Parametric Simulation Optimization for Multistage Stochastic Programming

Saeed Ghadimi ∗ Raymond T. Perkins † Warren B. Powell ‡
January 1, 2019

Abstract

A widely used heuristic method for solving multistage stochastic linear programming problems, as arises in many inventory and resource allocation problems, is to use a deterministic rolling horizon procedure which has been modified to handle uncertainty (e.g. buffer stocks, schedule slack). This approach has been criticized for its use of a deterministic approximation of a stochastic problem, which is the major motivation for using stochastic approximations of the future within the context of stochastic programming. We recast this debate by identifying both deterministic and stochastic approaches as policies for solving a stochastic base model, which may be a simulator or the real world. Stochastic lookahead models (stochastic programming) require a range of approximations to keep the problem tractable. By contrast, so-called deterministic models are actually parametrically modified cost function approximations which use parametric adjustments to the objective function and/or the constraints. These parameters are then optimized in a stochastic base model which does not require making any of the types of simplifications required by stochastic programming. We formalize this strategy and describe simulation-based optimization algorithms to optimize the parameters.

Keywords
Stochastic Optimization, Policy Search, Stochastic Programming, Simulation-based Optimization, Parametric Cost Function Approximation

1 Introduction

There has been a long history in industry of using deterministic optimization models to make decisions that are then implemented in a stochastic setting. Energy companies use deterministic forecasts of wind, solar and loads to plan energy generation (Wallace and Fleten (2003)); airlines use deterministic estimates of flight times to schedule aircraft and crews (Lan et al. (2006)); and retailers use deterministic estimates of demands and travel times to plan inventories (Harrison and Van Mieghem (1999)). These models have been widely criticized in the research community for not accounting for uncertainty, which often motivates the use of large-scale stochastic programming models which explicitly model uncertainty in future outcomes (Mulvey et al. (1995) and Birge and Louveaux (2011)). These large-scale stochastic programs have been applied to unit commitment

∗Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ; email: sghadimi@princeton.edu.
†Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ; email: raymondp@princeton.edu.
‡Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ; email: powell@princeton.edu
(Jin et al. (2011)), hydroelectric planning (Carpentier et al. (2015)), and transportation (Lium et al. (2009)). These models use large scenario trees to approximate potential future events, but result in very large-scale optimization models that can be quite hard to solve in practice.

We make the case that these previous approaches ignore the true problem that is being solved, which is always stochastic. The so-called “deterministic models” used in industry are almost always parametrically modified deterministic approximations, where the modifications are designed to handle uncertainty. Both the “deterministic models” and the “stochastic models” (formulated using the framework of stochastic programming) are examples of lookahead policies to solve a stochastic optimization problem. The stochastic optimization problem is to find the best policy which is typically tested using a simulator, but may be field tested in an online environment (the real world).

In this paper, we characterize these modified deterministic models as parametric cost function approximations which puts them into the same category as other parameterized policies that are well known in the research community working on policy search (Ng and Jordan (2000), Peshkin et al. (2000), Hu et al. (2007), Deisenroth et al. (2013), and Mannor et al. (2003)). A parallel community has evolved under the name simulation-optimization (see the recent edited volume Fu (2015)), where powerful tools have been developed based on the idea of taking derivatives of simulations (see the extensive literature on derivatives of simulations covered in Glasserman (1991), Ho (1992), Kushner and Yin (2003), Cao (2008)); a nice tutorial is given in Chau et al. (2014).

Our use of modified linear programs is new to the policy search literature, where “policies” are typically parametric models such as linear models (“affine policies”), structured nonlinear models (such as (s,S) policies for inventories) or neural networks. The process of designing the modifications (in this paper, these modifications always appear in the constraints) requires the same art as the design of any statistical model or parametric policy. The heart of this paper is the design of gradient-based search algorithms, which are nontrivial in this setting.

This paper formalizes the idea, used for years in industry, that an effective way to solve complex stochastic optimization problems is to shift the modeling of uncertainty from a lookahead approximation, where even deterministic lookahead models can be hard to solve, to the stochastic base model, typically implemented as a simulator but which might also be the real world. Tuning a model in a stochastic simulator makes it possible to handle arbitrarily complex dynamics, avoiding the many approximations (such as two-stage models, scenario trees, exogenous information that is independent of decisions) that are standard in stochastic programming. In the process, we are expanding the range of problems considered by the simulation-optimization community to the entire class of vector-valued multistage stochastic optimization problems considered in the stochastic programming literature.

Parametric cost function approximations make it possible to exploit structural properties. For example, it may be obvious that the way to handle uncertainty when planning energy generators in a unit commitment problem is to require extra reserves at all times of the day. A stochastic programming model encourages this behavior, but the requirement for a manageable number of scenarios will produce the required reserve only when one of the scenarios requires it. Imposing a reserve constraint (which is a kind of cost function approximation) allows us to impose this requirement at all times of the day, and to tune this requirement under very realistic conditions.

Designing a parametric cost function approximation closely parallels the design of any parametric statistical model, which is part art (creating the model) and part science (fitting the model). In fact, the design of the parameterization and the tuning of the parameters, represent important
areas of research. To illustrate the process of designing a parametric cost function approximation, we use the setting of a time-dependent stochastic inventory planning problem that arises in the context of energy storage, but could represent any inventory planning setting. We assume we have access to rolling forecasts where forecast errors are based on careful modeling of actual and predicted values for energy loads, generation from renewable sources, and prices. The combination of the time-dependent nature and the availability of rolling forecasts which are updated each time period make this problem a natural setting for lookahead models, where the challenge is how to handle uncertainty. We have selected this problem since it is relatively small, simplifying the extensive computational work. However, our methodology is scalable to any problem setting which is currently being solved using a deterministic model.

Most important, the parametric CFA opens up a fundamentally new approach for providing practical tools for solving high-dimensional, stochastic programming problems. It provides an alternative to classical stochastic programming with its focus on optimizing a stochastic lookahead model which requires a variety of approximations to make it computationally tractable. The parametric CFA makes it possible to incorporate problem structure, such as the recognition that robust solutions can be achieved using standard methods such as schedule slack and/or buffer stocks. The parametric CFA makes it possible to incorporate problem structure for handling uncertainty. Some examples include:

- Supply chains handle uncertainty by introducing buffer stocks.
- Airlines handle uncertainty due to weather and congestion by using schedule slack.
- Air freight companies plan for equipment problems by maintaining spare aircraft at different locations around the country.
- Hospitals can handle uncertainty in blood donations and the demand for blood by maintaining supplies of O-minus blood, which can be used by anyone.
- Grid operators handle uncertainty in generator failures, as well as uncertainty in energy from wind and solar, by requiring generating reserves.

Central to our approach is the ability to manage uncertainty by recognizing effective strategies for responding to unexpected events. We would argue that this structure is apparent in many settings, especially in complex resource allocation problems. At a minimum, we offer that our approach represents an interesting, and very practical, alternative to stochastic programming.

This paper makes the following contributions. First, we introduce and develop the idea of parameterized cost function approximations as a tool for solving important classes of multistage stochastic programming problems, shifting the focus from solving complex, stochastic lookahead models to optimizing a stochastic base model. This approach is computationally comparable to solving deterministic approximations, with the exception that the parametric modifications have to be optimized, typically in a simulator that avoids the many approximations made in stochastic lookahead models. Second, we describe different simulation-based optimization algorithms for performing the policy search drawing on the tools of the simulation-optimization community. We note that, as is typically the case, the objective function is nonconvex in the parameters, but that we typically know the optimal values of the parameters for the special case of perfect forecasts. This provides a natural starting point for stochastic search algorithms. Third, we use extensive simulations on a complex, nonstationary energy storage problem to demonstrate that a simple parameterization can provide results that are 30 percent better than a base policy using unmodified...
rolling forecasts. We also show that starting from the optimal parameter values if forecasts were perfect represents a robust starting point for problems with uncertain forecasts.

Our presentation is organized as follows. The modeling framework is given in Section 2. We then provide an overview of the different classes of policies in Section 3. We defer the literature review until Section 3 which allows us to put the literature in the framework of the different classes of policies. We formally introduce the parametric CFA approach in Section 4 along with the derivation of the gradient of the base model with respect to the policy parameters and algorithms for optimizing these parameters in Section 5. In Section 6, we specialize the parametric CFA approach for a complex, nonstationary inventory problem that arises in energy storage. Finally, Section 7 reports on the results of extensive numerical experiments for solving this application problem.

2 Canonical Model

Sequential, stochastic decision problems require a richer notation than standard linear programs and deterministic problems. For the sake of notational consistency, we follow the canonical model in Powell (2011) which breaks dynamic programs into five dimensions:

- The state variable, $S_t$, is the minimally dimensioned function of history that, combined with the exogenous information process, contains the information needed to compute the cost function, the constraints, and the transition function, from time $t$ onward. We use the initial state, $S_0$, to represent any deterministic parameters, as well as probabilistic beliefs (if needed).

- A decision, $x_t$, is an $n$-dimensional vector that must satisfy $x_t \in X_t$, which is typically a set of linear constraints. Decisions are determined by a decision function (policy) which we denote by $X^\pi_t(S_t)$, where $\pi$ carries the information that determines the structure and parameters that define the function.

- The exogenous information, $W_t$, describes the information that first becomes known at time $t$. We let $\omega \in \Omega$ be a sample path of $W_1, \ldots, W_T$. Let $\mathcal{F}$ be the sigma algebra on $\Omega$, and let $\mathcal{P}$ be a probability measure on $(\Omega, \mathcal{F}, \mathcal{P})$. Next let $\mathcal{F}_t = \sigma(W_1, \ldots, W_t)$ be the sigma-algebra generated by $W_1, \ldots, W_t$, where $(\mathcal{F}_t)_{t=1}^T$ forms a filtration. The information $W_t$ may depend on the state $S_t$ and/or the action $x_t$, which means it depends on the policy. If this is the case, we write our probability space as $(\Omega^\pi, \mathcal{F}^\pi, \mathcal{P}^\pi)$, with the associated expectation operator $\mathbb{E}^\pi$.

