Chapter 1

Control of Diffusions via Linear Programming

Jiarui Han and Benjamin Van Roy

In this chapter we present an approach that leverages linear programming to approximate optimal policies for controlled diffusion processes, possibly with high-dimensional state and action spaces. The approach fits a linear combination of basis functions to the dynamic programming value function; the resulting approximation guides control decisions. Linear programming is used here to compute basis function weights.

What we present extends the linear programming approach to approximate dynamic programming, previously developed in the context of discrete-time stochastic control [19, 20, 7, 8, 9]. One might question the practical merits of such an extension relative to descretizing continuous-time models and treating them using previously developed methods. As will be made clear in this chapter, there are indeed important advantages in the simplicity and efficiency of computational methods made possible by working directly with a diffusion model.

We begin in Section 1.1 by presenting a problem formulation and a linear programming characterization of optimal solutions. The numbers of variables and constraints defining this linear program are both infinite. In Section 1.2, we describe algorithms that apply finite-dimensional linear programming to approximately solve this infinite-dimensional problem. To illustrate practical qualities, we discuss in Section 1.3 computational results from case studies in dynamic portfolio optimization. A closing section summarizes merits of the new approach.

1.1 Preliminaries

This section offers a context for algorithms to be presented in Section 1.2. To avoid the technical complexities that often arise when dealing with controlled diffusion models, we make simplifying assumptions that can be restrictive and difficult to verify. The most notable ones concern compactness of state and action spaces and regularity of the optimal value function. These assumptions are stronger than those put forth by more thorough treatments of controlled diffusion models [14, 2]. Our motivation is not to develop a general model, but rather a context for explaining practical algorithms that can be applied more broadly. Indeed, the portfolio optimization models addressed in Section 1.3 fail to satisfy our assumptions.

1.1.1 Controlled Diffusion Processes

Let $B = (B^{(1)}, \ldots, B^{(D)})^\top$ be a standard Brownian motion in $\Re^D$ defined with respect to a complete probability space $(\Omega, \mathcal{F}, P)$, and let $\mathbf{F} = \{\mathcal{F}_t | t \geq 0\}$ be the augmented filtration generated by B.

We consider a controlled diffusion process with state at each time t, denoted by X_t, taking values in a state space $\mathcal{S}$ that is a compact subset of $\Re^N$. At each time t, an action $\psi \in \Psi$ is taken, where Ψ is a compact subset of $\Re^M$. The process evolves according to

$$dX_t = \mu(X_t, \psi)\, dt + \sigma(X_t, \psi)\, dB_t,$$

where μ and σ are Lipschitz continuous. The process terminates at a time $T_{\mathcal{S}}$, where for any closed set A, T_A is the first time at which X_t hits the boundary of A.

For each time t, a control function $\theta_t : \mathcal{S} \to \Psi$ maps states to actions. A policy θ is a continuum of control functions: $\theta = \{\theta_t\}_{t=0}^{\infty}$. Such a policy is said to be stationary if the same control function is used at each time. Under a policy θ, the state X_t evolves according to

$$dX_t = \mu(X_t, \theta_t(X_t))\, dt + \sigma(X_t, \theta_t(X_t))\, dB_t.$$

Under the assumptions we have made, it is easy to show that this stochastic differential equation admits a solution over the interval $[0, T_{\mathcal{S}}]$.

Let $\overline{\Theta}$ denote the set of admissible policies and Θ the set of admissible stationary policies. For any function f, we will use $f(x, \theta)$ as shorthand for $f(x, \theta(x))$.

A utility function $u : \mathcal{S} \times \Psi \mapsto \Re$ captures preferences among state-action pairs. The objective of our control problem takes the form

$$\sup_{\theta \in \overline{\Theta}} E_{x,\theta} \left[\int_{t=0}^{T_{\mathcal{S}}} e^{-\alpha t} u(X_t, \theta_t) dt \right], \tag{1.1}$$

for some discount rate $\alpha > 0$. The subscripts in the expectation indicate that the initial state is $X_0 = x$ and that the policy $\theta \in \overline{\Theta}$ is employed.

1.1.2 Value Functions and the HJB Equation

For each policy $\theta \in \overline{\Theta}$, the value function associated with θ is defined by

$$J_\theta(x) = E_{x,\theta} \left[\int_{t=0}^{T_{\mathcal{S}}} e^{-\alpha t} u(X_t, \theta_t) dt \right],$$

and the optimal value function is defined by

$$J^*(x) = \sup_{\theta \in \overline{\Theta}} J_\theta(x).$$

Let C^2 denote the set of twice continuously differentiable functions mapping $\mathcal{S}$ to $\Re$. We make the following regularity assumption:

Assumption 1.1.1. $J^* \in C^2$

We define dynamic programming operators H_θ and H by

$$\begin{aligned} H_\theta J(x) &= J_x(x)^\top \mu(x, \theta) + \frac{1}{2} \mathrm{tr} \left(J_{xx}(x) \sigma(x, \theta) \sigma(x, \theta)^\top \right) - \alpha J(x) + u(x, \theta), \\ HJ(x) &= \sup_{\theta \in \overline{\Theta}} H_\theta J(x), \end{aligned}$$

for any $J \in C^2$. With some abuse of notation, we will sometimes use H_ψ in place of H_θ when $\theta(x) = \psi$ for all x. We refer to $\theta_J \in \overline{\Theta}$ as a greedy policy with respect to J if $H_{\theta_J} J = HJ$. It is easy to see that for any $J \in C^2$, θ_J is a well-defined admissible stationary policy.

Our goal for the remainder of this subsection is to establish that the optimal value function satisfies the HJB equation and that policies that are greedy with respect to the optimal value function are optimal. The following lemma captures a fundamental property of H that will be used to establish our desired results.

Lemma 1.1.1. *For all $J \in C^2$, $\theta \in \overline{\Theta}$, $x \in \mathcal{S}$, and closed bounded sets $A \subset \mathcal{S}$,*

$$J(x) = E_{x,\theta}\left[\int_0^T e^{-\alpha t}\big(u(X_t, \theta_t) - (H_{\theta_t} J)(X_t)\big)\, dt + e^{-\alpha T} J(X_T)\right],$$

where $T = T_A \wedge T_{\mathcal{S}}$.

Proof. Let X_t be a sample path generated by the policy θ. By Ito's formula,

$$d\left(e^{-\alpha t} J(X_t)\right) = e^{-\alpha t}\left((H_{\theta_t} J)(X_t) - u(X_t, \theta_t)\right)\, dt + e^{-\alpha t} J_x(X_t)\sigma(X_t, \theta_t)\, dB_t.$$

Therefore

$$e^{-\alpha T} J(X_T) - J(X_0) = \int_0^T e^{-\alpha t}\left((H_{\theta_t} J)(X_t) - u(X_t, \theta_t)\right)\, dt + \int_0^T \gamma_t\, dB_t,$$

where

$$\gamma_t = e^{-\alpha t} J_x(X_t)\sigma(X_t, \theta_t).$$

Note that γ_t is bounded for $t \leq T$, since X_t is bounded and J is continuously differentiable and σ is continuous. It follows that

$$E_{x,\theta}\left[e^{-\alpha T} J(X_T) - J(X_0)\right] = E_{x,\theta}\left[\int_0^T e^{-\alpha t}\big((H_{\theta_t} J)(X_t) - u(X_t, \theta_t)\big) dt\right].$$

Rearranging terms, we obtain the desired result. □

The following theorem verifies that J^* is a solution to the HJB equation.

