Stochastic Optimal Control Based on Monte Carlo Simulation and Least-Squares Regressiond

Cong, Fei

DOI
10.4233/uuid:43aceb1b-b125-4f05-ad37-102fa1c388f7

Publication date
2016

Document Version
Final published version

Citation (APA)
Cong, F. (2016). *Stochastic Optimal Control Based on Monte Carlo Simulation and Least-Squares Regressiond*. [Dissertation (TU Delft), Delft University of Technology].
https://doi.org/10.4233/uuid:43aceb1b-b125-4f05-ad37-102fa1c388f7

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Stochastic Optimal Control Based on Monte Carlo Simulation and Least-Squares Regression

STOCHASTIC OPTIMAL CONTROL BASED ON MONTE CARLO SIMULATION AND LEAST-SQUARES REGRESSION

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K. C. A. M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 19 december 2016 om 10:00 uur

door

Fei CONG

wiskundig ingenieur
geboren te Xi'an, China.

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. C. W. Oosterlee

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. ir. C. W. Oosterlee	Technische Universiteit Delft, promotor

Onafhankelijke leden:

Prof. dr. P. A. Forsyth	University of Waterloo, Canada
Prof. dr. D. Sevcovic	Comenius University, Slowakije
Prof. dr. A. A. J. Pelsser	Maastricht University
Prof. dr. J. M. Schumacher	Tilburg University
Prof. dr. ir. K. I. Aardal	Technische Universiteit Delft
Prof. dr. ir. A. W. Heemink	Technische Universiteit Delft
Prof. dr. ir. G. Jongbloed	Technische Universiteit Delft, reservelid

Stochastic Optimal Control Based on Monte Carlo Simulation and Least-Squares Regression
Dissertation at Delft University of Technology

ISBN 978-94-6186-753-7

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Summary

In the financial engineering field, many problems can be formulated as stochastic control problems. A unique feature of the stochastic control problem is that uncertain factors are involved in the evolution of the controlled system and thus the objective function in the stochastic control is typically formed by an expectation operator. There are in general two approaches to solve this kind of problems. One can reformulate the problem to be a deterministic problem and solve the corresponding partial differential equation. Alternatively, one calculates conditional expectations occurring in the problem by either numerical integration or Monte Carlo methods.

We focus on solving various types of multi-period stochastic control problems via the Monte Carlo approach. We employ the Bellman dynamic programming principle so that a multi-period control problem can be transformed into a composition of several single-period control problems, that can be solved recursively. For each single-period control problem, conditional expectations with different filtrations need to be calculated. In order to avoid nested simulation (i.e. Monte Carlo simulation within a Monte Carlo simulation), which may be very time consuming, we implement Monte Carlo simulation and cross-path least-squares regression. So-called "regress-later" and "bundling" approaches are introduced in our algorithms to make them highly accurate and robust. In most cases, high quality results can be obtained within seconds.

Chapter 1 gives a general introduction of the multi-period stochastic control problem and the Bellman dynamic programming principle. We elaborate on the special features, i.e. the "regress-later" and "bundling" approaches, of the simulation-based numerical algorithms implemented by us. We utilize this algorithm to solve four types of problems, including: (1) a Bermudan option pricing problem, (2) a multi-period utility-based portfolio optimization problem, (3) a multi-period target-based mean-variance optimization problem and (4) a multi-period time-consistent mean-variance optimization problem.

Chapter 2 deals with Bermudan option pricing problems with Merton jump-diffusion asset dynamics. We compare the newly-designed regression method with the standard regression method in an error analysis. Regarding the choice of basis functions for regression and the bundling technique, we propose a uniform way to configure our numerical method. This uniform setting is implemented throughout this thesis. Control variates are introduced for achieving effective variance reduction for Bermudan option pricing problems.

Starting with Chapter 3, we investigate dynamic portfolio optimization problems. In these problems, in order to achieve the optimal performance of a portfolio which is measured by a target function, an investor dynamically manages the portfolio by specifying the fraction of the wealth in the risk-free asset and that in the risky assets. In general, risky assets yield higher expected returns than the risk-free asset but also lead to higher risk. We first work on a utility-based portfolio optimization problem, in which a constant relative risk aversion (CRRA) utility function is considered as the target function.

Different from the literature, we introduce a new way to approximate the CRRA utility functions with a Taylor expansion. Combining this Taylor expansion with our numerical algorithms for calculating conditional expectations yields more accurate and robust results compared to existing literature. These results are confirmed by a benchmark algorithm, by which we calculate conditional expectation with a numerical integration method using a Fourier cosine transformation technique.

A multi-period mean-variance portfolio optimization problem is discussed in Chapter 4. Due to the occurrence of the variance operator in the target function, this kind of problem cannot be solved directly by using the Bellman dynamic programming principle. We first apply an embedding technique to transform the multi-period mean-variance optimization problem to an equivalent multi-period quadratic optimization problem. This quadratic optimization problem is also termed a *target-based* or a *pre-commitment* optimization problem. Since the traditional Monte Carlo approaches cannot be applied to solve this problem, we develop a forward-backward numerical scheme for solving this problem. In the forward process, we perform Monte Carlo simulation with a sub-optimal strategy. In the backward process, we locally solve the optimality to improve the sub-optimal strategy, which is used in the forward process. Iterating the forward-backward approach can yield convergent results for the target-based optimization. In the numerical tests, it is shown that this result is identical to optimal solutions from the literature. Our method is highly efficient and generates high quality results in just a few seconds on a basic personal computer.

We still work on a multi-period mean-variance optimization problem in Chapter 5. Instead of adopting the embedding technique, we introduce time-consistency conditions into the problem so that the dynamic programming principle can be used. This problem is termed the *time-consistent* mean-variance optimization problem. We again utilize the forward-backward algorithm for solving the time-consistent problem and some variants of it. We find that, although in the literature the time-consistent strategy is known to generate lower mean-variance efficient frontiers than the target-based strategy, the time-consistent strategy is not always inferior since two strategies generate significantly different terminal wealth distributions.

