Tutorial on Stochastic Optimization in Energy II: An energy storage illustration

Warren B. Powell, *Member, IEEE,* Stephan Meisel

***Abstract*—In Part I of this tutorial, we provided a canonical modeling framework for sequential, stochastic optimization (control) problems. A major feature of this framework is a clear separation of the process of modeling a problem, versus the design of policies to solve the problem. In Part II, we provide additional discussion behind some of the more subtle concepts such as the construction of a state variable. We illustrate the modeling process using an energy storage problem. We then create five variations of this problem designed to bring out the features of the different policies. The first four of these problems demonstrate that each of the four classes of policies is best for particular problem characteristics. The fifth policy is a hybrid that illustrates the ability to combine the strengths of multiple policy classes.**

***Index Terms*—Stochastic optimization, dynamic programming, approximate dynamic programming, reinforcement learning, optimal control, stochastic programming, robust optimization, energy systems, energy storage**

I. Introduction

THE real challenge of stochastic optimization involves taking an actual problem, creating a complete and accurate model, and then searching for practical, high quality policies for controlling the system. In contrast with classical approaches for stochastic optimization, we maintain a clear separation between modeling a problem and designing policies.

We begin our presentation in section II with a quick summary of the canonical model given in part I. Section III provides a more in-depth discussion of state variables, which is an important but largely overlooked dimension of modeling. After a brief summary of the four classes of policies in section IV, section V discusses lookahead policies in more detail, including a summary of five different types of approximations that are used in most lookahead models. To highlight current practice, section VI describes how a number of papers in IEEE Trans. on Power Systems approach the modeling of stochastic optimization problems, where we highlight what we feel are differences in the approaches used by the literature versus the modeling strategy we are proposing.

We then illustrate our modeling framework in section VII using a relatively simple but rich class of energy storage problems. We have designed this problem to illustrate different modeling principles. We use this basic problem to create a series of five variations simply by altering the characteristics of the data, covering issues such as the nature of the variability of loads and energy from renewables, and the accuracy of forecasts. Then, section VIII creates four policies based on policy function approximations (PFAs), a myopic cost function approximation (CFA), a policy based on a value function approximation (VFAs) and a deterministic lookahead policy. A fifth hybrid policy is also designed. We then test each policy on each of the five problems, and show that each policy is best on one of the five classes of problems.

Warren B. Powell is with the Department of Operations Research and Financial Engineering, Princeton University, e-mail: (powell@princeton.edu).

Stephan Meisel is with the Department of Information Systems, University of Muenster, Germany, e-mail: (stephan.meisel@uni-muenster.de)

Manuscript received xxx, 2014

II. Our canonical model

We briefly summarize the elements of our canonical model that were presented in Part I. In addition to fixed parameters such as physical constants, we model a problem by specifying five core components:

- State variable S_t, which is the information that is necessary and sufficient to compute the cost function, decision function (policy) and transition function from time t onward.
- Decision variable (control) x_t (or a_t or u_t), which is specified by some policy $X^\pi(S_t)$ (or $A^\pi(S_t)$ or $U^\pi(S_t)$) which we determine later.
- Exogenous information W_t, which is the information that is learned for the first time at time t (or more precisely, between $t-1$ and t).
- The transition function $S^M(S_t, x_t, W_{t+1})$ which determines the transition from S_t to S_{t+1}. The transition function is sometimes referred to as the state model, system model, plant model, or simply model. If the equations behind the transition function are unknown, then we have to turn to what is called *model free* dynamic programming.
- The objective function, which we write as

$$\min_{\pi\in\Pi} \mathbb{E} \sum_{t=0}^{T} C(S_t, X_t^\pi(S_t)), \tag{1}$$

where $S_{t+1} = S^M(S_t, x_t, W_{t+1})$.

There is considerable interest in replacing the expectation in equation (1) with a quantile, various risk measures, or perhaps the worst outcome (the foundation of robust optimization).

III. State variables

While state variables represent a fundamental concept in the modeling of sequential, stochastic systems, finding a clear, usable definition has proved to be surprisingly difficult. One might think that the literature on Markov decision processes would be a good source, but Richard Bellman, the father of dynamic programming (in operations research) states only that "... we have a physical system characterized at any stage by a small set of parameters, the *state variables*" [1]. The award-winning book on Markov decision processes by Puterman [2] introduces state variables with "At each decision epoch, the system occupies a *state*." (Italics are in the original manuscripts). Wikipedia (in 2013) offers "State commonly refers to either the present condition of a system or entity," which is true but hopelessly vague, or "A state variable is one of the set of variables that are used to describe the mathematical state of a dynamical system" (it is not appropriate to use the word you are defining in the definition).

While many optimal control books similarly do not attempt to define a state variable [3], [4], [5], a few do, in fact, provide explicit definitions. [6](p. 16-17) provides the definition "The *state of a system* is a set of quantities $x_1(t), x_2(t), \ldots, x_n(t)$ which if known at time $t = t_0$ are determined for $t \geq t_0$ by specifying the inputs to the system for $t \geq t_0$." We agree with this definition, but find it offers little guidance to someone designing a state variable from a fundamental description of a problem (for example, what are the inputs?).

