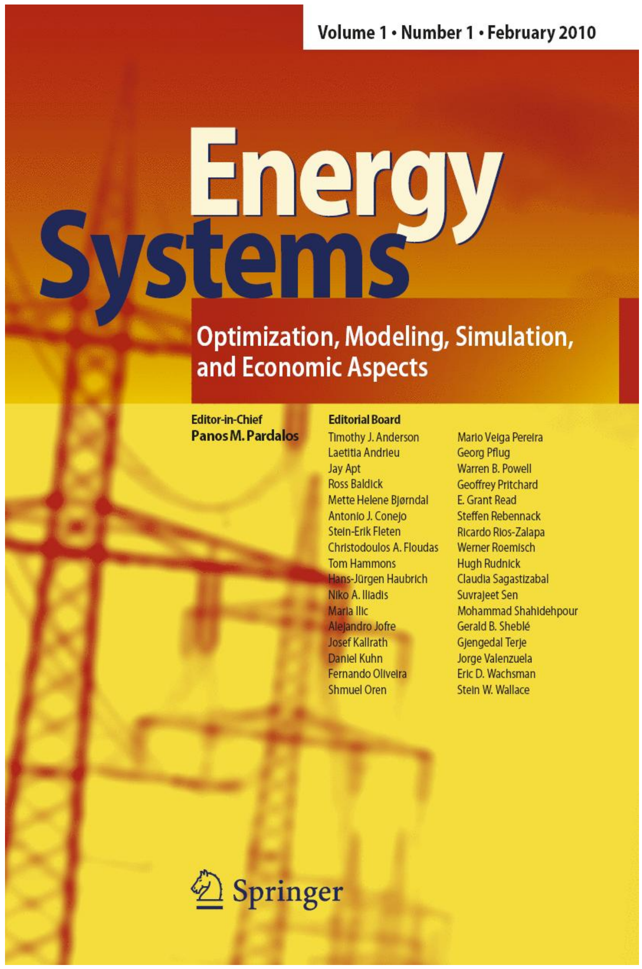
ISSN 1868-3967, Volume 1, Number 1



Energy Syst (2010) 1: 31–59
DOI 10.1007/s12667-009-0006-5

ORIGINAL PAPER

A dynamic model for the failure replacement of aging high-voltage transformers

Johannes Enders · Warren B. Powell · David Egan

Received: 26 May 2009 / Accepted: 14 December 2009 / Published online: 4 January 2010

Abstract As the electric transmission system in the U.S. ages, mitigating the risk of high-voltage transformer failures becomes an increasingly important issue for transmission owners and operators. This paper introduces a model that supports these efforts by optimizing the acquisition and the deployment of high-voltage transformers dynamically over time. We formulate the problem as a Markov Decision Process which cannot be solved for realistic problem instances. Instead we solve the problem using approximate dynamic programming using three different value function approximations, which are compared against an optimal solution for a simplified version of the problem. The methods include a separable, piecewise linear value function, a piecewise linear, two-dimensional approximation, and a piecewise linear function based on an aggregated inventory that is shown to produce solutions within a few percent with very fast convergence. The application of the best performing algorithm to a realistic problem instance gives insights into transformer management issues of practical interest.

1 Introduction

In the 1960's and 70's, the electric power industry underwent a major expansion requiring a significant investment in high-voltage transformers. This investment has produced an age distribution with a bubble of older transformers that will begin failing at a higher than average rate. We consider the problem faced by PJM Interconnections, the largest regional transmission operator in the U.S. responsible for

J. Enders · W.B. Powell (✉)
Department of Operations Research and Financial Engineering, Princeton University, Princeton, USA
e-mail: powell@princeton.edu

D. Egan
PJM Interconnection, Norristown, USA

controlling the flow of energy through the electric power grid covering states from New Jersey, New York, Pennsylvania and Maryland through Ohio and Illinois. In this paper we focus on high-voltage transformers that step up and down the voltage between the backbone transmission system operating at 500 kV and the lower voltage transmission system operating at 230 kV. Currently, approximately 50 percent of the high-voltage transformers in PJM's transmission grid are 30 or more years old. The transformers can cost $5 million each, and require one to two years for delivery.

It is important to maintain a reliable set of transformers since the cost of a failure can be quite high. If a transformer fails, the network incurs costs by needing to purchase energy from more expensive utilities to avoid a bottleneck. These costs can range from several million to as high as $100 million per year. For an industry which operates with very thin profit margins, it is necessary to plan a replacement strategy that strikes a careful balance between potential congestion costs and equipment replacement costs. Given capacity constraints on the industry which makes this equipment, it is necessary to design time-dependent safety stocks to respond to potential failures.

Our model optimizes transformer purchasing and deployment over time taking into account failure uncertainty which changes with time. The model can be used to support the capital investment and risk mitigation planning performed by regional transmission organizations and transmission owners. There are two important trade-offs that we capture with this model:

- The model balances the minimization of adverse system effects due to failures against the efficient use of financial resources. This is the acquisition problem.
- Once a replacement transformer is purchased there is the need to weigh its use to replace a failure now against a possible but uncertain use for a more important failure later. This is the deployment problem.

Such a model can be framed as a special case of an inventory problem. In the language of inventory problems the safety stock of replacement transformers is the "inventory" and failures are "demands." The transformer acquisition and deployment model is an inventory problem with the following four characteristics. (1) Any unserved demand is fully backlogged. (2) There is a multi-period lead time. (3) The inventory serves multiple demand classes, characterized by different shortage costs. (4) The demand process is fairly complex: demands are state dependent and correlated.

We only considered the possibility of replacing transformers that have failed. This is not a fundamental limitation of the model or algorithm. We can handle such a situation by using a rule to designate a transformer as "at risk" (a type of failure) with no congestion cost (but a higher risk of incurring congestion costs). But the reality is that failure tails are extremely long. A high risk transformer can easily last another 10 years, and inventories of spares are going to be sparse given the high capital cost. The decision to swap out an aging but working transformer would be based primarily on issues of risk rather than expected cost.

Figure 1 shows a schematic representation of the transformer acquisition and deployment model at time t. To the left of the dashed line in the figure is the state of the model before decisions at time t. To the right is the state of the model after decisions

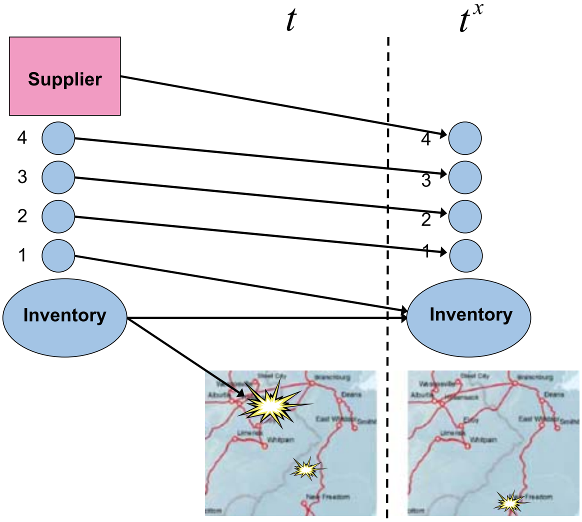


Fig. 1 Schematic representation of the transformer acquisition and deployment model

at time t which we denote t^x. Arrows represent decisions. At time t transformers in inventory can be used to meet a failure or be held in inventory to be available the next time period. If a failure is not met it remains part of the state. The circles represent the pipeline of ordered transformers that are due to arrive to the system in one, two, three, and four periods. High-voltage transformers are not the only domain where such a model can be applied. It is generally applicable to spare provisioning for capital intensive failure prone equipment with long lead times such as jet engines and ship turbines.

Modeling transformer acquisition and deployment as a dynamic stochastic optimization problem is fundamentally different from the approach that is currently being taken by transmission owners. The strategy currently used in practice seeks a spare inventory level such that the expected time between spare shortages is bigger than a threshold value. This approach does not make the cost trade-offs explicit, and nor does it optimize them. The current approach does not provide insight into deployment decisions and does not consider that failure rates change over time.

Backlogging and long lead times are standard features of inventory models. Zipkin [27] gives a thorough account of these topics. It is well known that lead times that are longer than a single period introduce the “curse of dimensionality” into an inventory model. The state of the system needs to track not only how much inventory is available and on order but also when ordered inventory is scheduled to arrive. This fact leads to a high-dimensional state variable.

The fact that there are multiple demand classes gives rise to the deployment problem described above. When is it desirable to backlog lower priority failures in order to have more replacement transformers available for uncertain future high priority failures? The pioneering work in inventory rationing is Topkis [24] who introduces a model with n demand classes, backlogging, and non-stationary demand distributions. This model is to this date the closest match in the inventory rationing literature to the system in this paper. Other contributions to the inventory rationing problem with backlogging include Deshpande et al. [7], De Véricourt et al. [6], Tempelmeier [22], and Ha [10]. The number of demand classes is large in our model because the substation locations have widely varying congestion costs. This contributes to the "curse of dimensionality" in our model.

The failure probability of a transformer in operation depends on its age and physical condition. Therefore these attributes need to be part of the system state which introduces further complexity. Failures are also correlated due to the existence of transformer banks. A bank is a set of transformers that always operate together. Once a transformer in a bank fails, the entire bank shuts down and no further failures in the same bank are possible. This correlation complicates the computation of expectations, such as expected congestion costs.

The issue of transformer spares has received some attention in the literature. Chowdhury and Koval [5] and Kogan et al. [12] develop procedures to determine the optimal number of spares in a static model that assumes a stationary failure process and that all failures have the same priority. Li et al. [14] go beyond the previously mentioned papers as they allow failure events of different priorities and also consider the timing of spare additions. Their approach is rather cumbersome as it does not use any optimization, requires the enumeration and evaluation of all possible failure events, and needs to be repeated whenever the underlying transformer population changes. Our approach uses a realistic model of the physical system, which we solve in a compact way using approximate dynamic programming (ADP).

There is a larger literature on the management of spare parts, although transformers tend to be fairly unique given their size and cost. Simão and Powell [21] use approximate dynamic programming to handle high-value spare parts for an aircraft manufacturer which shares some of the same characteristics (and where we use a similar approach). Wong et al. [25] and Wong et al. [26] propose heuristic policies for multi-item spare parts in the presence of lateral transshipments. Huiskonen [11] provides a nice discussion of the issues surrounding maintenance spare parts, but perhaps the most thorough modern discussion is [15] which considers in depth the use of inventory policies adapted to the management of spare parts, motivated by military applications.

In this paper we propose a series of solution algorithms based on dynamic programming and approximate dynamic programming. The contributions of this paper are twofold: (1) We show that the use of separable, piecewise linear approximations, which has proven to be successful in prior work [8, 23] does not work for this problem class. (2) We introduce two classes of non-separable approximations and show, through comparisons against the exact solution of a simplified MDP, that they produce results between 0 and 4 percent of optimal.