In Chapter 6, we make a comparison between the target-based strategy and the time-consistent strategy when model prediction errors occur. Here the existence of the "model prediction error" means that the real-world market does not evolve as the model predicts and therefore the optimal controls determined based on the model information may be problematic in the real-world market. The time-consistent strategy is found to be robust in terms of model prediction errors. In some cases, the time-consistent strategy can even generate higher mean-variance efficient frontiers than the pre-commitment strategy, since "time-consistency" may serve as a protection for the investor. We also perform an analysis on the robust counterparts of both strategies. The robust strategies are required to perform well in the worst-case scenarios. We find that for both strategies the worst-case scenarios can be generated by solving a specific equation at each time step.

We note that all the work presented in this thesis is based on published or submitted papers written during the PhD research.

Samenvatting

In de financiële wiskunde kunnen veel problemen worden geformuleerd als stochastische regeltechniekproblemen. Een uniek kenmerk van een probleem binnen de stochastische regeltechniek is dat onzekere factoren een rol spelen in het te regelen systeem. Dit impliceert dat de doelfunctie die bij het probleem hoort vaak wordt weergegeven door een verwachtingsoperator. In het algemeen zijn er twee manieren om dergelijke stochastische regeltechniekproblemen op te lossen. Men kan het probleem herformuleren tot een deterministisch probleem en de bijbehorende partiële differentiaalvergelijking oplossen. Als alternatief kan men de conditionele verwachtingen in de probleemstelling berekenen door gebruik te maken van numerieke integratietechnieken of Monte-Carlosimulatie.

We richten ons op het oplossen van verschillende multi-periode stochastische regeltechniekproblemen met Monte Carlo benaderingen. We maken gebruik van Bellmans dynamisch programmeerprincipe, waardoor een multi-periode vraagstuk kan worden omgevormd tot een compositie van verschillende één-periode problemen, die recursief kunnen worden opgelost. Voor elk afzonderlijke één-periode probleem uit de regeltechniek moeten conditionele verwachtingen met verschillende filtraties worden uitgerekend. We kunnen een geneste simulatiemethode (dat wil zeggen, een Monte-Carlosimulatie binnen een Monte-Carlosimulatie), die zeer tijdrovend kan zijn, vermijden door middel van een combinatie van Monte-Carlosimulatie en kleinste-kwadratenregressie. Op deze manier kunnen de conditionele verwachtingen efficiënt berekend worden. Om de algoritmen zeer nauwkeurig en robuust te maken, introduceren we de volgende twee technieken “regress-later” en “bundling”. In de meeste gevallen kost het slechts seconden om resultaten met een hoge kwaliteit te verkrijgen met de verbeterde technieken.

Hoofdstuk 1 geeft een algemene introductie over multi-periode stochastisch regeltechniekproblemen en Bellmans dynamisch programmeerprincipe om deze problemen op te lossen. We beschrijven de speciale technieken, dat wil zeggen de “regress-later” en “bundling” benaderingen, die worden gebruikt in onze op simulatie gebaseerde numerieke algoritmen. We maken gebruik van dit algoritme om vier typen dynamische stochastische regeltechniekproblemen op te lossen: (1) het prijzen van een Bermuda optie, (2) het optimaliseren van de nutsfunctie van een multi-periode portefeuille, (3) een multi-periode gemiddelde-variantie optimalisatievraagstuk en (4) een multi-periode tijdsconsistent gemiddelde-variantie optimalisatieprobleem.

Hoofdstuk 2 houdt zich bezig met het prijzen van Bermuda opties, waarbij de onderliggende een Merton sprong-diffusie stochastisch model volgt. In een foutenanalyse vergelijken we de nieuw ontworpen regressietechniek met de standaard regressiemethode. Onze numerieke methode kan op een uniforme manier worden geconfigureerd door de basisfuncties voor de regressie en de manier van het bundelen van stochastische roosterpunten correct te kiezen. Een uniform algoritme wordt in dit proefschrift voorgesteld. Variabelen worden geïntroduceerd om een effectieve reductie van de variantie te verkrijgen bij het prijzen van Bermuda opties.

Vanaf Hoofdstuk 3 starten we met het onderzoeken van vraagstukken met betrekking tot het optimaliseren van tijdsafhankelijke portefeuilles. In deze vraagstukken beheert een investeerder een portefeuille met aandelen en obligaties op een dynamische manier, door te bepalen welk deel van van het vermogen in risicovrije en welk deel in risicovolle financiële producten geïnvesteerd zal worden, om zo een door een doelfunctie gegeven optimaal beleggingsresultaat te behalen. Over het algemeen hebben risicovolle financiële producten hogere verwachte opbrengsten dan risicovrije producten, maar de investeerder loopt dan wel meer risico. Allereerst werken we aan een het optimaliseren van een nutsfunctie gerelateerd aan een portefeuille. Hierin nemen we aan dat de optimaliseringsdoelfunctie gegeven wordt door een constante relatieve risico-aversie (CRRA) nutsfunctie. We introduceren een nieuwe manier om de CRRA nutsfuncties te benaderen, namelijk via een speciale taylorontwikkeling. Door deze taylorontwikkeling en de ontwikkelde numerieke algoritmen samen te voegen, verkrijgen we resultaten die nauwkeuriger en robuuster zijn dan reeds gepresenteerde resultaten in de literatuur. Onze resultaten worden bevestigd met behulp van een referentie-algoritme, waarin we de conditionele verwachting berekenen met een numerieke methode die gebaseerd is op Fourier cosinustransformaties.

