REINFORCEMENT LEARNING AND STOCHASTIC OPTIMIZATION

A unified framework for sequential decisions

Warren B. Powell

August 22, 2021

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2021 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created ore extended by sales representatives or written sales materials. The advice and strategies contained herin may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department with the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data:

Optimization Under Uncertainty: A unified framework
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CHAPTER 13

COST FUNCTION APPROXIMATIONS

Parametric function approximations (chapter 12) can be a particularly powerful strategy for problems where there is a clear structure to the policy. For example, buying when the price is below θ^{min} and selling when it is above θ^{max} is an obvious structure for many buy/sell problems. But PFAs do not scale to larger, more complex problems such as, say, scheduling an airline or managing an international supply chain. PFAs cannot even help you plan the path you will take with your car.

The problem with PFAs is that you either have to be able to identify a simple structural form (which means some form of linear or nonlinear model), or you can specify a high-dimensional architecture (locally constant or linear, full nonparametric, or a deep neural network) which will require a substantial number of training iterations (possibly in the millions or tens of millions). There are many problems, however, where the policy is high-dimensional, which means that lots of variables interact, such as the location of pieces on a chess board, or the effect of surplus blood inventories in one region on the allocation of blood around the country. Learning these interactions in the presence of noise is especially difficult.

CFAs are a form of parameterized optimization model. Imagine that you have a problem that suggests a natural approximation as a deterministic optimization problem. These may be myopic (assigning available drivers in a ride-sharing fleet to waiting customers), or they may involve optimizing a deterministic approximation of the future (technically a form of direct lookahead approximation, but a simple one). An example is the use of deterministic shortest path problems in navigation systems, or optimizing inventory decisions over a planning horizon given a point forecast of demands.

Copyright © 2021 John Wiley & Sons, Inc.

The optimization problem may be as complicated as scheduling an airline, or as simple as trying to pick a medical treatment $x \in \mathcal{X} = \{x_1, \ldots, x_M\}$ that will treat a patient's high blood sugar. Let $\bar{\mu}_x^n$ be the estimated reduction in blood sugar from treatment x_m after we have run n different tests, and let $\bar{\sigma}_x^n$ be the standard deviation of our estimate $\bar{\mu}_x^n$. Assuming our beliefs are independent of each other, our current state (belief state) is given by $S^n = B^n = (\bar{\mu}_x^n, \bar{\sigma}_x^n), x \in \mathcal{X}$. A greedy ("pure exploitation") policy would use the policy

$$X^{Explt}(S^n) = \arg\max_x \bar{\mu}_x^n.$$

Such a policy uses the treatment that appears to be best, but fails to recognize that after choosing x^n and observing $\hat{F}^{n+1} = F(x^n, W^{n+1})$ we can use this information to update our belief state (captured by S^n). The problem is that we may have an estimate $\bar{\mu}_x^n$ that is too low that would discourage us from trying it again. One way to fix this (which we introduced in chapter 7 as interval estimation) is by using the modified policy

$$X^{IE}(S_t|\theta) = \arg\max_{x \in \mathcal{X}} \left(\bar{\mu}_x^n + \theta \bar{\sigma}_x^n\right), \tag{13.1}$$

where θ is a parameter that has to be tuned through our usual objective function

$$\max_\theta F(\theta) = \mathbb{E} \sum_{t=0}^{T} C(S_t, X^\pi(S_t|\theta)). \tag{13.2}$$

We have tweaked the pure exploitation policy by adding an "uncertainty bonus" in (13.1) which encourages trying alternatives where $\bar{\mu}_x$ might be lower, but where there is sufficient uncertainty that it might actually be higher. This is a purely heuristic way of enforcing a tradeoff between exploration and exploitation (but a heuristic that enjoys some nice theoretical properties).

While our interval estimation policy is limited to discrete action spaces, parametric CFAs can actually be extended to very large-scale problems. Once you introduce an "$\arg\max_x$" into the policy, you open the door to using solvers for large linear, integer, nonlinear and even nonlinear-integer programs as we illustrate later in this chapter. Suddenly we can now allow x_t to be vectors with hundreds of thousands of variables (dimensions).

The idea of using a parameterized optimization model is a widely used engineering heuristic, but has been completely overlooked as a valid way of building a policy for solving stochastic sequential decision problems. The point of departure between an ad-hoc heuristic and a formal optimization model is equation (13.2). Normally we do our parameter tuning in a simulator (presumably with a final reward objective), or online in the field (presumably with a cumulative reward objective). Either way, we need to explicitly formulate the parameter tuning process as an explicit optimization problem (such as (13.2)); if this is not done, then what you are doing is, in fact, just an engineering heuristic.

While using a parameterized optimization model is quite common in practice, using equation (13.2) to tune the parameters is not. As with PFAs, there are three dimensions in the use of parametric CFAs:

1) Designing the parameterization - This is the art of any parametric model (including statistical models). CFAs begin as some form of deterministic optimization model, where the parameterization should be chosen to improve what can be achieved with the original deterministic approximation.

2) Evaluating a parametric CFA - The most common way to evaluate a policy is a simulator, but there are many settings where simulators are either too time consuming or expensive to develop, or because we simply cannot create a mathematical model of the problem, requiring evaluation to be done in the field.

3) Tuning the parameters - As we have seen in the chapters on stochastic search (chapters 5 and 7) and policy search (chapter 12), tuning the parameters θ using the objective function (13.2) is not easy. For this reason, it is quite common in industry for someone to use intuition to simply pick values for θ. While the performance of the resulting policy may be reasonable, this is not optimization.

The research community has largely dismissed parameterized deterministic models as an "industrial heuristic." We claim that a parameterized optimization model is a powerful strategy for solving certain classes of stochastic optimization problems, and is just as valid as using any PFA, or any of the strategies that we are going to present later in this book. It all boils down to exploiting problem structure and insights into how uncertainty affects the solution.

We need to pause and make an important observation: PFAs and CFAs both look like parameterized policies, but they tend to be different in a critical way, especially when the PFA uses a generic architecture such as a linear model or neural network. PFAs using a generic architecture will provide no guidance in terms of the scaling of the vector θ. By contrast, if we start with a deterministic approximation, it introduces a tremendous amount of structure, which has the effect of scaling the problem. This dramatically simplifies the parameter search process.

The remainder of this chapter will focus on illustrating different ways to create parametric CFAs. Section 13.1 sets up some general notation. Then, section 13.2 presents examples of parameterizing the objective function, followed by section 13.3 which presents examples of parameterized constraints.

13.1 GENERAL FORMULATION FOR PARAMETRIC CFA

There are two ways to parameterize an optimization problem: through the objective function, and through the constraints. To capture these changes we define

$$\begin{aligned}\bar{C}^{\pi}(S_t, x_t|\theta) &= \text{the modified objective function as determined by the policy } \pi, \text{ where } \theta \text{ represents the tunable parameters,}\\ \mathcal{X}_t^{\pi}(\theta) &= \text{the modified set of constraints (that is, the feasible region) determined by policy } \pi, \text{ with tunable parameters } \theta.\end{aligned}$$

A parametric CFA can be written in its most general form as

$$X^{CFA}(S_t|\theta) = \arg \max_{x_t \in \mathcal{X}_t^{\pi}(\theta)} \bar{C}^{\pi}(S_t, x_t|\theta), \tag{13.3}$$

where $\bar{C}^{\pi}(S_t, x_t|\theta)$ is a parametrically modified cost function, subject to a (possibly modified) set of constraints $\mathcal{X}^{\pi}(\theta)$, where θ is the vector of tunable parameters.

We now have a tunable policy $X^{CFA}(S_t|\theta)$ where we face the same challenge of finding θ as we did with PFAs in chapter 12. Note that θ might be a scalar, or may have dozens, even hundreds or thousands, of dimensions. We anticipate that the most common search procedures will be those based on either derivative-based stochastic search using numerical

derivatives such as the SPSA algorithm described in section 12.5 (or section 5.4.4) or derivative-free stochastic optimization such as the methods outlined in section 12.6. It is possible that we might apply the exact gradient described in section 12.7, but taking the derivative of the policy when the policy is an optimization problem is likely going to be daunting.