Theorem 1.1.1. *Under Assumptions 1.1.1, $HJ^* = 0$.*

Proof. We begin by showing that $HJ^* \geq 0$. Let x be in the interior of $\mathcal{S}$, and assume for contradiction that $(HJ^*)(x) < 0$. By Assumption 1.1.1, HJ^* is continuous, and therefore, there exists a compact set A containing x in its interior, such that $(HJ^*)(x') < (HJ^*)(x)/2$ for all $x' \in A$. Let $T = T_A \wedge T_{\mathcal{S}}$. Since the state requires time to escape,

$$\inf_{\theta \in \overline{\Theta}} E_{x,\theta}\left[\int_{t=0}^{T} e^{-\alpha t} dt\right] > 0.$$

Let θ^ϵ be such that $J^*(x) - J_{\theta^\epsilon}(x) \leq \epsilon$. Then,

$$\begin{aligned}
0 &\leq J_{\theta^\epsilon}(x) - J^*(x) + \epsilon \\
&\leq E_{x,\theta^\epsilon}\left[\int_{t=0}^{T} e^{-\alpha t} u(X_t, \theta_t) dt + e^{-\alpha T} J^*(X_T)\right] - J^*(x) + \epsilon \\
&= E_{x,\theta^\epsilon}\left[\int_{t=0}^{T} e^{-\alpha t} (H_{\theta_t^\epsilon} J^*)(X_t) dt\right] + \epsilon \\
&\leq E_{x,\theta^\epsilon}\left[\int_{t=0}^{T} e^{-\alpha t} (HJ^*)(X_t) dt\right] + \epsilon \\
&\leq \frac{(HJ^*)(x)}{2} E_{x,\theta^\epsilon}\left[\int_{t=0}^{T} e^{-\alpha t} dt\right] + \epsilon \\
&\leq \frac{(HJ^*)(x)}{2} \inf_{\theta \in \overline{\Theta}} E_{x,\theta}\left[\int_{t=0}^{T} e^{-\alpha t} dt\right] + \epsilon,
\end{aligned}$$

where the one equality follows from Lemma 1.1.1. Since ϵ can be chosen arbitrarily small, we have a contradiction. It follows that $HJ^* \geq 0$.

We will now establish that $HJ^* \leq 0$. Let x be in the interior of $\mathcal{S}$, and assume for contradiction that $(HJ^*)(x) > 0$. Note that

$$E_{x,\theta_{J^*}}\left[e^{-\alpha T}\left(J^*(X_{T_{\mathcal{S}}}) - J_{\theta_{J^*}}(X_{T_{\mathcal{S}}})\right)\right] = 0.$$

By Assumption 1.1.1, HJ^* is continuous, and therefore, there exists a closed bounded set A containing x in its interior, such that $(HJ^*)(x') > (HJ^*)(x)/2$ for all $x' \in A$. Since the state requires time to escape,

$$E_{x,\theta_{J^*}}\left[\int_{t=0}^{T \wedge T_A} e^{-\alpha t} dt\right] > 0.$$

By Lemma 1.1.1, we obtain

$$
\begin{aligned}
0 &\leq J^*(x) - J_{\theta_{J^*}}(x) \\
&= E_{x,\theta_{J^*}}\left[\int_{t=0}^{T} e^{-\alpha t}(u(X_t,\theta_t) - (H_{\theta_{J^*}}J^*)(X_t))dt + e^{-\alpha T}J^*(X_T)\right] \\
&\quad -E_{x,\theta_{J^*}}\left[\int_{t=0}^{T} e^{-\alpha T}u(X_t,\theta_t) + e^{-\alpha T}J_{\theta_{J^*}}(X_T)\right] \\
&= -E_{x,\theta_{J^*}}\left[\int_{t=0}^{T} e^{-\alpha t}(HJ^*)(X_t)dt\right] \\
&\leq -E_{x,\theta_{J^*}}\left[\int_{t=0}^{T\wedge T_A} e^{-\alpha t}(HJ^*)(X_t)dt\right] \\
&\leq -\frac{(HJ^*)(x)}{2}E_{x,\theta_{J^*}}\left[\int_{t=0}^{T\wedge T_A} e^{-\alpha t}dt\right],
\end{aligned}
$$

where the second-to-last inequality follows from the fact that $HJ^* \geq 0$. We have a contradiction. It follows that $HJ^* \leq 0$. □

We close this section with a characterization of optimal policies in terms of greedy policies with respect to the optimal value function.

Theorem 1.1.2. *Under Assumptions 1.1.1, a stationary policy* $\theta \in \overline{\Theta}$ *is optimal if* $H_\theta J^* = 0$.

Proof. Suppose $H_\theta J^* = 0$. Then θ is a greedy strategy with respect to J^*. By Lemma 1.1.1,

$$
\begin{aligned}
J^*(x) &= E_{x,\theta}\left[\int_0^{T_S} e^{-\alpha t}(u(X_t,\theta) - (H_\theta J^*)(X_t))dt\right] \\
&= E_{x,\theta}\left[\int_0^{T_S} e^{-\alpha t}u(X_t,\theta)dt\right] \\
&= J_\theta(x).
\end{aligned}
$$

Hence, θ is an optimal stationary policy. □

1.1.3 A Linear Programming Characterization

The HJB Equation offers a characterization of the optimal value function. In this subsection, we discuss an alternative though closely related characterization which motivates the approximation algorithms of the next section. In particular, we consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & \int J(x)\rho(dx) \\ \text{subject to} & HJ \leq 0 \\ & J \in C^2 \end{array} \tag{1.2}$$

where the function J is the variable to be optimized and ρ is a pre-specified positive measure. Note that the objective functional is linear. Furthermore, for each θ, $H_\theta J$ is affine in J, so each constraint $(HJ)(x) \leq 0$ can be converted into a continuum of linear constraints, each taking the form $(H_\theta J)(x) \leq 0$. As such, this optimization problem can be viewed as a linear program.

The following theorem establishes that the unique optimal solution to this linear program is the optimal value function J^*. Here, uniqueness is in the sense of equivalence with respect to the measure ρ.

Theorem 1.1.3. *Under Assumption 1.1.1, J^* uniquely attains the optimum in (1.2).*

Proof. From Theorem 1.1.1 we know that $HJ^* = 0$, so J^* is a feasible solution.

Consider an arbitrary feasible solution $J \in \mathcal{J}$. By Lemma 1.1.1, we have

$$J(x) = E_{x,\theta_{J^*}} \left[\int_0^{T_\mathcal{S}} e^{-\alpha t} \left(u(X_t, \theta_t) - (H_{\theta_{J^*}} J)(X_t) \right) dt \right].$$

Moreover, since $HJ \leq 0$, we also have

$$J(x) \geq E_{x,\theta_{J^*}} \left[\int_0^{T_\mathcal{S}} e^{-\alpha t} u(X_t, \theta_t) \, dt \right] = J_{\theta_{J^*}}(x) = J^*(x).$$

Since ρ is a positive measure and $\int J^*(x)\rho(dx) < \infty$, only J^* can obtain the optimum. □

1.2 Approximate Dynamic Programming

It is not generally possible to solve the the linear program (1.2), as there are infinite numbers of variables and constraints. In this section, we present approximation methods that extend analogous ones previously developed for discrete-time stochastic control problems [19, 20, 7, 8, 9].

1.2.1 Approximation via Basis Functions

The approach we consider approximates the optimal value function J^* using a linear combination $\sum_{k=1}^K r_k \phi_k$ of pre-selected basis functions $\phi_1, \ldots, \phi_K \in C^2$. Here, $r \in \Re^K$ denotes a vector of weights. The basis functions are hand-crafted using intuition about the form of the value function, and the choice may be improved via trial and error. Given a set of basis functions, weights are computed to fit the optimal value function. We will present in this section algorithms for computing weights. The case studies of Section 1.3 will serve to illustrate, among other things, examples of basis functions.