Een multi-periode gemiddelde-variantie optimalisatieprobleem wordt behandeld in Hoofdstuk 4. Vanwege de aanwezigheid van een variantie-operator in de doelfunctie, kan dit probleem niet direct opgelost worden met Bellmans dynamisch programmeerprincipe. Om te beginnen transformeren we het probleem daarom naar een gelijkwaardig multi-periode kwadratisch optimalisatieprobleem. Dit gelijkwaardige probleem noemt men ook wel een *doelgericht* optimalisatieprobleem. Aangezien de traditionele Monte-Carlomethoden niet toegepast kunnen worden bij het oplossen van dit probleem, ontwikkelen we een numerieke methode die gebaseerd is op een voorwaartse-terugwaartse recursie. We voeren eerst een voorwaartse Monte-Carlosimulatie uit en verkrijgen een suboptimale regeltechniekstrategie. In het daaropvolgende, terugwaarts-in-de-tijd, proces optimaliseren we lokaal, om de suboptimale oplossing, die we in het voorwaartse proces verkregen, te verbeteren. Het itereren met deze voorwaartse-terugwaartse optimaliseringsmethode kan zorgen voor convergerende resultaten voor de optimalisatie van het specifieke doel. In de numerieke voorbeelden wordt getoond dat de gevonden oplossingen identiek zijn aan de optimale oplossingen, die gegeven zijn in de literatuur. Onze methode is uiterst efficiënt en genereert op een standaard computer in slechts enkele seconden optimale resultaten van een hoge kwaliteit.

Ook in Hoofdstuk 5 werken we aan een multi-periode gemiddelde-variantie optimalisatieprobleem. In plaats van de inbeddingstechniek uit het vorige hoofdstuk, introduceren we hier zogenaamde tijdsconsistente voorwaarden in het probleem. Hierdoor kan het dynamisch programmeerprincipe weer worden gebruikt. Dit resulterende probleem wordt ook wel het tijdsconsistente gemiddelde-variantie optimalisatieprobleem genoemd. Opnieuw gebruiken we het voorwaartse-terugwaartse recursie algoritme om een oplossing voor dit probleem en verwante vraagstukken te vinden. Volgens de literatuur genereert een tijdsconsistente optimaliseringsstrategie gewoonlijk minder efficiënte gemiddelde-variantie oplossing dan de andere, doelgerichte, strategie. Desondanks tonen wij dat een tijdsconsistente strategie niet altijd minder presteert, omdat de twee optimaliseringsstrategieën vermogensverdelingen genereren, die significant verschillen.

In Hoofdstuk 6 vergelijken we de twee strategieën in het scenario dat er onnauwkeurigheden in de voorspellingen voor de aandelenmodel in de toekomst voor komen. Met andere woorden, de concrete aandelenkoersen evolueren niet zoals het model voorspelt had. Daardoor is de optimale investeringsstrategie, zoals die bepaald was op basis van de informatie van het model, mogelijk problematisch in de concrete markt. De tijdsconsistente optimaliseringsstrategie blijkt het meest robuust met betrekking tot voorspellingsfouten van het model. In sommige gevallen genereert de tijdsconsistente strategie zelfs efficiëntere gemiddelde-variantie oplossingen dan de doelgerichte strategie, omdat "tijdsconsistentie" als een mogelijke bescherming kan dienen voor een investeerder. Ook verrichten we een foutenanalyse voor de robuuste tegenhanger van beide strategieën. Robuuste strategieën worden gebruikt om acceptabele investeringsstragiën te genereren in het slechtst mogelijke aandelenscenario. We vinden dat voor beide strategieën het slechtste scenario gegenereerd kan worden door het oplossen van een specifieke vergelijking op iedere tijdstap.

Het werk in dit proefschrift is gebaseerd op gepubliceerde of ingediende artikelen die voltooid zijn tijdens het promotieonderzoek.

Contents

CHAPTER 1

Multi-period Stochastic Optimization and Dynamic Programming

1.1. INTRODUCTION

Stochastic optimization refers to the problem of finding optimal controls in the presence of randomness in an optimization process. Compared to a common optimization problem in the engineering field, the main difficulty encountered in stochastic optimization is that the uncertainty has to be taken into account and thus the optimization object is usually formed by using an expectation operator and perhaps also a variance operator. In this chapter, we briefly describe the formulation of multi-period stochastic optimization problems and the dynamic programming approach, which can be utilized to solve this kind of problems. We elaborate on the special features of a newly developed numerical method based on Monte Carlo simulation and least-squares regression, which can be adopted in the dynamic programming. This method is the main technique implemented by us for solving the problems described in the following chapters. An overview of this thesis is provided at the end of this chapter.

1.2. MULTI-PERIOD STOCHASTIC OPTIMIZATION

We illustrate the problem in the one-dimensional case. We assume that the financial market is defined on a complete filtered probability space $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{0\le t\le T}, \mathbb{P})$ with finite time horizon $[0, T]$. The state space Ω is the set of all realizations of the financial market within the time horizon $[0, T]$, $\mathscr{F}$ is the sigma algebra of events till time T, i.e. $\mathscr{F} = \mathscr{F}_T$. We assume that the filtration $\{\mathscr{F}_t\}_{0\le t\le T}$ is generated by the price processes of the financial market and augmented with the null sets of $\mathscr{F}$. The probability measure $\mathbb{P}$ is defined on $\mathscr{F}$. We first introduce a single-period stochastic optimization problem, which can be formulated by[1]:

$$c^{\text{opt}} = \arg\max_{c} \mathbb{E}[V_T(X_T(\omega))|X_0, c]$$

and

$$V_0(X_0) = \mathbb{E}[V_T(X_T(\omega))|X_0, c^{\text{opt}}].$$

Here c_{opt} denotes the optimal control for maximizing the expectation of $V_T(X_T(\omega))$ based on the information at time $t_0 = 0$. X_T stands for the *state variable*[2] at time T, c for the

[1] We show the formulation for a maximization problem. The formulation for a minimization problem can be established by changing the maximization operator to a minimization operator.