13.2 OBJECTIVE-MODIFIED CFAS

We begin by considering problems where we modify the problem through the objective function to achieve desired behaviors. Including bonuses and penalties is a widely used heuristic approach to getting cost-based optimization models to produce desired behaviors, such as balancing real costs against penalties for poor service. Not surprisingly, we can use this approach to also produce robust behaviors in the presence of uncertainty.

We begin by presenting a general way of including linear cost correction models in the objective function. We then present three application settings: a dynamic assignment problem for assigning drivers to loads, a stochastic, dynamic shortest path problem, and a financial trading problem.

13.2.1 Linear cost function correction

Although we favor parameterizations that are guided by the structure of the problem, a general approach to improving the performance of an optimization-based policy is to add a linear term to the objective, which gives us

$$X^{CFA-cost}(S_t|\theta) = \arg\max_{x_t \in \mathcal{X}_t} \left(C(S_t, x_t) + \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t, x_t) \right). \tag{13.4}$$

where $(\phi_f(S,x))_{f\in\mathcal{F}}$ is a set of features that depend first and foremost on x, and possibly on the state S. If a feature does not depend on the decision, then it would not affect the choice of optimal solution.

Designing the features for equation (13.4) is no different than designing the features for a linear policy function approximation (or, for that matter, any linear statistical model which we introduced in chapter 3). It is always possible to simply construct a polynomial comprised of different combinations of elements of x_t and S_t with different transformations (linear, square, ...), but many problems have very specific structure.

13.2.2 CFAs for dynamic assignment problems

The truckload trucking industry requires matching drivers to loads, just as ride-sharing companies match drivers to riders. The difference with truckload trucking is that the customer is a load of freight, and sometimes the load has to wait a while (possibly several hours) before being picked up.

To model our problem we begin with defining the sets of resources and tasks which make up the state variable

$\mathcal{D}_t$ = The set of all drivers (with tractors) available at time t,
$\mathcal{L}_t$ = the set of all loads waiting to be moved at time t,
S_t = $(\mathcal{D}_t, \mathcal{L}_t)$ = the state of our system at time t.

Our decision variables and costs are given by

$x_{td\ell}$ = 1 if we assign driver d to load ℓ at time t, 0 otherwise,

$c_{td\ell}$ = the contribution of assigning driver $d \in \mathcal{D}_t$ to load $\ell \in \mathcal{L}_t$ at time t, including the revenue generated by the load, the cost of moving empty to the load, as well as penalties for late pickup or delivery.

Finally, we have the post-decision sets of loads and drivers which we represent using

$\mathcal{L}_t^x$ = set of loads that were served at time t, which is to say all ℓ where $\sum_{d\in\mathcal{D}_t} x_{td\ell} = 1$,

$\mathcal{D}_t^x$ = set of drivers that were dispatched at time t, which is to say all d where $\sum_{\ell\in\mathcal{L}_t} x_{td\ell} = 1$.

A myopic policy for assigning drivers to loads would be formulated as

$$X^{Assign}(S_t) = \arg\max_{x_t} \sum_{d\in\mathcal{D}_t} \sum_{\ell\in\mathcal{L}_t} c_{td\ell} x_{td\ell}. \tag{13.5}$$

Once we dispatch a driver (that is, $x_{td\ell} = 1$ for some $\ell \in \mathcal{L}_t$), we assume the driver vanishes (this is purely for modeling simplification). We then model drivers becoming available as an exogenous stochastic process along with the new loads. This is modeled using

$\hat{L}_{t+1}$ = exogenous process describing random loads (complete with origins and destinations) that were called in between t and $t+1$,

$\hat{D}_{t+1}$ = exogenous process describing drivers calling in between t and $t+1$ to say they are available (along with location).

In practice $\hat{D}_t$ will depend on prior decisions, but this simplified model will help us make the point. The transition function would be given by

$$\mathcal{L}_{t+1} = \mathcal{L}_t \setminus \mathcal{L}_t^x \cup \hat{L}_{t+1}, \tag{13.6}$$

$$\mathcal{D}_{t+1} = \mathcal{D}_t \setminus \mathcal{D}_t^x \cup \hat{D}_{t+1}, \tag{13.7}$$

where $\mathcal{A} \setminus \mathcal{B}$ means we subtract set $\mathcal{B}$ from set $\mathcal{A}$. In real settings, however, loads that have been waiting too long may drop out and look for another carrier, which means we lose the load (and the revenue). Our myopic policy simply is not taking the value of what might happen in future time periods into account.

One way to handle this is to put a positive bonus for moving loads that have been delayed. Let

$\tau_{t\ell}$ = the time that load $\ell \in \mathcal{L}_t$ has been delayed as of time t.

Now consider the modified policy

$$X^{CFA-Assign}(S_t|\theta) = \arg\max_{x_t} \sum_{d\in\mathcal{D}_t} \sum_{\ell\in\mathcal{L}_t} (c_{td\ell} + \theta\tau_{t\ell}) x_{td\ell}. \tag{13.8}$$

Now we have a modified cost function (we use the term "cost function" even though we are maximizing) that is parameterized by θ which places a bonus (assuming $\theta > 0$) on loads that have been delayed. The next challenge is to tune θ: Too large, and we move long

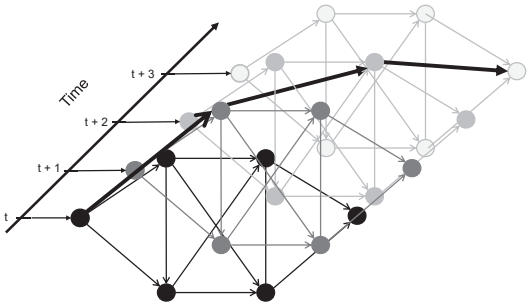


Figure 13.1 Illustration of a shortest path over a time-dependent network.

distances to pull loads that have been waiting; too small, and we end up losing loads that have to wait too long. Our optimization problem is given by

$$\max_{\theta} \mathbb{E} \sum_{t=0}^{T} C(S_t, X^{CFA-Assign}(S_t|\theta)), \tag{13.9}$$

where

$$C(S_t, x_t) = \sum_{d \in \mathcal{D}_t} \sum_{\ell \in \mathcal{L}_t} c_{td\ell} x_{td\ell}.$$

We now face the problem of tuning θ to maximize profits. We may also set a target on, say, the number of loads that have been delayed more than 4 hours.

This is a classical use of a parametric cost function approximation for finding robust policies for a very high-dimensional resource allocation problem. The delay penalty parameter θ can be tuned in a simulator that represents the objective (13.9) along with the dynamics (13.6) and (13.7). In real applications, this tuning is often done (albeit in an ad hoc way) in an online setting based on real observations.

13.2.3 Dynamic shortest paths

Consider the problem of finding the best path through a network over time, as illustrated in figure 13.1. Our navigation system uses best estimates of the times for each link in the network to plan a path to the destination, but as we progress along the path, new information arrives and the path is updated. This is a form of direct lookahead policy (that we consider in depth in chapter 19) using forecasts of future travel times.

The idea of planning paths into the future using a deterministic forecast is so familiar to us that we do not even challenge it, but this is a fully sequential, stochastic decision problem, with rolling forecasts. Now imagine that our shortest path takes us over a toll bridge which has to be lifted periodically to allow taller boats to traverse underneath. When this happens, traffic can be stopped for up to 20 minutes. It will take you 40 minutes to get to this bridge, and if you are delayed, you will miss your appointment.

When link times have distributions with long tails, we may wish to consider, for example, the 90th percentile of the time to traverse each link rather than the expectation. This is a form of parametric cost function approximation where we use a modified objective function.