Let us first briefly discuss how an approximation may be used to control the system. The idea is to employ a greedy policy with respect to this approximation in lieu of an optimal one, which would be greedy with respect to the optimal value function. Note that when at a state $x \in \mathcal{S}$, a greedy action with respect to $\sum_{k=1}^K r_k \phi_k$ can be obtained in real-time by solving

$$\max_{\psi \in \Psi} \left\{ \sum_{k=1}^K r_k \left((\nabla_x \phi_k)^\top (x) \mu(x, \psi) + \frac{1}{2} \mathrm{tr} \left((\nabla_x^2 \phi_k)(x) \sigma(x, \psi) \sigma(x, \psi)^\top \right) \right) + u(x, \psi) \right\}. \tag{1.3}$$

This is a constrained nonlinear program, and one might consider solving for a local optimum using an interior point method. It is worth noting, though, that in many contexts of practical relevance, such expressions are amenable to efficient global optimization. For example, in the portfolio optimization problems to be discussed in Section 1.3, this optimization problem is a convex quadratic program.

As a framework for computing weights, we consider a variation of the linear program (1.2) that characterizes the optimal value function. We maintain the same constraints but restrict the solution space to the span of the basis functions. The resulting linear program takes the form

$$\begin{array}{ll} \text{minimize} & \int (\Phi r)(x) \rho(dx) \\ \text{subject to} & H \Phi r \leq 0. \end{array} \tag{1.4}$$

where $\Phi r = \sum_{k=1}^K r_k \phi_k$. Note that the C^2 constraint is not needed here because the basis functions are themselves in C^2.

This new linear program has a finite number of variables. However, the number of constraints remains infinite. There is one linear constraint per state-action pair. Further, the objective function involves an integral over states. In the next subsection, we present a method for alleviating the dependence on the number of states through randomized sampling. We will later discuss methods that alleviate the dependence on the number of actions.

It is worth mentioning that the choice of measure ρ influences the set of optimal weight vectors for our new linear program (1.4). This is in contrast with the previous linear program (1.2) for which, by Theorem 1.1.3, the unique optimal solution is the optimal value function J^* regardless of the choice of ρ. There is no clear understanding of how ρ should be chosen in (1.4), though results in [7, 9] suggest desirability of a discounted relative frequency measure associated with the greedy policy that will ultimately be derived from the resulting weight vector. The case studies reported in Section 1.3 make use of measures chosen in this spirit.

1.2.2 Sampling States

We consider approximating the linear program (1.4) using one that replaces the integral with a sum over a sampled set of states and retains only constraints associated with this sampled set. In particular, this method generates a set of Q independent identically distributed samples $x^{(1)}, \ldots, x^{(Q)} \in \mathcal{S}$, drawn according to the positive measure ρ, and then solves a linear program of the form

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{Q} (\Phi r)(x^{(i)}) \\ \text{subject to} & (H\Phi r)(x^{(i)}) \leq 0, \quad \forall i = 1, \ldots, Q. \end{array} \tag{1.5}$$

This method is entirely analogous to one developed for discrete-time problems in [8], which also offers theoretical results to motivate why sampling may not distort the solution too much. Here we only provide some less formal heuristic motivation. For any r, the sum $\sum_{i=1}^{Q} (\Phi r)(x^{(i)})$ offers an unbiased estimate of the integral $\int (\Phi r)(x)\rho(dx)$. Intuitively, as the number of samples grows the estimate should become accurate uniformly across relevant values of r, and therefore, replacing the integral with the sum seems reasonable. Regarding constraints, note that the linear program (1.4) defines a set in $\Re^K$ using an infinite number of constraints. As such, one may expect almost all of the constraints to be irrelevant. A tractable subset may lead to a reasonable approximation.

Our new linear program (1.5) alleviates the dependence on the number of states. The objective involves a sum over Q sampled states rather than the entire state space. The number of constraints $(H\Phi r)(x^{(i)}) \leq 0$

also scales with the number of sampled states. However, there are still as many linear constraints as there are actions. In particular, each constraint $(H\Phi r)(x^{(i)}) \leq 0$ is equivalent to a set of linear constraints: $(H_\psi \Phi r)(x^{(i)}) \leq 0$ for all $\psi \in \Psi$. In the next subsection, we explain two methods for alleviating this dependence on the number of actions.

1.2.3 Dealing with the Action Space

The two methods we will present pose different merits. The first is guaranteed to obtain an optimal solution to (1.5). However, it requires solving a nonlinear convex program which cannot always be represented in a manner amenable to efficient computation. Further, even when an appropriate representation exists, though this convex program can be solved in polynomial time, for the large instances that arise in practical applications, computational requirements can be onerous. The second method is heuristic and comes with no guarantee of finding an optimal solution to (1.5). However, it relies on solving linear rather than nonlinear convex programs, and in computational studies, this method has proved to be effective and more efficient. The method solves a sequence of linear programs, each with Q constraints. The set of constraints is adapted based on results of each iteration.

Convex Programming

Note that (1.5) is itself a convex program since the objective function is linear and the constraint associated with each sampled state is convex. However, the left-hand-side of each constraint $(H\Phi r)(x^{(i)}) \leq 0$ is the result of an optimization problem:

$$
\begin{aligned}
(H\Phi r)(x^{(i)}) &= \max_{\psi \in \Psi} (H_\psi \Phi r)(x^{(i)}) \\
&= \max_{\psi \in \Psi} \left\{ \sum_{k=1}^{K} r_k \Big((\nabla_x \phi_k)^\top (x^{(i)}) \mu(x^{(i)}, \psi) \right. \\
&\quad + \frac{1}{2} \mathrm{tr}\left((\nabla_x^2 \phi_k)(x^{(i)}) \sigma(x^{(i)}, \psi) \sigma(x^{(i)}, \psi)^\top \right) \\
&\quad \left. - \alpha \phi_k(x^{(i)}) \Big) + u(x^{(i)}, \psi) \right\}.
\end{aligned}
$$

If this optimization problem can be solved in closed form to obtain a simple expression for the maximum as a function of r, there is hope of representing the convex constraint in a way that is amenable to efficient computation. However, this is unlikely to be the case for a constrained optimization problem. As such, we consider converting it to an unconstrained problem by pricing out the constraints.

We will assume that the action set Φ is a polytope, represented by L linear inequalities. In particular, there is a matrix A and vector b such that

$$\Psi = \left\{\psi \in \Re^M | A\psi \leq b\right\}.$$

Then, by Lagrangian duality,

$$(H\Phi r)(x^{(i)}) = \min_{\lambda \in \Re^L_+} \max_{\psi \in \Re^M} ((H_\psi \Phi r)(x^{(i)}) - \lambda^\top (A\psi - b)).$$

Now, for fixed λ, the maximization problem is unconstrained. Our convex programming approach is applicable in cases where this unconstrained maximization problem can be solved in closed form to obtain a simple expression for the maximum as a function of r and λ. The case studies of Section 1.3, for example, fall into this category because in each case the expression being maximized is a concave quadratic.