[2] In the following chapters, we may consider other notations for the state variables.

control variable and ω denotes the *randomness* in the optimization process, which may have an impact on the state variable. In this thesis, we omit the letter ω in case of no confusion. V_0 and V_T are termed the *value functions*, respectively, at time 0 and time T.

For a single-period stochastic optimization problem, the control c usually has an explicit impact on the values of the state variables. An optimal control for this kind of problems can thus be generated by first transforming the stochastic optimization problem into an optimization problem without stochastic factors and then solving the new problem with basic optimization techniques, for example, as discussed in [14]. However, in the multi-period case, the problem becomes much more involved. The challenge comes since the controls at different time steps may have different impact on the state variable X_T and even the control at one time step may be dependent on the controls at other time steps. Therefore, it is usually not trivial to generate a series of optimal controls for the multi-period stochastic optimization directly. A common approach to solve this multi-period optimization problem is to divide this problem into several static problems and to solve them sequentially in time. This approach is termed *dynamic programming*, which will be introduced in the next section.

1.3. DYNAMIC PROGRAMMING

We consider a multi-period stochastic control problem, in which the control variables at multiple time steps have to be optimized. This multi-period optimization problem can be formulated by:

$$V_0(X_0) = \max_{\{c_t\}_{t=0}^{T-\Delta t}} \mathbb{E}[V_T(X_T)|X_0, \{c_t\}_{t=0}^{T-\Delta t}]. \tag{1.1}$$

with a given value function $V_T(X_T)$ at the terminal time step. Here we need to determine the control variables $\{c_t\}_{t=0}^{T-\Delta t}$ at M time steps $t \in [0, \Delta t, 2\Delta t, \ldots, T-\Delta t]$, where $\Delta t = T/M$. In general, optimizing with respect to all controls $\{c_t\}_{t=0}^{T-\Delta t}$ at one run is a difficult task. One possible to way to simplify this problem is to use the Bellman dynamic programming principle [5].

1.3.1. BACKWARD RECURSIVE CALCULATION

The main idea of the Bellman dynamic programming principle is to transform a multi-period optimization problem into several static optimization problems, which can be formulated and solved in a backward recursive fashion. Since the value function $V_T(X_T)$ at time T is known, the optimal control $c_{T-\Delta t}^{\text{opt}}$ at time step $T-\Delta t$ can be generated by:

$$c_{T-\Delta t}^{\text{opt}} = \arg\max_{c_{T-\Delta t}} \mathbb{E}[V_T(X_T)|X_{T-\Delta t}, c_{T-\Delta t}].$$

After the optimal control $c_{T-\Delta t}^{\text{opt}}$, which is dependent on $X_{T-\Delta t}$, is obtained, the value function $V_{T-\Delta t}(X_{T-\Delta t})$ reads:

$$V_{T-\Delta t}(X_{T-\Delta t}) = \mathbb{E}[V_T(X_T)|X_{T-\Delta t}, c_{T-\Delta t}^{\text{opt}}].$$

The multi-period stochastic control problem can then be solved recursively backward in time and the optimal controls and the value functions at sequential time steps read:

$$c_t^{\text{opt}} = \arg\max_{c_t} \mathbb{E}[V_{t+\Delta t}(X_{t+\Delta t})|X_t, c_t], \tag{1.2}$$

$$V_t(X_t) = \mathbb{E}[V_{t+\Delta t}(X_{t+\Delta t})|X_t, c_t^{\text{opt}}], \tag{1.3}$$

for $t = T-\Delta t, T-2\Delta t, \ldots, \Delta t, 0$.

1.3.2. TIME-CONSISTENCY

In some cases, the optimization problem is formulated by:

$$J_0(X_0) = \max_{\{c_t\}_{t=0}^{T-\Delta t}} g(\mathbb{E}[V_T(X_T)|X_0, \{c_t\}_{t=0}^{T-\Delta t}]).$$

where the function $g(\cdot)$ is a nonlinear function, for example a quadratic function. In this case, the backward recursive formulation cannot be established as in Equations (1.2) and (1.3). We use $J_0(\cdot)$ rather than $V_0(\cdot)$ as in Equation (1.1) to denote the value function of this problem. In order to solve this kind of optimization problems, there are in general two approaches. We can try to reformulate the problem to be an equivalent problem which can be solved by the backward recursive technique. Otherwise, we can introduce a time-consistency condition into the optimization process so that optimization targets at intermediate time steps can be established.

A time-consistency condition means that "given the optimal control for a multi-period optimization problem, the truncated optimal control should also constitute the optimal control for the truncated problem". Mathematically, given any $t \in [0, \Delta t, 2\Delta t, \ldots, T-\Delta t]$, a set of time-consistent optimal controls $\{c_s^{\text{tc}}\}_{s=t}^{T-\Delta t}$ is defined by the optimal control for:

$$J_t(X_t) = \max_{\{c_s\}_{s=t}^{T-\Delta t}} g(\mathbb{E}[V_T(X_T)|X_t, \{c_s\}_{s=t}^{T-\Delta t}]),$$

with an additional requirement that the subsets $\{c_s^{\text{tc}}\}_{s=\tau}^{T-\Delta t}$, $\tau = t+\Delta t, t+2\Delta t, \ldots, T-\Delta t$, also form the optimal controls for

$$J_\tau(X_\tau) = \max_{\{c_s\}_{s=\tau}^{T-\Delta t}} g(\mathbb{E}[V_T(X_T)|X_\tau, \{c_s\}_{s=\tau}^{T-\Delta t}]), \quad \tau = t+\Delta t, t+2\Delta t, \ldots, T-\Delta t.$$

Since the time-consistency condition gives us information about how the problem can be established at all intermediate time steps, a multi-period stochastic control problem with time-consistency requirements can be solved by dynamic programming in a backward recursive fashion as well.

1.4. STOCHASTIC GRID BUNDLING METHOD

When the multi-period stochastic optimization problem needs to be solved, a challenge is that conditional expectations with different filtrations need to be calculated. To accurately and efficiently calculate the conditional expectations forms an important topic in the financial engineering field. For example, in order to proceed the recursive programming process, the value functions, as in Equation (1.3), at several intermediate time steps have to be computed.