- The transition function, $S^M(\cdot)$, explicitly describes the relationship between the state of the model at time $t$ and $t+1$,

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

- The objective function is used to evaluate the effectiveness of a policy or sequence of decisions. It minimizes the expected sum of the costs $C(S_t, x_t)$ in each time period $t$ over a finite horizon, where we seek to find the policy that solves

$$\min_{\pi \in \Pi} \mathbb{E}^\pi \left[ \sum_{t=0}^T C(S_t, X^\pi_t(S_t)) \right] \bigg| S_0, \tag{2}$$

where $S_{t+1} = S^M(S_t, X^\pi_t(S_t), W_{t+1})$. We use $\mathbb{E}^\pi(\cdot)$ since the exogenous variables in the model may be affected by the decisions generated by our policy. Therefore we express the expectation
as dependent on the policy. Since stochastic problems incorporate uncertainty in the model a variety of risk measures can be used in replacement of expectation. Equation (2), along with the transition function and the exogenous information process, is called the base model.

This canonical model can be used to model virtually any sequential, stochastic decision problem as long as we are using expectations instead of risk measures. We use this setting to put different policies onto a standard footing for comparison. In the next section we describe the major classes of policies that we can draw from to solve the problem. We use this framework to review the literature.

We state this canonical model because it sets up our modeling framework, which is fundamentally different than the standard paradigm of stochastic programming (for multistage problems). However, it sets the foundation for searching over policies which is fundamental to our approach.

3 Solution Strategies

There are two fundamental strategies for identifying policies. The first is policy search, where we search over different classes of functions \( f \in F \) and different parameters \( \theta \in \Theta_f \) in each class. Policy search is written as

\[
\min_{\pi=(f,\theta) \in (F,\Theta_f)} \mathbb{E} \left\{ \sum_{t=0}^{T} C(S_t, X_t^\pi(S_t|\theta_f)) \mid S_0 \right\}
\]

Policies that can be identified using policy search come in two classes:

Policy function approximations (PFAs) These include linear or nonlinear models, neural networks, and locally parametric functions. For example a linear model, also known as an affine policy, might be written

\[
X^{PFA}(S_t|\theta) = \theta_0 + \theta_1 \phi_1(S_t) + \theta_2 \phi_2(S_2) + \ldots
\]

PFAs (using any of a wide range of approximation strategies) have been widely studied in the computer science literature under the umbrella of policy search (which we expand in this paper), most commonly using parametric functions. A few examples of parametric policies are the Boltzmann exploration policy (Sutton et al. (1999)), linear decision rules (see Bertsimas and Goyal (2012), Hadjiyiannis et al. (2011), and Bertsimas et al. (2011)), and neural networks (Lillicrap et al. (2015) and Levine and Abbeel (2014)). See Ng and Jordan (2000), Peshkin et al. (2000), Hu et al. (2007), Deisenroth et al. (2013), and Mannor et al. (2003) for a sample.

Cost function approximations (CFAs) Here we use parametrically modified costs and constraints that are then minimized. These are written

\[
X^{CFA}(S_t|\theta) = \arg\min_{x_t \in \mathcal{X}(\theta)} \tilde{C}^\pi(S_t, x_t|\theta).
\]

CFAs are widely used in industry for complex problems such as scheduling energy generation or planning supply chains, but they have not been studied formally in the research literature.

In special cases, PFAs and CFAs may produce optimal policies, although generally we are looking for the best within a class.
The second strategy is to construct policies based on lookahead models, where we capture the value of the downstream impact of a decision $x_t$ made while in state $S_t$. An optimal policy can be written

$$X_t^*(S_t) = \arg\min_{x_t \in X_t} \left( C(S_t, x_t) + \mathbb{E} \left\{ \min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t' = t+1}^{T} C(S_{t'}, X^\pi(S_{t'})) | S_{t+1} \right\} | S_t, x_t \right\} \right).$$  (4)

Equation (4) is basically Bellman’s equation, but it is computable only for very special instances. For example, to model a decision tree the policies $\pi$ in (4) would be a lookup table expressing the action to be taken out of every decision node.

There are two core strategies for approximating the lookahead portion in (4):

**Value function approximations (VFAs)** Here we approximate the lookahead portion using a value function. Standard practice is to write the value function $V_t(S_t)$ around the (pre-decision) state $S_t$ as

$$V_{t+1}(S_{t+1}) = \min_{\pi \in \Pi} \mathbb{E} \left\{ \sum_{t' = t+1}^{T} C(S_{t'}, X^\pi(S_{t'})) | S_{t+1} \right\}.$$

Since we typically cannot compute $V_{t+1}(S_{t+1})$ exactly, we replace it with a value function approximation $\hat{V}_t(S_t)$, in which case we would write our policy as

$$X_t^{VFA}(S_t) = \arg\max_{x_t \in X_t} \left( C(S_t, x_t) + \mathbb{E}\{\hat{V}_{t+1}(S_{t+1})|S_t\} \right).$$

Often it is easier to use the post-decision state $S^x_t$ (the state immediately after a decision has been made) which captures the entire lookahead term in equation (4). This allows us to write our policy without the imbedded expectation

$$X_t^{VFA}(S_t) = \arg\max_{x_t \in X_t} \left( C(S_t, x_t) + \hat{V}_t^x(S^x_t) \right).$$

Eliminating the expectation opens the door to solving problems where $x_t$ is high-dimensional (but only if $\hat{V}_t^x(S^x_t)$ is concave). Value function approximations have been widely studied under the umbrellas of approximate dynamic programming (see Powell (2011), and Bertsekas (2011)) and reinforcement learning (Sutton and Barto 1998). Specialized methods have evolved for handling convex problems that arise in multistage linear programming such as stochastic dual dynamic programming (SDDP) (see Pereira and Pinto (1991), Shapiro et al. (2014), Shapiro (2011), and Philpott and Guan (2008)) or piecewise linear, separable value functions (Powell et al. (2004), Topaloglu and Powell (2006)).

**Direct lookahead approximations (DLAs)** When the lookahead problem cannot be reasonably approximated by a value function, it is often necessary to turn to a direct lookahead approximation, where we replace the model with an approximation for the purpose of approximating the future. In this case our policy (4) can be written

$$X_t^{DLA}(S_t) = \arg\min_{x_t \in X_t} \left( C(S_t, x_t) + \hat{E} \left\{ \min_{\pi \in \Pi} \hat{E} \left\{ \sum_{t' = t+1}^{t+H} \hat{C}(\hat{S}_{t,t'}, \hat{X}^\pi(\hat{S}_{t,t'})) | \hat{S}_{t,t+1} \right\} | S_t, x_t \right\} \right).$$  (5)

Here, all variables (states and decisions) in the lookahead model are indicated with tilde’s, and are indexed by $t$ (the time at which the lookahead model is instantiated) and $t'$ (the
time period within the lookahead horizon). Lookahead models are typically characterized by five types of approximations: 1) the horizon, 2) the staging of information and decisions (multistage problems may be approximated by two-stages), 3) the outcome space (we may use a deterministic lookahead or a sampled stochastic), 4) discretization (of states, actions, and time periods), and 5) holding some information static that varies in the base model (a common assumption is to hold a forecast constant within the lookahead model). Lookahead models can take a variety of forms: deterministic lookahead models, also referred to as rolling horizon procedures (Sethi and Sorg 1991) or model predictive control (Camacho and Alba 2013), decision trees (which can be approximated using Monte Carlo tree search) for discrete actions, or stochastic programming models using scenario trees (see Birge and Louveaux (2011) and Donohue and Birge (2006)).

Policy search, whether we are using PFAs or CFAs, requires tuning parameters in our base objective function (2). By contrast, policies based on lookahead approximations depend on developing the best approximation of the future that can be handled computationally, although these still need to be evaluated using (2).

An often overlooked challenge is the presence of forecasts. These inherently require some form of lookahead, but are universally ignored when using value function approximations (the forecast would be part of the state variable). However, stochastic lookahead approximations in (5) are typically computationally very demanding.

In this paper, we are going to propose a hybrid comprised of a deterministic lookahead (a form of DLA), which is modified with parameters that have to be tuned using policy search (a form of CFA). This idea has been widely used in industry in an ad-hoc fashion without formal tuning of the parameters. We develop this idea in the context of a multistage linear program using the context of a complex, time-dependent energy storage problem with time-varying forecasts (the forecasts themselves evolve over time) of varying quality.

4 The Parametric Cost Function Approximation

We extend the concept of policy search to include parameterized optimization problems. The parametric Cost Function Approximation (CFA) draws on the structural simplicity of deterministic lookahead models and myopic policies, but allows more flexibility by adding tunable parameters. This puts this methodology in the same class as parametric policy function approximations widely used in the policy search literature, with the only difference that our parameterized functions are inside an optimization problem, making them more useful for high dimensional problems.

4.1 Basic Idea

Since the idea of a parametric cost function approximation is new, we begin by outlining the general strategy and then demonstrate how to apply it for our energy storage problem in Section 6. We propose using parameterized optimization problems such as

\[
X^\pi_t(S_t|\theta) = \arg\min_{x_t \in X^\pi(S_t)} \left\{ C(S_t, x_t) + \sum_{f \in F} \theta^c f(S_t, x_t) : A_t x_t = \bar{b}^c(\theta^b) \right\}
\]

(6)
as a type of parameterized policy. Here the index \(\pi\) signifies the structure of the modified set of constraints, \(\theta^c\) is the vector of cost function parameters, \(\theta^b\) is the vector of constraint parameters, and \(\phi_f\) are the basis functions corresponding to features \(f \in F\).
Parametric terms can be added to the cost function or constraints of a myopic or deterministic lookahead model. In the following example, parameters have been added as an error correction term to the objective function as well as to the model constraints, giving us

\[ X_t^\pi(S_t|\theta) = \arg\min_{x_t \in X_t} \left( C(S_t, x_t) + \sum_f \theta_f \phi_f(S_t, x_t) \right) \]  

subject to

\[ A_t x_t = b_t + D \theta, \]

where \( D \) is a scaling matrix. We emphasize that the cost correction term should not be confused as a value function approximation, because we make no attempt to approximate the downstream value of being in a state.