A sketch of the model using our standard framework is as follows:

State variables - We represent the location of a traveler by

$$R_t = \text{the next node where the traveler has to make a decision.}$$

Estimated travel costs are represented by

$$\begin{aligned}\tilde{c}_t &= (\tilde{c}_{t,i,j})_{(i,j)\in\mathcal{N}},\\ &= \text{the vector of estimates of the cost to traverse link } (i,j) \text{ at time } t,\\ &\quad \text{given what is known at time } t.\end{aligned}$$

We are also going to assume that we have a historical dataset that tells us the *distribution* of travel costs. Since these distributions would be compiled based on many observations, we are going to assume that these are static (we would include these distributions in our initial state S_0, not in our dynamic state S_t).

The traveler's state S_t at time t is then

$$S_t = (R_t, \tilde{c}_t).$$

A common mistake is to assume that the state of our system is the location of the traveler. In a dynamic network, you have to include the estimates of the travel times on *every* link of the network in the state variable, since these are being updated every time period.

Decision variables - The decision variables are given by

$$x_{tij} = \begin{cases} 1 & \text{if we traverse link } i \text{ to } j \text{ when we are at } i \text{ at time } t,\\ 0 & \text{otherwise.}\end{cases}$$

These are subject to constraints that ensure that from any node i, we have to go somewhere (until we reach our destination).

As always, we let $X^\pi(S_t|\theta)$ be our policy for determining which link (i,j) to traverse given that we are at node i.

Exogenous information - There are two types of exogenous information for this problem:

$$\hat{c}_{t+1,ij} = \text{This is the observed cost of traversing link } (i,j) \text{ after the traveler made the decision at time } t \text{ and traversed this link.}$$

The second type of new information is the updated estimates of the link costs. We are going to model the exogenous information as the change in the estimates:

$$\begin{aligned}\delta\tilde{c}_{t+1,ij} &= \tilde{c}_{t+1,ij} - \tilde{c}_{t,ij},\\ \delta\tilde{c}_{t+1} &= (\tilde{c}_{t+1,ij})_{(i,j)\in\mathcal{N}}.\end{aligned}$$

Our exogenous information variable, then, is given by

$$W_{t+1} = (\hat{c}_{t+1}, \delta\tilde{c}_{t+1})$$

Transition function - The transition function for the forecasts evolves according to

$$\tilde{c}_{t+1,ij} = \tilde{c}_{t,ij} + \delta\tilde{c}_{t+1,ij}. \tag{13.10}$$

We update the physical state R_t using

$$R_{t+1} = \{j|x_{t,R_t,j} = 1,\}. \tag{13.11}$$

In other words, if we are at node $i = R_t$ and we make the decision $x_{tij} = 1$ (which requires that we be at node i, since otherwise $x_{tij} = 0$), then $R_{t+1} = j$.

Equations (13.10) and equation (13.11) make up our transition function:

$$S_{t+1} = S^M(S_t, X^\pi(S_t|\theta), W_{t+1}).$$

Objective function - We now write our objective function as

$$\min_\pi F^\pi(\theta) = \mathbb{E}\left\{\sum_{t=0}^{T}\sum_{(i,j)\in\mathcal{N}} \hat{c}_{t+1,ij}X^\pi(S_t|\theta)|S_0\right\}. \tag{13.12}$$

Note that our policy $X^\pi(S_t|\theta)$ is an indicator variable that is 1 if it specifies that the traveler should move over link (i, j) at time t, incurring the cost $\hat{c}_{t+1,ij}$.

Designing policies - There will always be some academic interest in solving the stochastic shortest path problem that we sketched above, but we are not aware of any practical algorithms for solving, even approximately, the full dynamic shortest path problem that recognizes that the state variable captures the state of the entire graph.

For now, however, we are focusing on simple, practical solutions. The reality is that deterministic shortest path problems are exceptionally easy to solve (see our discussion in section 2.3.3). What we are going to propose is that instead of solving a deterministic shortest path problem using the estimates $\tilde{c}_t$, we are going to use the θ-percentile of the distribution for each link. Let

$$\tilde{c}^\pi_{t,ij}(\theta) = \text{the } \theta\text{-percentile of the travel time for link } (i,j) \text{ given our estimate at time } t.$$

We are going to solve a deterministic shortest path problem (as before), but using these modified link costs. Let $X^\pi(S_t|\theta)$ be the policy for choosing the next link based on solving the shortest path with these modified link costs.

Figure 13.2 demonstrates the process of solving shortest paths (which we illustrated in figure 13.1) on a rolling basis. Each time we look ahead, we solve a deterministic shortest path using the θ-percentile costs $\tilde{c}^\pi_t(\theta)$. The solution to the shortest path problem at time t, when we are at a node i, simply tells us which node j to traverse to. By the time that we arrive at node j, the costs $\tilde{c}^\pi_t(\theta)$ would be updated, and we repeat the process.

All that remains is choosing θ. We do this by simulating our policy as we have done in the past where we have to estimate $F^\pi(\theta)$ in equation (13.12). Here we just have to apply our usual tools for stochastic search, recognizing that θ is a scalar, which means we just have a one-dimensional search. This would be fairly easy without the potentially high level of noise in the policy simulations.

13.2.4 Dynamic trading policy

We are going to describe a dynamic trading policy for determining which financial instruments to purchase that uses forecasts of stochastic prices that incorporate additional industrial statistics. The policy needs to balance risk with expected asset performance.

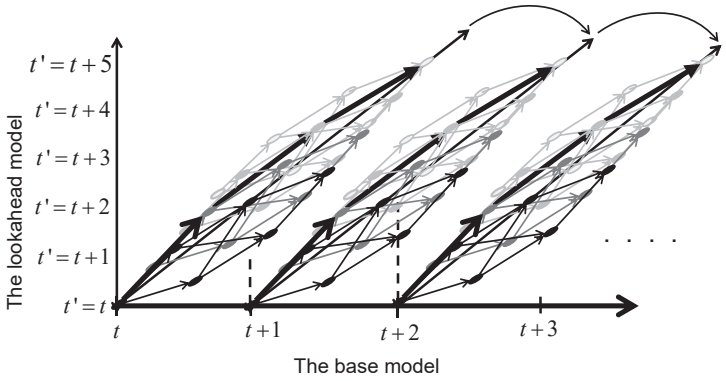


Figure 13.2 Illustration of rolling solution of deterministic shortest path problems using costs $\tilde{c}_t^{\pi}(\theta)$.

We briefly present a model of the problem using our standard framework. Of particular interest, however, is the policy that we suggest at the end that uses a modified objective function.

State variables - We represent the assets we may purchase using

$$\begin{aligned}
\mathcal{I} &= \text{the set of stocks we may hold a position in, with } i = 0 \text{ referring to cash,} \\
R_{ti} &= \text{our position (in shares) in a particular stock } i \in \mathcal{I}, \text{ where } R_{ti} \text{ can be either positive (for a long position) or negative (for a short position), and where } R_{t,0} \text{ is the amount in cash,} \\
R_t &= (R_{ti})_{i \in \mathcal{I}}.
\end{aligned}$$

Other information variables are

$$\begin{aligned}
p_{ti} &= \text{the price of stock } i, \\
p_t &= (p_{ti})_{i \in \mathcal{I}}, \\
f_{tt'i} &= \text{the forecast, generated at time } t, \text{ of the price of stock } i \text{ at time } t' \text{ over a horizon } t' = t, \ldots, t + H, \\
f_t &= (f_{tt'i})_{i \in \mathcal{I}, t' = t, \ldots, t+H.}
\end{aligned}$$

Our state variable is then

$$S_t = (R_t, p_t, f_t).$$

Decision variables - The decision variable is

$$x_{ti} = \text{the number of shares that we trade for each of the stocks. We use } x_{ti} > 0 \text{ to represent the number of shares we buy for stock } i, \text{ and } x_{ti} < 0 \text{ to represent a selling decision.}$$

The decision is constrained by the requirement that we have enough cash on hand to finance the purchasing decisions, given by

$$\sum_{i=1}^{M} x_{ti} p_{ti} \leq R_{t,0}.$$

We let $X^\pi(S_t|\theta)$ be the policy that determines x_t which satisfies this constraint.