In this thesis, we will calculate the conditional expectations based on Monte Carlo simulation and cross-path least-squares regression. We adopt the idea of a recently developed method, the Stochastic Grid Bundling method (SGBM) [57]. Compared to other commonly used regression-based approaches like [16, 65], our method exhibits a different way to setup the regression step. First, when we perform the least-squares regression, we choose the regressands and the regressors collected from the same time step. This approach is termed the "Regress-Later" technique. Compared to its counterpart the "Regress-Now" technique, which relies on choosing the regressands and the regressors collected from different time steps, Regress-Later is more involved and requires more information of the stochastic processes. However, Regress-Later is proved to be more stable than Regress-Now in [48]. Besides, by using the "bundling" technique, we decompose a global regression problem, as considered in [16, 65], into several local regression problems. In general, with the same basis functions for regression, more satisfactory polynomial fitting can be achieved in the local domain than in the global domain.

In the following two subsections, we will explain the "Regress-Later" and "bundling" techniques in detail.

1.4.1. "REGRESS-LATER" TECHNIQUE

Assume that we need to compute $V_t(X_t)$ as in Equation (1.3). When using the Regress-Now technique, we directly approximate the function $V_{t+\Delta t}(X_{t+\Delta t})$ with basis functions formed by X_t and c_t^{opt}:

$$V_{t+\Delta t}(X_{t+\Delta t}) \approx \sum_{k=0}^{K} \alpha_k \cdot \psi_k(X_t, c_t^{\text{opt}}), \tag{1.4}$$

where $\{\psi_k(X_t, c_t^{\text{opt}})\}_{k=0}^{K}$ denote the basis functions and $\{\alpha_k\}_{k=0}^{K}$ are the coefficients computed via least-squares regression. By replacing $V_{t+\Delta t}(X_{t+\Delta t})$ in Equation (1.3) with its approximation as shown in Equation (1.4) and using the basic property of conditional expectations, we have:

$$V_t(X_t) = \mathbb{E}[V_{t+\Delta t}(X_{t+\Delta t})|X_t, c_t^{\text{opt}}] = \sum_{k=0}^{K} \alpha_k \cdot \psi_k(X_t, c_t^{\text{opt}}). \tag{1.5}$$

Remark 1.4.1. *We can thus also interpret the right-hand-side of Equation (1.4) as an approximation of $V_t(X_t)$, which is defined by the conditional expectation given X_t and thus forms a function of X_t.*

When we implement the Regress-Later technique, $V_{t+\Delta t}(X_{t+\Delta t})$ is first approximated with basis functions formed by $X_{t+\Delta t}$:

$$V_{t+\Delta t}(X_{t+\Delta t}) \approx \sum_{k=0}^{K} \beta_k \cdot \phi_k(X_{t+\Delta t}). \tag{1.6}$$

Then by inserting this approximation into Equation (1.3), we obtain:

$$\begin{aligned} V_t(X_t) &= \mathbb{E}[\sum_{k=0}^{K} \beta_k \cdot \phi_k(X_{t+\Delta t}) | X_t, c_t^{\text{opt}}] \\ &= \sum_{k=0}^{K} \beta_k \cdot \mathbb{E}[\phi_k(X_{t+\Delta t}) | X_t, c_t^{\text{opt}}]. \end{aligned} \tag{1.7}$$

According to Equation (1.7), in order to implement the Regress-Later technique, accurate knowledge of the conditional expectations of the basis functions is essential. This is a limitation for implementing the Regress-Later approach but at the same time this is an interesting research challenge.

1.4.2. "BUNDLING" APPROACH

The other feature of our method is the "bundling" approach, which provides additional information in the regression step. We again consider the problem of computing the conditional expectations shown in Equation (1.3). Assume that $X_{t+\Delta t} \in \Phi_{t+\Delta t}$, where $\Phi_{t+\Delta t}$ denotes a sub-domain of $\mathbb{R}$ (note that we are considering the one-dimensional case in this chapter). Based on the information about the sub-domains, using the Regress-Later technique, we can approximate $V_{t+\Delta t}(X_{t+\Delta t})$ by:

$$V_{t+\Delta t}(X_{t+\Delta t}) \approx \sum_{k=0}^{K} \hat{\beta}_k \cdot \phi_k(X_{t+\Delta t}). \tag{1.8}$$

Here we still consider the basis functions $\{\phi_k(X_{t+\Delta t})\}_{k=0}^{K}$ as used in Equation (1.6). Since we have the information that $X_{t+\Delta t}$ belongs to a specified sub-domain, the coefficients $\{\hat{\beta}_k\}_{k=0}^{K}$ may be different from $\{\beta_k\}_{k=0}^{K}$. Approximating value functions in a sub-domain usually requires fewer basis functions for achieving satisfactory accuracy.

1.5. OUTLINE OF THIS THESIS

In this thesis, we develop, analyze and implement simulation-based numerical algorithms to solve various kinds of multi-period stochastic control problems.

In Chapter 2, we consider a Bermudan option pricing problem. At a given time step, the prices of risky assets, whose dynamics are random, are the state variables and the control variables can be chosen to be either 0 or 1, which respectively stand for exercising an option or not. For the Bermudan option pricing problem, since the dynamics of the state variables are not affected by the choice of the control variables, this problem is not involved. It constitutes the point of departure for us to investigate the multi-period stochastic control problems. In this chapter, a general way to configure our algorithm is discussed.

In Chapter 3, we solve a utility-based multi-period portfolio optimization problem. Similar to the option pricing problem, the state variables in the utility-based portfolio problem are not influenced by the control variables. However, this problem is more involved since the control variables may take any real number as their values. To successfully solve this problem requires an efficient and robust numerical algorithm. The

advantage of using our numerical algorithm over other simulation-based approaches will be shown in that chapter.