Whether the parameterizations are in the objective function, or in the constraints, the specification of a parametric CFA parallels the specification of any statistical model (or policy). The structure of the model is the “art” that draws on the knowledge and insights of the modeler. Finding the best CFA, which involves finding the best \( \theta \), is the science which draws on the power of classical search algorithms.

### 4.2 A hybrid Lookahead-CFA policy

There are many problems that naturally lend themselves to a lookahead policy (for example, to incorporate a forecast or to produce a plan over time), but where there is interest in making the policy more robust than a pure deterministic lookahead using point forecasts. For this important class (which is the problem we face), we can create a hybrid policy where a deterministic lookahead has parametric modifications that have to be tuned using policy search. When parameters are applied to the constraints it is possible to incorporate easily recognizable problem structure. For example, a supply chain management problem can handle uncertainty through buffer stocks, while an airline scheduling model might handle stochastic delays using schedule slack. A grid operator planning energy generation in the future might schedule reserve capacity to account for uncertainty in forecasts of demand, as well as energy from wind and solar. As with all policy search procedures, there is no guarantee that the resulting policy will be optimal unless the parameterized space of policies includes the optimal policy. However, we can find the optimal policy within the parameterized class, which may reflect operational limitations. We note that while parametric cost function approximations are widely used in industry, optimizing within the parametric class is not done.

### 4.3 Structure of the cost function approximation

Assume that a lookahead policy is given as

\[ X_t^{D-LA}(S_t) = \arg\min_{x_t, (\tilde{x}_{t,t'}, t' = t+1, \ldots, t+H)} \left( C(S_t, x_t) + \left[ \sum_{t' = t+1}^{t+H} \tilde{c}_{t,t'} \tilde{x}_{t,t'} \right] \right) \]

where \( \tilde{S}_{t,t'+1} = \tilde{S}^M(\tilde{S}_{t,t'}, \tilde{x}_{t,t'}, \tilde{W}_{t,t'+1}) \) and \( H \) is the size of the lookahead horizon. If the contribution function, transition function, and constraints of \( X^*(\cdot) \) are all linear, this policy can be expressed as the following linear program

\[ X_t^{D-LA}(S_t) = \arg\min_{x_t, \tilde{x}_{t,t'}, t' = t+1, t+H} c_t x_t + \tilde{c}_{t,t} \tilde{x}_t : A_t x_t \leq b_t, \tilde{A}_t x_t \leq \tilde{b}_t, \ x_t \geq 0, \]

where

\[ \tilde{S}_{t,t'+1} = \tilde{S}^M(\tilde{S}_{t,t'}, \tilde{x}_{t,t'}, \tilde{W}_{t,t'+1}) \]
where $\tilde{c}_t = \{\tilde{c}_{t,t'} : t' = t + 1, \ldots, t + H\}$, $\tilde{A}_t = \{\tilde{A}_{t,t'} : t' = t + 1, \ldots, t + H\}$, and $\tilde{b}_t = \{\tilde{b}_{t,t'} : t' = t + 1, \ldots, t + H\}$.

Parametric terms can be appended to the cost function and existing constraints, and new ones can be also added to the existing model. Often the problem setting will influence how the policy should be parameterized. In particular, there are different ways to parameterize the above-mentioned policy including parameterizing the cost vector, the matrix of coefficient, and the right hand side vector. In particular, assuming that all the uncertainty in our problem is restricted to the mentioned policy including parameterizing the cost vector, the matrix of coefficient, and the right hand side vector. In particular, there are different ways to parameterize the above-

ones can be also added to the existing model. Often the problem setting will influence how the becomes

Although it can greatly increase the parameter space, the upper and lower bounds can also depend on $t' - t$, as in

$\theta^L_{t,t'} \leq \theta^U_{t,t'}$ for $t' > t$. 

(11)

The resulting modified deterministic problem is no harder to solve than the original deterministic problem (where $\theta^L = 0$ and $\theta^U = R_{max}$). We now have to use stochastic search techniques to solve the policy search problem to optimize $\theta$.

There are also different policies for parameterizing the right hand side adjustment. A simple form is a lookup table indexed by time as in equation (12). Although it may be simple, a lookup table model for $\theta$ means that the dimensionality increases with the horizon which can complicate the policy search process. This type of parameterization is not limited to just modifying the point estimate of exogenous information. If the modeler has sufficient information such as the cumulative distribution function, one can even exchange the point estimate with the quantile function. The lookup table in time parameterization is best if the relationship between parameters in different periods is unknown.

Instead of having an adjustment $\theta_r = \theta^U_{t,t'}$ for each time $t + \tau$ in the future, one can use instead a parametric function of $\tau$, which reduces the number of parameters that we have to estimate. For example, we might use the parametric adjustment given by

$\theta^L_{t,t'} \cdot e^{\alpha \theta^L_{t,t'}} \leq \theta^U_{t,t'} \cdot e^{\beta \theta^U_{t,t'}}$ for $t' > t$ and $\alpha, \beta \in \mathbb{R}$. 

(13)

These parametric functions of time can also be used to directly modify the approximated future in the lookahead model.

We should point out that the parameterization scheme can be more general to better capture the underlying uncertainty in the model. In particular, one can also use the parameterized forms of $b(\theta(\sigma)), c(\theta(\sigma)), A(\theta(\sigma))$, where $\sigma$ represents an estimated variance of noise corresponding to the all sources of uncertainty. We let $\sigma = 0$ correspond to the case of having perfect information about the future over the horizon, which is the same as assuming that the base model is deterministic.
In this case, the parameterized model should satisfy $b_t(\theta(0)) = b_1$, $c_t(\theta(0)) = c_1$, $A_t(\theta(0)) = A_1$. An important example of parameterization would be affine with the general form of $b_t(\theta) = b_0 + \theta^t b_1$ (with similar modifications for the elements of the matrix $A$ and the cost vector $c$). If $\sigma = 0$, we would use $\theta_0^t = 0$ and $\theta_1^t = 1$. Finding the proper choice of parameterization is an art of modeling the problem which we believe that itself is an important line of research within the CFA approach.

5 Optimizing the parameters of the CFA model

To tune our parameterized policy in the CFA model, we need to solve the following optimization problem:

$$\max_{\theta \in \Theta} \left\{ F(\theta) := \mathbb{E}_\omega \left[ F(\theta, \omega) \right] = \mathbb{E} \left[ \sum_{t=0}^{T} C \left( S_t(\omega), X_t^\pi(S_t(\omega), \theta) \bigg| S_0 \right) \right] \right\},$$

(14)

where $S_{t+1}(\omega) = S^M(S_t(\omega), X_t^\pi(S_t(\omega)), W_{t+1}(\omega))$ denotes the transition function for every $\omega \in \Omega$. If $F(\theta)$ is well defined, finite valued, concave, and continuous at every $\theta$ in the nonempty, closed, and convex set $\Theta \subset \mathbb{R}^n$, then an optimal $\theta^* \in \Theta$ exists and can be found by stochastic approximation (SA) algorithms. However, when $F(\theta)$ is possibly nonconcave, SA-type methods can be modified to find stationary points of the above problem. In Subsection 5.1, we propose the classical SA algorithm discuss its convergence when noisy gradient of the objective function in (14) is available. We then provide a randomized SA algorithm in Subsection 5.1 which only requires noisy values of objective function.

5.1 Stochastic Gradient method for optimizing the CFA model

Our goal in this subsection is to solve problem (14) under specific assumptions on $F(\theta)$. Stochastic approximation algorithms require computing stochastic (sub)gradients of the objective function iteratively. Due to the special structure of $F(\theta)$, its (sub)gradient can be computed recursively under certain conditions as shown in the next result.

**Proposition 5.1** Assume $\bar{F}(\cdot, \omega)$ is convex/concave for every $\omega \in \Omega$, $\theta$ is an interior point of $\Theta$, and $\bar{F}(\cdot)$ is finite valued in the neighborhood of $\theta$, then

$$\nabla_\theta F(\theta) = \mathbb{E}[\nabla_\theta \bar{F}(\theta, \omega)]$$

where

$$\nabla_\theta \bar{F}(\theta) = \left( \frac{\partial C_0}{\partial X_0^\pi} \cdot \frac{\partial X_0^\pi}{\partial \theta} \right) + \sum_{t=1}^{T} \left[ \left( \frac{\partial C_t}{\partial S_t} \cdot \frac{\partial S_t}{\partial \theta} \right) + \left( \frac{\partial C_t}{\partial X_t^\pi} \cdot \left( \frac{\partial X_t^\pi}{\partial S_t} \cdot \frac{\partial S_t}{\partial \theta} + \frac{\partial X_t^\pi}{\partial \theta} \right) \right) \right],$$

(15)

and

$$\frac{\partial S_t}{\partial \theta} = \frac{\partial S_t}{\partial S_{t-1}} \cdot \frac{\partial S_{t-1}}{\partial \theta} + \frac{\partial S_t}{\partial X_{t-1}^\pi} \cdot \left[ \frac{\partial X_{t-1}^\pi}{\partial S_{t-1}} \cdot \frac{\partial S_{t-1}}{\partial \theta} + \frac{\partial X_{t-1}^\pi}{\partial \theta} \right],$$

where the $\omega$ is dropped for simplicity.