Exogenous information - The exogenous information includes both the change in price and the change in forecasts given by

$$\begin{aligned} \hat{p}_{t+1,i} &= \text{the change in the price of stock } i \text{ between } t \text{ and } t+1, \\ \hat{p}_t &= (\hat{p}_{t+1,i})_{i \in \mathcal{I}}. \end{aligned}$$

For the forecasts, the new information is contained in the new forecasts $f_{t+1,t',i}$. We would then write our exogenous information W_{t+1} as

$$W_{t+1} = (\hat{p}_{t+1}, f_{t+1}).$$

To simulate our process, we need to assume a probability model for $\hat{p}_{t+1,i}$. A simple model would be to assume that $\hat{p}_{t+1,i}$ is normally distributed with mean 0 and variance σ_i^2. Modeling these stochastic processes is important and can be quite challenging, but our interest right now is on the design of the policy.

Transition function - The transition equation for the position in a stock R_{ti} is given by

$$R_{t+1,i} = R_{ti} + x_{ti}. \tag{13.13}$$

The transition equation for the cash position $R_{t,0}$ is given by

$$R_{t+1,0} = R_{t0} - \sum_{i=1}^{M} x_{ti} p_{ti}. \tag{13.14}$$

The transition function for the price p_t would be given by

$$p_{t+1,i} = p_{ti} + \hat{p}_{t+1,i}. \tag{13.15}$$

Also, since the new forecasts are contained in the exogenous information, we can combine equations (13.13), (13.14), and (13.15) as

$$S_{t+1} = S^M(S_t, X^\pi(S_t|\theta), W_{t+1}), \tag{13.16}$$

where $X^\pi(S_t|\theta)$ denotes a policy that maps a state to a decision.

Objective function - Our single-period contribution function is given by

$$c^{trans} = \text{the transaction cost per dollar.}$$

The transaction cost per period is given by

$$C_t(S_t, x_t) = -c^{trans} \sum_{i=1}^{M} |x_{ti}| p_{ti}, \text{ for } t = 0, \ldots, T-1,$$

where $|x_{ti}|$ is the absolute value of x_{ti}, which gives us the quantity of the trade (it does not matter whether we are buying or selling).

At the end of the day, we evaluate our risk using the quadratic function

$$\rho(R_T) = R_T' \Sigma R_T, \tag{13.17}$$

where Σ denotes the covariance matrix of the returns, which we assume we have estimated from historical data in advance. The final-period contribution function is then given by

$$C_T(S_T, x_T) = R_{T0} + \sum_{i=1}^{M} R_{Ti} p_{Ti} - \rho(R_T).$$

The objective function can now be written

$$\max_{\pi} \mathbb{E}\left\{ \sum_{t=0}^{T} C_t(S_t, X_t^{\pi}(S_t)) \middle| S_0 \right\}. \tag{13.18}$$

In practice, the expectation is approximated by using historical prices, which avoids the need to develop an underlying stochastic model.

Designing policies - We propose the following policy

$$X_t^{\pi}(S_t|\theta) = \arg\max_{x_t} \left(\sum_{i=1}^{M} \left((R_{ti} + x_{ti})(\tilde{f}_{ti}(\theta) - p_{ti}) - c^{trans}|x_{ti}|p_{ti} \right) - \rho(R_t + x_t) \right), \tag{13.19}$$

where $\tilde{f}_{ti}(\theta) = \sum_{s=1}^{H} \theta_s f_{t,t+s,i}$ represents an overall prediction of the future price using all available forecasts with different horizons and a tunable parameter vector $\theta = (\theta_1, \ldots, \theta_H)$. This policy maximizes a utility function that balances the trade-off between return and risk. It can be seen that for the risk function (13.17), the policy can be computed efficiently by solving a convex optimization problem.

A popular approach for tuning policies in financial trading settings is to use historical prices, otherwise known as "back-testing." It is possible to tune the policy on a single, long series of prices pulled from history. As always, the danger is that the policy adapts to the vagaries of a particular price sequence from history that may not be replicated in the future. However, using a historical set of prices avoids the modeling approximations inherent in any mathematical model.

13.2.5 Discussion

Care has to be used if you want to use a stochastic gradient method for optimizing cost-modified CFAs since the objective function $F(\theta)$ (see equation 13.2) is generally not going to be differentiable with respect to θ. Small changes in θ may produce sudden jumps, with intervals where there is no change at all. However, the expectation does help to smooth surfaces, so it all boils down to trying different methods to see which works the best.

13.3 CONSTRAINT-MODIFIED CFAS

A particularly powerful approach to CFAs is to modify the constraints, since this provides the analyst with direct control over the solution. It helps if there is some intuition how

uncertainty is likely to affect the final solution. While this is not always the case, it often is, and the idea of parametrically modifying constraints makes it possible to build this understanding into our solution.

The examples below provide some illustrations:

■ **EXAMPLE 13.1**

Airlines routinely use deterministic scheduling models to plan the movements of aircraft. Such models have to be designed to represent the travel times between cities, which can be highly uncertain. To handle this, the airline uses travel times equal to the θ-percentile of the travel time distribution between each pair of cities (there may be different values of θ for different types of markets).

■ **EXAMPLE 13.2**

A retailer has to manage inventories for a long supply chain extending from the far East to North America. Uncertainties in production and shipping require that the retailer maintain buffer stocks. Let θ be the amount of buffer stock planned in the future (inventory is allowed to go to zero at the last minute), which enters the model through the constraints.

■ **EXAMPLE 13.3**

Independent system operators (ISOs) for the power grid have to plan how much energy to generate tomorrow based on a forecast of loads, as well as energy to be generated from wind and solar. They use a forecast factored by a vector θ with elements for each type of forecast.

We begin our discussion by describing how a set of linear constraints can be modified. We then present a study of a realistic, time-dependent energy storage problem in the presence of rolling forecasts of energy from wind.

13.3.1 General formulation of constraint-modified CFAs

Constraint-modified CFAs can be written in the form

$$X^{Con-CFA}(S_t|\theta) = \arg \max_{x_t \in \mathcal{X}_t^\pi(\theta)} C(S_t, x_t), \tag{13.20}$$

where we are using a modified feasible region $\mathcal{X}_t^\pi(\theta)$ defined by

$$\begin{aligned} A_t^\pi(\theta^a)\tilde{x}_t &= \theta^b \otimes b_t + \theta^c, && (13.21)\\ \tilde{x}_t &\leq u_t - \theta^u, && (13.22)\\ \tilde{x}_t &\geq 0 + \theta^\ell. && (13.23) \end{aligned}$$

where $\theta^b \otimes b_t$ is the element by element product of the vector b with the similarly dimensioned vector of coefficients θ^b, plus a shift vector θ^c. The parameterization of the matrix $A_t^\pi(\theta^a)$ is how we would insert schedule slack for travel times, as well as any other

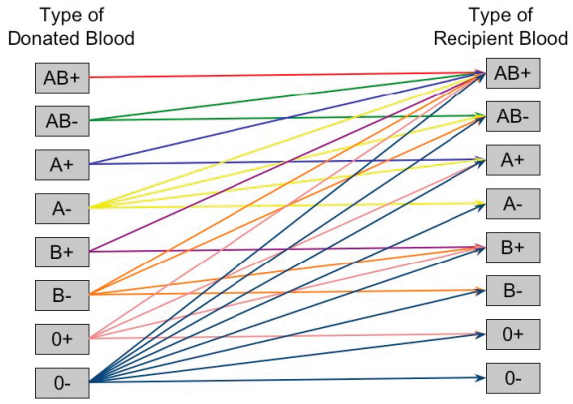


Figure 13.3 Allowable substitutions of different blood types.

adjustments that seem appropriate to the application. We then reduce the upper bounds u_t by a shift vector θ^u, and possibly raise the lower bounds by θ^ℓ. Our constraints are now parameterized by the (possibly high-dimensional) vector $\theta = (\theta^a, \theta^b, \theta^c, \theta^\ell, \theta^u)$.