A more involved portfolio management problem will be discussed in Chapters 4 and 5. In Chapter 4, a multi-period mean-variance optimization problem, which cannot be solved directly via dynamic programming, is first transformed to a multi-period target-based quadratic optimization problem, which can be solved in a backward recursive fashion. Since the control variables for this problem have a significant impact on the state variables, the state variables cannot be simulated without taking control variables into account. We introduce a forward-backward numerical algorithm for solving this problem. Using this algorithm, we first simulate paths with sub-optimal controls in a forward process and then update these controls in a backward process. Highly satisfactory results are obtained. In Chapter 5, the multi-period mean-variance problem with time-consistency requirements is discussed. By carefully choosing value functions, we utilize a modified forward-backward numerical algorithm to solve this problem. A link between the problems respectively discussed in Chapters 4 and 5 is established. The mean-variance frontiers obtained by performing the optimal controls for these two problems are compared.

In Chapter 6, we discuss the robust counterparts of the problems solved in Chapters 4 and 5. Introducing the robustness requirement changes the original mean-variance problem, which is either a minimization problem or a maximization problem, to be a minimax optimization problem. Analysis on the robust optimization problems is given and the robustness of the mean-variance strategies discussed in Chapters 4 and 5 is checked numerically.

CHAPTER 2

Bermudan Options under Merton Jump-Diffusion Dynamics

In this chapter, we utilize the Stochastic Grid Bundling Method (SGBM) for pricing multi-dimensional Bermudan options. We compare SGBM with a traditional regression-based pricing approach and present detailed insight in the application of SGBM, including how to configure it and how to reduce the uncertainty of its estimates by control variates. We consider the Merton jump-diffusion model, which performs better than the geometric Brownian motion in modeling the heavy-tailed features of asset price distributions. Our numerical tests show that SGBM with appropriate setup works highly satisfactorily for pricing multi-dimensional options under jump-diffusion asset dynamics.

Keywords: Monte Carlo simulation · Least-squares regression · Jump-diffusion process · Bermudan option · High-dimensional problem

2.1. INTRODUCTION

Pricing high-dimensional Bermudan options is a challenging topic. For this type of problem, the traditional methods based on solving partial differential equations or on Fourier transformation may fail, because the complexity of these techniques grows exponentially as the dimensionality of the problem increases. Pricing methods based on simulation generally do not suffer from the curse of dimensionality and, therefore, have become increasingly attractive for high-dimensional pricing problems.

Simulation-based pricing for Bermudan options took off in 1993 when Tilley [81] introduced a bundling algorithm to estimate the continuation values of the option at intermediate time steps. In 1996, an option pricing method based on regression was introduced by Carriere in [20]. The basic idea was to estimate the option's continuation values at all time points by projections of the future option values on finite-dimensional subspaces spanned by pre-selected basis functions. Depending on the procedure of generating basis functions, regression methods can be categorized into two types: Regress-Now and Regress-Later approaches, as in [48]. More details of these two methods will be discussed in Section 2.3. Following Carriere's work [20] many papers discussing regression methods based on the Regress-Now feature appeared, for example [65] and [82]. However, the investigation on Regress-Later methods is not abundant.

This chapter is based on the article 'Pricing Bermudan options under Merton jump-diffusion asset dynamics', published in *International Journal of Computer Mathematics*, 92(12):2406–2432, 2015 [24].

The Stochastic Grid Bundling Method (SGBM), introduced in [57], belongs to the type of Regress-Later approaches. In SGBM, both "bundling" and "regression" are utilized to estimate the continuation values. Similar to [18], SGBM produces two estimators: one biased high and the other biased low, which respectively correspond to the "value function approximation" and the "stopping time approximation" discussed in [79]. Compared to the well-known least-squares method (LSM), introduced in [65], for pricing Bermudan options, SGBM typically yields estimates with significantly lower variances, according to [57, 59]. In our numerical tests, we obtain similar results: for achieving comparable accuracy, many more paths and higher computational times are required in LSM compared to SGBM. Moreover, according to [57], SGBM generates upper and lower bounds for the option price and also accurate sensitivities or Greeks of the option price, while the original LSM is only applicable for calculating the lower bound of the option price.

In this chapter, we extend the discussion of SGBM in four directions. First, we gain insight into the essential components of SGBM. According to our analysis, it is sufficient to choose the basis functions for regression of polynomial type, which ensures that conditional expectations of the basis functions can be calculated exactly. Secondly, in the error analysis we explicate that the number of bundles used is a "trade-off" factor of two types of biases in SGBM. Thirdly, we combine SGBM with control variates to reduce the variance of the biased low estimator. We implement the traditional control variates and an improved approach proposed in [73]. According to the tests, the improved control variates work uniformly better in the one-dimensional case, but for higher-dimensional problems the cost of calculating the improved control variates is significant and the traditional control variates appear favorable. Instead of considering plain geometric Brownian motion we focus our discussion on assets with their dynamics following the Merton jump-diffusion process for high-dimensional Bermudan option pricing.

This chapter is organized as follows. Section 2.2 gives the formulation of the problem. In Section 2.3 we compare SGBM with the standard regression method. In Section 2.4 we focus on the features of SGBM and explain how we can configure SGBM. In Section 2.5 the sources of errors in SGBM are compared to those in the standard regression method. Section 2.6 discusses traditional control variates and the improved versions. In Section 2.7 the Merton jump-diffusion model is introduced and in Section 2.8 the corresponding numerical results are presented.