Let

$$g_i(r, \lambda) = \max_{\psi \in \Re^M} ((H_\psi \Phi r)(x) - \lambda^\top (A\psi - b)).$$

Recall that the constraint we wish to impose is $(H\Phi r)(x^{(i)}) \leq 0$. This is equivalent to imposing $\min_{\lambda \in \Re^L_+} g_i(r, \lambda) \leq 0$, which is equivalent in turn to imposing a pair of constraints

$$\begin{aligned} & g_i(r, \lambda) \leq 0 \\ & \lambda \geq 0. \end{aligned}$$

In particular, r satisfies the constraint if there exists a $\lambda \in \Re^L_+$ such that $g_i(r, \lambda) \leq 0$. Now, if $g_i(r, \lambda)$ is a simple expression, we arrive at a convex program represented in a manner amenable to efficient computation:

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^Q (\Phi r)(x^{(i)}) & \\ \text{subject to} & g_i(r, \lambda^{(i)}) \leq 0, & \forall i = 1, \ldots, Q \\ & \lambda^{(i)} \geq 0, & \forall i = 1, \ldots, Q. \end{array} \quad (1.6)$$

Adaptive Constraint Selection

In many cases of practical interest $g_i(r, \lambda)$ can indeed be written as a simple expression. However, when this is not the case or when the resulting convex program (1.6) is computationally burdensome, an alternative method is called for. We now describe a heuristic which in computational studies has proved to be effective and much more efficient.

Our heuristic solves a sequence of linear programs, each with Q linear constraints. In each iteration, one constraint is chosen per sampled state $x^{(i)}$. For each of these states, the choice is governed by the action that is greedy with respect to the weight vector generated in the previous iteration. A more detailed description of the heuristic follows:

Algorithm 1 Adaptive constraint selection.

1: **for** $t = 1$ to ∞ **do**

2: **for** $i = 1$ to Q **do**

3:

$$\psi^{(i)} \in \underset{\psi \in \Psi}{\text{argmax}} (H_\psi \Phi r^{(t-1)})(x^{(i)})$$

4: **end for**

5:

$$r^{(t)} \in \begin{array}{ll} \text{argmin} & \sum_{i=1}^{Q} (\Phi r)(x^{(i)}) \\ \text{subject to} & (H_{\psi^{(i)}} \Phi r)(x^{(i)}) \leq 0, \qquad \forall i = 1, \ldots, Q. \end{array}$$

6: **end for**

The linear program solved in line 5 involves K variables and Q linear constraints. Use of this heuristic requires computation of greedy actions as well. The associated optimization problem of line (3) may be an arbitrary nonlinear program, but in many contexts of practical interest, the problem is a simple convex program. In the case studies of Section 1.3, for example, this problem is a convex quadratic program.

The adaptive constraint selection algorithm does not necessarily generate an optimal solution to the linear program (1.5) it aims to solve. However, if the sequence $r^{(t)}$ converges, it must converge to an optimal solution. To see why, consider a limit of convergence $\bar{r}$. This limit must attain the minimum of

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{Q} (\Phi r)(x^{(i)}) \\ \text{subject to} & (H_{\psi^{(i)}} \Phi r)(x^{(i)}) \leq 0, \qquad \forall i = 1, \ldots, Q, \end{array} \tag{1.7}$$

where each action $\psi^{(i)}$ is greedy with respect to $\Phi\bar{r}$. Now assume for contradiction that $\bar{r}$ is not an optimal solution of (1.5) and let r^* be an optimal solution. Then,

$$\sum_{i=1}^{Q} (\Phi r^*)(x^{(i)}) < \sum_{i=1}^{Q} (\Phi \bar{r})(x^{(i)}),$$

and

$$(H_{\psi^{(i)}} \Phi r^*)(x^{(i)}) \leq (H\Phi r^*)(x^{(i)}) \leq 0 \quad \forall i = 1, \ldots, Q.$$

It follows that r^* is a feasible solution to (1.7) with a smaller objective value than $\bar{r}$, yielding a contradiction.

1.3 Case Studies in Portfolio Optimization

In this section we will formulate a dynamic portfolio optimization model and present two case studies that apply the approximation approach of the previous section. There is significant prior work on approximate dynamic programming methods for portfolio optimization [13, 3, 18, 1, 16, 4, 21, 12]. The primary source of difference in our work, which was originally reported in the first author's dissertation [11], is the algorithm used. In particular, it is based on linear programming and works directly with the controlled diffusion (without requiring discretization). These features enable more efficient computations and a more streamlined experimental process, involving far less tinkering.

1.3.1 Market Model

We consider a market with M risky assets and one risk-free asset. Prices follow a diffusion processes with drift and diffusion coefficients modulated by market state. Let $B = (B^{(1)}, \ldots, B^{(D)})^\top$ be a standard Brownian motion in $\Re^D$, and let $\mathbf{F} = \{\mathcal{F}_t | t \geq 0\}$ be the associated filtration.

At each time t, the state of the market is described by a vector $Z_t \in \Re^N$, which evolves according to

$$dZ_t = \mu_z(Z_t)\, dt + \sigma_z(Z_t)\, dB_t,$$

where $\mu_z : \Re^N \to \Re^N$ and $\sigma_z : \Re^N \to \Re^{N \times D}$ are Lipschitz continuous functions. We assume that Z_0 is constant.

There is an instantaneously risk-free asset, which we will refer to as the money market, whose price $S_t^{(0)}$ follows

$$dS_t^{(0)} = r(Z_t)S_t^{(0)}\,dt,$$

where $r : \Re^N \to \Re$ is a Lipschitz continuous function such that $r(z) \geq 0$, $\forall z \in \Re^N$, meaning that the interest rate is always nonnegative.

Further, there are M risky assets, whose prices $S_t = (S_t^{(1)}, \ldots, S_t^{(M)})^\top$ follow

$$\frac{dS_t}{S_t} = \mu_s(Z_t)\,dt + \sigma_s(Z_t)\,dB_t,$$

where $\mu_s : \Re^N \to \Re^M$, $\sigma_s : \Re^N \to \Re^{M\times D}$, $\mathrm{tr}(\sigma_s(\cdot)\sigma_s(\cdot)^\top) : \Re^N \to \Re$ are Lipschitz continuous functions, and $\sigma_s(z)$ has full rank for all $z \in \Re^N$.

Note that Merton's classical intertemporal model is a special case of this model in which r, μ_s, and σ_s are constants. The model we consider generalizes this classical model by allowing the set of investment opportunities to vary with a stochastic market state process.

1.3.2 Portfolio Choice

We consider an investor who manages a portfolio, which he can rebalance at any time without incurring any transaction cost. The portfolio can include long and short positions in any asset, though there can be margin constraints. In mathematical terms, the portfolio is represented as a vector $\psi \in \Psi$, where Ψ is a polytope in $\Re^M$ containing the origin. Each component $\psi^{(m)}$ indicates the fraction of an investor's wealth invested in the mth risky asset. The fraction of wealth invested in the money market is denoted by

$$\psi^{(0)} = 1 - \sum_{m=1}^{M} \psi^{(m)}(x).$$

Note that these fractions can be greater than one or negative, because an investor may trade on margin or sell short.

From the perspective of an investor with wealth $W_t > 0$, the state at time t is $X_t = (Z_t, W_t) \in \Re^N \times \Re_{++}$. Let the state space be denoted by $\mathcal{S} = \Re^N \times \Re_{++}$. A *portfolio function* $\theta : \mathcal{S} \to \Psi$ maps the state to a portfolio. Let the set of portfolio functions be denoted by Θ.

A *portfolio strategy* is a continuum of portfolio functions $\theta = \{\theta_t \in \Theta | t \geq 0\}$. If an investor employs a portfolio strategy θ, his wealth process follows

$$\frac{dW_t}{W_t} = \big(r(Z_t) + \theta_t(X_t)^\top \lambda(Z_t)\big)\, dt + \theta_t(X_t)^\top \sigma_s(Z_t)\, dB_t, \tag{1.8}$$

where $\lambda(z) = \mu(z) - r(z)\mathbf{1}$ is the vector of excess rates of return.