**Proof.** If $\bar{F}(\cdot, \omega)$ is convex or concave for every $\omega \in \Omega$, $\theta$ is an interior point of $\Theta$, and $\bar{F}(\cdot)$ is finite valued in the neighborhood of $\theta$, then by theorem 7.47 of Shapiro et al. (2009), we have

$$\nabla_\theta \left( \mathbb{E} \left[ \bar{F}(\theta, \omega) \right] \right) = \mathbb{E} \left[ \nabla_\theta \bar{F}(\theta, \omega) \right].$$
Applying the chain rule, we find
\[
\nabla_\theta \bar{F}(\theta) = \frac{d}{d\theta} \left[ C_0(S_0, X^\pi_0) + \sum_{t=1}^T C(S_t, X^\pi_t) \right] = \left( \frac{\partial C_0}{\partial S_0} \frac{dX^\pi_0}{d\theta} \right) + \left[ \sum_{t=1}^T \frac{d}{d\theta} C(S_t, X^\pi_t) \right]
\]
\[
= \left( \frac{\partial C_0}{\partial S_0} \cdot \frac{dX^\pi_0}{d\theta} \right) + \sum_{t=1}^T \left[ \left( \frac{\partial C_t}{\partial S_t} \cdot \frac{dS_t}{d\theta} \right) + \left( \frac{\partial C_t}{\partial X^\pi_t} \cdot \frac{dX^\pi_t}{d\theta} \right) \right] \] ,
\[
\text{where}
\]
\[
\frac{\partial S_t}{\partial \theta} = \frac{\partial S_t}{\partial S_{t-1}} \cdot \frac{\partial S_{t-1}}{\partial \theta} + \frac{\partial S_t}{\partial X^\pi_{t-1}} \cdot \left[ \frac{\partial X^\pi_{t-1}}{\partial S_{t-1}} \cdot \frac{\partial S_{t-1}}{\partial \theta} + \frac{\partial X^\pi_{t-1}}{\partial \theta} \right] .
\]

Note that if \( \bar{F}(\theta) \) is not differentiable, then its subgradient can be still computed using (15). However, when \( \bar{F}(\theta) \) is not concave (convex), its subgradient may not exist and the concept of generalized subgradient should be employed.

If \( \nabla_\theta \bar{F}(\theta, \omega) \) exists for every \( \omega \in \Omega \), the ability to calculate its unbiased estimator allows us to use stochastic approximation techniques to determine the optimal parameters, \( \theta^* \), of the CFA policy model. Below is an iterative SA algorithm for optimizing the CFA model.

Algorithm 1 The stochastic gradient method for the CFA model (SG-CFA)

1: Input: \( \theta^0, N \).
2: For \( n = 1, 2, \ldots, N \):
   2.1: Generate a trajectory \( \omega^n \) where
   \[
   S^n_{t+1}(\omega^n) = S^M(S^n_t(\omega^n), X^n_t(S^n_t(\omega^n), \theta_n, \omega^n)), W_{t+1}(\omega^n) \quad t = 0, 1, \ldots, T - 1,
   \]
   2.2: Compute the gradient estimator using equation (15).
   2.3: Update policy parameters as
   \[
   \theta^n = \theta^{n-1} + \alpha_n \nabla_\theta \bar{F}(\theta^{n-1}, \omega^n)|_{\theta=\theta^{n-1}}
   \]
3: End For

For the convergence of the above algorithm, we need to assume that at the \( n \)-th iteration, (see Robbins and Monro (1951)) the computed stochastic subgradient, \( g^n \) satisfies the following condition:
\[
\text{a) } \mathbb{E} \left[ g^{n+1} \cdot (\bar{\theta}^n - \theta^*) \right] = 0,
\]
\[
\text{b) } |g^n| \leq B_g,
\]
\[
\text{c) For any } \theta \text{ where } |\theta - \theta^*| > \delta, \delta > 0, \text{ there exists } \epsilon > 0 \text{ such that } \mathbb{E}[g^{n+1}|F^n] > \epsilon.
\]

Furthermore, we need to assume that stepsizes \( \alpha_n \) satisfy
\[
\alpha_n > 0, \text{ a.s.}
\]

11
\[
\sum_{n=0}^{\infty} \alpha_n = \infty, \text{ a.s.} \quad (18)
\]
\[
\sum_{n=0}^{\infty} \alpha_n^2 < \infty. \quad (19)
\]

If the above conditions hold, \( F(\cdot) \) is continuous and finite valued in the neighborhood of every \( \theta \), in the nonempty, closed, bounded, and convex set \( \Theta \subseteq \mathbb{R}^d \) such that \( \bar{F}(\cdot, \omega) \) is concave for every \( \omega \in \Omega \) where \( \theta \) is an interior point of \( \Theta \), then
\[
\lim_{n \to \infty} \theta^n \to \theta^* \text{ a.s.}
\]

Although any stepsize rule that satisfies the previous conditions will guarantee asymptotic convergence, we prefer parameterized rules that can be tuned for quicker convergence rates. Therefore, we limit our evaluation of the algorithm to how well it does within \( N \) iterations. The CFA Algorithm can be described as a policy, \( \theta^\pi(S^n) \), with a state variable, \( S^n = \theta^n \) plus any parameters needed to compute the stepsize policy, and where \( \pi \) describes the structure of the stepsize rule. If \( \theta^\pi,n \) is the estimate of \( \theta \) using stepsize rule \( \pi \) after \( n \) iterations, then our goal is to find the rule that produces the best performance (in expectation) after we have exhausted our budget of \( N \) iterations. Thus, we wish to solve
\[
\max_{\pi} \mathbb{E} \left[ \bar{F}(\theta^\pi,N, \omega) \right],
\]

i.e., finding the best stepsize rule that maximizes the terminal value within \( N \) iterations. For our numerical experiments, we use two well-known stepsize policies, namely, the adaptive gradient algorithm (AdaGrad) Duchi et al. (2011) and the Root Mean Square Propagation (RMSProp) Tieleman and Hinton (2012). AdaGrad modifies the individual stepsize for the updated parameter, \( \theta \), based on previously observed stochastic gradients using
\[
\alpha_n = \frac{\eta}{\sqrt{G_n} + \epsilon} \quad (21)
\]
where \( \eta \) is a scalar learning rate, \( G_n \in \mathbb{R}^{d \times d} \) is a diagonal matrix where each diagonal element is the sum of the squares of the stochastic gradients with respect to \( \theta \) up to the current iteration \( n \), and \( \epsilon \) is a smoothing term that avoids division by zero. RMSProp uses a running average of previously observed stochastic gradients to scale the current stepsize as
\[
\alpha_n = \frac{\eta}{\sqrt{\bar{g}_n}}, \quad \bar{g}_n = \beta \bar{g}_{n-1} + (1 - \beta) \| g_n \|^2,
\]
where \( \eta \) is the learning rate, \( \beta \in (0,1) \) is the running weight, and \( g_n \) is the observed stochastic gradient at time \( n \).

Note that \( F(\theta) \) can be generally nonsmooth and nonconvex and hence, its subgradient does not exist everywhere. While one can define generalized subgradients for this nonsmooth, nonconvex function, one can also define a smooth approximation of \( F(\theta) \) and then try to apply a stochastic approximation algorithm to maximize this function. We pursue this idea in the next subsection.

### 5.2 Stochastic Gradient-free method for optimizing the CFA model

As mentioned in the previous subsection, gradient of \( F(\theta) \) in (15) is only computable under restricted conditions. However, noisy values of \( F(\theta) \) can be obtained through simulation. This motivates using techniques from simulation-based optimization where even shape of the function
may not be known (see e.g., Fu (2015) and references therein). In this subsection, we provide a zeroth-order SA algorithm and establish its finite-time convergence analysis to solve problem (14). For simplicity, we allow $\theta$ to take arbitrary values i.e, $\Theta = \mathbb{R}^d$ throughout this subsection.

A smooth approximation of the function $F(\theta)$ can be defined as the following convolution

$$F_\eta(\theta) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int F(\theta + \eta v) e^{-\frac{1}{2}\|v\|^2} \, dv = \mathbb{E}_v[F(\theta + \eta v)].$$  \hspace{1cm} (22)

where $\eta > 0$ is the smoothing parameter and $v \in \mathbb{R}^d$ is a Gaussian random vector whose mean is zero and covariance is the identity matrix.

Nesterov and Spokoiny (2017) provide the following result about properties of $F_\eta(\cdot)$.

\textbf{Lemma 5.1} The following statements hold for any Lipschitz continuous function $F$ with constant $L_0$.

a) The function $F_\eta$ is differentiable and its gradient is given by

$$\nabla F_\eta(\theta) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int \frac{F(\theta + \eta v) - F(\theta)}{\eta} v e^{-\frac{1}{2}\|v\|^2} \, dv,$$  \hspace{1cm} (23)

b) The gradient of $F_\eta$ is Lipschitz continuous with constant $L_\eta = \sqrt{d} \eta L_0$.

c) For any $\theta \in \mathbb{R}^d$,

$$|F_\eta(\theta) - F(\theta)| \leq \eta L_0 \sqrt{d},$$  \hspace{1cm} (24)

$$\mathbb{E}_v[\|F(\theta + \eta v) - F(\theta)\|_2^2] \leq \eta^2 L_0^2 (d + 4)^2.$$  \hspace{1cm} (25)

We now present a SA-type algorithm which only uses noisy values of $F$ to solve problem (14).

\textbf{Algorithm 2} The stochastic gradient-free method for the CFA model (SGF-CFA)

1: Input: $\theta^0 \in \mathbb{R}^d$, an iteration limit $N$, a positive sequence convergent to zero $\{\eta_k\}_{k \geq 1}$, and a probability mass function $P_R(\cdot)$ supported on $\{1, \ldots, N\}$ given by

$$P_R(R = k) = \frac{\alpha_k}{\sum_{k' = 1}^N \alpha_{k'}} \quad k = 1, \ldots, N.$$  \hspace{1cm} (26)

For $k = 1, \ldots, N$:

2: Generate a trajectory $\omega^k$ where $S_{t+1}^k(\omega^k) = S_t^M(S_t^k(\omega^k), X_t^\pi(S_t^k(\omega^k)|\theta^{k-1}), W_{t+1}(\omega^k))$, and a random Gaussian vector $v_k$ to compute the gradient estimator as

$$G_{\eta_k}(\theta^{k-1}, \omega^k) = \frac{F(\theta^{k-1} + \eta_k v_k, \omega^k) - F(\theta^{k-1}, \omega^k)}{\eta_k} v_k.$$  \hspace{1cm} (27)

3: Update policy parameters

$$\theta^k = \theta^{k-1} + \alpha_k G_{\eta_k}(\theta^{k-1}, \omega^k).$$  \hspace{1cm} (28)

End For

4: Generate a random index $R$ according to $P_R$ and output $\theta^R$. 