In Sect. 2 we introduce notation and formulate the problem as a multi-period stochastic optimization model. Section 3 discusses the general algorithmic approach and Sect. 4 specifies the value function approximations and the ADP algorithms. In Sect. 5 we describe the numerical work performed to validate the algorithm and an application of our best performing algorithm to a real high-voltage transformer population.

2 Model formulation

A transformer is described by a vector of attributes a consisting of the elements

$$\begin{pmatrix} \text{substation id} \\ \text{bank id} \\ \text{phase} \\ \text{failed} \\ \text{age} \\ \text{condition} \\ \text{availability time} \end{pmatrix}.$$

An example of an attribute vector a is

$$a = \begin{pmatrix} \text{Branchburg} \\ 52 \\ \text{A} \\ \text{false} \\ 39.5 \\ \text{average} \\ 0 \end{pmatrix}.$$

Each transformer in operation belongs to a substation identified by a substation identifier. A substation contains one or more transformer banks and a bank is identified by a bank id. Within a bank a particular transformer is identified by the power phase that it handles; current moves in one of three power phases, which are denoted A, B, or C. Another attribute indicates if a transformer has failed or not. The failure probability over a time interval depends on the age and the condition of the transformer. The age is measured in time periods and the condition can be "good," "average," or "watch." All the attributes described so far are needed to model transformers in operation. One attribute that is important for replacement transformers is the estimated time of arrival, since transformers have to be ordered 12 to 24 months in advance. We refer to the time when a transformer is scheduled to arrive as the available time. An available time of 0 indicates that the transformer is available to be used as a replacement.

We let $\mathcal{A}$ be the set of all possible attribute vectors. For this research, we need to separate transformers that are in operation (that is, in working order), and transformers that have failed and are in need of replacement. The sample attribute vector above describes a transformer that is in working order. An example of an attribute vector for a transformer that is not in working order is

$$a = \begin{pmatrix} - \\ - \\ - \\ \text{false} \\ 0 \\ \text{good} \\ 3 \end{pmatrix},$$

where "–" denotes a missing attribute. To separate transformers that are in operation (in working order) and failed transformers that are in need of replacement, we define

$\mathcal{A}^{\text{rep}}$ = set of attribute vectors for transformers that can be used as replacements,

$\mathcal{A}^{\text{op}}$ = set of attribute vectors for transformers in operation,

$\mathcal{A}^{\text{fail}}$ = set of attributes vectors for transformers that have already failed,

$\mathcal{A} = \mathcal{A}^{\text{op}} \cup \mathcal{A}^{\text{rep}}$ = set of transformer attribute vectors, $a \in \mathcal{A}$.

We need to represent attributes of transformers that have failed. For this purpose, we define

$a^f =$ attribute vector of a failed transformer after we have acted on the transformer d to replace it.

We define

$a_s =$ attribute vector of replacement transformers that are s periods away from being available,
$\tau =$ scalar lead time, giving the number of months that a transformer has to be ordered in advance.

We note that a_s is a particular instance of an attribute vector for a transformer that will arrive in s time periods. There can be a number of attribute vectors describing transformers that are s time periods away. We use this notation to indicate the value of the time at which a transformer will become available. All the replacement transformers have attributes in the set $\{a_0, a_1, \ldots, a_{\tau-1}\} = \mathcal{A}^{\text{rep}}$. a_0 is the attribute vector of a replacement transformer with age 0, that has not failed, is in "good" condition, and has availability time 0, i.e. is in inventory. $a_{\tau-1}$ is the attribute vector of a transformer that has just been ordered. a_τ is the attribute vector of a transformer that has not been purchased but is available for purchase. Note that a_τ is not an element of $\mathcal{A}^{\text{rep}}$.

We now introduce the notation for the system state measured immediately before the decisions at time t are made. This is called the pre-decision state.

$R_{ta} =$ number of transformers with attribute vector a at time t before the time t decisions are made,
$R_t = (R_{ta})_{a \in \mathcal{A}}$ = vector of pre-decision transformers at time t.

Throughout the paper we will also use the state captured immediately after the decisions at time t are made. We use the notation

$R_t^x = (R_{ta}^x)_{a \in \mathcal{A}}$ = vector of post-decision transformers at time t.

Transformers can be acted upon with three decision types:

$\mathcal{D}^{\text{rep}} =$ set of types of decisions to use a replacement transformer in inventory to replace a type of transformer that has failed and is in need of repair. The set contains one decision d_a for each transformer that can be used for replacements, $a \in \mathcal{A}^{\text{rep}}$,
$d^b =$ decision to buy a replacement transformer,

$d^{\emptyset} =$ decision to hold a replacement transformer in inventory to the next period.

For example, imagine that we have three failed transformers at different locations, with different attributes. Denote these by a, a' and a''. The set $\mathcal{D}^{\mathrm{rep}}$ would include the decisions $d_a, d_{a'}$ and $d_{a''}$ representing the decision to replace the transformer with attribute a, a' or a'', respectively.

We define the set of all possible types of decisions using

$$\mathcal{D} = \mathcal{D}^{\mathrm{rep}} \cup d^b \cup d^{\emptyset}.$$

The decision variables and their contributions are defined as:

$x_{tad} =$ number of transformers with attribute a to be acted on with decision d at time t,

$x_t = (x_{tad})_{a\in\mathcal{A}, d\in\mathcal{D}} =$ decision vector at time t,

$x_t^{\mathrm{rep}} = (x_{tad})_{a\in\mathcal{A}^{\mathrm{rep}}, d\in\mathcal{D}^{\mathrm{rep}}} =$ vector of replacement decisions at time t.

The one period contribution function of our model is

$c_{tad} =$ cost of acting on one unit of resource with attribute a with decision d at time t,

$$C_t(x_t) = c_{ta_\tau d^b} x_{ta_\tau d^b} - \sum_{d\in\mathcal{D}^{\mathrm{rep}}} c_{ta_0 d} x_{ta_0 d} + c_{ta_0 d^{\emptyset}} x_{ta_0 d^{\emptyset}}.$$

The first term constitutes the purchase cost of new replacement transformers. The second term is avoided congestion costs for each failure for which the model provides a replacement transformer. The third term is an inventory holding cost charged for all transformers in inventory that are held from one period to the next. We note that congestion costs depend on the number of failed transformers. In practice, the number of failed transformers is very small, since utilities manage their networks very conservatively.

Randomness enters the system through random transformer failures and random transitions from "good" to "average" and from "average" to "watch." A random event always produces one more transformer of a certain type and one less transformer of another type, for example, one more failed transformer and one less working transformer. We denote

$\hat{R}_{ta} =$ the change in the number of transformers with attribute vector a due to random events during period t, and

$\hat{R}_t = (\hat{R}_{ta})_{a\in\mathcal{A}} =$ vector of random changes during time t.

In any given period a working transformer that is in use at a substation can fail, change condition, or remain unaffected by randomness. Each of these events happens with a certain probability which depends on the age of the transformer and its condition. The failure probability also depends on the status of the other transformers in the bank. If one transformer of a bank failed then the other transformers in the same bank cannot fail until the failed transformer is replaced. This dependence arises from the fact that a single transformer failure will cause the entire bank to be shut down. And if a bank is shut down no further failures in that bank can occur.

The transition from R_t to R_t^x is described by the transition equations

$$R^x_{ta_{\tau-1}} = x_{ta_\tau d^b}, \tag{1}$$

$$R^x_{ta_s} = R_{ta_{s+1}} \quad \text{for } s \in \{1, \dots, \tau-2\}, \tag{2}$$

$$R^x_{ta_0} = x_{ta_0 d^\emptyset} + R_{ta_1}, \tag{3}$$

$$R^x_{ta^f} = R_{ta^f} - x_{ta_0 d} \quad \text{for } a^f \in \mathcal{A}^{\text{op}}, d \in \mathcal{D}^{\text{rep}}. \tag{4}$$

Equation (1) models the decision to purchase a transformer. Equation (2) models the transition of a transformer that has not yet arrived from being τ time periods away to $\tau - 1$ time periods away. Equation (3) models the accumulation of transformers in inventory, and (4) models the decision to repair a transformer. The post- to pre-decision transition function simply is

$$R_{t+1} = R_t^x + \hat{R}_{t+1}. \tag{5}$$

Since the model is a multi-period model under uncertainty we wish to find an optimal policy, π, and the corresponding decision rule, $X^\pi(R_t)$ that returns optimal decisions as a function of the system state R_t. The optimization problem is to find

$$\min_\pi \mathbb{E}\left\{\sum_{t=0}^{T} \gamma^t C_t(X^\pi(R_t))\right\}, \tag{6}$$

where γ is a discount factor. Our challenge now is to find a good replacement policy, something that we do with approximate dynamic programming.

3 Algorithmic approach

Stochastic integer programming algorithms have been developed to solve discrete optimization problems under uncertainty. One approach is to formulate the problem as a large-scale deterministic (mixed) integer program, the deterministic equivalent. Parija et al. [18] and Ahmed et al. [1] solve the deterministic equivalent directly. Parija et al. [18] show that taking into account special problem structure results in algorithms that are more efficient than a straightforward application of a general purpose MIP solver. At the same time this work shows that solving the deterministic equivalent is computationally infeasible for the transformer acquisition and deployment model. The largest problem solved in Parija et al. [18] has 18,006 binary variables, 3 time periods, and 500 scenarios. We will be solving the transformer model for up to 200 time periods which would result in much larger deterministic problems.

Another computational approach to solving stochastic integer programs uses decomposition. All major work has been on two-stage problems [3, 13, 17] and does not directly apply to the multi-stage transformer problem.

The starting point for our computational strategy is the classic dynamic programming recursion

$$V_t(R_t) = \min_{x_t \in \mathcal{X}_t(R_t)} \{C(x_t) + \gamma \mathbb{E}[V_{t+1}(R_{t+1})] | R_t\}, \tag{7}$$

where

$$\mathcal{X}_t = \text{ feasible region at time } t,$$
$$V_{t+1}(R_{t+1}) = \text{ value function—expected discounted cumulative costs for time periods } t+1, t+2, \ldots, T.$$

This recursion cannot be solved using classic backward dynamic programming because of the dimensionality of the problem. The long lead time for replacement transformers increases the dimensionality as does the backlogging of failures of varying importance and the fact that transformer age and condition are part of the attribute vector.