The structure of the modified set of constraints hints at how we can expect to scale the vector θ. If our deterministic model closely matches what actually happens, then we would expect that $\theta^b \approx 1$, while $\theta_c, \theta^u, \theta^\ell \approx 0$. As uncertainty increases, we would expect θ^b to move away from 1 (but not too far), while we might expect $\theta^u, \theta^\ell \leq u_t$ while $\theta^c \leq b_t$. When you start doing stochastic search, you will appreciate that this type of scaling information is extremely valuable.

13.3.2 A blood management problem

In section 8.3.2 we described a blood management problem where we have to manage eight types of blood, which can be only held for five weeks (the model works in one-week increments). Figure 13.3 provides all the ways that blood types can be substituted. Note that O-minus blood can be used for any blood type (this is the universal donor), but the supplies of O-minus do not come close to covering the entire demand for blood.

Our challenge is deciding which blood type to use for each patient, given the random demands for blood in the future. A mathematical model of this problem was already provided in section 8.3.2. Here, we provide an illustration of the model in figure 13.4 which shows two time periods of a dynamic network, where all the different demands have been aggregated purely to streamline the graph and highlight decisions to use blood (if allowed) or to hold blood, where we have to keep track of aging. If we had perfect forecasts of blood demands, this would be a simple, time-dependent linear program.

We may assume that we will solve this problem once each week, using the forecasts for blood demands given by

$$f^D_{t,t'b} = \text{forecast for the demand for blood with attribute vector } b \text{ made at time } t, \text{ to serve a demand at time } t'.$$

If we use point forecasts (that is, assume that our forecasts $f^D_{tt'}$ are perfect), then we have a deterministic lookahead, just as we used for our dynamic shortest path problem in section 13.2.3. With the dynamic shortest path problem, we offered a solution for

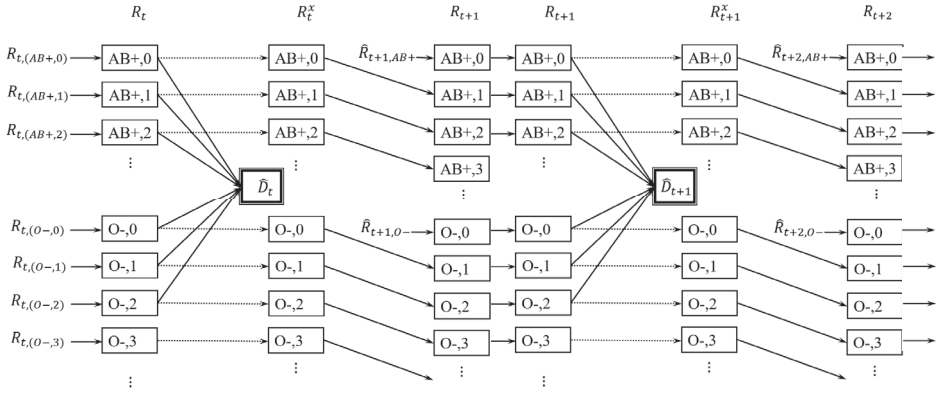


Figure 13.4 Multiperiod model of blood management, focusing on holding leftover blood over time.

handling uncertainty in travel times by modifying the costs, using $\theta-$percentiles instead of the means, which is a form of modified objective function.

With our blood management problem, ignoring the uncertainty in the forecasts might produce a solution where we use our entire inventory of O-minus blood. Intuition would say that we want to conserve our O-minus blood because it can be used to serve any form of random demand. One way to do this would be to inflate the demand for O-minus blood, which would encourage the model to maintain reserves of O-minus. To estimate the inflation, we might aggregate all the other blood types, and then take the difference between the mean and the $\theta-$percentile of the aggregate demand for the other blood types. This difference could then be added to the O-minus forecast.

With this modification, let $X_t^\pi(S_t|\theta)$ be the solution of how to allocate blood supplies at time t, given the modified demand for O-minus blood. We have to tune θ, ideally using a simulator, although it is not out of the question to experiment in the field (using, of course, a cumulative-reward objective).

13.3.3 An energy storage example with rolling forecasts

Consider a general energy storage system depicted in figure 13.5 which consists of energy from a wind farm, energy from the grid, a battery storage, and a load which could be a building, a university campus, or an entire city. The flows of energy have to be managed to meet a fairly consistent, if noisy, demand that depends on time of day (figure 13.6(a)), which has to be planned in the presence of rolling forecasts of the energy from wind (figure 13.6(b)). The demand follows familiar daily patterns, but the wind does not. In addition, the wind forecasts are not very accurate, and change quickly as the forecasts are updated.

We present our model in the usual five components: state variables, decision variables, exogenous information variables, transition function and the objective function. We note that understanding the details of the model is not important. After presenting the model, we are going to present a policy that uses a deterministic lookahead which depends on forecasts of energy from the wind farm, as well as the demand for energy over the course of the day. We are going to parameterize these forecasts as a way of handling the uncertainty

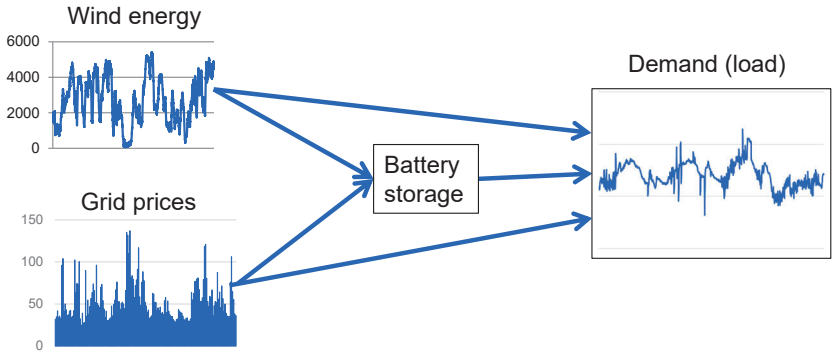


Figure 13.5 Energy storage system, including a renewable source (wind), energy from the grid at real-time prices, battery storage, and a load.

in the forecasts.

State variables - The planning of the system has to respond to the following information that is evolving over time:

D_t = Demand ("load") for power during hour t.
E_t = Energy generated from renewables (wind/solar) during hour t.
R_t = Amount of energy stored in the battery at time t.
u_t = Limit on how much generation can be transmitted at time t (this is known in advance).
p_t = Price to be paid for energy drawn from the grid at time t.

We have access to rolling forecasts of the demand D_t and the energy from wind E_t, given by:

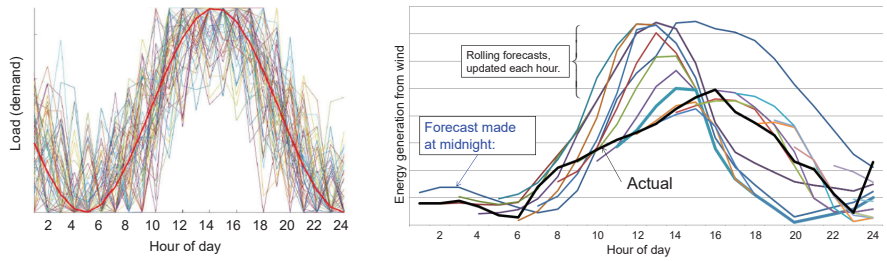
$f^D_{tt'}$ = Forecast of $D_{t'}$ made at time t.
$f^E_{tt'}$ = Forecast of $E_{t'}$ made at time t.

These variables make up our state variable:

$$S_t = (R_t, (f^D_{tt'})_{t' \geq t}, (f^E_{tt'})_{t' \geq t}).$$

Decision variables: - These are the flows between each of the elements of our energy system:

x_t = Planned generation of energy during hour t which consists of the following elements:
x^{ED}_t = flow of energy from wind to demand,
x^{EB}_t = flow of energy from wind to battery,
x^{GD}_t = flow of energy from grid to demand,
x^{GB}_t = flow of energy from grid to battery,
x^{BD}_t = flow of energy from battery to demand.