13
Theorem 5.1

Let $H$.

Hence, we now provide its main convergence property.

Algorithm 2 for solving problem (14). It should be mentioned that the framework of this algorithm has been first proposed in Nesterov and Spokoiny (2017) and then widely used in the literature of stochastic optimization using zeroth-order information (see e.g., Ghadimi and Lan (2013)). However, our choice of stepsize policy and using adaptive smoothing parameter makes Algorithm 2 different from its existing variants.

The algorithm uses an idea first proposed by Ghadimi and Lan (2013) of choosing randomly from the set $\{\theta_1, \ldots, \theta_k, \ldots, \theta_N\}$, instead of using the ending value $\theta^N$. This step is needed to establish the rate of convergence result like (33) for SA-type algorithms when applied to nonconvex stochastic optimization problems. This kind of randomization scheme seems to be the only way for this purpose since $\min_{k=1,2,\ldots} \|\nabla F(\theta_k)\|^2$ is not computable for this class of problems. Note that the probability mass function of the random index of the output solution depends on the choice of stepsize policy which will be discussed later in this subsection.

Since Algorithm 2 uses adaptive smoothing parameter, its convergence analysis is slightly different than the one presented in Nesterov and Spokoiny (2017) when $F$ is nonsmooth and nonconvex. Hence, we now provide its main convergence property.

**Theorem 5.1** Let $\{\theta_k\}$ be generated by Algorithm 2 and $\bar{F}(\theta)$ be Lipschitz continuous with constant $L_0$. If $F(\theta)$ is bounded above by $F^*$, we have

$$
\mathbb{E}[\|\nabla F_{\eta R}(\theta^R)\|^2] \leq \frac{1}{\sum_{k=1}^{N} \alpha_k} \left[ F^* - F(\theta^0) + L_0 \sqrt{d} \left( \eta_1 + \eta_N + \sum_{k=1}^{N-1} |\eta_k - \eta_{k+1}| + L_0^2 (d + 4)^2 \sum_{k=1}^{N} \frac{\alpha_k^2}{2\eta_k} \right) \right],
$$

where the expectation is taken w.r.t the randomness arises from the nature of the problem $\omega$, and the ones imposed by the algorithm, namely, Gaussian random vector $v$ and random integer number $R$ whose probability distribution is given by (26).

**Proof.** First note that $F(\theta)$ is Lipschitz continuous with constant $L$ due to the same assumption on $\bar{F}(\theta)$. Hence, the gradient of $F_\eta(\theta)$ is Lipschitz continuous with constant $L_\eta$ due to Theorem 5.1.b which together with (28) imply that

$$
F_{\eta k}(\theta^k) \geq F_{\eta k}(\theta^{k-1}) + \langle \nabla F_{\eta k}(\theta^k), \theta^k - \theta^{k-1} \rangle - \frac{L_{\eta k}}{2} \|\theta^k - \theta^{k-1}\|^2 \\
= F_{\eta k}(\theta^{k-1}) + \alpha_k \langle \nabla F_{\eta k}(\theta^k), G_{\eta k}(\theta^{k-1}, \omega^k) \rangle - \frac{L_{\eta k} \alpha_k^2}{2} \|G_{\eta k}(\theta^{k-1}, \omega^k)\|^2.
$$

Taking expectations of both sides of the above inequality, noting (25), (27), and (29), we obtain

$$
\mathbb{E}[F_{\eta k}(\theta^k)] \geq F_{\eta k}(\theta^{k-1}) + \alpha_k \|\nabla F_{\eta k}(\theta^k)\|^2 - \frac{L_0^3 \sqrt{d} (d + 4)^2 \alpha_k^2}{2\eta_k}.
$$
Summing up both sides of the above inequality and re-arranging the terms, we have
\[ \sum_{k=1}^{N} \alpha_k \mathbb{E}[\|\nabla F_{\eta_k}(\theta^k)\|^2] \leq \Delta_N + L_0^2 \sqrt{d}(d + 4)^2 \sum_{k=1}^{N} \frac{\alpha_k^2}{2 \eta_k}, \]  
(31)

where \( \Delta_N = F_{\eta_N}(\theta^N) - F_{\eta_1}(\theta^0) + \sum_{k=1}^{N-1} [F_{\eta_k}(\theta^k) - F_{\eta_{k+1}}(\theta^k)] \). Noting (24), the fact that \( F(\theta) \leq F^* \) for any \( \theta \in \mathbb{R}^d \), (22), and Lipschitz continuity of \( F \), we have
\[ F_{\eta_N}(\theta^N) - F_{\eta_1}(\theta^0) \leq F^* - F(\theta^0) + (\eta_1 + \eta_N) L_0 \sqrt{d}, \]
\[ F_{\eta_k}(\theta^k) - F_{\eta_{k+1}}(\theta^k) = \mathbb{E}_v[F(\theta^k + \eta_k v) - F(\theta^k + \eta_{k+1} v)] \leq L_0 |\eta_k - \eta_{k+1}| \mathbb{E}[|v|] \leq L_0 \sqrt{d} |\eta_k - \eta_{k+1}|. \]

Combining (31) with the above two relations, (30) follows by noting that in the view of (26) we have
\[ \mathbb{E}[\|\nabla F_{\eta_R}(\theta^R)\|^2] = \frac{\sum_{k=1}^{N} \alpha_k \mathbb{E}[\|\nabla F_{\eta_k}(\theta^k)\|^2]}{\sum_{k=1}^{N} \alpha_k}. \]

Corollary 5.1 Let the smoothing parameters and stepsizes of Algorithm 1 are chosen as
\[ \eta_k = \frac{L_0(d + 4)}{k^\beta}, \quad \alpha_k = \frac{1}{\sqrt{k}}, \quad k = 1, \ldots, N, \]
(32)
for any \( \beta \in (0, \frac{1}{2}) \). Then we have
\[ \mathbb{E}[\|\nabla F_{\eta_R}(\theta^R)\|^2] \leq \frac{1}{2 \sqrt{N}} \left[ F^* - F(\theta^0) + L_0^2 \sqrt{d}(d + 4) \left( 2 + \frac{(N + 1)^{\beta}}{\beta} \right) \right], \]
(33)

where the probability distribution of \( R \) is given in (26).

Proof. Note that by (32), we have
\[ \sum_{k=1}^{N-1} |\eta_k - \eta_{k+1}| = \sum_{k=1}^{N-1} (\eta_k - \eta_{k+1}) = \eta_1 - \eta_N, \]
\[ \sum_{k=1}^{N} \alpha_k \geq 2 \sqrt{N}, \quad \sum_{k=1}^{N} \frac{\alpha_k^2}{\eta_k} = \frac{1}{L_0(d + 4)} \sum_{k=1}^{N} k^{\beta - 1} \leq \frac{(N + 1)^{\beta}}{\beta L_0(d + 4)}, \]

which, with (30), clearly imply (33).

It should be mentioned that if a fixed smoothing parameter \( \eta_k = \eta \) and a fixed stepsize \( \alpha_k = \alpha \) are properly employed, then the upper bound would be on the order of \( 1/\sqrt{N} \). This rate has been obtained in Nesterov and Spokoiny (2017) for the weighted average of \( \mathbb{E}[\|\nabla F_{\eta}(\theta^k)\|^2] \) without introducing the random index \( R \). However, to ensure convergence to the stationary point of the original problem, one should allow the smoothing parameter \( \eta_k \) and stepsize \( \alpha_k \) converge to 0. Therefore, we think choosing these two quantities adaptively converging to 0 would be more desirable in practice than setting them to very small numbers at the starting point.
5.3 The linear CFA model

Our goal in this subsection is to specialize some results from the previous subsection and provide more properties for the linear CFA model. In particular, if the objective function in (14) is a linear function of the decisions, $x_t$, the parametric CFA policy, $X_t^\pi(S_t|\theta)$, which determines the decision, $x_t$, can be written as the following linear program

$$X_t^\pi(S_t|\theta) = x_t^*,$$

$$\tilde{x}_t = [x_t^*, \ldots, \tilde{x}_{t,T}^*] = \arg \max_{x_t, (\tilde{x}_{t,t'}), t'=t+1, \ldots, T} c_t x_t + \sum_{t'=t+1}^{T} \tilde{c}_{t,t'} \tilde{x}_{t,t'}, \quad \tilde{T} = \min(t+H, T),$$

where $\tilde{A}_t \tilde{x}_t \leq \tilde{b}_t(\theta)$ for given $\tilde{A}_t$ and $\tilde{b}_t$. The state variable, $S_t$, includes the point estimates, $(\tilde{W}_{t,t'})_{t'=t+1, \ldots, T}$, that are used to approximate exogenous information. If this policy is written as a linear program where the state and approximated exogenous information is only in the right hand side constraints, $\tilde{b}_t(\theta)$, then Proposition 5.1 for computing a stochastic subgradient of $F(\theta)$ can be simplified as follows.