The algorithmic approach taken in this paper is approximate dynamic programming (ADP) [2, 19]. This algorithmic strategy has recently been applied with success to large-scale, complex optimization problems under uncertainty [19]. We solve the dynamic program approximately by estimating a value function approximation recursively using Monte Carlo simulation. We use the following notation:

$$\bar{V}_t(R_t^x) = \text{ approximation of value function } V_t(R_t^x).$$

In order to solve the problem using a Monte Carlo based technique we need a recursion that can be solved for a single sample realization of the random information process at time t. Equation (7) cannot be used for this purpose as the knowledge of a time $t+1$ sample realization, $\hat{R}_{t+1}(\omega)$, at time t would violate non-anticipativity. In order to overcome this problem we formulate the following recursion around the post-decision state:

$$V_{t-1}\big(R_{t-1}^x\big) = \mathbb{E}\Big[\min_{x_t \in \mathcal{X}_t(R_t)} \big\{C(x_t) + \gamma V_t\big(R_t^x\big)\big\}\Big] \tag{8}$$

where $\mathcal{X}_t$ is defined by the following constraints:

$$x_{tad^\emptyset} + \sum_{d \in \mathcal{D}^{\text{rep}}} x_{ta_0 d} = R_{ta_0}, \tag{9}$$

$$R^x_{t,a_{\tau-1}} - x_{t,a-\tau d^b} = 0, \tag{10}$$

$$R^x_{ta_0} - x_{tad^\emptyset} = R_{ta_1}, \tag{11}$$

$$x_{ta_0 d} \le R_{ta^f} \quad \text{for } d \in \mathcal{D}^{\text{rep}}, \tag{12}$$

$$x_{tad} \ge 0 \quad \forall a, d. \tag{13}$$

Here, (9) enforces flow conservation given current inventories. Equation (10) defines the in-transit inventory in terms of the amount ordered, while (11) defines the new inventory as the sum of what was left over plus new arrivals. Equation (12) limits the number of decisions to replace transformers by the number of transformers that have failed, and (13) provides the necessary non-negative constraints.

The recursion in (8) can be solved approximately by using an approximation of the value function. The corresponding approximate problem is

$$\hat{V}_t^n\big(R_{t-1}^{x,n}, \hat{R}_t(\omega^n)\big) = \hat{V}_t^n\big(R_t^n\big) = \min_{x_t \in \mathcal{X}_t(R_t^n)} \big\{C(x_t) + \gamma \bar{V}_t^{n-1}\big(R_t^{x,n}\big)\big\} \tag{14}$$

Step 0. Initialize $\bar{V}_t^0(R_t^x)$ for $t \in \{0, 1, \dots, T\}$.
Step 1. Set $n = 1$ and $t = 0$.
Step 2. Solve problem (14) and transition function as in (1)–(4).
Step 3. Sample $\hat{R}_{t+1}(\omega^n)$ and transition function as in (5).
Step 4. If $t < T$, set $t = t + 1$, and go to Step 2.
Step 5. Update $\bar{V}_t^{n-1}(R_t^x)$ for $t \in \{0, 1, \dots, T - 1\}$.
Step 6. If $n < N$ set $n = n + 1$, $t = 0$, and got to Step 2.

Fig. 2 Sketch of approximate dynamic programming algorithm for the transformer acquisition and deployment model

where $n \in \{0, 1, \dots, N\}$ is the iteration counter. Note that this optimization problem does not violate non-anticipativity because in order to make the time t decisions x_t the formulation only uses the information that is indeed available at that time. The idea of ADP is to produce a good approximation $\bar{V}_t^N(R_t^x)$ which results in near optimal decisions. Figure 2 gives a sketch of the ADP approach.

Note that the algorithm steps forward in time. This avoids the enumeration of states and actions that is typical for classic backward dynamic programming. The use of simulation avoids the need to evaluate the expectation explicitly. In the next section we introduce functional forms for $\bar{V}_t(R_t^x)$ and show how this function can be estimated iteratively using Monte Carlo methods.

The next section describes a family of value function approximations for solving the dynamic program.

4 Value function approximations

We tested three different functional forms for $\bar{V}_t(R_t^x)$. The starting point is a separable piecewise linear function that has been used successfully in Topaloglu and Powell [23], Powell et al. [20], and Godfrey and Powell [8]. We will show that this function is inappropriate in our problem setting and explain why. The second functional form extends the first approach by using pre-decision state information. The third algorithm uses an explicit two-dimensional, piecewise linear value function approximation (VFA). We show how to estimate a two-dimensional value function surface.

4.1 Piecewise linear separable value function approximation

Topaloglu and Powell [23], Powell et al. [20], and Godfrey and Powell [8] all show the merits of a VFA that is separable in the elements of the resource vector R_t^x. Recalling that the set $\mathcal{A}^{\text{fail}}$ is the attributes of transformers that have already failed (these are the demands that we need to satisfy), we can approximate the value of all transformers (failed and replacements held in inventory) using a separable approximation of the form

$$\bar{V}_t(R_t^x) = \sum_{a \in \mathcal{A}^{\text{rep}}} \bar{V}_{ta}(R_{ta}^x) + \sum_{a \in \mathcal{A}^{\text{fail}}} \bar{V}_{ta}(R_{ta}^x). \qquad (15)$$

The first term on the right hand side is the value derived from replacement transformers and the second term is the value of deferring the replacement of a failed transformer. We follow the methods in Topaloglu and Powell [23] and use piecewise linear approximations for the value of each type of transformer.

Let us focus on replacement transformers first. Since there can be multiple replacement transformers ordered and in inventory, we choose piecewise linear functions with integer breakpoints to represent the value of such replacement transformers. These piecewise linear functions have to be convex to reflect the decreasing marginal value of additional resources of the same type.

Note that the purchase decision produces a newly purchased transformer $a_{\tau-1}$ and the holding decision is associated with transformers a_0. The corresponding value function components $\bar{V}_{ta_0}(R^x_{ta_0})$ and $\bar{V}_{ta_{\tau-1}}(R^x_{ta_{\tau-1}})$ are of particular importance because they are related to decisions. All the intermediate transformers are not subject to decision making. In fact we introduce an algorithm that only requires the estimation of $\bar{V}_{ta_0}(R^x_{ta_0})$ and $\bar{V}_{ta_{\tau-1}}(R^x_{ta_{\tau-1}})$.

We now turn to failed transformers, the second term on the right hand side of (15). There is future value associated with replacing a failure, namely avoiding the congestion costs that the failure would have caused in the future. The VFA component associated with replacing a failure is the expected future congestion cost savings. Figure 3 illustrates the subproblem at time t using piecewise linear VFA components for replacement transformers in the future, and linear VFA components for the value of replacing a failure. The nodes labeled $a_4, \dots, a_1$ represents transformers that are $4, \dots, 1$ time periods away from being used. The value of transformers at each of these nodes, and at the inventory node, is captured by a piecewise linear value function. In addition, there are linear functions capturing the value of covering a failed transformer which depends on the congestion cost.

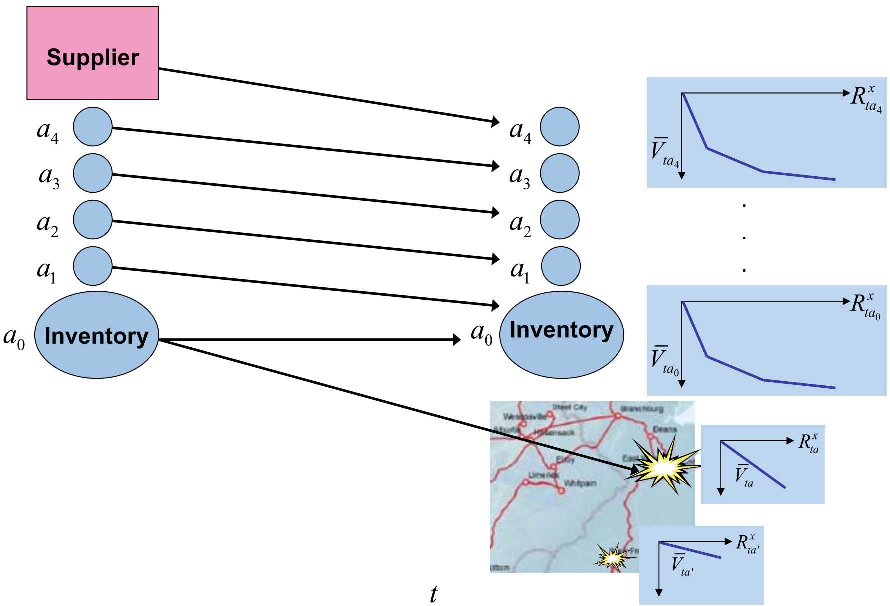


Fig. 3 Schematic representation of the piecewise-linear separable VFA for the transformer acquisition and deployment model

With the VFA described above problem (14) becomes an easily solvable linear program. Using the notation

$M =$ number of segments of a piecewise linear VFA component,
$y_{tam} =$ variable representing the m^{th} segment of the piecewise linear VFA $V_{ta}(R^x_{ta})$,
$y_t = (y_{tam})_{a\in\{a_0,a_{\tau-1}\},t\in\{0,\dots,T\}} =$ vector of variables representing piecewise linear VFAs,
$\bar{v}_{tam} =$ slope of the m^{th} segment of the piecewise-linear VFA.

The problem is to solve

$$\hat{V}_t(R_t) = \min_{x_t, y_t}\left\{C(x_t) + \gamma\left(\sum_{d\in\mathcal{D}^{\text{rep}}} \bar{v}_{taf}\, x_{ta_0d} + \sum_{a\in\{a_{\tau-1},a_0\}}\sum_{m=1}^{M} \bar{v}_{tam}\, y_{tam}\right)\right\} \tag{16}$$

subject to

$$x_{ta_0d^{\emptyset}} + \sum_{d\in\mathcal{D}^{\text{rep}}} x_{ta_0d} = R_{ta_0}, \tag{17}$$

$$R^x_{ta_{\tau-1}} - x_{tad^b} = 0, \tag{18}$$

$$R^x_{ta_0} - x_{tad^{\emptyset}} = R_{ta_1}, \tag{19}$$

$$\sum_{m=1}^{M} y_{tam} - R^x_{ta} = 0 \quad \text{for } a \in \{a_{\tau-1}, a_0\}, \tag{20}$$

$$x_{ta_0d} \le R_{taf} \quad \text{for } d \in \mathcal{D}^{\text{rep}}, \tag{21}$$

$$0 \le y_{tam} \le 1 \quad \text{for } a \in \{a_{\tau-1}, a_0\},\ m = 1, \dots, M, \tag{22}$$

$$x_{tad} \ge 0 \quad \forall a, d. \tag{23}$$

The minimization problem in (16) includes the one-period costs in $C(x_t)$, and the value function approximation that captures the impact of decisions on the future. These impacts include a linear approximation of the downstream cost of not replacing a failed transformer now, and a piecewise-linear approximation of the value of replacement transformers. Equation (17) is the flow conservation constraint for transformers that can be acted on now. Equation (18) defines the post-decision resource state for transformers that are in-transit, while (19) defines the post-decision state for transformers that are available now. Equation (20) defines the sum of the variables (y_{tam}), $m = 1, \dots, M$, used in the piecewise linear value function, to be the total number of transformers that are now available, while (21) is the demand constraint which limits our ability to replace failed transformers by the number of transformers that have failed. Equation (22) limits the y variables to be between 0 and 1, and (23) imposes non-negativity.