The book [7](p. 8) provides a similar definition with one small addition: "The *state* of a system at time t_0 is the information required at t_0 such that the output $y(t)$ for all $t \geq t_0$ is uniquely determined from this information and from [the control] $u(t), t \geq t_0$." The addition is the specification of the control $u(t)$, and while unspecified in the definition, the context implies that the problem (and therefore the control) is deterministic, which means that it would be fully specified at time t_0. This characterization would not work in a sequential stochastic problem, since it does not require that we have the information that tells us *how* to make a decision (now or in the future).

Both of these definitions are effectively saying that the state is all the information needed at time t to model the system from time t onward. But, this does not explain the characterization of $M/G/1$ queueing systems as "non-Markovian" in [7](Chapter 8). In many conversations with experts in the field, it appears that there is a consensus that a state variable S_t includes all the information needed to model a system from any time t onward, yet leading books in Markov decision processes [2] and stochastic control [5] continue to make a distinction between "Markovian" and "history-dependent" systems. Yet, even the authoritative text [8] states "... every stochastic process can be made Markovian by enhancing its state space sufficiently." Clearly, if the state variable does, in fact, include all the information needed to model a system from time t onward, then *every* properly modeled system must be Markovian.

We have come to the conclusion that there are two approaches to working with state variables. The most common, which pervades the research literature, is to assume that the state variable is given (presumably by someone who has already prepared a model of the problem). When this is the case, it makes more sense to characterize the properties of a state variable. For example, if the state variable is incomplete (that is, it does not include all the information needed to model the system from time t onward), then we would say that the system is non-Markovian (and typically history dependent, assuming all inputs are observable).

We adopt what could be called a *constructive* definition of a state variable, wherein it provides a specific recipe for designing a state variable from a raw description of a problem. Thus, we use the definition given originally in [9] (Chapter 5): "The state variable is the minimally dimensioned function of history that is necessary and sufficient to calculate the cost function, the decision function and the transition function." The definition should be ended with "from time t onward" since there are some problems where information from history is only needed at a later point in time. Some notes:

- The decision function (or policy) would include the constraints on the decisions/controls, including integrality requirements.
- The definition does not specify that the policy has to be optimal, and as we see in our example below, different policies may require different information (such as forecasts). We could modify the definition to include only the information needed to compute costs, constraints and the transition function (from time t onward), since this is all that any optimal policy would need. But many problems are solved using particular policy classes, and if information is needed by a policy class, then it needs to be added to the state variable.
- Including the transition function in our list of functions introduces some circular logic, because the transition function is the function that maps S_t to S_{t+1}. We could add extraneous information to this mapping, at which point it becomes a part of our state variable (even though the information is never needed). We would argue that the information does not satisfy the "necessary" requirement, and our example below shows why we need to include the transition function.

There is a tendency in some communities (but not all) to interpret a state variable as the physical state of the system. However, we feel that there is a fairly widespread consensus that a state variable should include any information needed to model the system from time t onward. We have found it useful to identify three classes of information:

- The physical state R_t - This might include the water in a reservoir, the energy in the battery, the temperature of the water in a steam generator, or the number of customers waiting to have their power restored.
- The information state I_t - This would include other information (other than R_t which, technically, is information). This might include the price of electricity on the grid, the speed of wind, the pattern of clouds, or the history of recent decisions (or physical states).
- The state of knowledge K_t - Often called the belief state, this includes a set of probability distributions describing unobservable parameters such as the degradation of transformers due to past storms, or the remaining lifetime of a battery.

Technically, R_t is a type of information that would be included in I_t, while the information in I_t is part of our knowledge but with a distribution that has a spike at one point. We present R_t, I_t and K_t as distinct types of state variables purely because it seems to make modeling more intuitive. We emphasize that it does not matter if a variable is listed as part of R_t or I_t, although there are many systems where R_t is controllable while I_t is exogenous. However, there are problems where information in I_t, such as the price of electricity, can be influenced by decisions (such as charging/discharging grid-scale batteries).

IV. Designing policies

Part I of this tutorial summarizes four classes of policies. For completeness, these are:

- Policy function approximations (PFAs) - These are analytic functions mapping states to actions. These may be lookup tables, parametric or nonparametric approximations. A parametric PFA would be
$$X^\pi(S_t|\theta) = \theta_0 + \theta_1 S_t + \theta_2 S_t^2. \tag{2}$$
- Cost function approximations (CFAs) - A cost function approximation is a parametric modification of a cost function (or constraints) that can be written in the general form
$$X_t^\pi(S_t|\theta) = \arg\min_{x\in\mathcal{X}_t^\pi(\theta)} \bar{C}^\pi(S_t, x|\theta). \tag{3}$$
There are many ways to create modified cost functions. One is to add an error-correction term, giving us
$$X_t^\pi(S_t|\theta) = \arg\min_{x\in\mathcal{X}_t} \Big(C(S_t,x) + \sum_{f\in\mathcal{F}} \theta_f \phi_f(S_t,x)\Big).$$
Other modifications could include energy reserves (this would occur in the constraints) or upper and lower bounds on the energy stored in a battery (to handle surges or drops in energy from renewables).
- Policies based on value function approximations (VFAs) - Here we approximate the cost/value of being in a downstream state resulting from an action now.
$$X_t^\pi(S_t|\theta) = \arg\min_{x\in\mathcal{X}_t} \big(C(S_t,x) + \mathbb{E}\{\overline{V}_{t+1}(S_{t+1}|\theta)|S_t\}\big).$$
If we use a post-decision state (see Part I, or [9](Chapter 4)), we can remove the expectation, creating a more tractable policy
$$X_t^\pi(S_t|\theta) \;=\; \arg\min_{x\in\mathcal{X}_t} \big(C(S_t,x) + \overline{V}_t(S_t^x|\theta)\big). \tag{4}$$
- Lookahead policies - This encompasses any policy that solves a (typically approximate) lookahead model. A deterministic lookahead model (often referred to as model predictive control, or open loop control) would be written
$$X_t^\pi(S_t|\theta) = \arg\min_{\tilde{x}_{tt},\ldots,\tilde{x}_{t,t+H}} \sum_{t'=t}^{t+H} C(\tilde{S}_{tt'}, \tilde{x}_{tt'}), \tag{5}$$
where θ captures the horizon and any other parameters involved in approximating the lookahead model. Recall that we use tilde's on variables in the lookahead model. Important variations include stochastic lookahead models (from the field of stochastic programming) which represent multiple scenarios in the future, and robust optimization, which requires solving a deterministic approximation using an uncertainty set. All of these strategies involve some approximation of the information model.

We note that "dynamic programming" and "optimal control" are typically associated with VFAs and PFAs; "stochastic programming" uses a stochastic lookahead policy; "robust optimization" solves a deterministic problem over an uncertainty set. There is a history of criticizing each of these fields because of the weaknesses that exist in each approach. Dynamic programming suffers from the "curse of dimensionality" (only true if you use lookup tables for value functions); "optimal control" only works for special problems (there is a long history of exploiting the ability to solve certain problems analytically); "stochastic programming" is computationally expensive and is limited by the need to use small samples; and "robust optimization" designs for worst-case outcomes that would never happen. Ultimately, all of these criticisms may be true for particular problems, but each policy deserves its day in court, for which we need a standard benchmark.

A number of authors have recognized that it is possible to construct hybrid policies. Examples include

- Combine a lookahead policy (possibly full tree search) with VFAs to limit the depth of the tree.
- Incorporate a PFA (which can be used to guide low-dimensional decisions) in a high-dimensional cost function as a penalty term.
- Combine a deterministic lookahead with CFA-inspired modifications that are tuned in a stochastic simulator, producing a robust form of deterministic lookahead that is easy to solve and which can handle forecasts.

Below, we use a simple energy storage application to demonstrate that *all* of these policies work best on specific problem instances.

V. Lookahead policies

An optimal policy can always be written in the form

$$X_t^*(S_t) \;=\; \arg\min_{x_t}\Bigg(C(S_t,x_t) + \min_{\pi\in\Pi}\mathbb{E}\left\{\sum_{t'=t+1}^{T} C(S_{t'}, X_{t'}^\pi(S_{t'}))\,\middle|\, S_t, x_t\right\}\Bigg). \tag{6}$$

Since it is almost always impossible to solve (6) exactly, we replace the model within the expectation with an approximation. This approximation is called the *lookahead model*.

Lookahead policies are very popular in practice. Deterministic lookahead policies (also known as model predictive control), which solve deterministic lookahead models, are perhaps the most popular class of policies because they are (typically) easy to solve and handle nonstationary behavior with forecasts in a natural way. Stochastic programming (popular for stochastic unit commitment) uses a stochastic lookahead policy that solves a sampled approximation of the information process. Robust optimization is a lookahead policy using a particular approximation of the information model (the uncertainty set), combined with the use of the max operator over the uncertainty set.

Lookahead policies can be the most confusing, because they involve solving a lookahead model, which makes it easy to forget that it is the base model (equation (1)) that is the real problem that we are solving. This is particularly true in stochastic programming, since solving a stochastic lookahead model can be quite hard. In fact, virtually all lookahead policies use five types of approximations when formulating the lookahead model:

1) Limiting the horizon - We typically reduce the horizon from (t, T) to $(t, t+H)$, where H is a suitable short horizon that is chosen to capture important behaviors.
2) Stage aggregation - A stage represents the process of revealing information followed by the need to make a decision. Since multistage models can be hard to solve, the most popular strategy is to approximate the lookahead model with a two-stage approximation (make decision, see all future information, then make all future decisions with perfect information), but multistage applications can be found in hydro reservoir planning [10]. See [11], [12] and [13] for treatments of two-stage and multistage models.
3) Outcome aggregation or sampling - We replace all future events with a sample $\tilde{\Omega}_t$ that is constructed at time t, given the current state of the system. This is typically referred to as the *scenario tree* [14], [15].
4) Discretization - Time, states, and decisions may all be discretized in a way that makes the resulting model computationally tractable.
5) Dimensionality reduction - We may ignore some variables in our lookahead model as a form of simplification. The most common form of dimensionality reduction is to fix the forecast within the lookahead model, but other variables may be held constant within the lookahead model, yet they evolve in the base model. These are known as *latent variables*.