**Proposition 5.2** Let $\tilde{F}(\theta, \omega)$ be concave in $\theta$ for every $\omega \in \Omega$, $\theta$ be an interior point of $\Theta$, and the contribution function $C(x)$ be a linear function of $x$, and the transition function $S_t = S^M(S_{t-1}, x_{t-1}, W_t)$ be linear in $S_{t-1}$ and $x_{t-1}$. Moreover, assume that $F(\theta)$ is finite valued in the neighborhood of $\theta$, and the policy, $X_t^\pi(S_t|\theta)$ is given by (34) in which $B_t$ is the basis matrix corresponding to the basic variables for the optimal solution. Then

$$\nabla_{\theta} \tilde{F}(\theta, \omega) = \sum_{t=1}^{T} \left( \nabla_{\theta} \tilde{b}_t(\theta) + \nabla_S \tilde{b}_t(\theta) \cdot \nabla_{\theta} S_t \right) T \cdot \left( B_t^{-1} \right)^T \cdot c_t,$$

where

$$\nabla_{\theta} S_t = \nabla S_{t-1} S^M(S_{t-1}, x_{t-1}, W_t) \cdot \nabla_{\theta} S_{t-1} + \nabla x_{t-1} S^M(S_{t-1}, x_{t-1}, W_t) \cdot \nabla_{\theta} x_{t-1}.$$  

**Proof.** Proof follows from the proof of Proposition 5.1 and the basic properties of the linear program. Hence, we skip the details. ■

For the linear CFA model, we can also provide some properties of the objective function as follows.

**Lemma 5.2** Assume that problem (14) belongs to a class of linear programs such that

$$C_t(S_t(\omega), X_t^\pi(S_t, \theta)) = c_t(\omega) X_t^\pi(S_t, \theta),$$

where $X_t^\pi(S_t, \theta)$ is set to (34). Further assume that the transition function $S_t = S^M(S_{t-1}, x_{t-1}, W_t)$ is linear in $S_{t-1}$ and $x_{t-1}$, and $\tilde{b}_t$ is also linear in $\theta$ and $S_t$. Moreover, let $\Theta_t(\omega)$ be the largest convex subset of $\Theta_t(\omega)$ such that the optimal basis corresponding to $\tilde{x}_t^*$ remains the same for any $\theta$ belonging to this subset with the definition of $\Theta_t(\omega) := \mathbb{R}^d$. Then $F(\theta)$, defined in (14), is linear in $\omega$ over $\Theta := \bigcap_{\omega \in \Omega} \Theta_T(\omega)$.

**Proof.** First note that by the definition of $\Theta_t(\omega)$, for a fixed $\omega \in \Omega$ we have $\Theta_T \subseteq \Theta_{T-1} \subseteq \cdots \Theta_0 \subseteq \mathbb{R}^d$. Noting that $S_0$ does not depend on $\theta$ and $\tilde{b}_0(\theta, S_0)$ is linear in $\theta$, we conclude that $x_0^*$ is linear in $\theta$ for any $\theta \in \Theta_0(\omega)$. This also implies that $S_1$ is linear in $\theta$. Hence, $\tilde{b}_1(\theta, S_1)$ is also linear in $\theta$ for any $\theta \in \Theta_0(\omega)$ implying that $x_0^*$ is linear in $\theta$ for any $\theta \in \Theta_0(\omega)$ and $x_0^*$ is also linear in $\theta$ for any $t \geq 2$. Therefore, taking the intersection of $\bigcap_{t=0, \ldots, T} \Theta_t(\omega) = \Theta_T(\omega)$ over all $\omega \in \Omega$, we conclude that $F(\theta)$ is linear for all $\theta$ belonging to this intersection set. ■

A more general form of the above Lemma can be stated as follows.
Lemma 5.3  Consider problem (14) together with (37) and (34) such that only one of the right hand side of the constraints (other than inventory ones), say the i-th constraint, is parameterized with a $\theta$ for all periods. Further assume that the transition function $S_t = S^M(S_{t-1},x_{t-1},\omega_t)$ is linear in $S_{t-1}$ and $x_{t-1}$, and $b_i^t$ is also linear in $\theta$ and $S_t$. Assuming that an interval $[a^l,a^u] \subset \mathbb{R}$ is given for the range of $\theta$, one can partition $[a^l,a^u]$ into subintervals such that $F(\theta)$ is a piecewise linear concave function of $\theta$ on each of these subintervals.

Proof. Assume that the linear programs in (34) are solved for all $t = 0, 1, \ldots, T$ with the choice of $\theta = a^u$. Moreover, assume that $[a^l_0,a^u]$ be the subinterval that the optimal basis for the linear program in (34) corresponding to $t = 0$ does not change for any $\theta$ belonging to this subinterval. Hence, $X^\pi_0(S_0,\theta)$ is a linear function of $\theta$ on this subinterval. This also implies that the inventory constraint for the linear program of $t = 1$ are also linear functions of $\theta$ on the aforementioned subinterval. Hence, there exists $a^l_1 \geq a^l_0$ such that the optimal basis for the linear program in (34) corresponding to $t = 1$ does not change for any $\theta \in [a^l_1,a^u]$. Moving forward with this argument we obtain the non-decreasing sequence $\{a^l_t\}_{t=0,1,\ldots,T-1}$ such that the solution of the $t$-th linear program is a linear function of $\theta$ on the subinterval $[a^l_t,a^u]$. Consequently, the right hand side of the $T$-th linear program including the inventory constraint and the i-th constrains will be a linear function for all $\theta \in [a^l_1,a^u]$. Hence, the optimal solution of this linear program is a piecewise linear concave function for any $\theta \in [a^l_1,a^u]$ and so is $F(\theta)$.

Repeating the above argument for the interval $[a^l_0,a^l_1]$, we obtain a new subinterval over which $F(\theta)$ is a piecewise linear concave function of $\theta$. This process can be continued till covering the whole initial interval $[a^l,a^u]$.

Note that the result of the above lemma can be extended when more than one constraint is linearly parameterized by $\theta$. In this case, instead of subintervals, we have subsets of the parameter space over which $F(\theta)$ is a piecewise linear concave function of $\theta$.

Our last result in this subsection provides the optimal policy for the special case of having perfect information about the future.

Lemma 5.4  Assume that $H = T - 1$ and we are given perfect information for all sources of uncertainty. Then the optimal policy is not to parameterize the model i.e., $\theta^* = \theta(0)$.

Proof. Given perfect forecasts for the case $H = T - 1$ means that the forecast is no longer rolling over the horizon and is fixed. Hence the optimal solution of the linear program solved at $t = 0$ is also optimal for all linear programs solved over the horizon.

5.4 The Static CFA model

In this subsection, we propose an alternative approach for solving the base model. In particular, we consider a static variant of problem (14) given by

$$\max_{\theta \in \Theta} \left\{ F^S(\theta) := \mathbb{E} \left[ \bar{F}^S(\theta) = \sum_{t=0}^{T} C(X_t^\pi(\theta)) \right] \right\}, \quad (38)$$

where

$$[X^\pi_0(\theta),X^\pi_1(\theta),\ldots,X^\pi_T(\theta)]^T = \operatorname{argmax}_{x_0,x_1,\ldots,x_T} \sum_{t=0}^{T} c_t x_t,$$

subject to $A_t x_t \leq b_t(\theta) \quad t = 0, 1, \ldots, T. \quad (39)$
Indeed, to evaluate the objective function of the model for a given $\theta$, we only solve one linear program at time $t = 0$. In this case to better capture forecast changes for all periods, we allow $b_t$ to be any concave function of $\theta$. We also allow each period to have its own parameterization independent of other periods. The next result provides conditions where problem (38) is a convex programming problem.

**Lemma 5.5** Let $F^S(\theta)$ and $X^\pi$ be defined in (38) and (39), respectively. Also assume that inventory constraints are not parameterized by $\theta$ and $b_t(\theta)$ is concave in $\theta$ for any $t \in \{0,1,\ldots,T\}$. Then $F^S(\theta)$ is concave in $\theta$.

**Proof.** Note that if inventory constraints are not parameterized by $\theta$ and $b_t(\theta)$ is concave in $\theta$ for any $t \in \{0,1,\ldots,T\}$, the feasible set of problem (39) is clearly convex in $(x,\theta)$. Hence, since the objective function is only linear (concave) in $x$, the optimal value of (39), which is indeed $F^S(\bar{\theta},\omega)$, is also concave in $\theta$. Therefore, $F^S(\theta) = \mathbb{E}[F^S(\bar{\theta},\omega)]$ is clearly a concave function of $\theta$. $lacksquare$

Note that in the above lemma, we only need the parametrization of constraints (except the inventory ones) be concave. In this case, subgradients of $F^S(\theta)$ are available everywhere and one can use the standard stochastic approximation algorithms for convex programming such as Algorithm 1 to solve problem (38) with convergence guarantees. We should also point out that to easily compute subgradients of $F^S(\theta)$, we also assume that $b_t(\theta)$ is differentiable in $\theta$. Moreover, under the aforementioned static setting, we can significantly simplify the subgradient computations since the right-hand-side of the constraints do not depend on the state variables. In particular, (35) is reduced to

$$\nabla_\theta F^S(\theta,\omega) = \sum_{t=0}^T \nabla_\theta b_t(\theta) \mathbf{\top} \cdot \left(B_t^{-1}\right) \mathbf{\top} \cdot c_t. \quad (40)$$

### 6 An Energy Storage Application

In this section, we use the setting of an energy storage application to show how we can use a parametric CFA to produce robust policies using rolling forecasts of varying quality. In Subsection 6.1, we introduce the energy storage problem and its associated optimization problem. We then provide a lookahead approximation model in Subsection 6.2 and discuss different ways of policy parameterization to improve the performance of the approximation model.

#### 6.1 Problem description

Consider a smart grid manager who must satisfy a recurring power demand with a stochastic supply of renewable energy, limited supply of energy from the main power grid at a stochastic price, and access to a local rechargeable storage devices. This system is illustrated in Figure 1.