In our numerical work we will provide the data showing that a separable VFA is inappropriate for our problem class. In this section we focus on providing intuition why a separable approximation may not work. Figure 3 shows that the value contribution of a newly purchased spare $a_{\tau-1}$, $\bar{V}_{ta_{\tau-1}}(R^x_{ta_{\tau-1}})$, is independent of all the other replacement transformers, $R^x_{ta_0}, R^x_{ta_1}, \dots, R^x_{ta_{\tau-2}}$. This is implied by separability. This assumption is only true if there is no spare inventory holding over from one period to the next. In our problem replacement transformers will typically be held in inventory for some time because congestion costs are substantially higher than inventory holding costs. In this case a spare that is in inventory or in the pipeline at time t can easily still be around when a newly purchased spare arrives in inventory at time $t+\tau$. The marginal value of one more or one less transformer of type $a_{\tau-1}$ at time t may very well depend on how many transformers $R^x_{ta_0}, R^x_{ta_1}, \dots, R^x_{ta_{\tau-2}}$ are in the system. If there is an abundance of spares in inventory and in the pipeline at time t then the value of ordering one more spare is probably low. If transformer inventory is exhausted the value of ordering one more spare is probably higher. We need a VFA that captures this behavior while retaining the computational advantages of problem (16)–(23). The next section introduces such an approximation as well as a complete ADP algorithm.

4.2 Piecewise linear VFA with pre-decision state information

For computational purposes it is desirable to maintain the network LP structure of problem (16)–(23). To this end we define

$$\tilde{R}_t = \max\left(\sum_{s=0}^{\tau-1} R_{ta_s} - \sum_{a\in\mathcal{A}^{\text{fail}}} R_{ta}, 0\right). \tag{24}$$

In this equation, $\tilde{R}_t$ is capturing the total number of transformers that are either in transit or available to be assigned.

We have $\tilde{R}_t \approx \sum_{s=0}^{\tau-1} R^x_{ta_s}$ where the approximation is an equality unless the model is holding back available spare inventory from low-priority failures. This approximation allows us to replace the VFA component $\bar{V}_{ta_{\tau-1}}(R^x_{ta_{\tau-1}})$ with $\bar{V}_{ta_{\tau-1}}(R^x_{ta_{\tau-1}} + \tilde{R}_t)$ without destroying the network structure of problem (16)–(23). Making the change only requires calculating $\tilde{R}_t$ and adding it as a supply to the node representing $R^x_{ta_{\tau-1}}$. Figure 4 gives an example of the procedure. In problem (16)–(23) only (18) changes to

$$R^x_{ta_{\tau-1}} - x_{ta_\tau d^b} = \tilde{R}_t. \tag{25}$$

With the optimization problem solved we turn to the issue of estimating the parameters $\bar{v}_{ta}$, the slopes of $\bar{V}_t(R^x_t)$. We simplify this task by setting the linear value function components of the failures to a fixed multiple of the one-period congestion costs in the various locations. We denote

$$\bar{v}_{ta^f} = \rho^f c_{ta^f d} \quad \forall d \in \mathcal{D}^{\text{rep}} \tag{26}$$

where

$\rho^f =$ tunable parameter that penalizes the value of not satisfying a failure at a particular point in time.

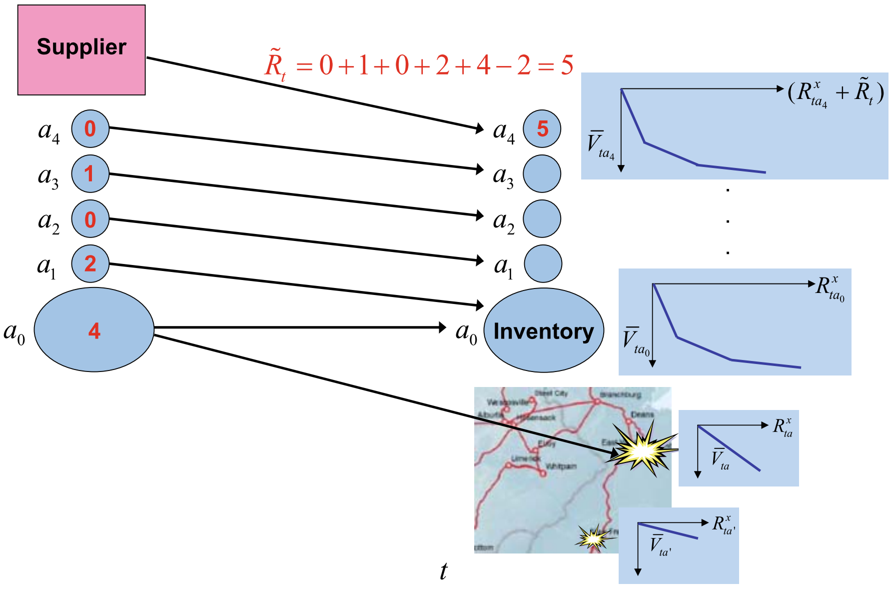


Fig. 4 Schematic representation of the piecewise-linear VFA with pre-decision state information for the transformer acquisition and deployment model

Fixing the linear VFA components that capture the value of pushing a failure to the future allows us to focus on the piecewise-linear VFA components representing the value of replacement transformers.

The estimation of the piecewise-linear value functions is done in an iterative procedure where each iteration consists of stepping forward through the time horizon, simulating a sample path, and solving the following approximate problem. Using n to denote the iteration index the approximate problem (16)–(23) becomes:

$$\hat{V}_t^n(R_t^n) = \min_{\{x_t,y_t\}\in\mathcal{X}_t''} \left\{ C(x_t) + \gamma \left(\sum_{d\in\mathcal{D}^{\mathrm{rep}}} \bar{v}_{taf}\, x_{ta_0d} + \sum_{a\in\{a_\tau,a_0\}} \sum_{m=0}^{M} \bar{v}^n_{tam}\, y_{tam} \right) \right\} \quad (27)$$

where $\mathcal{X}_t''$ is the feasible region defined by (17), (25), and (19)–(23). By solving perturbations of (27) with respect to R_t^n we can obtain stochastic sample gradients that represent a sample realization of the value of one more or one less replacement transformer. Let e_a be a vector of zeros and a one at element a. We define the stochastic right gradient as

$$\hat{v}^{+,n}_{ta_0} = \hat{V}_t^n(R_t^n + e_{a_0}) - \hat{V}_t^n(R_t^n), \quad (28)$$

and the left stochastic gradient as

$$\hat{v}^{-,n}_{ta_0} = \hat{V}_t^n(R_t^n) - \hat{V}_t^n(R_t^n - e_{a_0}) \quad \text{if } R^n_{ta_0} > 0. \quad (29)$$

Note that we only calculate stochastic gradients for replacement transformers in inventory (attribute vector a_0). This is because these transformers are the only source of (additional) rewards. In that sense the replacement transformers in inventory drive the system. The stochastic gradients are sample realizations of value function slopes and need to be smoothed into the current estimates of the slopes in order to "average" across sample paths. Which slope estimates should be updated with $\hat{v}^{+,n}_{ta_0}$ and $\hat{v}^{-,n}_{ta_0}$? Since a stochastic gradient is the value of one more or one less transformer in inventory at time t, the gradients need to update the value functions that lead to one more or one less transformer in inventory at time t.

To obtain one more or one less transformer in inventory at time t the model would have to hold one more transformer in inventory at time $t-1$ or order one more transformer at time $t-\tau$. Therefore $\hat{v}^{+,n}_{ta_0}$ and $\hat{v}^{-,n}_{ta_0}$ update $\bar{V}^{n-1}_{t-1,a_0}$ and $\bar{V}^{n-1}_{t-\tau,a_{\tau-1}}$. The stochastic gradients at time t are passed backwards in time to the VFAs that provide transformer spares for time t. $\bar{V}^{n-1}_{t-1,a_0}$ and $\bar{V}^{n-1}_{t-\tau,a_{\tau-1}}$ share the same role as the mechanism for providing spares for time t but they are not the same functions. The difference lies in what segments of the VFAs components are updated by the stochastic gradients.

Updating the VFA components $\bar{V}^{n-1}_{t-\tau,a_{\tau-1}}$ and $\bar{V}^{n-1}_{t-1,a_0}$ is a two-step procedure. We pick $\bar{V}^{n-1}_{t-\tau,a_{\tau-1}}$ to illustrate the steps. In the first step the stochastic gradients are smoothed into the current value function estimate in the following way to obtain an intermediate vector of slopes $(u^n_{t-\tau,a_{\tau-1},m})_{m=0,1,\ldots,M}$. The calculation is

$$u^n_{t-\tau,a_{\tau-1},m} = \begin{cases} (1-\alpha_{n-1})\bar{v}^{n-1}_{t-\tau,a_{\tau-1},m} + \alpha_{n-1}\hat{v}^{+,n}_{ta_0} & \text{if } m = R^{x,n}_{t-\tau,a_{\tau-1}} + \tilde{R}_t, \\ (1-\alpha_{n-1})\bar{v}^{n-1}_{t-\tau,a_{\tau-1},m} + \alpha_{n-1}\hat{v}^{-,n}_{ta_0} & \text{if } m = R^{x,n}_{t-\tau,a_{\tau-1}} + \tilde{R}_t - 1, \\ \bar{v}^{n-1}_{t-\tau,a_{\tau-1}} & \text{otherwise,} \end{cases} \tag{30}$$

where α_{n-1} is a step size between 0 and 1. Equation (30) updates the slopes of our piecewise linear approximation using the right ($\hat{v}^{+,n}_{ta_0}$) and left ($\hat{v}^{+,n}_{ta_0}$) derivatives around the current number of transformers. At this stage, we only update the slopes adjacent to the number of transformers currently in a particular state.

Unfortunately, this computation can cause convexity violations. To restore convexity we apply the convexity restoring procedure of Godfrey and Powell [9] to the intermediate vector of slopes $(u^n_{t-\tau,a_{\tau-1},m})_{m=0,1,\ldots,M}$. The updated value function slopes are obtained as

$$\bar{V}^n_{t-\tau,a_{\tau-1}} = \left(\bar{v}^n_{t-\tau,a_{\tau-1},m}\right)_{m=0,1,\ldots,M} = \Pi\left(u^n_{t-\tau,a_{\tau-1},m}\right)_{m=0,1,\ldots,M} \tag{31}$$

where Π is the convexity restoring operator defined by the equations

$$\min_{\bar{v}^n_{t-\tau,a_{\tau-1}}} \left\| \bar{v}^n_{t-\tau,a_{\tau-1}} - \bar{u}^n_{t-\tau,a_{\tau-1}} \right\|^2 \tag{32}$$

subject to:

$$\bar{v}^n_{t-\tau,a_{\tau-1},m+1} - \bar{v}^n_{t-\tau,a_{\tau-1},m} \geq 0 \quad \text{for } m = 0,\ldots,M-1. \tag{33}$$

The idea of ADP is that after a finite number of iterations, N, the VFA $\bar{V}^N_t$ is a good approximation of the true value function for every t and that the collection of all value functions across time implies a policy that is close to optimal.