We denote the set of admissible portfolio strategies by $\overline{\Theta}$. A portfolio strategy θ is *stationary* if for all x, t, and s, $\theta_t(x) = \theta_s(x)$. When referring to a portfolio function $\theta_t \in \Theta$ associated with a stationary strategy θ, we will drop the subscript, denoting the function by $\theta \in \Theta$. Since there is a one-to-one correspondence between portfolio functions and stationary strategies, we will also denote the set of stationary strategies by Θ.

1.3.3 Utility Function

We represent investor preferences using a power utility function over his wealth w:

$$u_\beta(w) = \frac{w^{1-\beta}}{1-\beta}.$$

Here, the parameter $\beta > 0$ captures an investor's level of risk aversion. In fact, the Arrow-Pratt coefficient of relative risk aversion for this power utility function is

$$-\frac{w u''_\beta(w)}{u'_\beta(w)} = \beta,$$

which is a constant. Note that the limiting case of $\beta \to 1$ concurs with the logarithmic utility function, whose coefficient of relative risk aversion is 1. It is well known that when the investor has logarithmic utility, myopic strategies are optimal.

We aim to maximize expected utility at a random time $\tilde{\tau}$, which is independent of the evolution of the market and is exponentially distributed with mean $\tau > 0$. Hence, the optimization problem of interest is

$$\sup_{\theta \in \overline{\Theta}} E_{x,\theta}\left[u_\beta(W_{\tilde{\tau}})\right],$$

or equivalently,

$$\sup_{\theta \in \overline{\Theta}} E_{x,\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u_\beta(W_t) dt\right].$$

The subscripts in the expectation indicate that the initial state is $X_0 = x = (z, w)$ and that the portfolio strategy $\theta \in \overline{\Theta}$ is employed. We will generally suppress the parameter β, and use $u(w)$ to denote $u_\beta(w)$.

1.3.4 Relation to the Controlled Diffusion Framework

The correspondence between our dynamic portfolio optimization problem and the controlled diffusion framework described in Section 1.1 is straightforward. The portfolio optimization problem is a controlled diffusion for which states are comprised of market states and wealth levels and for which actions are portfolio choices. The drift and diffusion functions μ and σ can be derived from those of the market state (μ_z and σ_z) and those of the asset prices (μ_s and σ_s). Utility in the portfolio optimization problem is a function of state through wealth. The discount rate is $\alpha = 1/\tau$.

Assumptions on the problem primitives posed in Section 1.1 carry over with one notable exception that the state space in the portfolio optimization problem is not compact. When the state space is noncompact, additional technical conditions are required to support HJB Equation and linear programming characterizations of the optimal value function. Such technical conditions and results for the portfolio optimization context are provided in the first author's dissertation [11]. We will not discuss them here. Rather, we will focus on application of our approximation algorithms, which apply readily to problems with compact or noncompact state spaces.

1.3.5 Special Structure

The portfolio optimization problem possesses special structure that simplifies computation of greedy actions and facilitates basis function selection. First of all, $(H_\psi J)(x)$ is a concave quadratic function of ψ. Specifically,

$$
\begin{aligned}
(H_\psi J)(x) \quad = \quad & J_w(x) w \big(\psi^\top \lambda(z) + r(z)\big) + \frac{1}{2} J_{ww}(x) w^2 \psi^\top \sigma_s(z) \sigma_s(z)^\top \psi \\
& + J_z(x)^\top \mu_z(z) + w J_{wz}(x) \sigma_z(z) \sigma(z)^\top \psi \\
& + \frac{1}{2} \mathrm{tr}\big(J_{zz}(x) \sigma_z(z) \sigma_z(z)^\top\big) \\
& - J(x)/\tau + u(w),
\end{aligned}
$$

where $x = (z, w)$. Since Ψ is a polytope, the problem of computing a greedy action is a convex quadratic program.

The use of a power utility function gives rise to some useful special structure in the optimal value function, as captured by the following well-known result.

Theorem 1.3.1. *Under Assumption 1.1.1, there exists a twice continuously differentiable function $V^* : \Re^N \to \Re$ such that*

$$J^*(z,w) = u(w)V^*(z), \quad \forall w > 0, z \in \Re^N.$$

Proof. By definition, we have

$$\begin{aligned} J^*(z,w) &= \sup_{\theta \in \overline{\Theta}} J_\theta(z,w) \\ &= \sup_{\theta \in \overline{\Theta}} E_{(z,w),\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u(w_t)\, dt\right] \\ &= \sup_{\theta \in \overline{\Theta}} w^{1-\beta} E_{(z,w),\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u\left(\frac{w_t}{w}\right)\, dt\right] \\ &= w^{1-\beta} \sup_{\theta \in \overline{\Theta}} E_{(z,1),\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u(w_t)\, dt\right]. \end{aligned}$$

So if we define $V^* : \Re^N \to \Re$ by

$$V^*(z) = (1-\beta) \sup_{\theta \in \overline{\Theta}} E_{(z,1),\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u(w_t)\, dt\right],$$

then we have

$$J^*(z,w) = u(w)V^*(z).$$

□

For value functions J that factor in the same way as J^*, the following theorem reflects the dependence of $H_\psi J$ on wealth.

Theorem 1.3.2. *For all $\psi \in \Psi$ and $J \in C^2$ for which $J(z,w) = u(w)V(z)$,*

$$\frac{(H_\psi J)(z,w)}{u(w)} = \frac{(H_\psi J)(z,\overline{w})}{u(\overline{w})},$$

for any $(z,w) \in \mathcal{S}$, and $\overline{w} \in \Re_+$, *Proof.* We have

$$
\begin{aligned}
(H_\psi J)(x) &= J_w(x)w\big(\psi^\top\lambda(z)+r(z)\big)+\frac{1}{2}J_{ww}(x)w^2\psi^\top\sigma_s(z)\sigma_s(z)^\top\psi \\
&\quad +J_z(x)^\top\mu_z(z)+wJ_{wz}(x)\sigma_z(z)\sigma(z)^\top\psi \\
&\quad +\frac{1}{2}\mathrm{tr}\big(J_{zz}(x)\sigma_z(z)\sigma_z(z)^\top\big) \\
&\quad -J(x)/\tau+u(w) \\
&= (1-\beta)u(w)V(z)\big(\psi^\top\lambda(z)+r(z)\big)+\frac{1}{2}\beta(\beta-1)u(w)V(z)\psi^\top\sigma_s(z)\sigma_s(z)^\top\psi \\
&\quad +u(w)V_z(z)^\top\mu_z(z)+(1-\beta)u(w)V_z(z)\sigma_z(z)\sigma(z)^\top\psi \\
&\quad +\frac{1}{2}\mathrm{tr}\big(u(w)V_{zz}(a)\sigma_z(z)\sigma_z(z)^\top\big) \\
&\quad -u(w)V(z)/\tau+u(w) \\
&= u(w)\Bigg(1-\beta)V(z)\big(\psi^\top\lambda(z)+r(z)\big)+\frac{1}{2}\beta(\beta-1)V(z)\psi^\top\sigma_s(z)\sigma_s(z)^\top\psi \\
&\quad +V_z(z)^\top\mu_z(z)+(1-\beta)V_z(z)\sigma_z(z)\sigma(z)^\top\psi \\
&\quad +\frac{1}{2}\mathrm{tr}\big(V_{zz}(a)\sigma_z(z)\sigma_z(z)^\top\big) \\
&\quad -u(w)V(z)/\tau+1\Bigg).
\end{aligned}
$$

The result follows. □

An immediate corollary of Theorem 1.3.2 is a well-known result concerning optimal portfolio strategies for investors with power utility.