(a) (b)

Figure 13.6 (a) Energy load by hour of day and (b) rolling forecast, updated hourly.

We would normally write out the constraints that these flows have to satisfy. These consist of the flow conservation constraints, as well as upper bounds due to transmission constraints, as well as nonnegativity constraints on all the variables except x_t^{GB} since energy is allowed to flow both ways between the grid and the battery. For compactness, we are going to represent the constraints using

$$\begin{aligned} A_t x_t &= R_t, \\ x_t &\leq u_t, \\ x_t &\geq 0. \end{aligned}$$

Exogenous information - For the variables with forecasts (demand and wind energy), the exogenous information is the change in the forecast, or the deviation between forecast and actual:

$\varepsilon^D_{t+1,\tau}$ = Change in the forecast of demand (for $\tau > 1$ periods in the future) that we first learn at time $t+1$, or the deviation between actual and forecast (for $\tau = 1$).

$\varepsilon^E_{t+1,\tau}$ = Change in the forecast of wind energy (for $\tau > 1$ periods in the future) that we first learn at time $t+1$, or the deviation between actual and forecast (for $\tau = 1$).

We assume that prices evolve purely exogenously with deviations:

$\hat{p}_{t+1}$ = Change in grid prices between t and $t+1$.

Our exogenous information is then

$$W_{t+1} = ((\varepsilon^D_{t+1,\tau}, \varepsilon^E_{t+1,\tau})_{\tau \geq 1}, \hat{p}_{t+1}).$$

Transition function - The variables that evolve exogenously are

$$\begin{aligned} f^D_{t+1,t'} &= f^D_{tt'} + \varepsilon^D_{t+1,t'-t-1}, \quad t' = t+2, \ldots, \\ D_{t+1} &= f^D_{t+1,t'} + \varepsilon^D_{t+1,1}, \\ f^E_{t+1,t'} &= f^E_{tt'} + \varepsilon^E_{t+1,t'-t-1}, \quad t' = t+2, \ldots, \\ E_{t+1} &= f^E_{t+1,t'} + \varepsilon^E_{t+1,1}, \\ p_{t+1} &= p_t + \hat{p}_{t+1}. \end{aligned}$$

The energy in storage evolves according to

$$R_{t+1,t'} = R_{tt'} + x^{EB}_{tt'} + x^{GB}_{tt'} - x^{BD}_{tt'}.$$

The estimate $\tilde{R}_{t+1,t+1}$ becomes the actual energy in the battery as of time $t+1$, while $\tilde{R}_{t+1,t'}$ for $t' \geq t+2$ are projections that may change. These equations make up our transition function $S_{t+1} = S^M(S_t, x_t, W_{t+1})$.

Objective function - Our single-period contribution function is

$$C(S_t, x_t) = p_t\big(x^{GB}_t + x^{GD}_t\big).$$

Our objective function, then, would be

$$\max_{\pi} F^{\pi}(\theta) = \mathbb{E}\left\{\sum_{t=0}^{T} C(S_t, X^{\pi}(S_t|\theta))|S_0\right\}. \tag{13.24}$$

As in the past, we can estimate this objective function by simulating our policy, which we present next.

Designing the policy - Given the complex interactions of time-dependent demands, time-varying energy from wind, and the constraints on transmission, we are going to develop a deterministic lookahead model (a form of DLA). Although we do not deal with DLAs in depth until chapter 19, a deterministic lookahead is fairly simple, and we are going to show how to parameterize the policy to handle the uncertainty in the forecasts.

We distinguish the decision we make at time t, x_t, and the *planned decisions* we make at time t over our planning horizon, which we indicate by $\tilde{x}_{tt'}$. Our planned decisions are given by

$$\begin{aligned}
\tilde{x}_{tt'} &= \text{planned generation of energy during hour } t' > t, \text{ where the plan is made at time } t, \text{ which is comprised of the following elements:}\\
\tilde{x}^{ED}_{tt'} &= \text{flow of energy from renewables to demand,}\\
\tilde{x}^{EB}_{tt'} &= \text{flow of energy from renewables to battery,}\\
\tilde{x}^{GD}_{tt'} &= \text{flow of energy from grid to demand,}\\
\tilde{x}^{GB}_{tt'} &= \text{flow of energy from grid to battery,}\\
\tilde{x}^{BD}_{tt'} &= \text{flow of energy from battery to demand.}
\end{aligned}$$

We have to create projections of the energy in the battery over the horizon $t' > t$:

$$\tilde{R}_{t+1,t'} = \tilde{R}_{tt'} + \tilde{x}^{EB}_{tt'} + \tilde{x}^{GB}_{tt'} - \tilde{x}^{BD}_{tt'}.$$

The estimate $\tilde{R}_{t+1,t+1}$ becomes the actual energy in the battery as of time $t+1$, while $\tilde{R}_{t+1,t'}$ for $t' \geq t+2$ are projections that may change.

Our policy, then, is to optimize deterministically using point forecasts over a planning horizon $t, t+1, \ldots, t+H$:

$$X^{DLA}(S_t) = \arg \max_{x_t, (\tilde{x}_{tt'}, t'=t+1,\ldots,t+H)} \left(p_t(x^{GB}_t + x^{GD}_t) + \sum_{t'=t+1}^{t+H} \tilde{p}_{tt'}(\tilde{x}^{GB}_{tt'} + \tilde{x}^{GD}_{tt'})\right) \tag{13.25}$$

subject to the following constraints: First, for time t we have

$$\begin{aligned} x_t^{BD} - x_t^{GB} - x_t^{EB} &\leq R_t, && (13.26)\\ \tilde{R}_{t,t+1} - (x_t^{GB} + x_t^{EB} - x_t^{BD}) &= R_t, && (13.27)\\ x_t^{ED} + x_t^{BD} + x_t^{GD} &= D_t, && (13.28)\\ x_t^{EB} + x_t^{ED} &\leq E_t, && (13.29)\\ x_t^{GD}, x_t^{EB}, x_t^{ED}, x_t^{BD} &\geq 0. && (13.30) \end{aligned}$$

Then, for $t' = t+1, \ldots, t+H$ we have

$$\begin{aligned} \tilde{x}_{tt'}^{BD} - \tilde{x}_{tt'}^{GB} - \tilde{x}_{tt'}^{EB} &\leq \tilde{R}_{tt'}, && (13.31)\\ \tilde{R}_{t,t'+1} - (\tilde{x}_{tt'}^{GB} + \tilde{x}_{tt'}^{EB} - \tilde{x}_{tt'}^{BD}) &= \tilde{R}_{tt'}, && (13.32)\\ \tilde{x}_{tt'}^{ED} + \tilde{x}_{tt'}^{BD} + \tilde{x}_{tt'}^{GD} &= f_{tt'}^{D}, && (13.33)\\ \tilde{x}_{tt'}^{EB} + \tilde{x}_{tt'}^{ED} &\leq f_{tt'}^{E}. && (13.34) \end{aligned}$$

We are now going to focus on equations (13.33) and (13.34) since both depend on forecasts which are uncertain. In chapter 19 we are going to propose a general approach for creating lookahead policies that capture uncertainty. Here, we are going to do something simple (and very practical), which may even outperform the more complicated lookahead strategies we will describe later.