A major advantage of lookahead models is that there is no need to develop an approximation of a function (as arises with PFAs, CFAs and VFAs), but the price is that you have to solve the lookahead model, which is typically some approximation of the base model (from time t onward). What is important is that whatever approximation is chosen, it still needs to be tested as a policy using the base model such as the model given by equation (1). In fact, we would argue that any attempt to design a lookahead policy should be evaluated by comparing it to a pure deterministic lookahead policy to assess the value of explicitly handling uncertainty. While some authors have adopted this strategy, there is a surprising number of papers that never develop a base model (typically referred to as a "simulator").

VI. Modeling issues from the IEEE literature

We surveyed a series of highly cited papers drawn from IEEE Trans. on Power Systems, spanning problems in energy markets, energy storage, and stochastic unit commitment. Arguably the most popular approach to handling uncertainty uses the framework of stochastic programming, where a sampled (typically two-stage) approximation of a lookahead model is used to produce a policy that reflects future uncertainties [16], [17], [18], [19], [20], [21]. As is common in this community, all of these papers form a policy without writing out a base model for the original problem (equation (1)). The same is true for the growing number of papers that use robust optimization [22], [23], [24], [25], where the lookahead policy uses a deterministic model built around an uncertainty set. Other papers simply use a classical deterministic lookahead policy (e.g. [26]), a strategy widely referred to as model predictive control [27]. We argue that the failure to present the base model means that you are proposing a policy to solve a problem without ever presenting a model of the problem. We also note that while lookahead models may be hard to solve, an optimal solution of a lookahead model is, with rare exceptions, not an optimal policy.

Not all papers use a lookahead policy. In very different settings, [28] (power) and [29] (finance) use simple deterministic policies to solve a stochastic model (which would be the base model). Energy storage has become a popular topic for dynamic programming, where it is standard to present a base model and then suggest either a VFA-based policy [30] or a PFA-based policy [31]. However, a lack of an understanding of the proper modeling of a state variable [31] can invalidate claims of optimality.

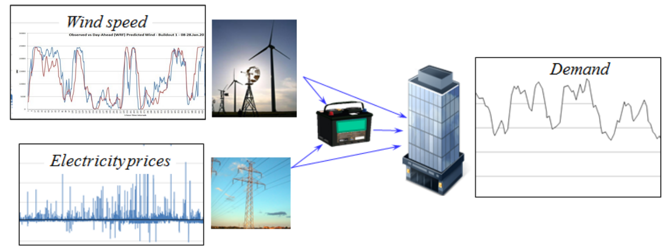

Fig. 1. Storage system with energy from renewables and the grid to serve a time-varying load, with a storage device.

VII. Modeling an energy storage problem

Energy storage represents a fundamental problem class that arises throughout applications in energy systems. We use an elementary storage system, illustrated in figure 1, where our storage might be a battery, a water reservoir or pumped hydro facility, a compressed air storage facility, or other forms of storage such as a building that can be pre-cooled to minimize air conditioning during peak periods.

We are going to use this simple system to show that we can create variations that are best suited to *each* of our four classes of policies. We start by presenting the basic model, and then describe how we can change the character of the problem by the nature of the underlying data.

A. A basic storage model

We use our framework to model our energy system, which consists of states, decision/controls, exogenous information, the transition function and the objective function. We use this example to illustrate some modeling devices that can be used in other applications.

State variable: Although we list state variables as the first element of our problem, creating the state variable evolves from the modeling process. Recall that the state variable S_t is all the information we need (and only the information we need) to compute the cost function, decision function (in particular, the constraints) and the transition function from time t onward. This means that we build up the state variable by first reading the entire model, and pulling the information we need into the state variable.

The cost function needs only p_t:

$$p_t \;=\; \text{The price of electricity from the grid at time } t.$$

For the decision function, we need to compute the cost function and the constraints, which require

$$\begin{aligned} R_t &= \text{The amount of energy in storage at time } t, \\ E_t &= \text{The level of energy from renewables at time } t, \\ L_t &= \text{The load at time } t, \\ \eta_t &= \text{The rate of energy loss (e.g. due to evaporation} \\ &\quad\ \text{or temperature) between } t-1 \text{ and } t. \end{aligned}$$

If we use a lookahead policy, we will need any available forecasts. We represent our forecast of loads using

$$\begin{aligned} f^L_{tt'} &= \text{Forecast of the load } L_{t'} \text{ at time } t' > t \text{ given} \\ &\quad\ \text{what we know at time } t, \\ &= \mathbb{E}_t L_{t'}, \\ f^L_t &= (f^L_{tt'})_{t'>t}. \end{aligned}$$

Similarly, we might have forecasts of energy from renewables f^E_t, forecasts of exogenous inputs f^R_t (represented as $\hat{R}_t$ below), and forecasts of losses f^η_t. We can represent our set of forecasts using

$$f_t = (f^L_t, f^E_t, f^R_t, f^\eta_t).$$

Finally, we see that the transition function also requires p_{t-1} and p_{t-2}. Thus, our state variable is

$$S_t = \big(R_t, (p_t, p_{t-1}, p_{t-2}), E_t, L_t, \eta_t, f_t\big).$$

This state variable captures all the information we are going to need (this needs to be verified as we present the rest of the model). Note that this consists of a controllable physical state R_t, and information $I_t = ((p_t, p_{t-1}, p_{t-2}), E_t, L_t, \eta_t, f_t)$. In this model, we do not have any unobservable parameters that we would put in our knowledge state K_t.