Step 0. Initialize $\bar{V}_t^0$ for $t = 1, \ldots, T$ and set $n = 1$.
Step 1. Do for $t = 0, \ldots, T-1$:
Step 1a. Calculate $\tilde{R}_t^n$ as in (24) and solve time t subproblem (27). Store $\tilde{R}_t^n$ and $R_t^{x,n}$ for later use.
Step 1b. Calculate sample gradients as in (28) and (29) and store them for later use.
Step 1c. Sample $\hat{R}_{t+1}^n(\omega_{t+1}^n)$ and calculate R_{t+1}^n as in (5).
Step 2a. For $t = \tau, \ldots, T-1$ calculate $\bar{V}_{t-\tau,a_{\tau-1}}^n(R_{t-\tau,a_{\tau-1}}^{x,n} + \tilde{R}_{t-\tau t}^n) = (\bar{v}_{t-\tau,a_{\tau-1},m}^n)_{m=0,1,\ldots,M}$ using (30) and (31).
Step 2b. For $t = 1, \ldots, T-1$ calculate $\bar{V}_{t-1,a_0}^n(R_{ta_0}^{x,n}) = (\bar{v}_{t-1,a_0,m}^n)_{m=0,1,\ldots,M}$ using

$$u_{t-1,a_0,m}^n = \begin{cases} (1-\alpha_{n-1})\bar{v}_{t-1,a_0,m}^{n-1} + \alpha_{n-1}\hat{v}_{ta_0}^{+,n} & \text{if } m = R_{t-1,a_0}^{x,n}, \\ (1-\alpha_{n-1})\bar{v}_{t-1,a_0,m}^{n-1} + \alpha_{n-1}\hat{v}_{ta_0}^{-,n} & \text{if } m = R_{t-1,a_0}^{x,n} - 1, \\ \bar{v}_{t-1,a_0}^{n-1} & \text{otherwise} \end{cases} \tag{34}$$

and

$$(\bar{v}_{t-1,a_0,m}^n)_{m=0,1,\ldots,M} = \Pi(u_{t-1,a_0,m}^n)_{m=0,1,\ldots,M}. \tag{35}$$

Step 3. Set $n = n+1$. If $n < N$, go to Step 1.

Fig. 5 ADP training algorithm for the replacement transformer acquisition and deployment model

Step 0. Set $n = 1$.
Step 1. Do for $t = 0, \ldots, T-1$:
Step 1a. Calculate $\tilde{R}_t^n$ as in (24) and solve time t subproblem (27).
Step 1b. Sample $\hat{R}_{t+1}^n(\omega_{t+1}^n)$ and calculate R_{t+1}^n as in (5).
Step 2. Set $n = n+1$. If $n < N$, go to Step 1.

Fig. 6 Evaluation algorithm for the replacement transformer acquisition and deployment model

Figure 5 gives the complete ADP algorithm for estimating the VFA. Equations (34) and (35) are analogous to (30) and (31) for the inventory value functions $\bar{V}_{ta_0}$. Once the VFA estimation is accomplished running the model involves executing a simplified version of Fig. 5. We call this the evaluation algorithm because this algorithm is used to evaluate policies based on a number of sample paths. Figure 6 specifies the steps of the evaluation algorithm with fixed VFAs.

4.3 Two-dimensional piecewise linear VFA

Given the non-separable nature of the value function $V(R_t^x)$ the thought of a multi-dimensional non-separable VFA is attractive. In this section we explore this idea for the two-dimensional case. The two-dimensional case is particularly interesting in our problem setting because it allows us to capture the joint future value of acquisition and holding decisions. We present an algorithm to estimate a VFA surface, $\bar{V}_t(R_{ta_0}^x, R_{ta_1}^x)$. Such a VFA would allow us to express the true relationship between resource states and their value if we restrict the problem to two dimensions, which in our application means we are restricting the problem to a two-period lead time.

To minimize notational clutter let us denote $R^x_{ta_0}$ by y_1 and $R^x_{ta_1}$ by y_2. Given $M_i \in \mathbb{N}_0$ for $i = 1, 2$, let $\mathcal{I} = \{0, 1, \ldots, M_1\}$, $\mathcal{J} = \{0, 1, \ldots, M_2\}$ and define

$$\mathcal{Y}^1_k = \{y = (y_1, y_2) | y_1 = [0, M_1], y_2 = k\} \quad \forall k \in \mathcal{J},$$
$$\mathcal{Y}^2_l = \{y = (y_1, y_2) | y_1 = l, y_2 = [0, M_2]\} \quad \forall l \in \mathcal{I},$$

to be the horizontal and vertical grid lines in the rectangle $(0, 0)$, $(0, M_2)$, (M_1, M_2), $(M_1, 0)$. Let

$$\mathcal{G} = \bigcup_{k=0}^{M_2} \mathcal{Y}^1_k \cup \bigcup_{l=0}^{M_1} \mathcal{Y}^2_l,$$

be the grid itself and

$$\mathcal{V} = [0, 1, \ldots, M_1] \times [0, 1, \ldots, M_2],$$

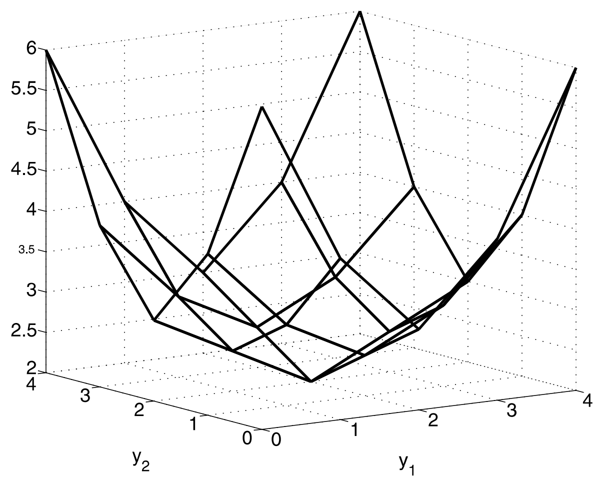
the set of vertices in $\mathcal{G}$. We define $\bar{V}_t : \mathcal{G} \to \mathbb{R}$. We assume $\bar{V}_t$ is submodular on $\mathcal{V}$ and piecewise-linear convex with integer breakpoints along each grid line. Submodularity means that the slopes in the y_2 direction are increasing along the y_1 dimension and the slopes in the y_1 direction are increasing along the y_2 dimension. Formally, the definition of submodularity is

$$\bar{V}_t(y_1, y_2 + 1) - \bar{V}_t(y_1, y_2) \le \bar{V}_t(y_1 + 1, y_2 + 1) - \bar{V}_t(y_1 + 1, y_2).$$

Submodularity can be thought of as a form of convexity for discrete functions. For a detailed discussion of submodularity, see [16].

Figure 7 gives an example of the function $\bar{V}_t$. As can be seen, we are taking advantage of the natural convexity (expressed as submodularity) of this function. It captures the interaction between acquisition and holding decisions in a non-separable function, allowing us to test the errors introduced using a separable approximation.

Fig. 7 Example of function $\bar{V}_t$



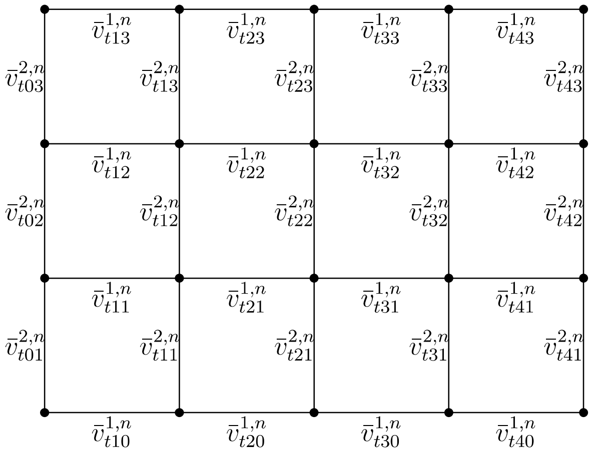


Fig. 8 Illustration of $\bar{V}_t^n$ in terms of its slopes on the *grid lines*

We show how to solve problem (14) using the VFA defined here. To simplify notation let $\bar{V}_{tij}^n = \bar{V}_t^n(i, j)$. Define

$$\bar{v}_{tij}^{1,n} = \bar{V}_{tij}^n - \bar{V}_{t,i-1,j}^n, \quad i \in \mathcal{I} \setminus 0, j \in \mathcal{J},$$
$$\bar{v}_{tij}^{2,n} = \bar{V}_{tij}^n - \bar{V}_{t,i,j-1}^n, \quad i \in \mathcal{I}, j \in \mathcal{J} \setminus 0,$$

the slopes of $\bar{V}_t^n$ along the grid lines. Figure 8 illustrates $\bar{V}_t^n$ in terms of its slopes.

The problem is

$$\hat{V}_t^n\left(R_{ta_0}^n, R_{ta_1}^n\right) = \min_{x_t, y_t}\left\{C(x_t) + \gamma\left(\bar{V}_{t00} + \bar{V}_t^1\left(x_t^{\text{rep}}\right) + \bar{V}_t^2(y_1, y_2)\right)\right\}, \tag{36}$$

with

$$\bar{V}_t^1 = \sum_{d \in \mathcal{D}^{\text{rep}}} \bar{v}_{ta_0 d} x_{ta_0 d}, \tag{37}$$

$$\bar{V}_t^2 = \sum_{i=1}^{M_1} \bar{v}_{ti0}^{1,n-1} y_{t1i} + \sum_{j=1}^{M_2} \bar{v}_{t0j}^{2,n-1} y_{t2j} + \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left(\bar{v}_{tij}^{2,n-1} - \bar{v}_{t,i-1,j}^{2,n-1}\right) y_{t1i} y_{t2j}, \tag{38}$$

subject to

$$x_{ta_0 d^{\emptyset}} + \sum_{d \in \mathcal{D}^{\text{rep}}} x_{ta_0 d} = R_{ta_0}^n, \tag{39}$$

$$y_2 - x_{ta_\tau d^b} = 0, \tag{40}$$

$$y_1 - x_{ta_0 d^{\emptyset}} = R_{ta_1}^n, \tag{41}$$

$$x_{ta_0 d} \le R_{ta^f}^n \quad \text{for } d \in \mathcal{D}^{\text{rep}}, \tag{42}$$

$$\sum_{i \in \mathcal{I}} y_{1i} - y_1 = 0, \tag{43}$$

$$\sum_{j\in\mathcal{J}} y_{2j} - y_2 = 0, \tag{44}$$

$$x_{tad} \geq 0 \quad \forall a \in \mathcal{A}, d \in \mathcal{D} \tag{45}$$

$$y_{1i}, y_{2j} \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{J}. \tag{46}$$

Equation (36) includes the one-period cost plus the value function approximation given by (37)–(38). Equations (39)–(46) closely follow the constraints (17)–(23).