Our parameterized policy replaces equations (13.33) and (13.34) with

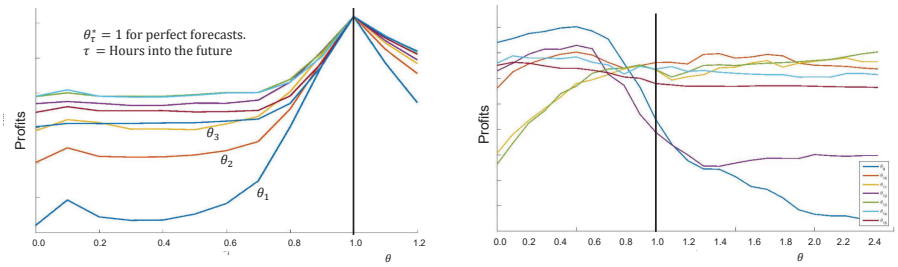
$$\begin{aligned} \tilde{x}_{tt'}^{ED} + \tilde{x}_{tt'}^{BD} + \tilde{x}_{tt'}^{GD} &= \theta_{t'-t}^{D} f_{tt'}^{D}, && (13.35)\\ \tilde{x}_{tt'}^{EB} + \tilde{x}_{tt'}^{ED} &\leq \theta_{t'-t}^{E} f_{tt'}^{E}. && (13.36) \end{aligned}$$

Now let $X_t^{CFA}(S_t|\theta)$ be the policy that solves the optimization problem in (13.25) subject to the constraints (13.31)-(13.32) and (13.35)- (13.36). We have introduced the parameters $\theta = (\theta_\tau^E, \theta_\tau^D), \tau = 1, 2, \ldots, H$ as a form of "discount factor" on the forecasts f_t^D and f_t^E.

We now face the problem of tuning θ, which means optimizing $F^\pi(\theta)$ in (13.24). For this we draw on our foundation of stochastic search. For this problem, we used the SPSA algorithm described in section 12.5 (see section 5.4.4 for a more detailed description) because it is well suited to handling multidimensional problems (θ has two 23-dimensional vectors).

We will not repeat any of the algorithmic steps (they have already been covered), but we share the following experiences with the numerical work:

- Simulations of the policy are relatively fast, requiring solving 24 relatively small linear programs (allowing us to perform an entire simulation in just a few seconds).
- Simulations of the policy are *very* noisy. It is necessary to average 1000 repetitions to get a reasonable estimate of the function (but always use whatever parallel computing capabilities you have available).
- This does not mean that we need to use a mini-batch with 1000 simulations in the SPSA calculation, but we did need mini-batches on the order of 20 to 40, which means we needed 40 to 80 function evaluations for each gradient.
- Do not forget the need to tune your stepsize formula (we used RMSProp from chapter 6). The tuning matters, and the tuning even depends on your choice of starting point.



(a) (b)

Figure 13.7 Objective vs. θ_τ for (a) perfect forecasts and (b) stochastic forecasts.

- The problem is highly time-dependent, but our parameterized lookahead policy is completely stationary. For example, θ_τ depends on how many time periods into the future we are forecasting, but does not depend on the time t at which we are making the decision. This is the value of imbedding the forecast within the policy.

A nice property of the policy is that if the forecasts are perfect, then the optimal solution should be $\theta^* = 1$. Figure 13.7(a) tests this idea for a problem with perfect forecasts by setting $\theta_\tau = 1$ for all τ and then varying each θ_τ individually. The graph shows that $\theta^*_\tau = 1$ for each value of τ.

We ran the SPSA algorithm for a problem with imperfect (in fact, highly imperfect) forecasts. We then fixed θ_τ to the values produced by the SPSA algorithm, and repeated the exercise of varying θ_τ for individual values of τ. The results are shown in figure 13.7(b), which shows that the optimum values have now moved well away from 1.0.

When doing stochastic search with any algorithm (derivative-based or derivative-free) is that it helps to understand the behavior of the surface $\mathbb{E}F^\pi(\theta, W)$. While the one-dimensional plots in figure 13.7 hint at the behavior of the surface (for example, the function appears to have a single optimum in each dimension), but seeing the function in higher dimensions contributes to our understanding.

Figure 13.8 shows four sets of two-dimensional heatmaps, where darker red reflects higher values. Each heatmap shows the two values of θ_τ between 0 and 2, so the center is $\theta_\tau = 1$, which was optimal for the deterministic problem. Note the ridges in 13.8(a) and (b), which will cause problems for a gradient-based algorithm. These ridges would also create challenges for derivative-free search methods.

Figure 13.9 shows how much the profits improved by optimizing θ using the SPSA algorithm compared to the performance using $\theta = 1$. The runs were performed for different starting points θ^0, drawn from four different regions:

1) The first region started from $\theta^0 = 1$.
2) The second region was $\theta^0 \in [0, 1]$.
3) The third region was $\theta^0 \in [.5, 1.5]$.
4) The fourth region was $\theta^0 \in [1.0, 2.0]$.

We can draw several conclusions from this graph:

- The optimized CFA outperforms the basic deterministic lookahead (with $\theta = 1$) by 20 to 50 percent, which we consider significant.

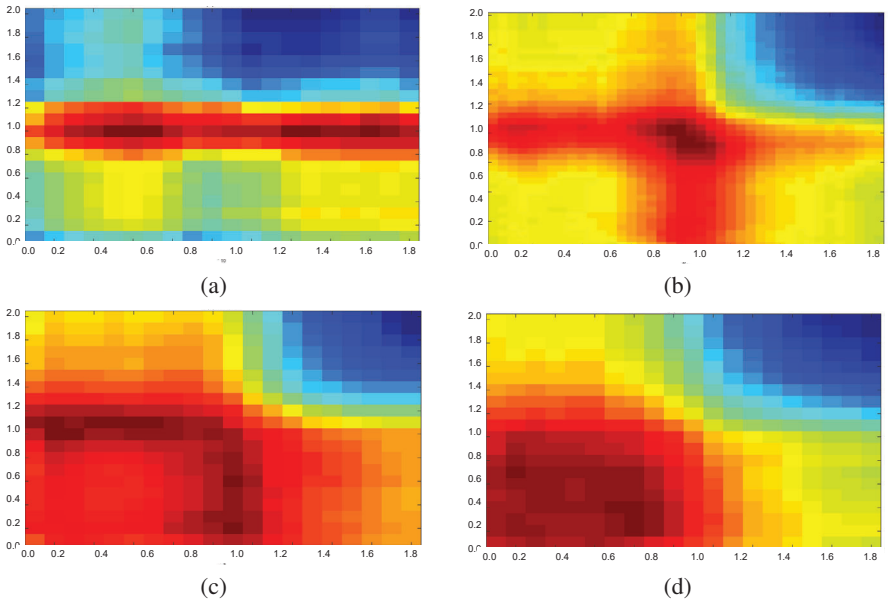


Figure 13.8 2-d heatmaps of the objective function for four different pairs of (θ_i, θ_j). Each dimension of each plot ranges from 0 to 2.

- The performance can vary widely as we randomize the starting points. However, starting with $\theta^0 = 1$ produced results that are comparable to or better than 12 out of 15 runs, but noticeably underperformed 3 of the runs. It is very nice to have a natural starting point, but more experiments are needed to understand the robustness of the optimized solutions.
- Not shown is the effect of tuning the stepsize policy, which was significant. Stepsize tuned for one starting region $[0, 1]$ but used for another starting region $[1, 2]$ could produce optimized values of θ that underperformed $\theta = 1$.
- The tuning process requires serious algorithmic work, but the resulting policy is no more complicated than a basic deterministic lookahead (that is, with $\theta = 1$).
- This is a highly nonstationary problem, with a dynamic, rolling forecast. However, our parametric CFA policy with an imbedded forecast is stationary (none of the parameters depend on time of day), which is very valuable for a problem which has strong time-of-day behavior. By imbedding the forecast, we turn a highly nonstationary problem into a stationary one that responds immediately to evolving forecasts.