Decision/control variable:
In our storage system, let G refer to grid, E refer to our renewable energy, B is the "battery" (storage), and L be the load. We would then designate, for example, the flow of energy from the grid to the load as x^{GL}_t. Our control is the vector

$$x_t \;=\; (x^{GL}_t, x^{GB}_t, x^{EL}_t, x^{EB}_t, x^{BL}_t).$$

The decisions are subject to several constraints:

$$\begin{aligned} x^{EL}_t + x^{EB}_t &\le E_t, &(7)\\ x^{GL}_t + x^{EL}_t + \eta_t x^{BL}_t &= L_t, &(8)\\ x^{BL}_t &\le R_t, &(9)\\ x^{EB}, x^{GB}, x^{BL} &\le \rho^{chrg}, &(10)\\ x^{GL}_t, x^{EL}_t, x^{EB}_t, x^{BL}_t &\ge 0. &(11) \end{aligned}$$

Equation (7) limits the energy from renewables to what is being generated, while (8) limits the amount that we can send to the customer by the load at that point in time. Equation (9) limits what we can draw from the battery to what is in the battery at time t. Equation (10) limits the charge/discharge rate for the battery (we assume that ρ^{chrg} is a fixed parameter). Note that we apply nonnegativity to every flow variable except x^{GB}_t, since we can sell to the grid. We refer to the feasible region defined by (7)-(11) as $\mathcal{X}_t$, where this is implicitly a function of S_t.

Exogenous information:

$$\begin{aligned} \hat{R}_t &= \text{Exogenous change to the energy stored between} \\ &\quad\ t-1 \text{ and } t \text{ (e.g. rainfall, chemical leakage)}, \\ \epsilon^p_t &= \text{A noise term in the transition equation for prices} \\ &\quad\ \text{(see below) that is revealed at time } t, \\ \hat{E}_t &= \text{The change in energy produced from renewable} \\ &\quad\ \text{sources between } t-1 \text{ and } t, \\ \hat{L}_t &= \text{Deviation from the forecasted load between } t-1 \\ &\quad\ \text{and } t, \\ \hat{\eta}_t &= \text{Change in the rate of energy conversion loss} \\ &\quad\ \text{(e.g. due to temperature) between } t-1 \text{ and } t. \end{aligned}$$

We assume that our vector of forecasts f_t is provided by an exogenous (commercial) forecasting service, so we are given the forecast directly (rather than the change in the forecast). Our exogenous information W_t is then

$$W_t \;=\; (\hat{R}_t, \epsilon^p_t, \hat{E}_t, \hat{L}_t, \hat{\eta}_t, f_t).$$

To complete the model, we would have to either provide the probability distribution for the exogenous variables, or to specify the source for actual observations.

Transition function:

$$\begin{aligned} R_{t+1} &= R_t + \eta_{t+1}(x^{GB}_t + x^{EB}_t) - x^{BL}_t + \hat{R}_{t+1}, &(12)\\ p_{t+1} &= \theta_0 p_t + \theta_1 p_{t-1} + \theta_2 p_{t-2} + \epsilon^p_{t+1}, &(13)\\ E_{t+1} &= E_t + \hat{E}_{t+1}, &(14)\\ L_{t+1} &= L_t + \hat{L}_{t+1}, &(15)\\ \eta_{t+1} &= \eta_t + \hat{\eta}_{t+1}. &(16) \end{aligned}$$

In equation (13), ϵ^p_{t+1} is a random variable independent of any history. These equations represent the transition function $S^M(S_t, x_t, W_{t+1})$. Note that our resource transition function depends on stochastic losses η_{t+1} (such as evaporation) and exogenous input $\hat{R}_{t+1}$ (such as rainfall) that only become known at time $t+1$.

Objective function:
For our simple storage problem, we do nothing but buy and sell from the grid at a price p_t, which gives us a cost function

$$C(S_t, x_t) = p_t x^{GB}_t,$$

where we pay money if we are charging our device ($x_t > 0$) and we receive money if we are selling back to the grid ($x_t < 0$).

Our objective function is our canonical objective originally given in equation (1):

$$\min_{\pi \in \Pi} \mathbb{E}^{\pi} \sum_{t=0}^{T} C(S_t, X_t^{\pi}(S_t)), \quad (17)$$

where $S_{t+1} = S^M(S_t, X_t^{\pi}(S_t), W_{t+1})$, which is given by equations (12)-(16). This is now a fully specified sequential decision *problem*; we just have to find an effective policy.

B. Problem variations

We are going to create a series of problem variations that are specifically designed to bring out the strengths of each of the four classes of policies. Our sample problems include:

A) A stationary problem with heavy-tailed prices, relatively low noise, moderately accurate forecasts and a reasonably fast storage device.