2.2. PROBLEM FORMULATION: BERMUDAN OPTION PRICING

This section describes the Bermudan option pricing problem mathematically and sets up the notations used in this chapter. We assume that a risk-neutral measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ exists under which the asset prices are martingales with appropriate numeraire. The Bermudan option considered can be exercised within a set of prescribed time points $\mathbb{T} = [t_0 = 0, \ldots, t_m, \ldots, t_M = T]$. The d-dimensional state variable is represented by an $\mathscr{F}_t$-adapted Markovian process $\mathbf{S}_t = (S_t^1, \ldots, S_t^d) \in \mathbb{R}^d$, where $t \in \mathbb{T}$. Let $h(\mathbf{S}_t)$ be the intrinsic value of the option, i.e. the holder of the option receives payoff $g(\mathbf{S}_t) = \max(h(\mathbf{S}_t), 0)$ if the option is exercised at time t. With the money savings account process $\mathscr{D}_t = \exp(\int_0^t r_s \mathrm{d}s)$, where r_s denotes the instantaneous risk-free rate of return, we define the discounting

process as

$$D_{t_m} = \frac{\mathscr{D}_{t_m}}{\mathscr{D}_{t_{m+1}}}.$$

For simplicity, we consider the special case where r_s is equal to a constant r. The problem of valuing a Bermudan option is to find the optimal exercise strategy (or equivalently the optimal stopping time, $\tau \in \mathbb{T}$) and calculating the expected discounted payoff following this strategy, that is:

$$V_0(\mathbf{S}_0) = \sup_{\tau \in \mathbb{T}} \mathbb{E}\left[\frac{h(\mathbf{S}_\tau)}{\mathscr{D}_\tau} \middle| \mathscr{F}_0 \right]. \tag{2.1}$$

The expectation $\mathbb{E}[\cdot]$ is computed under the risk-neutral measure $\mathbb{Q}$. Here we write the option value in the form $V_0(\mathbf{S}_0)$ to emphasize that when the asset dynamics are fixed the option value is uniquely determined by the initial asset value.

If we consider a dynamic programming approach, the optimal exercise strategy can be determined via a recursive process, by which the option values, $V_{t_m}(\mathbf{S}_{t_m})$, at intermediate time steps can be computed correspondingly. The value of the Bermudan option at maturity state[1] $(T, \mathbf{S}_T)$ is equal to its payoff, i.e.

$$V_T(\mathbf{S}_T) = g(\mathbf{S}_T) = \max(h(\mathbf{S}_T), 0). \tag{2.2}$$

In the recursive process, the conditional continuation value $Q_{t_m}(\mathbf{S}_{t_m})$ associated to state $(t_m, \mathbf{S}_{t_m})$, i.e. the discounted expected option value at time t_{m+1} conditioned on filtration $\mathscr{F}_{t_m}$, is given by:

$$Q_{t_m}(\mathbf{S}_{t_m}) = D_{t_m} \mathbb{E}[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) | \mathbf{S}_{t_m}]. \tag{2.3}$$

We write $\mathbf{S}_{t_m}$, which means that the stock price at time t_m is equal to $\mathbf{S}_{t_m}$. In the following discussions, the condition of the expectation may also be formulated as $\mathbf{S}_{t_m} = \hat{\mathbf{S}}$ to emphasize that the stock price at time t_m is known as a realization $\hat{\mathbf{S}}$.

The option value at state $(t_m, \mathbf{S}_{t_m})$ is then given by taking the maximum of its continuation value and the direct exercise value,

$$V_{t_m}(\mathbf{S}_{t_m}) = \max(Q_{t_m}(\mathbf{S}_{t_m}), g(\mathbf{S}_{t_m})). \tag{2.4}$$

We are interested in finding the option value at initial state $(t_0, \mathbf{S}_{t_0})$, using either Equation (2.1) or the recursive process mentioned above.

2.3. REGRESSION METHODS FOR BERMUDAN OPTION PRICING

We consider the Bermudan option with M equally distributed exercise opportunities before maturity T, i.e. the option can be exercised at time $t_m = m\Delta t$, where $m = 1, \ldots, M$ and $\Delta t = T/M$. When the Monte Carlo generation for the sample of N paths is done and the function values $h(\cdot)$ are determined, we find the option value associated to each path at maturity directly via Equation (2.2). Similarly for the ith path we obtain the direct exercise value $g(\mathbf{S}_{t_m}(i))$ at each exercise time t_m. The remaining problem is to calculate

[1] It denotes the realization at time T with the values of the option's underlying assets equal to $\mathbf{S}_T$.

the conditional continuation value $Q_{t_m}(\mathbf{S}_{t_m})$ as in Equation (2.3). To settle this problem, regression methods are employed.

As mentioned, regression methods can be classified into two categories: Regress-Now and Regress-Later approaches. In the remaining part of this section, we consider the standard regression method, which resembles the method introduced in [20, 46], as a typical case of Regress-Now methods and the Stochastic Grid Bundling Method as a representative of Regress-Later methods.

2.3.1. STANDARD REGRESSION METHOD

The classical standard regression method (SRM), described in [20, 46], has been widely discussed for pricing Bermudan options. The pricing procedure can be described as:

Step I: Get the option value at maturity time for each path:

$$V_{t_M}(\mathbf{S}_{t_M}(i)) = \max(h(\mathbf{S}_{t_M}(i)), 0), i = 1, \ldots, N.$$

Recursively moving backward in time from maturity time t_M, the following steps are performed at time t_m, $m < M$.