Every hour the manager must determine what combination of energy sources to use to satisfy the power demand, how much energy to store, and how much to sell back to the grid. The state variable at time $t$, $S_t$, includes the level of energy in storage, $R_t$, the amount of energy available from wind and its forecast, $\{f_{E_{t'}}\}_{t'\geq t}$, the spot price of electricity from the grid and its forecast, $\{f_{P_{g_{t'}}}\}_{t'\geq t}$, the market price of electricity $P_d$, the demand $D_t$ and its forecast $\{f_{D_{t'}}\}_{t'\geq t}$, and the energy available from the grid $G_t$ at time $t$. Hence the state of the system can be represented by the following vector,

$$S_t = (R_t, f_{E_{t'}}, P_{t'}, f_{P_{g_{t'}}}, f_{D_{t'}}) \quad \forall t' \geq t, \quad (41)$$
where $R_t \in [0, R_{\text{max}}]$ is the level of energy in storage at time $t$. In the Appendix, we describe how these forecasts can be generated.

At the beginning of every period $t$ the manager must combine energy from the following sources to satisfy the demand, $D_t$:

- Energy currently in storage (represented by a decision $x_{rd}^t$);
- Newly available wind energy (represented by a decision $x_{wd}^t$);
- Energy from the grid (represented by a decision $x_{gd}^t$).

Additionally, the manager must decide how much renewable energy to store, $x_{wr}^t$, how much energy to sell to the grid at price $P_t$, $x_{rg}^t$, and how much energy to buy from the grid and store, $x_{gr}^t$. Hence, the manager’s decision variable at $t$ is defined as the following vector

$$x_t = (x_{wd}^t, x_{gd}^t, x_{rd}^t, x_{wr}^t, x_{gr}^t, x_{rg}^t)^T \geq 0,$$  

which should satisfy the following constraints:

$$x_{wd}^t + \beta_d x_{rd}^t + x_{gd}^t \leq D_t,$$  
$$x_{rd}^t + x_{rg}^t \leq R_t,$$  
$$x_{wr}^t + x_{gr}^t - x_{rd}^t - x_{rg}^t \leq R_{\text{max}} - R_t,$$  
$$x_{wd}^t + x_{gd}^t \leq E_t,$$  
$$x_{rd}^t + x_{rg}^t \leq \gamma_c,$$  
$$x_{gr}^t + x_{rg}^t \leq \gamma_d,$$

where $\gamma_c$ and $\gamma_d$ are the maximum amount of energy that can be charged or discharged from the storage device. Typically, $\gamma_c$ and $\gamma_d$ are the same.

The transition function, $S^M(\cdot)$, explicitly describes the relationship between the state of the model at time $t$ and $t+1$ such that

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

where $W_{t+1} = (E_{t+1}, P_{t+1}, D_{t+1})$ is the exogenous information revealed at $t+1$. In our numerical experiments, we assumed that $W_{t+1}$ is independent of $S_t$, but the CFA algorithm can work with any sample path provided by an exogenous source. Indeed, the CFA method belongs to the class of *data driven* algorithms, where we do not need a model of the exogenous process.
The relationship of storage levels between periods is defined as a linear function of decision variables:

\[ R_{t+1} = R_t - x_t^{rd} + \beta^c x_t^{wr} + \beta^c x_t^{gr} - x_t^{rg}, \]  

(49)

where \( \beta^c \in (0, 1) \) and \( \beta^d \in (0, 1) \), are the charge and discharge efficiencies. For a given state \( S_t \) and decision \( x_t \), the profit realized at \( t \) is given by

\[
C(S_t, x_t) = P^m_t \cdot (x_t^{wd} + \beta^d x_t^{rd} + x_t^{gd}) + P^g_t \cdot (\beta^d x_t^{rg} - x_t^{gr} - x_t^{gd}) - C_{\text{penalty}} \cdot \left( D_t - x_t^{wd} - \beta^d x_t^{rd} - x_t^{gd} \right),
\]

(50)

where \( C_{\text{penalty}} \) is the penalty of not satisfying the demand. Our goal is to find the policy \( \pi \) that solves

\[
\max_{\theta} \mathbb{E} \left[ \sum_{t=0}^{T} C(S_t, X^\pi_t(S_t|\theta)) \bigg| S_0 \right] \quad \text{where} \quad S_{t+1} = S^M(S_t, X^\pi_t(S_t|\theta), W_{t+1}) \quad \forall t \in [0, T],
\]

(51)

and \( X^\pi_t(S_t|\theta) \) is given by (6).

### 6.2 Policy Parameterizations

If the contribution function, transition function and constraints are linear, a deterministic lookahead policy can be constructed as a linear program if point forecasts of exogenous information are provided. For our deterministic lookahead, by noting (50), we solve subproblem (8) subject to constraints (43) - (49) for \( t' \in [t + 1, t + H] \). We call this deterministic lookahead policy the benchmark policy, and use it to estimate the degree to which the parameterized policies are able to improve the results in the presence of uncertainty.

There are different ways of parameterizing the policy in this lookahead model. A few examples are as follows.

- **Capacity Constraints**: This parameterization limits the amount of energy in storage and guarantees there is capacity to purchase inexpensive energy. An upper bound constraint is easily created by multiplying the capacity of the storage device, \( R_{\max} \) by the parameter \( \theta_{t', t} \). This changes (45) to

\[
x_{t', t}^{wr} + x_{t', t}^{gr} \leq R_{\max} \cdot \theta_{t', t}^U - f_{t, t'}^R,
\]

(52)

where \( \theta_{t'} \in [0, 1] \) and \( t' \in [t, t + H] \). Parameterized lower constraints are incorporated into the policy by creating the additional linear constraints

\[
-x_{t'}^{rd} - x_{t'}^{rg} + R_t \geq R_{\max} \cdot \theta_{t'}^L
\]

(53)

where \( \theta_{t'}^L \in [0, 1] \) and \( t' \in [t + 1, t + H] \).

- **Constant forecast parameterization** - Instead of using a unique parameter for every period, this parameterization uses a single scalar to modify the forecast amount of renewable energy for the entire horizon. The policy constraints (46) are changed to

\[
x_{t, t'}^{wr} + x_{t, t'}^{wd} \leq \theta \cdot f_{t, t'}^E.
\]

(54)

- **Lookup table forecast parameterization** - Overestimating or underestimating forecasts of renewable energy influences how aggressively a policy will store energy. We modify the forecast of renewable energy for each period of the lookahead model with a unique parameter
This parameterization is a lookup table representation because there is a different \( \theta \) for each lookahead period, \( \tau = 0, 1, 2, ... \). This changes (46) to

\[
\bar{x}_{t,t'}^{wr} + x_{t,t'}^{wd} \leq f_{t,t'}^E \cdot \theta_{t'-t}.
\]  

(55)

where \( t' \in [t+1, t+H] \) and \( \tau = t' - t \). If \( \theta_{\tau} < 1 \) the policy will be more conservative and decrease the risk of running out of energy. Conversely, if \( \theta_{\tau} > 1 \) the policy will be more aggressive and less adamant about maintaining large energy reserves.

7 Numerical Results

To demonstrate the capability of the CFA approach and our proposed algorithms, we test the aforementioned parameterizations of the deterministic lookahead policy defined by equation (8) on variations of the previously described energy storage problem. We provide the same forecasts of exogenous information for the benchmark policy and parameterized policies. Our goal is to show that parameterizing the benchmark policy within the CFA framework and optimizing parameter values can improve the performance over that achieved by the benchmark policy. We say the parameterization \( \pi(\theta) \) outperforms the nonparametric benchmark policy if it has positive policy improvement \( \Delta F^\pi(\theta) \), given by

\[
\Delta F^\pi(\theta) = \frac{F^\pi(\theta) - F^{D-LA}}{|F^{D-LA}|},
\]  

(56)

where \( F^\pi(\theta) \) is the average profit generated by parametrization \( \pi(\theta) \) and \( F^{D-LA} \) is the average profit generated by the unparameterized deterministic lookahead policy described by equation (8). In all of our experiments, we compute these averaged profits over a testing data set including 1000 random samples.

In our first experiment, we evaluate the performance of the constant forecast parameterization for different levels of uncertainty in forecasting the supply from renewable energy. In this case, the optimization problem is one dimensional and hence, we use a grid search to find the optimal policy. As shown in Figure 2, performance of the constant parameterization policy is improved by increasing the value of \( \theta \) to one point and then decreased. It is also worth noting that under perfect forecast, \( \theta = 1 \) is the optimal value as mentioned before.

In our second experiment, we examine the performance of the lookup parameterization policy with \( H = 23 \) under perfect forecast i.e., \( \sigma_E = 0 \). In particular, we first set all values of \( \theta \) to 1 and then do a one-dimensional search over each coordinate of \( \theta \). As it can be seen from Figure 3, the optimal value for each coordinate of \( \theta \) is 1 while the others are set to 1 as suggested by Lemma 5.4. It is also worth noting that the last few coordinates of \( \theta \) have almost no effect on the performance of a given policy. Intuitively, the possible reason would be the fact that a given forecast for distant future should have small affect on the present decision.

In the next experiment, we evaluate performance of our proposed algorithms in solving problem (51) embedded with the lookup policy given in (55). In Algorithm 1, since the stochastic gradient given in (35) is not computable, we use numerical derivatives. Specifically, for each coordinate of the stochastic gradient, we use the finite-difference formula to estimate the corresponding partial derivatives by estimating the objective function at a given \( \theta \) and its perturbations for each coordinate. We call this variant of Algorithm 1 as Stochastic Numerical Gradient method for the CFA model (SNG-CFA) in this section. Since our optimization problem in nonconvex, we implement our algorithms for several different starting points with \( N = 800 \) number of iterations. We then
Figure 2: Averaged performance of constant parametrization policy over 1000 simulation.

Figure 3: Averaged performance of lookup parametrization policy with $H = 23$ under perfect forecast ($\sigma_E = 0$) over 1000 simulation. Each curves represents performance of the lookup policy over changing one $\theta_i$ while $\theta_j = 1 \ \forall j \neq i$. The left curve shows the results for $i = 1, \ldots, 9$, and the right one shows them for $i = 10, \ldots, 23$. 
Figure 4: Averaged performance of the output policy of the SGF-CFA method using RMSProp stepsize for $\sigma_E^2 = 40$ over 1000 simulation. The first starting point corresponds to the lookup policy $\theta(0)$ i.e., $\theta_i = 1 \forall i$, while the rest are randomly generated in the group of five such that $\theta_i \in [0, 1]$, $\theta_i \in [0.5, 1.5]$, and $\theta_i \in [1, 2]$, respectively.