The quadratic mixed binary program (36)–(46) can be transformed to a linear mixed binary program. We define $z_{tij} = y_{t1i}y_{t2j}$, rewrite (38) as

$$\bar{V}_t^2 = \sum_{i=1}^{M_1} \bar{v}_{ti0}^{1,n-1} y_{t1i} + \sum_{j=1}^{M_2} \bar{v}_{t0j}^{2,n-1} y_{t2j} + \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} \left(\bar{v}_{tij}^{2,n-1} - \bar{v}_{t,i-1,j}^{2,n-1}\right) z_{tij}, \tag{47}$$

and add the following constraints to the feasible region defined by (39)–(46)

$$y_{t1i} + y_{t2j} - z_{tij} \leq 1 \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{48}$$

$$z_{tij} - y_{t1i} \leq 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{49}$$

$$z_{tij} - y_{t2j} \leq 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{J}. \tag{50}$$

The procedure to estimate the slopes of the two-dimensional VFA is based on an iterative procedure similar to the one described above. Solving (36)–(46) repeatedly for perturbations of the resource state vector we obtain

$$\hat{V}_t^n\left(R_{ta_0}^n + 1, R_{ta_1}^n\right) \quad \text{if } R_{ta_0}^n < M_1, \tag{51}$$

$$\hat{V}_t^n\left(R_{ta_0}^n - 1, R_{ta_1}^n\right) \quad \text{if } R_{ta_0}^n > 0, \tag{52}$$

$$\hat{V}_t^n\left(R_{ta_0}^n, R_{ta_1}^n + 1\right) \quad \text{if } R_{ta_1}^n < M_2, \tag{53}$$

$$\hat{V}_t^n\left(R_{ta_0}^n, R_{ta_1}^n - 1\right) \quad \text{if } R_{ta_1}^n > 0, \tag{54}$$

$$\hat{V}_t^n\left(R_{ta_0}^n + 1, R_{ta_1}^n + 1\right) \quad \text{if } R_{ta_0}^n < M_1, R_{ta_1}^n < M_2, \tag{55}$$

$$\hat{V}_t^n\left(R_{ta_0}^n + 1, R_{ta_1}^n - 1\right) \quad \text{if } R_{ta_0}^n < M_1, R_{ta_1}^n > 0, \tag{56}$$

$$\hat{V}_t^n\left(R_{ta_0}^n - 1, R_{ta_1}^n - 1\right) \quad \text{if } R_{ta_0}^n > 0, R_{ta_1}^n > 0, \tag{57}$$

$$\hat{V}_t^n\left(R_{ta_0}^n - 1, R_{ta_1}^n + 1\right) \quad \text{if } R_{ta_0}^n > 0, R_{ta_1}^n < M_2. \tag{58}$$

Each sample observation is smoothed into the current estimate of $\bar{V}_{t-1}^{n-1}(R_{t-1,a_0}^x, R_{t-1,a_1}^x)$ in the following way:

$$\bar{Z}_{t-1}^n\left(R_{t-1,a_0}^{x,n}, R_{t-1,a_1}^{x,n}\right) = (1-\alpha_n)\bar{V}_{t-1}^{n-1}\left(R_{t-1,a_0}^{x,n}, R_{t-1,a_1}^{x,n}\right) + \alpha_n \hat{V}_t^n\left(R_{t,a_0}^n, R_{t,a_1}^n\right), \tag{59}$$

where α_n is a step size between 0 and 1. This updating step can introduce violations of submodularity and convexity along grid lines. The function $\bar{Z}_{t-1}$ is an intermediate

Step 0. Initialize $\bar{V}_t^0$ for $t = 1, \ldots, T$ and set $n = 1$.
Step 1. Do for $t = 0, \ldots, T - 1$:
Step 1a. Solve time t subproblem (36), (37), (47) subject to the constraints (39)–(46) and (48)–(50).
Step 1b. Calculate perturbations as in (51)–(54).
Step 2a. Use $\hat{V}_t^n(R_t^n)$ from Step 1a and perturbations from Step 1b for intermediate updates as in (59).
Step 2b. Solve (60)–(64)
Step 3. Set $n = n + 1$. If $n < N$, go to Step 1.

Fig. 9 ADP training algorithm for the replacement transformer acquisition and deployment model using a two-dimensional VFA

function that may no longer be convex along grid lines or submodular. We restore these properties using the following projection procedure. We use the notation: $\bar{V}_{tij}^n = \bar{V}_t^n(i, j)$, $\bar{Z}_{tij}^n = \bar{Z}_t^n(i, j)$, and $\bar{V}_t^n = (\bar{V}_{tij}^n)_{i \in \mathcal{I}, j \in \mathcal{J}}$. The problem is

$$\min_{\bar{V}_t^n} = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \left(\bar{V}_{tij}^n - \bar{Z}_{tij}^n\right)^2 \tag{60}$$

subject to

$$\left(\bar{V}_{tij}^n - \bar{V}_{t,i-1,j}^n\right) - \left(\bar{V}_{t,i+1,j}^n - \bar{V}_{tij}^n\right) \leq 0 \quad \forall i \in \mathcal{I} \setminus \{0, M_1\}, j \in \mathcal{J}, \tag{61}$$

$$\left(\bar{V}_{t,i,j+1}^n - \bar{V}_{tij}^n\right) - \left(\bar{V}_{tij}^n - \bar{V}_{t,i,j-1}^n\right) \leq 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{J} \setminus \{0, M_2\}, \tag{62}$$

$$\left(\bar{V}_{t,i,j+1}^n - \bar{V}_{tij}^n\right) - \left(\bar{V}_{t,i+1,j+1}^n - \bar{V}_{t,i+1,j}^n\right) \leq 0 \quad \forall i \in \mathcal{I} \setminus M_1, j \in \mathcal{J} \setminus M_2, \tag{63}$$

$$\left(\bar{V}_{t,i+1,j}^n - \bar{V}_{tij}^n\right) - \left(\bar{V}_{t,i+1,j+1}^n - \bar{V}_{t,i,j+1}^n\right) \leq 0 \quad \forall i \in \mathcal{I} \setminus M_1, j \in \mathcal{J} \setminus M_2. \tag{64}$$

Constraints (61)–(62) enforce convexity along grid lines and constraints (63)–(64) enforce submodularity.

Figure 9 gives the ADP algorithm for the two-dimensional non-separable VFA.

5 Numerical studies

The numerical work of this section includes two parts. First, we assess the solution quality of the ADP algorithms by comparing the resulting policy against the optimal policy for simplified problems. Second, we analyze a real-world electric power network with our best-performing algorithm. The numerical work was conducted using the data from a major grid operator, although simplifications were introduced for some of the algorithmic testing.

5.1 Solution quality assessment

In this subsection we make simplifying assumptions which allows us to compute the optimal policy of the transformer acquisition and deployment model using backward dynamic programming. We make the following simplifying assumptions:

1. All substations have the same congestion costs. That means there is a single class of failures.

Table 1 Characteristics of test data sets

Data set	Lead time (periods)	Expected failures	Holding cost ($ million)	Discount factor	Initial inventory	Max. inventory	Max. order
A1	2	2	0.5	0.9	6	6	4
A2	4	2	0.5	0.9	11	11	3
A3	6	2	0.5	0.9	12	12	3
A4	8	2	0.5	0.9	12	12	3
B1	2	4	2	0.9	10	10	6
B2	4	4	2	0.9	14	14	5
B3	6	4	2	0.9	15	15	5
B4	8	4	2	0.9	32	–	–
C1	2	1	0.1	0.098465	2	5	3
C2	4	1	0.1	0.098465	4	8	3
C3	6	1	0.1	0.098465	6	11	3
C4	8	1	0.1	0.098465	8	12	3
D1	2	4	0.1	0.098465	12	14	8
D2	4	4	0.1	0.098465	18	20	7
D3	6	4	0.1	0.098465	24	–	–
D4	8	4	0.1	0.098465	33	–	–
E1	2	2	0.1	0.098465	6	8	5
E2	4	2	0.1	0.098465	12	14	4
E3	6	2	0.1	0.098465	16	16	4
E4	8	2	0.1	0.098465	19	19	3

2. No backlogging. Failures that are not immediately met are lost to the system.
3. The number of failures per period is a binomial random variable with fixed parameters.

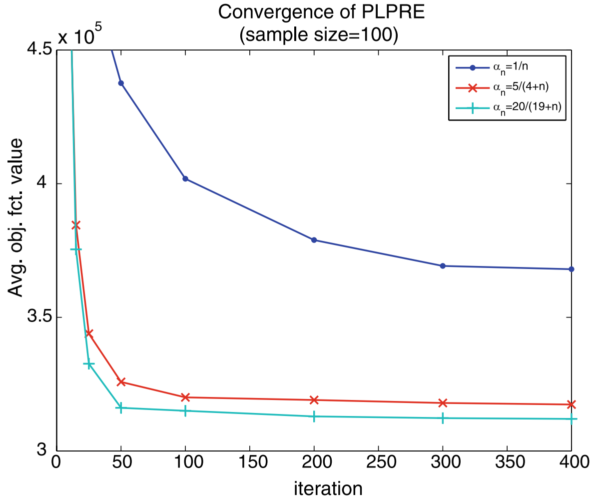
In our test data sets we hold the following parameters constant: the number of time periods is 40, the failure probability of a single transformer is 0.2, the purchase cost is $5 million, and the congestion cost parameter is $20 million. Note that the congestion cost parameter includes current period congestion costs ($5 million) and expected future congestion costs ($15 million, $\rho^f = 3$). The test data sets have varying lead times, holding costs, discount factors, and sizes of the transformer population. For each data set we specify an initial transformer inventory which in most cases is enough to cover failures until new purchases become available. A maximum inventory and the maximum order size are chosen such that the state space is minimized without cutting off an optimal solution. Table 1 summarizes the different test data sets.

The data sets are chosen such that it is feasible in most cases to calculate the optimal policy using backward dynamic programming (BDP). We use this optimal

Step 0. Initialize state space $\mathcal{R}$ and value function $V_T(R_T^x)$ for all $R_T^x \in \mathcal{R}$.
Step 1. Do for $t = T, \ldots, 1$:
Step 2. Do for each $R_{t-1}^x \in \mathcal{R}$:
Step 3. Compute $V_{t-1}(R_{t-1}^x) = \mathbb{E}[\min_{x_t}\{C(x_t) + \gamma V_t(R_t^x)\}]$.
Step 4. Next R_{t-1}^x.
Step 5. Next t.

Fig. 10 BDP training algorithm for the replacement transformer acquisition and deployment model

Fig. 11 Evaluation of three step size rules



solution to compare against ADP based on the following three VFAs: piecewise linear separable (PLS) of Sect. 4.1, piecewise linear with pre-decision state information (PLPRE) of Sect. 4.2, and two-dimensional (2D) of Sect. 4.3.