B) A time-dependent problem with daily load patterns, no seasonalities in energy and price, relatively low noise, less accurate forecasts and a very fast storage device.

C) A time-dependent problem with daily load, energy and price patterns, relatively high noise, less accurate forecasts using time series (errors grow with the horizon) and a reasonably fast storage device.

D) A time-dependent problem with daily load, energy and price patterns, relatively low noise, very accurate forecasts and a reasonably fast storage device.

E) Same as (C), but the forecast errors are stationary over the planning horizon.

VIII. Designing and comparing polices

We tested all four classes of policies (reviewed in section IV), plus a hybrid policy that combines a deterministic lookahead with a form of cost function approximation. We intentionally designed the problems to bring out the strengths of the different policies, but this necessarily also highlights the weaknesses of each policy class.

For the PFA policy, we use the following rule:

$$X_t^{PFA}(S_t|\theta) = \begin{cases} x_t^{EL} &= \min\{L_t, E_t\}, \\ x_t^{BL} &= \begin{cases} h_t & \text{If } p_t > \theta^U \\ 0 & \text{If } p_t < \theta^U \end{cases} \\ x_t^{GL} &= L_t - x_t^{EL} - x_t^{BL}, \\ x_t^{EB} &= \min\{E_t - x_t^{EL}, \rho^{chrg}\}, \\ x_t^{GB} &= \begin{cases} \rho^{chrg} - x_t^{EB} & \text{If } p_t < \theta^L \\ 0 & \text{If } p_t > \theta^L \end{cases} \end{cases}$$

where $h_t = \min\{L_t - x_t^{EL}, \min\{R_t, \rho^{chrg}\}\}$.

The first cost function approximation minimizes a one-period cost plus a tunable error correction term:

$$X^{CFA-EC}(S_t|\theta) = \arg\min_{x_t \in \mathcal{X}_t} C(S_t, x_t) + \theta * (x_t^{GB} + x_t^{EB} + x_t^{BL}) \quad (18)$$

where $\mathcal{X}_t$ is defined by (7)-(11). We use a linear correction term for simplicity (and it sometimes works).

Our VFA policy uses an approximate value function approximation, which we write as

$$X^{VFA}(S_t) = \arg\min_{x_t \in \mathcal{X}_t} C(S_t, x_t) + \overline{V}_t^x(R_t^x) \quad (19)$$

where $\overline{V}_t^x(R_t^x)$ is a piecewise linear function approximating the marginal value of the post-decision resource state. For a complete description, see [32].

The deterministic lookahead optimizes over a horizon H using forecasts of exogenous information:

$$X_t^{LA-DET}(S_t|H) = \arg\min_{(\tilde{x}_{tt}, \ldots, \tilde{x}_{t,t+H})} \sum_{t'=t}^{t+H} C(S_{tt'}, \tilde{x}_{tt'}) \quad (20)$$

subject to, for $t' = t, \ldots, T$:

$$\begin{aligned} \tilde{x}_{tt'}^{EL} + \tilde{x}_{tt'}^{EB} &\leq f_{tt'}^{E}, & (21)\\ f_{tt'}^{\eta}(\tilde{x}_{tt'}^{GL} + \tilde{x}_{tt'}^{EL} + \tilde{x}_{tt'}^{BL}) &= f_{tt'}^{L}, & (22)\\ \tilde{x}_{tt'}^{BL} &\leq \tilde{R}_{tt'}, & (23)\\ \tilde{R}_{t,t'+1} - (\tilde{R}_{tt'} + f_{t,t'+1}^{\eta}(\tilde{x}_{tt'}^{GB} + \tilde{x}_{tt'}^{EB}) - \tilde{x}_{tt'}^{BL}) &= f_{t,t'+1}^{R}, & (24)\\ \tilde{x}_{tt'} &\geq 0. & (25) \end{aligned}$$

The last policy, $X_t^{LA-CFA}(S_t|\theta^L, \theta^U)$, is a hybrid lookahead with a form of cost function approximation in the form of two additional constraints for $t' = t+1, \ldots, T$:

$$\begin{aligned} \tilde{R}_{tt'} &\geq \theta^L, & (26)\\ \tilde{R}_{tt'} &\leq \theta^U. & (27) \end{aligned}$$

These constraints provide buffers to ensure that that we do not plan on the energy level getting too close to the lower or upper limits, allowing us to anticipate that there will be times when the energy from a renewable source is lower, or higher, than we planned. We note that a CFA-lookahead policy is actually a hybrid policy, combining a deterministic lookahead with a cost function approximation (where the approximation is in the modification of the constraints).

The design of a hybrid policy, as with any parametric policy, is part art. It requires thinking about what a pure deterministic lookahead policy would do wrong when implemented in an uncertain environment. In this case, a pure deterministic lookahead policy might allow the energy in the battery to drop to zero, failing to recognize that the energy from wind or solar may be less than expected. At the same time, if the battery is allowed to be fully charged ignores the possibility that the energy from renewables may be more than expected. Our parametric modifications, then, are chosen specifically to address these issues.