To evaluate the quality of the outputs of the algorithms, we perform average over thousand simulations. Comparison of output solutions with the nonparametric benchmark policy, the unit policy in this case i.e., $\theta_i = 1 \forall i$, is summarized in Figures 4 and 5. This policy is also used as the first starting point while the others have been generated randomly. For both algorithms, the RMSProp stepsize outperforms the choice of AdaGrad and hence we only report the results associated to the former. Note that performance of both algorithms when started with the unit policy is almost as good as the best case with other random starting points. As mentioned before, this is an optimal policy when we are given perfect forecasts. It now also seems to be a good natural starting point for any given forecast. Moreover, as it can be seen, the SNG-CFA method has better performance than the SGF-CFA method for most of the starting points. The numerical derivatives computed at each iteration of the SNG-CFA method has only a few non-zero coordinates implying that all coordinates of the lookup policy are not updated at each iteration. Due to the interaction between the coordinates, this may result in better performance than updating all coordinates as is at each iteration of the SGF-CFA method. On the other hand, the best performance of the SNG-CFA method is slightly better than that of the SGF-CFA method. However, the run time for each iteration of the former is much larger than that of the latter due to the different number of function evaluations for both methods (24 vs. 2).

In our last experiment, we explored the structure of the response surface in our optimization problem. We first randomly generate 1000 lookup policies and evaluate the objective function for these policies. We then fit a quadratic regression model to these data points to find the most interactive pairs of coordinates of the lookup policy. Finally, we pick the best solution obtained by the proposed algorithms and do two-dimensional grid search for these pairs of coordinates while
keeping the other coordinates unchanged. We show the behaviour of the objective function in terms of improvement over the benchmark policy for eight pairs of coordinates in Figure 6. All graphs contain a ridge on which changing the coordinates does not improve the policy. While the shape of these ridges can be quite different from one pair of coordinates to another one, most of them share some kind of unimodularity. On the other hand, these are just a few two-dimensional graphs for a function of 23 variables. We believe that much more work has to be done in better understanding the behaviour of the objective function which would result in designing more efficient algorithms for optimizing this structured nonconvex problem. We leave this for future research.

8 Conclusion

This work builds upon a long history of using deterministic optimization models to solve sequential stochastic problems. Unlike other deterministic methods, our class of methods, parametric cost function approximations, parametrically modify deterministic approximations to account for problem uncertainty. Our particular use of modified linear programs within the framework of the parametric cost function approximation (CFA) approach represent a fundamentally new approach to solving multistage stochastic programming problems. Our method allows us to exploit the structural properties of the problem while capturing the complex dynamics of the full base model, rather than accepting the approximations required in a stochastic lookahead model. We have demonstrated this class of policies in the context of a complex, time-dependent energy storage problem with forecasts. For our numerical work we selected an energy storage problem that is relatively small to simplify the extensive computational work. However, our methodology is scalable.
Figure 6: Averaged performance of lookup policies with $H = 23$ and $\sigma_f^2 = 40$ over 1000 simulation. Each graph represents policy improvements over a two-dimensional grid surface of pairs of coordinates of $\theta$. 25
to any problem setting which is currently being solved using a deterministic model.

The parametric CFA approach indeed represents an alternative to stochastic lookahead models that represent the foundation of stochastic programming. It requires some intuition into how uncertainty might affect the optimal solution. We would argue that this requirement parallels the design of any parametric statistical model, and hence enjoys a long history. We believe there are many problems where practitioners have a good sense of how uncertainty affects the solution. This approach opens up entirely new lines of research. For example, while the parametric CFA provides a way of using insights into the structural properties of the solution of the stochastic problem, it requires that we be able to specify the parameterization, which may not be obvious for all problems. We might, for example, study the solution of a stochastic lookahead model (such as one using scenario trees) and compare the optimal solution to that obtained using a deterministic lookahead. Identifying an effective parameterization for an important problem class would represent an important research contribution.

Next, the process of optimizing a parameterization is likely to remain a challenging problem, although these issues are more familiar to the stochastic search and simulation-optimization communities. The presence of integer variables, for example, will complicate the parameter search problem considerably. In addition, there is the perennial challenge of building a simulator (which introduces its own approximations).

Acknowledgments

This research has been supported in part by the National Science Foundation, grant CMMI-1537427.

References


Tieleman, T. and Hinton, G. (2012), Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, *in ‘COURSERA: Neural Networks for Machine Learning’*.


Appendix I

1. Renewable energy and demand model

Our model is designed in part to create complex nonstationary behaviors to test the ability of our policy to exploit forecasts while managing uncertainty. We use a series of recursive equations to create a realistic model of the stochastic process describing the generation of renewable energy. In particular, letting \( f_{t,t'}^E \) be the forecast of the renewable energy at time \( t \) for time \( t' \) and assuming that \( \{ f_{0,t'}^E \}_{t'=0,...,\min(H,T)} \) is given, we define

\[
f_{t+1,t'}^E = f_{t,t'}^E + \epsilon_{t+1,t'} \quad t = 0, \ldots, T - 1, \quad t' = t + 1, \ldots, \min(t + H, T)
\]

where \( \epsilon_{t+1,t'} \) represents the level of noise whose distribution depends on \( f_{t,t'}^E \). To create such noise, we first construct a symmetric matrix \( \Sigma \in \mathbb{R}^{H \times H} \) such that \( \Sigma(i, j) = \sigma_E^2 e^{-\alpha|i-j|} \forall i, j \), where \( \sigma_E, \alpha > 0 \) are constant numbers. Indeed, we can manipulate the quality of the forecast by changing \( \sigma_E \). By construction, \( \Sigma \) acts as a covariance matrix representing less correlation between the \( i \)-th and \( j \)-th elements when they are far from each other. We then define a normal noise vector as

\[
\bar{\epsilon}_t = C \cdot Z_t,
\]

where \( C \) is the lower triangular Cholesky decomposition of \( \Sigma \) and \( Z_t \sim \mathcal{N}(0, I_{H \times H}) \). Hence, each element of \( \bar{\epsilon}_t \) has a normal distribution with zero mean and variance \( \sigma_E^2 \) due to the fact that \( \Sigma = C \cdot C^\top \) and \( \Sigma(i, i) = 1 \). However, these elements are correlated by construction. To avoid nonnegativity of the forecast, we set

\[
[
\epsilon_{t+1,t+1}, \ldots, \epsilon_{t+1,\min(t+H,T)}
\] = \( \Phi(\text{trc}_H^{-1}(\bar{\epsilon}_t)) \),

where the operator \( \text{trc}_H(y) \) truncates the first \( \min(H, T - t) \) elements of vector \( y \), \( \Phi(\cdot) \) is standard normal density function, and \( F_{EM}^{-1} \) is an empirical cumulative distribution function obtained from historical data. The choice of \( F_{EM} \) depends on \( f_{t,t'}^E \) for \( t' = t + 1, \ldots, \min(t + H, T) \). Figure 7 shows five examples of empirical cumulative distribution functions for the change in wind speed, used in our experiments. It should be mentioned that if \( f_{t,t'}^E \) becomes negative (by low chance), we just map it to 0. Finally, after generating all forecasts, we force the observed value of the renewable energy at time \( t \) to be \( f_{t,t'}^E \).

A generated sample of observed renewable energy and its prospective forecast can be viewed in Figure 8. This is an example of a complex stochastic process that causes problems for stochastic lookahead models. For example, it is very common when using the stochastic dual dynamic programming (SDDP) to assume interstage independence, which means that \( W_I \) and \( W_{t+1} \) are independent, which is simply not the case in practice (Shapiro et al. (2013) and Dušačová and Sladký (2002)). However, capturing this dynamic in a stochastic lookahead model is quite difficult. Our CFA methodology, however, can easily handle these more complex stochastic models since we only need to be able to simulate the process in the base model.

We use the above approach in a backward format to generate demand forecasts. More specifically, assuming that \( \{ f_{t,t'}^D \}_{t=0,...,T} \) is given, we define

\[
f_{t-1,t'}^D = f_{t,t'}^D + \text{trc}_H^{-1}(\epsilon_{t-1,t'}) \quad t = T, \ldots, 1, \quad t' = t, \ldots, \min(t + H, T).
\]

Note that since the observed demands are set in advance, we can avoid the nonnegativity issue without using the aforementioned inverse CDF of a (uniform) random variable. Indeed, since the
Figure 7: Empirical cumulative distribution functions from a real data set for the renewable energy. Observed demands are usually cyclic, their values are specified with a sinusoidal stochastic function:

\[ f_{t, t}^D = D_t = \max\{0, D - \dot{D} \sin \left(\frac{\bar{D} \pi t}{T}\right) + e_t\} \quad t = 0, 1, \ldots, T, \quad (59) \]

where \( \{e_t\}_{t \geq 0} \) are correlated standard normal random variables and \( D > \dot{D}, \bar{D} \) are positive constants. A generated sample of observed demand and its prospective forecast can be viewed in Figure 9.

2. Spot price model

We assume that spot price of electricity from the grid has a positive correlation with the demand. In particular, we set

\[ f_{t, t'}^P = a + b f_{t, t'}^D \quad t = 0, 1, \ldots, T, \quad t' = t, t + 1, \ldots, T, \]

where \( b > 0 \) and \( a \sim \mathcal{N}(\mu_p, \sigma_p) \) for some \( \mu_p, \sigma_p \geq 0 \). Moreover, the observed and forecasted market prices are fixed and set to the average of all forecasted grid prices assuming a long-term contract with the customer.
Figure 8: Generated sample of renewable energy ($E_t$)

Figure 9: Generated sample of demand ($D_t$)