A challenge of the BDP algorithm is to make sure that the underlying model exactly matches the model in (8). This is best achieved by using a BDP that uses a value function around the post-decision state variable. Figure 10 gives the BDP algorithm that we use.

In order to determine the number of iterations, N, and the step size rule, α_n, for the PLS and PLPRE algorithms we performed the following analysis. Choosing the step size rule $\alpha_n = \frac{1}{n}$ we ran the PLPRE algorithm on data set E2 varying the number of iterations in the set $\{5, 15, 25, 50, 100, 200, 300, 400\}$. The resulting VFAs were evaluated based on 100 sample paths. We repeated the analysis for the step size rules $\alpha_n = \frac{5}{4+n}$, $\alpha_n = \frac{20}{19+n}$. Figure 11 summarizes the results of this analysis. Based on these results we chose the step size rule $\alpha_n = \frac{20}{19+n}$ and $N = 200$. Figure 11 indicates that the PLPRE algorithm converges nicely as the number of training iterations increases. We used this approximation as our test bed for determining the best stepsize rule, which is influenced primarily by the overall level of uncertainty and how long a transformer is held in inventory.

Table 2 Numerical results comparing PLPRE and PLS with optimal (BDP). "–" indicate that the BDP did not complete due to memory constraints (2 GB). All experiments are run on a single processor (Intel P4), 3.06 GHz machine

Data set	BDP time (s)	PLPRE r_{BDP}^{PLPRE} (%)	95% CI (%)	Time (s)	PLS r_{BDP}^{PLS} (%)	95% CI (%)	r_{PLPRE}^{PLS} (%)	Time (s)
A1	0	−0.1	[−0.7, 0.6]	595	9.9	[7.4, 12.4]	10.0	565
A2	1	1.5	[0.9, 2.1]	580	5.7	[3.7, 7.6]	4.1	597
A3	22	4.1	[3.3, 4.9]	598	4.8	[3.1, 6.6]	0.7	575
A4	285	3.4	[2.8, 4.1]	541	2.4	[1.5, 3.2]	−1.0	581
B1	0	0.7	[0.3, 1.2]	604	8.0	[6.2, 9.9]	7.2	559
B2	10	2.9	[2.2, 3.6]	793	1.7	[1.0, 2.5]	−1.1	472
B3	504	3.2	[2.5, 3.8]	816	1.0	[0.4, 1.5]	−2.1	372
B4	–	–	–	778	–	–	−2.4	386
C1	0	−0.2	[−0.5, 0.1]	775	18.3	[15.2, 21.4]	18.6	354
C2	1	0.9	[−0.1, 1.8]	788	11.7	[9.2, 14.3]	10.8	360
C3	13	1.9	[0.6, 3.1]	828	10.3	[7.7, 12.8]	8.2	375
C4	146	3.0	[1.5, 4.4]	869	7.7	[5.4, 10.1]	4.6	376
D1	0	1.3	[0.8, 1.7]	873	15.8	[13.1, 18.5]	14.4	370
D2	54	2.8	[2.0, 3.7]	906	12.5	[10.0, 15.0]	9.4	371
D3	–	–	–	852	–	–	3.7	362
D4	–	–	–	762	–	–	4.7	375
E1	0	1.1	[0.5, 1.6]	680	16.2	[13.5, 19.0]	15.0	356
E2	2	0.7	[0.2, 1.1]	696	12.6	[10.0, 15.3]	11.9	361
E3	102	0.7	[0.2, 1.1]	709	11.5	[8.9, 14.1]	10.8	360
E4	454	1.6	[0.7, 2.6]	719	6.3	[4.4, 8.1]	4.5	368

To compare the different ADP algorithms against optimal we use the following performance metric

$$r_j^i = \frac{\text{Average cost of algorithm } i \text{ policy}}{\text{Average cost of algorithm } j \text{ policy}} - 1.$$

Averages are based on 100 sample paths that do not change between algorithms. Table 2 shows the computational results comparing the PLS, PLPRE and the optimal BDP algorithm. We see that PLS, which assumes separability, does not work well in almost all cases. The few cases where PLS performs better than or comparable to PLPRE (A3, A4, B2, B3) are characterized by longer lead times and high holding costs. PLPRE performs consistently well and both algorithms have very reasonable run times. We also see that BDP does not scale well: some of the small problems used in this section were too big to be solved to optimality.

The two-dimensional algorithm (2D) can only be applied without modifications to problems with two-period lead times and we restrict our analysis to these cases. We

Table 3 Size of the VFA and computational results for the 2D algorithm

Data set	$r_{\text{BDP}}^{\text{2D}}$ (%) 100 iter.	500 iter.	1000 iter.	2000 iter.	4000 iter.	Time (s) 100 iter.	500 iter.	1000 iter.	2000 iter.	4000 iter.
A1	46.1	3.1	2.7	3.6	2.5	367	2415	5673	12,875	28,581
B1	70.6	4.2	2.0	1.2	1.5	417	2891	7671	19,789	29,393
C1	77.8	3.5	3.6	3.9	3.0	316	1961	4064	8,817	15,610
D1	218.8	109.5	20.7	4.4	1.5	572	3510	9142	28,277	80,876
E1	154.7	11.7	2.7	1.7	1.8	348	2376	5512	14,155	35,018

run the 2D algorithm for 100, 500, 1000, 2000, and 4000 iterations which gives us an idea of the convergence behavior for different data sets. Generally, convergence is slower than for PLPRE which prompts the need for larger step sizes. The step size of choice is $\alpha_n = \sqrt{1 - (\frac{n}{N})^2}$ which declines slowly for most of the iterations and has the added benefit that it does not have a parameter requiring fine tuning.

Table 3 shows the results for the 2D algorithm. We see that the 2D algorithm produces good policies but shows slower convergence behavior than PLPRE and much higher run times. What drives the run times aside from the higher number of iterations is the choice of M_1 and M_2 which determines the size of the linear mixed integer program. The number of binary variables ranges from 77 for the smallest data set $C1$ to 230 for the biggest problem $D1$. Also note that the mixed integer program is solved up to nine times per iteration and time period in order to obtain all $\hat{V}_t^n$.

Since the PLPRE outperforms 2D we make PLPRE our algorithm of choice for the policy study in the following section.

5.2 Transformer acquisition and deployment policies for PJM's network

We use the ADP algorithm to analyze the high-voltage transformer population of PJM Interconnection, a major regional transmission organization. PJM controls the electric power grid in a large area of the Eastern U.S. PJM's transmission system includes 182 500–230 kV transformers in 42 substations. These are the transformers of interest for this study. In each period the outcome of a Bernoulli experiment indicates whether a particular working transformer has failed or not. The failure probabilities for the experiment are obtained from hazard rate curves that give the failure probability as a function of the transformer age. These curves are estimated by PJM based on historical failure data and certain assumptions on the shape of the curve (see [4]). There is one hazard rate curve for every transformer condition ("good," "average," "watch"). Figure 12 gives a general idea of the shape of the hazard rate curves. The condition attribute transitions randomly following a discrete Markov chain. Failures are correlated in the sense that a bank can only have one failed transformer at a time.

We analyze the system over a 50-year horizon using 3-month time periods. The lead time is 6 time periods (1.5 years). The transformer purchase cost is \$5 million. The congestion costs vary greatly across substations, ranging from less than

Fig. 12 Examples of hazard rate curves used for failure generation

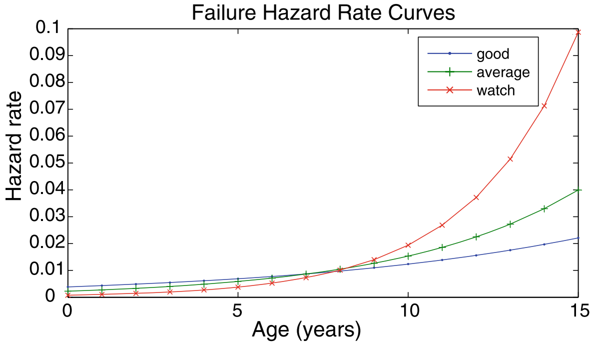
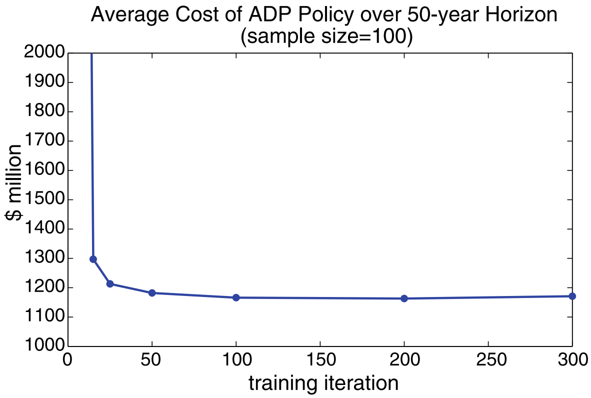


Fig. 13 Average cost of the ADP policy over the entire 50-year horizon as a function of the number of training iterations



$0.25 million to more than $50 million per period. Congestion costs, which are carefully computed by PJM, capture the increase in power generation costs when PJM has to use energy from a more expensive utility in order to avoid bottlenecks in the grid created by transformer failures. Congestion costs are computed as the incremental cost from a single transformer failure. Of course, if there are multiple transformer failures, interactions in the network imply that the cost of two failures may be greater than the sum of the congestion costs for individual failures. However, failures are rare, which means that using constant congestion costs is a reasonable first-order approximation. For our experiments, we used a congestion cost multiplier $\rho^f = 3$ and an inventory holding cost of $75,000 per period (1.5 percent of purchase costs). All costs grow over time at a rate of 1.5 percent per time period to reflect inflation.

We evaluate the ADP policy over 100 sample paths. The algorithmic parameters that need to be chosen are the number of training iterations, N, and the step size rule α_n. The step size rule of choice is $\alpha_n = \frac{20}{19+n}$ based on the analysis in the previous subsection. In order to determine the appropriate number of iterations we evaluate the policy on 100 sample paths after 5, 15, 25, 50, 100, 200, and 300 training iterations. Figure 13 shows no improvement after iteration 100. We therefore choose $N = 100$.