Most of these policies involve tunable parameters θ which have to be optimized using (17), where we replace the search over policies $\pi \in \Pi$ with a search over the parameters θ. The only exception is the VFA-based policy where the value function is estimated using Bellman's equation. For these experiments, the tuning was performed manually, but there is a wide range of stochastic search algorithms that can be applied to this problem (see [33] and [34] and the references cited there).

These policies were tested on a series of carefully chosen problems. The results, shown in table I, are summarized as a fraction of the posterior bound (that is, the optimal solution after all information is known). Note that the best CFA-LA is the same as the deterministic lookahead for problems A-D. The results show clearly that we can design realistic variations of our storage problem which make *each* of our policies work the best, sometimes by fairly significant margins.

Problem:	PFA	CFA-EC	VFA	LA-DET	LA-CFA
A	**0.959**	0.839	0.936	0.887	0.887
B	0.714	**0.752**	0.712	0.746	0.746
C	0.865	0.590	**0.914**	0.886	0.886
D	0.962	0.749	0.971	**0.997**	0.997
E	0.865	0.590	0.914	0.922	**0.934**

TABLE I
PERFORMANCE OF EACH CLASS OF POLICY ON EACH PROBLEM, RELATIVE TO THE OPTIMAL POSTERIOR SOLUTION.

IX. CONCLUSIONS

Central to our modeling strategy is equation (1) which involves a search over policies. This is our point of departure from classical modeling frameworks that fall under names such as dynamic programming, optimal control, stochastic programming, and robust optimization, which each imply a specific class of policy. In our approach, our modeling strategy does not presume any particular class of policy.

In practice, it is not going to be possible to test variations from all four classes of policies for every problem, since some are clearly more applicable to certain problems. From our experience, we offer the following general guidelines:

- Policy function approximations work well when the relationship between state and action is fairly clear. For example, charging a battery when prices are below one level, and discharge when prices are higher than other, is an example of policy structure. PFAs have also proven useful when the functions can be approximated accurately using perhaps a linear model ("affine policies" are popular, and are optimal for certain problem classes) or a neural network.
- Cost function approximations are popular when a deterministic model (this might be myopic or a lookahead model) produces good solutions, but where obvious adjustments are needed to make the solution more robust. One example is the imposition of constraints to force operating reserves in the unit commitment problem.
- Value function approximations work well when the downstream cost of being in a state is easy to approximate. For example, we might feel that the value function can be accurately approximated using a quadratic function, a neural network or a linear model.
- Lookahead models are needed in time-varying settings where an accurate forecast is necessary. They are also needed if the value of being in a state is a complex function that is hard to approximate.

The choice of policy will ultimately require striking a balance between solution quality (including robustness across a variety of problem settings) and computational complexity. Computational issues also need to consider off-line computing requirements (computing the value function, tuning a PFA or CFA) versus online computation. Lookahead policies typically do not have an offline learning component, but are harder to compute online (sometimes dramatically so). By contrast, a value function approximation may be hard to compute, but are then quite easy to use online. Also, a parametric model (for a PFA or VFA) is much easier to compute than a lookup table (although the lookup table may work better), which might be an issue if the policy has to be implemented in a small device. A robust cost function approximation for unit commitment models (which is standard industry practice) may prove to be quite effective in addition to being computationally much easier than a stochastic program or robust optimization formulation.

ACKNOWLEDGMENT

The work of the first author was supported by the National Science Foundation under grant ECCS-1127975 and the SAP initiative for energy systems research. The work of the second author was supported by the German Research Foundation.

REFERENCES

[1] R. E. Bellman, "Dynamic Programming," *Princeton University Press, Princeton, NJ.*, 1957.
[2] M. L. Puterman, *Markov Decision Processes*, 2nd ed. Hoboken, NJ: John Wiley and Sons, 2005.
[3] S. P. Sethi and G. L. Thompson, *Optimal Control Theory*, 2nd ed. Boston: Kluwer Academic Publishers, 2000.
[4] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Hoboken, NJ: John Wiley & Sons, 2012.
[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming*. Belmont, MA: Athena Scientific, 2012.
[6] D. E. Kirk, *Optimal Control Theory: An Introduction*. New York: Dover, 2004.
[7] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York: Springer, 2008.
[8] E. Cinlar, *Probability and Stochastics*. New York: Springer, 2011.
[9] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2011.
[10] J. Jacobs, G. Freeman, J. Grygier, D. P. Morton, G. Schultz, K. Staschus, and J. Stedinger, "SOCRATES: A system for scheduling hydroelectric generation under uncertainty," *Annals of Operations Research*, vol. 59, no. 1, pp. 99–133, Dec. 1995.
[11] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, 2nd ed. New York: Springer, 2011.
[12] A. J. King and S. Wallace, *Modeling with stochastic programming*. New York: Springer Verlag, 2012.
[13] A. Shapiro, D. Dentcheva, and A. Ruszczyski, *Lectures on stochastic programming: modeling and theory*, 2nd ed. Philadelphia: SIAM, 2014.
[14] H. Heitsch and W. Romisch, "Scenario tree modeling for multistage stochastic programs," *Mathematical Programming*, vol. 118, pp. 371–406, 2009.
[15] N. Growe-Kuska, H. Heitsch, and W. Romisch, "Scenario reduction and scenario tree construction for power management problems," *IEEE Bologna Power Tech Proceedings (Borghetti, A., Nucci, CA, Paolone, M. eds.)*, vol. 34, 2003.
[16] L. Wu, M. Shahidehpour, and T. Li, "Stochastic Security-Constrained Unit Commitment," *IEEE Trans. on Power Systems*, vol. 22, no. 2, pp. 800–811, 2007.
[17] J. M. Morales, S. Member, A. J. Conejo, and J. Pérez-ruiz, "Economic Valuation of Reserves in Power Systems With High Penetration of Wind Power," *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 900–910, 2009.