There is one particularly important point about parametric CFAs in general (and parameterized lookaheads in particular):

> Many problems have complex dynamics, such as the presence of rolling forecasts for this energy problem. It is typically impossible to build these dynamics into the policies in the lookahead classes that we cover starting in chapter 14 (but especially chapter 19 on direct lookaheads), but it is quite easy to capture them in the simulation of the

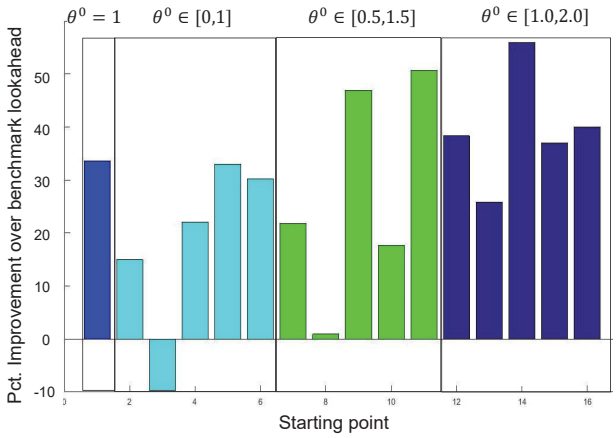


Figure 13.9 Improvement in profits using optimized θ over base results with $\theta = 1$, using starting point θ^0 drawn from each of four ranges: $\theta^0 = 1$, $\theta^0 \in [0, 1]$, $\theta^0 \in [.5, 1.5]$, and $\theta^0 \in [1, 2]$.

base model. For this reason, a carefully designed parametric CFA, tuned using the full base model which captures these dynamics, may outperform a much more complex stochastic lookahead policies that require approximations.

As with any parametric model (in optimization or statistics), there is always the question of the robustness of the model when it is implemented in new environments. These questions remain with both PFAs and CFAs. What seems to be most important is that the deterministic optimization model should capture important structural properties of the policy, which means that the tuning is just helping the policy to handle uncertainty.

13.4 BIBLIOGRAPHIC NOTES

Section 13.1 - The term "cost function approximation" was first proposed in Powell (2014). Powell & Meisel (2016) compared four classes of policies for an energy storage problem, with one being a simple version of a parameterized optimization model. The first paper to the idea of a CFA formally is Ghadimi et al. (2020). We note that the concept of parameterized optimization models is a widely used industry heuristic, but without the proper statement of an objective function.

Section 13.2 - The dynamic trading policy (section 13.2.4) was described by a graduate student based on his summer internship.

Section 13.2 - The energy storage problem with rolling forecasts (section 13.3.3) was first presented in Powell (2021). The model and algorithmic work was given in Ghadimi et al. (2020).

EXERCISES

Review questions

13.1 What are the two ways of parameterizing an optimization-based policy?

13.2 The dynamic assignment problem and the dynamic shortest path problem both parameterize the objective function, but motivated by completely different objectives. What are they?

13.3 What is the complete state variable of a dynamic shortest path problem?

Modeling questions

13.4 Using the model from section 8.3.2, write a model for the blood management problem in section 13.3.2 capturing the uncertainty in the forecasts for blood. Section 13.3.2 suggests a simple idea of inflating the demand for O-minus blood to have an adequate reserve in case we run short in our supply of other blood types. Of course, this ignores the ability to substitute across other blood types. You are going to develop a more general model for this problem based on a parameterized lookahead.

a) Write out the full, multiperiod model with random demands, including all five dimensions of a dynamic model.

b) Now introduce reserves θ_a for *each* blood type and write out this modified lookahead policy.

c) Write out the objective function for evaluating this policy.

d) The policy you have designed in (b) uses an additive adjustment. Now suggest a multiplicative adjustment as we used in section 13.3.3. How does this change the scaling of θ?

d) Since our tunable parameter θ is now a vector with eight dimensions, sketch the calculations required to estimate a gradient using the SPSA algorithm.

13.5 The energy storage problem in section 13.3.3 has to mange a highly time-dependent demand (with consistent peaks and valleys), along with rolling forecasts that can exhibit highs and lows at any time of day. Given these characteristics,

a) What does it mean to say that a "policy is stationary"?

b) Is the policy defined by (13.25) with constraints (13.26)-(13.32), (13.35) and (13.36) stationary? What allows you to make this determination?

c) Each element of the vector of parameters θ_τ was found to fall in the range [0,2]. In fact, if the forecasts were perfect then we know that $\theta_\tau = 1$. This is a very nice property. How is it that this CFA policy is so nicely scaled?

Computational exercises

13.6 From the supplementary materials page `https://castlelab.princeton.edu/rlso_supplementary/`, download the Python module (under Software) for the dynamic assignment problem. This software has modeled the dynamic assignment problem with the $\theta-$percent costs $\tilde{c}^{\pi}_{t,i,j}(\theta)$. Using this software, do the following:

a) Simulate the performance of the policy using $\theta = 1$. Repeat this 20 times and estimate the mean and standard deviation of the performance of the policy, and report the results. Normally we would use an initial experiment like this to determine how many times we need to run the simulation, but for now we evaluate a policy by averaging across 20 simulations.

b) Simulate the performance of the $\theta-$percentile policies using $\theta = 0, .2, .4, .6, .8, .9$, and report which value produces the best results.

Theory questions

13.7 Show that the optimization for the policy defined by the optimization problem (13.25)-(13.32), along with the modified constraints (13.35) - (13.36), is concave in θ. Note: this requires a background in linear programming.

13.8 Argue why the performance of the policy $F(\theta)$ produced by simulating the policy $X^{\pi}(S_t|\theta)$ given by (13.25), subject to constraints (13.26)-(13.32), is *not* concave in θ.

Problem solving questions

13.9 You would like to purchase a laptop. Price is a concern, but so is reliability, as well as service. There are some retail chains that offer service on the models they sell. You have found that that you buy laptops every two years. You do some research to develop a sense about reliability, but you will also learn from your own experience. Let

$$
\begin{aligned}
\mathcal{I} &= \text{The set of channels you can purchase the laptop from (retail outlets, websites),}\\
Q_i &= = 1 \text{ if channel } i \text{ offers repair service,}\\
\bar{\mu}_{ti} &= \text{estimated probability that the laptop purchased from channel } i \text{ will need service, given the experience as of time } t,\\
p_{ti} &= \text{price of a laptop purchased from channel } i \text{ at time } t,\\
R_{ti} &= = 1 \text{ if you are holding a laptop purchased from channel } i \text{ as of time } t,\\
z_{ti} &= = 1 \text{ if you purchase a laptop from channel } i \text{ at time } t,\\
\hat{F}_{ti} &= \text{if a laptop purchased from channel } i \text{ needs a repair at time } i.
\end{aligned}
$$

Use this notation to answer the following:

a) Define the state variable S_t.

b) Identify the decision variable and exogenous information variable. Create the notation for the policy for making the decision (we will design this below).

c) Give the equations for the transition function. Assume you are going to use exponential smoothing with parameter α to update your estimate of $\bar{\mu}_{ti}$.

d) You want to minimize how much you spend, and you put a weight ρ^{serv} on the value of purchasing the laptop from a channel that offers service. Finally, you would like to limit the probability of needing service to less than 0.05. Use these guidelines to create an objective function for evaluating your policy.

Diary problem

The diary problem is a single problem you chose (see chapter 1 for guidelines). Answer the following for your diary problem.

13.10 Do one of the following:

a) Pick a decision in your problem that lends itself to being made by solving a deterministic approximation over some horizon. Think about how uncertainty might affect the quality of this solution, and what you think should be done differently in the presence of uncertainty. Try to suggest a parametrization that would make the deterministic lookahead work better.

b) Pick a decision in your problem where a myopic optimization is a reasonable starting point. Now, think about how considering the downstream impact of the decision might affect the decision you are making now. Try to introduce a parametrization that would make the myopic model work better.

Bibliography

Ghadimi, S., Perkins, R. & Powell, W. B. (2020), 'Reinforcement Learning via Parametric Cost Function Approximation for Multistage Stochastic Programming'.

Powell, W. B. (2014), 'Clearing the Jungle of Stochastic Optimization', *Informs TutORials in Operations Research 2014*.

Powell, W. B. (2021), 'From reinforcement learning to optimal control: A unified framework for sequential decisions', *Handbook on Reinforcement Learning and Optimal Control, Studies in Systems, Decision and Control* pp. 29–74.

Powell, W. B. & Meisel, S. (2016), 'Tutorial on Stochastic Optimization in Energy - Part II: An Energy Storage Illustration', *IEEE Transactions on Power Systems*.