Corollary 1.3.1. *There exists a policy $\theta \in \Theta$ that is greedy with respect to J^* for which $\theta(z,w)$ does not depend on w for $(z,w) \in \mathcal{S}$.*

1.3.6 Factorization of Basis Functions

In light of Theorem 1.3.1, the function $J^*(z,w)$ that we wish to approximate factors into $u(w)V^*(z)$. We know u but not V^*, which is in essence what we need to approximate. As such, it is natural to choose basis functions that factor in the same way. In particular, we will use basis functions that take the form $\phi_k(z,w) = u(w)\tilde{\phi}_k(z)$ for some functions $\tilde{\phi}_k(z)$. Then, weights r are computed with an aim of approximating J^* by $\sum_{k=1}^K r_k\phi_k$ or, equivalently, V^* by $\sum_{k=1}^K r_k\tilde{\phi}_k$.

By Theorems 1.3.1 and 1.3.2, the linear program (1.5) we aim to solve can be rewritten as:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{Q} u(w^{(i)}) \sum_{k=1}^{K} r_k \tilde{\phi}_k(z^{(i)}) \\ \text{subject to} & (H_{\psi^{(i)}} \Phi r)(z^{(i)}, 1) \leq 0, \qquad\qquad \forall i = 1, \ldots, Q, \end{array}$$

where $x^{(i)} = (z^{(i)}, w^{(i)})$. Hence, the $w^{(i)}$ samples do not enter into the constraints and only influence the objective by weighting the values $\sum_{k=1}^{K} r_k \tilde{\phi}_k(z^{(i)})$ associated with $z^{(i)}$ samples.

1.3.7 Measure and Sampling

Loosely guided by the results of [7, 9], we consider a measure ρ associated with discounted relative state frequencies. First, define the discounted relative frequency measure for the market state process:

$$\tilde{\rho}(dz) = \frac{1}{\tau} E\left[\int_{t=0}^{\infty} e^{-t/\tau} \mathbf{1}(Z_t \in dz) dt\right].$$

The relative frequencies associated with X_t depend also on the evolution of wealth and therefore on the portfolio strategy in use. It is not clear how a measure should be defined here, but as a simple heuristic, we will use the measure

$$\rho(dz, dw) = \tilde{\rho}(dz) \mathbf{1}(1 \in dw).$$

Note that the conditional measure over w, conditioned on z, bears no impact on the sampling of constraints since they do not depend on the $w^{(i)}$s. There is, however, some impact on the objective through the weights $u(w^{(i)})$.

To construct the linear program, we must generate Q independent identically distributed state samples. Since wealth is constant under our measure, we need only to sample $z^{(1)}, \ldots, z^{(Q)}$. We generate each sample by simulating a discrete-time approximation to the market state dynamics, terminating the simulation at an exponentially distributed stopping time with expectation τ. Given the simulated trajectory, we uniformly sample a time between 0 and the stopping time and take as our state sample the state at that time.

1.3.8 Case Studies

In this subsection, we will present numerical results from two cases that make use of the adaptive constraint selection algorithm (1).[1] The first involves a problem that admits a simple closed-form solution and serves as a sanity test for our algorithm. For this problem, the algorithm appears to always deliver exact solutions. The second case study involves a ten-factor term structure model for which exact solution is likely to be intractable. Our computations make use of ILOG CPLEX to solve linear programs and quadratic programs.

Case Study 1: Constant Investment Opportunities

In Merton's classical dynamic portfolio optimization model [17], the set of investment opportunities is constant. This enables solution of the HJB equation and derivation of an optimal portfolio strategy in closed form. We will consider such a model. Specifically, we consider constant asset price drift and diffusion functions:

$$\mu_s(z) = \mu_s, \quad \sigma_s(z) = \sigma_s, \quad r(z) = r, \quad \lambda(z) = \lambda = \mu_s - r.$$

We impose no constraints on portfolio choice, so $\Psi = \Re^M$.

In this model, the drift and diffusion functions do not depend on any market state. It is therefore natural to think of the market state being constant. However, this would lead to a trivial computational problem as there would be no function V^* to approximate. Our intention here is to test our approximation algorithm, and as such, we model the market state as a five-dimensional Ornstein-Uhlenbeck process:

$$dz_t = -z_t\, dt + \sigma_z\, dB_t,$$

where $\sigma_z \in \Re^{5\times 5}$ is a constant and full rank matrix and B_t is a 5-dimensional Brownian motion. The function V^* now maps $\Re^5$ to $\Re$, but is constant. We try approximating V^* using a polynomial to see whether our algorithm produces the desired constant function.

We first provide an analytic derivation of the optimal value function and

[1]We experimented with the convex programming approach, but found that not to be efficient enough to address problems of practical scale.

policy. The HJB equation can be written in terms of V^* as

$$0 = \min_{\theta\in\Theta} \Big\{ (1-\beta)V^*(z)\big(\theta(x)^\top\lambda + r\big) + \frac{\beta(\beta-1)}{2}V^*(z)\theta(x)^\top\sigma_s\sigma_s^\top\theta(x) \\ + V_z^*(z)^\top\mu_z + (1-\beta)V_z^*(z)^\top\sigma_z(z)\sigma_s^\top\theta(x) \\ + \frac{1}{2}\mathrm{tr}\big(V_{zz}^*(z)\sigma_z(z)\sigma_z(z)^\top\big) - V^*(z)/\tau + 1 \Big\}.$$

The first-order condition gives a candidate for the optimal strategy:

$$\theta_V^*(z) = \frac{1}{\beta}\left(\sigma_s\sigma_s^\top\right)^{-1}\lambda + \frac{1}{\beta V^*(z)}\left(\sigma_s\sigma_s^\top\right)^{-1}\sigma_s\sigma_z(z)^\top V_z^*(z).$$

Plugging this portfolio strategy into the HJB equation leads to

$$0 = \frac{1}{V^*} + \big((1-\beta)r - \frac{1}{\tau}\big) + \frac{1-\beta}{2\beta}\lambda^\top(\sigma_s\sigma_s^\top)^{-1}\lambda + \frac{1-\beta}{\beta}\lambda^\top(\sigma_s\sigma_s^\top)^{-1}\sigma_s\sigma_z^\top\left(\frac{V_z^*}{V^*}\right) \\ + \frac{1-\beta}{2\beta}\left(\frac{V_z^*}{V^*}\right)^\top\sigma_z\sigma_s^\top(\sigma_s\sigma_s^\top)^{-1}\sigma_s\sigma_z^\top\left(\frac{V_z^*}{V^*}\right) + \mu_z^\top\left(\frac{V_z^*}{V^*}\right) + \frac{1}{2}\mathrm{tr}\left(\frac{V_{zz}^*}{V^*}\sigma_z\sigma_z^\top\right).$$

It is easy to check that this equation has a constant solution:

$$V^*(z) = \left(\frac{1}{\tau} - (1-\beta)r - \frac{1-\beta}{2\beta}\lambda^\top\left(\sigma_s\sigma_s^\top\right)^{-1}\lambda\right)^{-1}.$$

Hence, the HJB equation is solved by

$$J^*(z,w) = u(w)V^*(z).$$

Using techniques from [10, Chapter 9], it can be proved that J^* and θ^* are indeed the optimal value function and an optimal policy.

