A. Bowman and J. Schmee, "Pricing and managing a maintenance contract for a fleet of aircraft engines," *Simulation*, vol. 76, pp. 69–77, 2001.

[3] X. R. Cao, "From perturbation analysis to Markov decision processes and reinforcement learning," *Discrete Event Dyn. Syst.: Theor. Appl.*, vol. 13, pp. 9–39, 2003.

[4] X. R. Cao, Z. Y. Ren, S. Bhatnagar, M. Fu, and S. Marcus, "A time aggregation approach to Markov decision process," *Automatica J. IFAC*, vol. 38, pp. 929–943, 2002.

[5] H. S. Chang, R. Givan, and E. K. P. Chong, "Parallel rollout for online solution of partially observable Markov decision processes," *Discrete Event Dyn. Syst.: Theor. Appl.*, vol. 14, pp. 309–341, 2004.

[6] C. H. Chen, J. Lin, E. Yücesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization," *Discrete Event Dyn. Syst.*, vol. 10, pp. 251–270, 2000.

[7] D. I. Cho and M. Parlar, "A survey of maintenance models for multi-unit systems," *Eur. J. Oper. Res.*, vol. 51, no. 1, pp. 1–23, Mar. 6, 1991.

[8] R. Dekker, F. A. Van der Duyn Schouten, and R. E. Wildeman, "A review of multi-component maintenance models with economic dependence," *Math. Method. Oper. Res.*, vol. 45, pp. 411–435, 1997.

[9] R. Dekker, R. E. Wildeman, and R. van Egmond, "Joint replacement in an operational planning phase," *Eur. J. Oper. Res.*, vol. 91, no. 1, pp. 74–88, May. 24, 1996.

[10] A. Haurie and P. Lecuyer, "A stochastic control approach to group preventive replacement in a multicomponent system," *IEEE Trans. Autom. Contr.*, vol. AC-27, pp. 387–393, 1982.

[11] Y. C. Ho, Q. C. Zhao, and Q. S. Jia, *Ordinal Optimization: Soft Optimization For Hard Problems*. New York: Springer, 2007.

[12] W. J. Hopp and Y. L. Kuo, "Heuristics for multicomponent joint replacement: Applications to aircraft engine maintenance," *Nav. Res. Log.*, vol. 45, pp. 435–458, 1998.

[13] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 3rd ed. New York: McGraw-Hill, 2000, pp. 582–590.

[14] P. Lecuyer and A. Haurie, "Preventive replacement for multicomponent systems: An opportunistic discrete-time dynamic programming model," *IEEE Trans. Rel.*, vol. R-32, pp. 117–118, 1983.

[15] S. Özekici, "Optimal periodic replacement of replacement of multicomponent reliability system," *Oper. Res.*, vol. 36, pp. 542–552, Jul./Aug. 1988.

[16] M. L. Puterman, *Markov Decision Processes. Discrete Stochastic Dynamic Programming*. New York: Wiley, 1994.

[17] T. Sun, Q. Zhao, P. B. Luh, and R. N. Tomastik, "Joint replacement optimization for multi-part maintenance problems," in *Proc. IEEE Conf. Intelligent Robots Syst.*, 2004, pp. 1232–1237.

[18] J. S. Tan and M. A. Kramer, "A general framework for preventive maintenance optimization in chemical process operations," *Comput. Chem. Eng.*, vol. 21, no. 12, pp. 1451–1469, 1997.

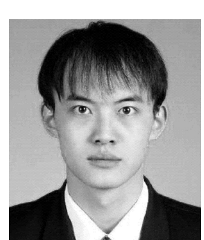
[19] G. Tu, P. B. Luh, and Q. Zhao, "An optimization method for joint replacement decisions in maintenance," in *Proc. IEEE Conf. Decision Control*, 2004, vol. 4, pp. 3674–3679.

[20] F. A. Van der Duyn Schouten and S. G. Vanneste, "Analysis and computation of (n, N) strategies for maintenance of a two-component system," *Eur. J. Oper. Res.*, vol. 48, pp. 260–274, 1990.

[21] H. Z. Wang, "A survey of maintenance policies of deteriorating systems," *Eur. J. Oper. Res.*, vol. 139, no. 3, pp. 469–489, Jun. 16, 2002.

[22] D. J. D. Wijnmalen and J. A. M. Hontelez, "Coordinated condition-based repair strategies for components of a multi-component maintenance system with discounts," *Eur. J. Oper. Res.*, vol. 98, pp. 52–63, Apr. 1997.

[23] L. Xia, Q. Zhao, and Q. Jia, "A structure property of optimal policies for a class of maintenance problem with safety-critical components," *IEEE Trans. Autom. Sci. Eng.*, to be published.



Tao Sun (S'03) received the B.S. degree in automatic control from Tsinghua University, Beijing, China, in 2003. He is currently pursuing the Ph.D. degree in the Center for Intelligent and Networked Systems, Tsinghua University.

His research interests include modeling and optimizing the discrete complex systems and scheduling of manufacturing systems.

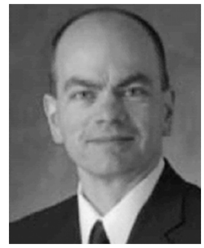
Qianchuan Zhao (M'06) received the B.E. degree in automatic control in July 1992, the B.S. degree in applied mathematics in July 1992, and Ph.D. degree in control theory and its applications in July 1996, all from Tsinghua University, Beijing, China.

He is currently a Professor and Associate Director of the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University. He was a Visiting Scholar at Carnegie Mellon University, Pittsburgh, PA, and Harvard University, Cambridge, MA, in 2000 and 2002, respectively. He was a Visiting Professor at Cornell University, Ithaca, NY, in 2006. His research interests include discrete event dynamic systems (DEDS) theory and applications, optimization of complex systems, and wireless sensor networks. He is an associate editor for the *Journal of Optimization Theory and Applications*.

Peter B. Luh (S'77–M'80–SM'91–F'95) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973; the M.S. degree in aeronautics and astronautics engineering from Massachusetts Institute of Technology, Cambridge, in 1977; and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1980.

Since then, he has been with the University of Connecticut, Storrs, and is currently the SNET Professor of Communications and Information Technologies and the Head of the Department of Electrical and Computer Engineering. He is also a Visiting Professor at the Center for Intelligent and Networked Systems, Department of Automation, Tsinghua University, Beijing, China. He is interested in planning, scheduling, and coordination of design, manufacturing, and supply chain activities; configuration and operation of elevators and HVAC systems for normal and emergency conditions; schedule, auction, portfolio optimization, and load/price forecasting for power systems; and decision-making under uncertain or distributed environments.

Dr. Luh is Vice President of Publication Activities for the IEEE Robotics and Automation Society, the founding Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, an Associate Editor of *IIE Transactions on Design and Manufacturing*, an Associate Editor of *Discrete Event Dynamic Systems*, and was the Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION (1999–2003).



Robert N. Tomastik (SM'01) received the B.S. degree from the University of Connecticut, Storrs, in 1989, the M.Eng. degree from Rensselaer Polytechnic Institute, Troy, NY, in 1992, and the Ph.D. degree from the University of Connecticut in 1995, all in electrical engineering.

He is currently Manager of Materials Management Center of Excellence at Pratt & Whitney, a division of United Technologies, where his group develops decision support tools to improve the company's commercial engine aftermarket supply chain. His research expertise includes mathematical programming, controls, and estimation.