[18] E. M. Constantinescu, V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu, "A Computational Framework for Uncertainty Quantification and Stochastic Optimization in Unit Commitment With Wind Power Generation," *IEEE Trans. on Power Systems*, vol. 26, no. 1, pp. 431–441, 2011.
[19] A. Papavasiliou, S. Member, S. S. Oren, and R. P. O. Neill, "Reserve Requirements for Wind Power Integration : A Scenario-Based Stochastic Programming Framework," *IEEE Trans. on Power Systems*, vol. 26, no. 4, pp. 2197–2206, 2011.
[20] V. S. Pappala, S. Member, I. Erlich, S. Member, K. Rohrig, and J. Dobschinski, "A Stochastic Model for the Optimal Operation of a Wind-Thermal Power System," *IEEE Trans. on P*, vol. 24, no. 2, pp. 940–950, 2009.
[21] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty Management in the Unit Commitment Problem," *IEEE Trans. on Power Systems*, vol. 24, no. 2, pp. 642–651, 2009.
[22] A. H. Hajimiragha, C. A. Cañizares, M. W. Fowler, S. Moazeni, A. Elkamel, and B. Indices, "A Robust Optimization Approach for Planning the Transition to Plug-in Hybrid Electric Vehicles," *IEEE Trans. on Power Systems*, vol. 26, no. 4, pp. 2264–2274, 2011.
[23] R. Jiang, S. Member, J. Wang, Y. Guan, and A. Sets, "Power and Pumped Storage Hydro," *IEEE Trans. on Power Systems*, vol. 27, no. 2, pp. 800–810, 2012.
[24] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive Robust Optimization for the Security Constrained Unit Commitment Problem," *IEEE Trans. on Power Systems*, vol. 28, no. 1, pp. 52–63, 2013.
[25] C. Zhao, J. Wang, and J. Watson, "Multi-Stage Robust Unit Commitment Considering Wind and Demand Response Uncertainties," *IEEE Trans. on Power Systems*, vol. 28, no. 3, pp. 2708–2717, 2013.
[26] L. Martinez and S. Soares, "Comparison Between Closed-Loop and Partial Open-Loop Feedback Control Policies in Long Term Hydrothermal Scheduling," *IEEE Trans. on Power Systems*, vol. 17, no. 2, pp. 330–336, 2002.
[27] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control : Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.
[28] K. Clement-Nyns, E. Haesen, S. Member, and J. Driesen, "The Impact of Charging Plug-In Hybrid Electric Vehicles on a Residential Distribution Grid," *IEEE Trans. on Power Systems*, vol. 25, no. 1, pp. 371–380, 2010.
[29] E. Y. Bitar, R. Rajagopal, P. P. Khargonekar, and K. Poolla, "Bringing Wind Energy to Market," *IEEE Trans. on Power Systems*, vol. 27, no. 3, pp. 1225–1235, 2012.
[30] X. Xi, R. Sioshansi, and V. Marano, "A Stochastic Dynamic Programming Model for Co-optimization of Distributed Energy," *Energy Sytstems*, 2014.
[31] H.-i. Su, S. Member, and A. E. Gamal, "Modeling and Analysis of the Role of Energy Storage for Renewable Integration : Power Balancing," *IEEE Trans. on Power Systems*, vol. 28, no. 4, pp. 4109–4117, 2013.
[32] D. F. Salas and W. B. Powell, "Benchmarking a Scalable Approximate Dynamic Programming Algorithm for Stochastic Control of Grid-Level Energy Storage," Princeton University, Princeton NJ, Tech. Rep. 2004, 2012.
[33] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control*. Hoboken, NJ: John Wiley & Sons, 2003.
[34] W. B. Powell and I. O. Ryzhov, *Optimal Learning*. Hoboken, NJ: John Wiley & Sons, 2012.

Warren B. Powell is a professor in the Department of Operations Research and Financial Engineering at Princeton University, where he has taught since 1981. His research specializes in computational stochastic optimization, with applications in energy, transportation, health and finance. He has authored/coauthored over 200 publications and two books. He founded and directs CASTLE Labs (www.castlelab.princeton.edu), specializing in fundamental contributions to computational stochastic optimization with a wide range of applications.

Stephan Meisel is an assistant professor in the Department of Information Systems at the University of Muenster, Germany. Before joining the department in 2013, he was a postdoctoral research associate in the Department of Operations Research and Financial Engineering at Princeton University. The focus of his research is on computational stochastic optimization with applications in energy and transportation.