We now consider application of our approximation algorithm. We will employ as basis functions tensor products of Chebyshev polynomials. The

Chebyshev polynomials are

$$\begin{aligned}
P_0(x) &= 1, \\
P_1(x) &= x, \\
P_2(x) &= 2x^2 - 1, \\
P_3(x) &= 4x^3 - 3x, \\
P_4(x) &= 8x^4 - 8x^2 + 1, \\
P_5(x) &= 16x^5 - 20x^3 + 5x, \\
P_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1, \\
&\vdots
\end{aligned}$$

Chebyshev polynomials are orthogonal, and this reduces numerical stabilities that can arise in computations. The basis functions we use are the complete polynomials up to third degree:

$$\left\{P_i(z^{(a)})P_j(z^{(b)})P_k(z^{(c)}), \quad i, j, k \geq 0, i + j + k \leq 3, a \neq b, b \neq c, c \neq a\right\}$$

We tried the adaptive constraint selection algorithm many times with different problem data and independently sampled sets each of 500, 5,000, or 50,000 states. Each time, the approximation converged to the correct function

$$V^*(z) = \left(\frac{1}{\tau} - (1-\beta)r - \frac{1-\beta}{2\beta}\lambda^\top \left(\sigma\sigma^\top\right)^{-1} \lambda\right)^{-1}$$

within two iterations.

Case Study 2: Ten-Factor Term Structure Model

In this section we consider a high-dimensional problem for which exact solution is likely to be intractable. We use a ten-factor CIR model [6], and our choice of problem data is guided by results of the empirical study of a three-factor CIR model [5].

The market state $Z_t \in \Re^{10}$ follows

$$dZ_t^{(i)} = \kappa_i(\zeta_i - Z_t^{(i)})\,dt + \sigma_i\sqrt{Z_t^{(i)}}\,dB_t^{(i)}, \quad i = 1, 2, \ldots, 10,$$

where κ_i, ζ_i, and σ_i are all positive constants. So

$$\begin{aligned}
\mu_z(z) &= \begin{pmatrix} \kappa_1 & 0 & \cdots & 0 \\ 0 & \kappa_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \kappa_{10} \end{pmatrix} \left(\begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_{10} \end{pmatrix} - z \right), \\
\sigma_z(z) &= \begin{pmatrix} \sigma_1\sqrt{z^{(1)}} & 0 & \cdots & 0 \\ 0 & \sigma_2\sqrt{z^{(2)}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{10}\sqrt{z^{(10)}} \end{pmatrix}.
\end{aligned}$$

The spot rate r as a function of market state is given by

$$r(z) = z^{(1)} + z^{(2)} + \ldots + z^{(10)}.$$

Let $P(z_t, T-t)$ be the price of a zero coupon bond with $T-t$ periods until maturity. This price is given by

$$P(z, T-t) = \prod_{i=1}^{10} \left(A_i(T-t) e^{-B_i(T-t) z^{(i)}} \right),$$

where

$$\begin{aligned}
A_i(T) &= \left[\frac{2\gamma_i e^{\frac{1}{2}(\kappa_i + \lambda_i - \gamma_i)T}}{2\gamma_i e^{-\gamma_i T} + (\kappa_i + \lambda_i + \gamma_i)(1 - e^{-\gamma_i T})} \right]^{\frac{2\kappa_i \zeta_i}{\sigma_i^2}}, \\
B_i(T) &= \frac{2(1 - e^{-\gamma_i T})}{2\gamma_i e^{-\gamma_i T} + (\kappa_i + \lambda_i + \gamma_i)(1 - e^{-\gamma_i T})},
\end{aligned}$$

where

$$\gamma_i = \sqrt{(\kappa_i + \lambda_i)^2 + 2\sigma_i^2}.$$

Price dynamics follow

$$\frac{dP(z_t, T-t)}{P(z_t, T-t)} = \sum_{i=1}^{10} \left(z^{(i)} [1 - \lambda_i B_i(T-t)]\, dt - B_i(T-t) \sigma_i \sqrt{z^{(i)}}\, dB_t^{(i)} \right).$$

So

$$\mu(z) = \begin{pmatrix} 1-\lambda_1 B_1(T_1-t) & 1-\lambda_2 B_2(T_1-t) & \cdots & 1-\lambda_{10} B_{10}(T_1-t) \\ 1-\lambda_1 B_1(T_2-t) & 1-\lambda_2 B_2(T_2-t) & \cdots & 1-\lambda_{10} B_{10}(T_2-t) \\ \vdots & \vdots & \ddots & \vdots \\ 1-\lambda_1 B_1(T_{10}-t) & 1-\lambda_2 B_2(T_{10}-t) & \cdots & 1-\lambda_{10} B_{10}(T_{10}-t) \end{pmatrix} z,$$

$$\sigma(z) = \begin{pmatrix} -B_1(T_1-t)\sigma_1\sqrt{z^{(1)}} & -B_2(T_1-t)\sigma_2\sqrt{z^{(2)}} & \cdots & -B_{10}(T_1-t)\sigma_{10}\sqrt{z^{(10)}} \\ -B_1(T_2-t)\sigma_1\sqrt{z^{(1)}} & -B_2(T_2-t)\sigma_2\sqrt{z^{(2)}} & \cdots & -B_{10}(T_2-t)\sigma_{10}\sqrt{z^{(10)}} \\ \vdots & \vdots & \ddots & \vdots \\ -B_1(T_{10}-t)\sigma_1\sqrt{z^{(1)}} & -B_2(T_{10}-t)\sigma_2\sqrt{z^{(2)}} & \cdots & -B_{10}(T_{10}-t)\sigma_{10}\sqrt{z^{(10)}} \end{pmatrix},$$

where T_i are the maturity of the i-th zero coupon bond. In this case study, we let $T_i = i$, $\forall i = 1, \ldots, 10$.

An empirical study [5] produced the following estimates for three-factor model of a real market:

$$\begin{aligned} \kappa &= (1.4298, 0.01694, 0.03510), \\ \zeta &= (0.04374, 0.002530, 0.003209), \\ \sigma &= (0.16049, 0.1054, 0.04960), \\ \lambda &= (-0.2468, -0.03411, -0.1569). \end{aligned}$$

Based on these estimates and an interest in considering a model of higher dimension, we will use the following parameter values:

$$\begin{aligned} \kappa &= (1.4298, 0.01694, 0.03510, 0.03510, \ldots, 0.03510), \\ \zeta &= (0.04374, 0.002530, 0.003209, 0.001, 0.001, \ldots, 0.001), \\ \sigma &= (0.16049, 0.1054, 0.04960, 0.04960, \ldots, 0.04960), \\ \lambda &= (-0.2468, -0.03411, -0.1569, -0.1569, \ldots, -0.1569). \end{aligned}$$

Note that with this problem data, the first component of Z_t is generally an order of magnitude larger than any other component. With this in mind, we selected the following basis functions:

1. Chebyshev polynomials for $z^{(1)}$ up to sixth degree:

$$\{1, P_1(z^{(1)}), P_2(z^{(1)}), P_3(z^{(1)}), P_4(z^{(1)}), P_5(z^{(1)}), P_6(z^{(1)})\}$$

2. Chebyshev polynomials for $z^{(i)}$ up to second degree, $\forall i = 2, \ldots, 10$:

$$\{P_1(z^{(i)}), P_2(z^{(i)}), i = 2, \ldots, 10\}$$

3. Second-degree cross-product terms between the first component and the other components:

$$\{P_1(z^{(1)})P_1(z^{(i)}), i = 2, \ldots, 10\}$$

This generates a total of 34 basis functions.