Figure 14 shows the average number of failures per year that result from following the ADP policy. We observe the pronounced "bubble" reflecting the failure of the older transformer cohorts that are currently in operation. The expected number of

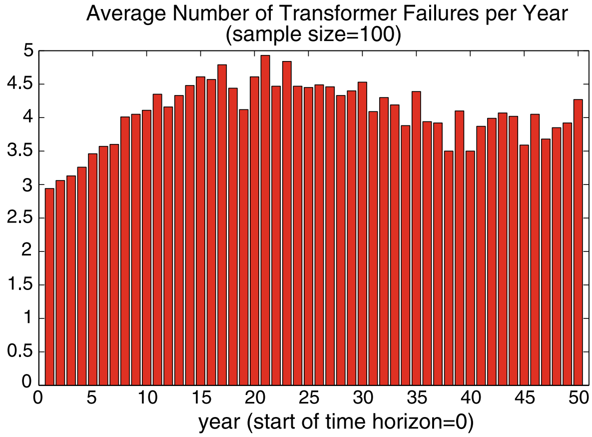


Fig. 14 Average number of transformer failures per year over 50-year time horizon

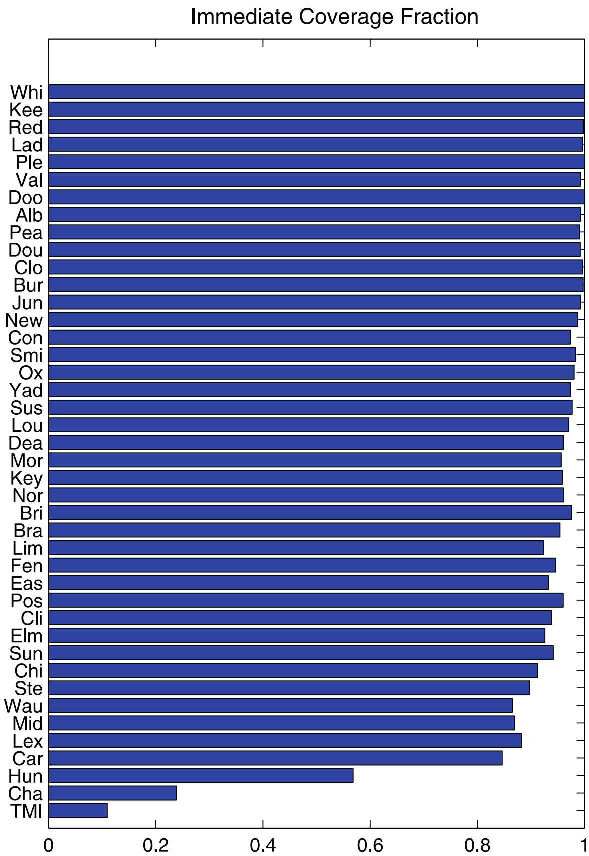


Fig. 15 Fraction of substation transformer failures that are replaced without delay

failures increases by 50 percent over the next 15 years, then remains high for another 15 years and declines afterwards. Figure 15 lists all 42 substations in descending order of congestion costs. The figure shows what fraction of the failures in each

Table 4 Sensitivity analysis for different parameter scenarios. Scenarios are listed in order of ascending total costs. τ is the lead time parameter and ρ^f is the congestion cost multiplier. Costs are aggregate values over the entire 50-year horizon and averaged over 100 sample paths. +/− columns list the relative cost change in percent compared to the base case which is indicated by *

τ (periods)	ρ^f (periods)	Purch. cost ($ mil)	+/− (%)	Cong. cost ($ mil)	+/− (%)	Inv. holding cost ($ mil)	+/− (%)	Total ($ mil)
4	3	996	0	81	−18	66	−7	1143
6	5	1012	+1	57	−41	82	+17	1151
6*	3	998	–	98	–	70	–	1166
8	3	1003	+1	126	+29	77	+9	1206
6	1	981	−2	303	+208	54	−23	1338

substation are met immediately at the end of the period when they happened. A valid model with the given parameter settings should make sure that failures in substations with the highest congestion costs are always met without any delay. This is what the figure shows. We also see that delays routinely occur when failures happen in the substations with the lowest congestion costs. This is what we expect given the model's inventory deployment feature and the widely varying congestion costs. Due to this feature the model incurs delays in serving low-priority failures in order to avoid uncertain future delays serving high-priority failures.

We vary the lead time parameter and the congestion cost multiplier in order to perform sensitivity analysis. The effect of changing lead times is of interest as lead times can vary depending on the state of the transformer manufacturing industry. The congestion cost multiplier is the key parameter to control inventory levels and the risk of delays. Starting with the base case we study the effect of one parameter at a time. The left side of Table 4 shows the 5 scenarios we analyze.

First note that the purchasing costs are almost constant across all scenarios. Reducing the lead time saves money, both in terms of congestion costs and inventory holding costs. Increasing the congestion cost multiplier leads to more conservative inventory policies that lead to lower congestion costs at the expense of higher inventory holding costs. Increasing lead times result in cost increases. The policy for this scenario leads to higher inventory levels and therefore inventory holding costs. Interestingly the congestion costs also increase considerably despite higher inventory levels. Apparently, the higher inventory levels do not compensate for the additional uncertainty that the lead time increase introduces into the model. Finally, a lower congestion cost multiplier leads to very aggressive inventory policies that result in very high congestion costs.

Table 5 shows inventory levels and the relative frequency of running out of spares for the congestion cost multiplier scenarios. These numbers are calculated by looking at the results of the 100 evaluation iterations. As the reward for meeting a failure grows (increasing ρ^f) the model adopts a more conservative policy leading to higher inventory levels and a lower shortage frequency. Even with the most conservative inventory policies the probability of running out of spares in any three month period is 1.7 percent. Note the connection between the relative shortage frequency in Ta-

Table 5 ρ^f is the congestion cost multiplier. The base case is indicated by *. The inventory statistic is averaged across all 200 time periods and all 100 sample paths, so the sample size is 20,000

ρ^f (periods)	Average inventory (# of transformers)	Relative frequency of “more failures than spares”
1	4.6	0.1236
3*	5.7	0.0358
5	6.51	0.017

ble 5 and the immediate coverage ratio in Fig. 15. Figure 15 considers delays due to shortages and intentional delays due to holding back inventory. The frequency in Table 5 only considers shortages and it also does not distinguish by substation. It is important to note that the inventory deployment feature will make sure that inventory shortages tend to affect lower-priority failures before they affect higher-priority ones. In other words, with an overall shortage frequency of 1.7 percent the likelihood that high-priority failures will be affected is much closer to 0.

6 Conclusions

We have introduced the spare transformer acquisition and deployment model which we solved using ADP. Our computational evidence shows that standard piecewise-linear separable value functions do not work in this problem setting. This observation is expected to generalize to similar problems in capital intensive industries where lead times are long and the shortage costs by far outweigh inventory holding costs. This paper introduces two novel value function approximation strategies which result in near optimal policies on a variety of test data sets. The PLPRE algorithm is also computationally efficient. Our analysis of a real power transmission system indicates that the expected number of transformer failures per time period will gradually rise by 50 percent over the course of the next 15 years. With an optimal replacement policy there will be considerable delays in meeting low-priority failures. Longer lead times increase congestion costs and inventory holding costs. With high inventory levels shortages are expected to occur 1.7 percent of the time.

In the current analysis the location of replacement transformers is not considered. Allowing replacement transformers to be stored at the substations would be an extension of the current model and a worthy topic of future research.

References

1. Ahmed, S., King, A.J., Parija, G.: A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. J. Glob. Optim. **26**, 3–24 (2003)
2. Bertsekas, D., Tsitsiklis, J.: Neuro-dynamic Programming. Athena Scientific, Belmont (1996)
3. Caroe, C.C., Tind, J.: L-shaped decomposition of two-stage stochastic programs with integer recourse. Math. Program. **83**(3), 139–152 (1998)
4. Chen, Q.M., Egan, D.M.: A Bayesian method for transformer life estimation using Perks’ hazard function. IEEE Trans. Power Syst. **21**(4), 1954–1965 (2006)
5. Chowdhury, A.A., Koval, D.O.: Development of probabilistic models for computing optimal distribution substation spare transformers. IEEE Trans. Ind. Appl. **41**(6), 1493–1498 (2005)

6. De Véricourt, F., Karaesmen, F., Dallery, Y.: Optimal stock allocation for a capacitated supply system. Manage. Sci. **48**(11), 1486–1501 (2002)
7. Deshpande, V., Cohen, M.A., Donohue, K.: A threshold inventory rationing policy for service-differentiated demand classes. Manage. Sci. **49**(6), 683–703 (2003)
8. Godfrey, G., Powell, W.: An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times. Transp. Sci. **36**, 21–39 (2002)
9. Godfrey, G.A., Powell, W.B.: An adaptive, distribution-free approximation for the newsvendor problem with censored demands, with applications to inventory and distribution problems. Manage. Sci. **47**, 1101–1112 (2001)
10. Ha, A.Y.: Stock-rationing policy for make-to-stock production system with two priority classes and backordering. Nav. Res. Logist. **44**, 457–472 (1997)
11. Huiskonen, J.: Maintenance spare parts logistics: special characteristics and strategic choices. Int. J. Prod. Econ. **71**, 125–134 (2001)
12. Kogan, V.I., Roeger, C.J., Tipton, D.E.: Substation distribution transformers failures and spares. IEEE Trans. Power Syst. **11**(4), 1905–1912 (1996)
13. Laporte, G., Louveaux, E.V.: The integer L-shaped method for stochastic integer programs with complete recourse. Oper. Res. Lett. **13**, 133–142 (1993)
14. Li, W., Vaahedi, E., Mansour, Y.: Determining number and timing of substation spare transformers using a probabilistic cost analysis approach. IEEE Trans. Power Deliv. **14**(3), 934–939 (1999)
15. Muckstadt, J.A.: Analysis and Algorithms for Service Parts Supply Chains. Springer, Berlin (2005)
16. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of approximations for maximizing submodular set functions—I. Math. Program. **14**, 265–294 (1978)
17. Norkin, V.I., Ermoliev, Y.M., Ruszczynski, A.: On optimal allocation of indivisibles under uncertainty. Oper. Res. **46**(3), 381–395 (1998)
18. Parija, G.R., Ahmed, S., King, A.J.: On bridging the gap between stochastic integer programming and MIP solver technologies. INFORMS J. Comput. **16**(1), 73–83 (2004)
19. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley, Hoboken (2007)
20. Powell, W.B., Shapiro, J.A., Simão, H.P.: An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem. Transp. Sci. **36**, 231–249 (2002)
21. Simão, H., Powell, W.: Approximate dynamic programming for management of high-value spare parts. J. Manuf. Technol. Manag. **20**, 147–160 (2009)
22. Tempelmeier, H.: Supply chain inventory optimization with two customer classes in discrete time. Eur. J. Oper. Res. **174**, 600–621 (2005)
23. Topaloglu, H., Powell, W.B.: Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems. INFORMS J. Comput. **18**(1), 31–42 (2006)
24. Topkis, D.M.: Optimal ordering and rationing policies in a non-stationary dynamic inventory model with *n* demand classes. Manage. Sci. **15**(3), 160–176 (1968)
25. Wong, H., Van Houtum, G., Cattrysse, D., Van Oudheusden, D.: Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints. J. Oper. Res. Soc. **56**, 1419–1430 (2005)
26. Wong, H., Van Houtum, G.J., Cattrysse, D., Oudheusden, D.V.: Multi-item spare parts systems with lateral transshipments and waiting time constraints. Eur. J. Oper. Res. **171**, 1071–1093 (2006)
27. Zipkin, P.: Foundations of Inventory Management. McGraw-Hill, New York (2000)