We sampled 10,000 market states by simulating 10,000 trajectories using a model with time discretized into steps of size 0.01. Each step here represents about half a calendar week, which seems like a reasonably small time period for fixed income portfolio rebalancing considerations. It takes about 20 minutes to execute the adaptive constraint selection algorithm and arrive at an approximation to the value function. We used C++ code in concern with ILOG CPLEX on a Sun Blade 2000 machine. We used as an initial state $z = (0.03, 0.00253, 0.003209, 0.001, 0.001, \ldots, 0.001)$, a horizon time $\tau = 2$, and several levels of risk aversion: $\beta \in \{1.1, 2, 3, 4, 5, 6\}$.

Since we do not know the optimal portfolio strategy, we will use simple heuristics as benchmarks for performance comparison. One heuristic we consider is the myopic strategy, which can be thought of as the greedy strategy with respect to an approximate value function $J(z, w) = u(w)$. A second heuristic we consider is the risk-free strategy, which maintains all funds in the money market, earning the instantaneous risk-free rate at each time.

Another basis for comparison is provided by the approximate value function Φr. In particular, if Φr satisfies the constraints $(H\Phi r)(x) \leq 0$ for all $x \in \mathcal{S}$ then $\Phi r \geq J^*$. In this case, $(\Phi r)(x_0)$ would provide an upper bound on performance of an optimal portfolio strategy. In our context, Φr satisfies only a sampled subset of such constraints. Regardless, one might think of $(\Phi r)(x_0)$ as an approximation to an upper bound and compare this to the performance of heuristic portfolio strategies.

One way to measure performance of a portfolio strategy θ is in terms of our objective

$$E_{x,\theta}\left[\int_{t=0}^{\infty} e^{-t/\tau} u_\beta(W_t) dt\right],$$

with $x = (z_0, 1)$. Let us denote this objective value by U. One issue with this measure of performance is that it can be difficult to interpret. We consider a more easily interpreted measure defined by the constant rate of return r^{ce} that would attain the same objective value. In particular, r^{ce} solves

$$U = \tau E \left[\int_{t=0}^{\infty} e^{-t/\tau} u_\beta(e^{r^{ce} t}) dt \right],$$

and can be written as

$$r^{ce} = \frac{1}{\tau(1-\beta)} - \frac{1}{U(1-\beta)^2}. \tag{1.9}$$

We will refer to r^{ce} as the *certainty equivalent return rate* of the associated portfolio strategy.

We use Monte Carlo simulation to assess certainty equivalent return rates for portfolio strategies resulting from solving the linear program, as well as myopic and risk-free strategies. This involves simulating sample paths for a discrete-time model to estimate the objective values and then converting them to estimates of certainty equivalent return rates according to (1.9). When simulating a discrete-time model, the portfolio is revised based on the strategy in use at the beginning of each time period. To obtain each objective value estimate, we simulate 40,000 sample paths for a model with time steps of duration 0.01. It takes hours to estimate certainty equivalent return rates for the three policies, and we have observed almost no difference in cases where we have compared estimates generated from 40,000 and 80,000 sample paths.

Table 1.1 presents certainty equivalent return rates from our experiments. Further, the right-most column offers approximate upper bounds given by the approximate value function, evaluated at the initial state and converted to units of a certainty equivalent return rate. Our results indicate that strategies generated by the linear programming approach significantly outperform myopic strategies across a broad range of levels of risk aversion. Both types of strategies greatly outperform risk-free strategies. Further, performance of the LP-based strategies generally exceed the approximate upper bounds.

1.4 Closing Remarks

The linear programming approach of this chapter extends one developed for discrete time problems [19, 20, 7, 8, 9]. One might alternatively discretize a continuous problem, say using the techniques of [15]. However, there are

β	Certainty equivalent return rate (%)			
	LP Strategy	Myopic Strategy	Risk-Free Strategy	LP Value
1.1	17.4218	17.2734	8.6298	17.3907
2	12.2776	12.1151	5.6263	12.2710
3	10.2067	10.0090	5.4624	10.1140
4	9.0417	8.8739	5.4563	8.9293
5	8.2736	8.0785	5.4578	8.1706
6	8.1728	7.9365	5.4653	8.1516

Table 1.1: Performance comparison

several advantages to working directly with a continuous-time model. First, constraints in the discrete-time version of the linear program involve the discrete-time dynamic programming operator and therefore expressions with one-step expectations. Dealing with these expectations becomes computationally taxing as the dimension of the state space grows. Second, the optimization problems that must be solved to determine greedy actions are often more complex for discrete-time models. As we saw with portfolio optimization, these problems often amount to simple convex programs in continuous-time contexts. Finally, one might argue for aesthetics of working directly with a continuous time model rather than an auxiliary model, especially when there are no practical benefits to discretization.

Many controlled diffusion problems can be addressed by the linear programming approach. In the area of dynamic portfolio optimization alone, it would be interesting to explore models involving transaction costs or taxes, either of which greatly increases model complexity. Some preliminary work that applies the approach to transaction cost optimization is reported in [22].

It is also worth noting that for a large variety of continuous-time control problems, our approach may be inappropriate. In particular, the approach relies on certain regularities and may not work well when state or control actions need to change abruptly or when the utility function is not smooth.

Bibliography

[1] N. Barberis. Investing for the long run when returns are predictable. *The Journal of Finance*, 55(1):225–265, 2000.

[2] V. Borkar. *Optimal Control of Diffusion Processes*. Longman, Harlow, England, 1989.

[3] M. Brandt. Estimating portfolio and consumption choice: A conditional Euler equation approach. *The Journal of Finance*, 54(5):1609–1645, 1999.

[4] M. Brandt, A. Goyal, P. Santa-Clara, and J. Stroud. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, forthcoming.

[5] R.-R. Chen and L. Scott. Multi-factor Cox-Ingersoll-Ross models of the term structure: Estimates and tests from a Kalman filter model. 2002.

[6] J. Cox, J. Ingersoll, and S. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385–408, Mar. 1985.

[7] D. P. de Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[8] D. P. de Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[9] D. P. de Farias and B. Van Roy. A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees. *Mathematics of Operations Research*, 31(3):597–620, 2006.

[10] D. Duffie. *Dynamic Asset Pricing Theory*. Princeton University Press, Princeton, New Jersey, third edition, 2001.

[11] J. Han. *Dynamic Portfolio Management: An Approximate Linear Programming Approach*. PhD thesis, Stanford University, 2005.

[12] M. Haugh, L. Kogan, and Z. Wu. Approximate dynamic programming and duality for portfolio optimization. In preparation, Forthcoming.

[13] K. Judd and J. Gaspar. Solving large-scale rational-expectations models. *Macroeconomic Dynamics*, 1:45–75, 1997.

[14] N. V. Krylov. *Controlled Diffusion Processes*. Springer-Verlag, New York, NY, 1980. Translated by A. B. Aries.

[15] H. J. Kushner and P. G. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer, New York, NY, 2001.

[16] A. Lynch. Portfolio choice and equity characteristics: characterizing the hedging demands induced by return predictability. *Journal of Financial Economics*, 62:67–130, 2001.

[17] R. Merton. Optimum consumption and portfolio rules in a continuous-time model. *Journal of Economic Theory*, 3:373–413, 1971.

[18] C. Munk. Optimal consumption/investment policies with undiversifiable income risk and liquidity constraints. *Journal of Economic dynamics and Control*, 24(9):1315–1343, 2000.

[19] P. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.

[20] M. Trick and S. Zin. Spline approximations to value functions: A linear programming approach. *Macroeconomic Dynamics*, 1:255277, 1997.

[21] A. Wang. *Approximate Value Iteration Approaches to Constrained Dynamic Portfolio Problems*. PhD thesis, MIT, 2003.

[22] X. Yan. Transaction-cost-conscious pairs trading via approximate dynamic programming, 2005. Computer Science Honors Thesis, Stanford University.