Step II: Regression step.
For all paths, we get the regression parameters $\{\alpha_k\}_{k=1}^{K}$ by regressing the option values $\{V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}(i))\}_{i=1}^{N}$ on basis functions $[\phi_1(\mathbf{S}_{t_m}(i)), \ldots, \phi_K(\mathbf{S}_{t_m}(i))]_{i=1}^{N}$, that are constructed using the asset values at time t_m. Linear regression offers us an approximation of the option value for any specified S_{t_m} , i.e.

$$V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) \approx \sum_{k=1}^{K} \alpha_k \phi_k(\mathbf{S}_{t_m}). \tag{2.5}$$

Step III: Calculate the continuation value and the option value at time t_m for the ith path:

$$\begin{aligned} Q_{t_m}(\mathbf{S}_{t_m}(i)) &= D_{t_m}\mathbb{E}\left[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m} = \mathbf{S}_{t_m}(i)\right] \\ &\approx D_{t_m}\mathbb{E}\left[\sum_{k=1}^{K} \alpha_k \phi_k(\mathbf{S}_{t_m})\middle|\mathbf{S}_{t_m} = \mathbf{S}_{t_m}(i)\right] \\ &= D_{t_m}\sum_{k=1}^{K} \alpha_k \phi_k(\mathbf{S}_{t_m}(i)). \end{aligned} \tag{2.6}$$

The first equality is immediate from the definition of the continuation value in Equation (2.3). The approximation is supported by Equation (2.5). The second equality is valid based on a property of conditional expectations. The option value $V_{t_m}(\mathbf{S}_{t_m}(i))$ can be computed as:

$$V_{t_m}(\mathbf{S}_{t_m}(i)) = \max(Q_{t_m}(\mathbf{S}_{t_m}(i)), g(\mathbf{S}_{t_m}(i))). \tag{2.7}$$

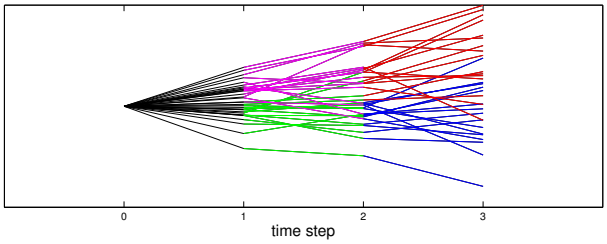


Figure 2.1: Paths from initial state to terminal time step 3. At the first backward recursion step, paths are bundled according to their state at time step 2, giving the "red" and "blue" paths in two bundles. At the next recursion step, the paths are bundled according to their state at time step 1, giving the "magenta" and "green" paths.

2.3.2. STOCHASTIC GRID BUNDLING METHOD

The Stochastic Grid Bundling Method (SGBM) introduced in [57] belongs to the category of Regress-Later approaches. After generating all paths by Monte Carlo simulation, the algorithm of SGBM can be described as:

Step I: Get the option value at maturity for each path:

$$V_{t_M}(\mathbf{S}_{t_M}(i)) = \max(h(\mathbf{S}_{t_M}(i)), 0), i = 1, \ldots, N.$$

The following steps are subsequently performed at time t_m, $m < M$.

Step II: Bundle paths at time t_m.
With a specified bundling criterion, we bundle all paths at time t_m into $\mathscr{B}_{t_m}(1), \ldots, \mathscr{B}_{t_m}(b), \ldots, \mathscr{B}_{t_m}(B)$ non-overlapping partitions. Figure 2.1 illustrates how bundling is performed in the one-dimensional case. The details of the bundling technique are discussed in the section to follow.

Step III: Regression step.
Assume that there are $N_B(b)$ paths in bundle $\mathscr{B}_{t_m}(b)$ and denote their asset values at time t_{m+1} as $\{\mathbf{S}^{(b)}_{t_{m+1}}(i)\}_{i=1}^{N_B(b)}$ and the option values as $\{V^{(b)}_{t_{m+1}}(i)\}_{i=1}^{N_B(b)}$. For these paths, we get the bundle regression parameters $\{\alpha_k(b)\}_{k=1}^{K}$ by regressing the option values $\{V^{(b)}_{t_{m+1}}(\mathbf{S}_{t_{m+1}}(i))\}_{i=1}^{N_B(b)}$ on the basis functions $[\phi_1(\mathbf{S}^{(b)}_{t_{m+1}}(i)), \ldots, \phi_K(\mathbf{S}^{(b)}_{t_{m+1}}(i))]_{i=1}^{N_B(b)}$, which are constructed using the asset values at time t_{m+1}. For assets whose values $\mathbf{S}_{t_m} = [S^1_{t_m}, \ldots, S^d_{t_m}]$ are covered by bundle $\mathscr{B}_{t_m}(b)$, the corresponding option value at time t_{m+1} can be approximated by[2]:

$$V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) \approx \sum_{k=1}^{K} \alpha_k(b)\phi_k(\mathbf{S}_{t_{m+1}}). \tag{2.8}$$

[2] The authors of [9] show that Regress-Later is fundamentally different from Regress-Now, noticing that the former does not introduce a projection error between two time steps in the regression stage. As a result, Regress-Later achieves a faster convergence rate than Regress-Now in terms of the sample size.

At each time step, the regression is repeated for all bundles. In each bundle, the same basis functions $[\phi_1(\cdot),\ldots,\phi_K(\cdot)]$ are utilized.

Step IV: Calculate the continuation value and the option value at time t_m for the ith path.
Assume that the ith path at time t_m belongs to bundle $\mathscr{B}_{t_m}(b)$. The continuation value at time t_m associated to this path is given by:

$$\begin{aligned} Q_{t_m}(\mathbf{S}_{t_m}(i)) &= D_{t_m}\mathbb{E}\left[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m}=\mathbf{S}_{t_m}(i)\right] \\ &\approx D_{t_m}\mathbb{E}\left[\sum_{k=1}^{K}\alpha_k(b)\phi_k(\mathbf{S}_{t_{m+1}})\Big|\mathbf{S}_{t_m}=\mathbf{S}_{t_m}(i)\right] \\ &= D_{t_m}\sum_{k=1}^{K}\alpha_k(b)\mathbb{E}\left[\phi_k(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m}=\mathbf{S}_{t_m}(i)\right]. \end{aligned}$$

Note that, compared to Equation (2.6), the last equation contains conditional expectations of the basis functions, which is typical for Regress-Later approaches.

The motivation for the equality and approximation signs above is the same as for Step III of SRM. To obtain a closed-form expression for $Q_{t_m}(\mathbf{S}_{t_m}(i))$, we need analytic conditional expectations of the basis functions, $\mathbb{E}\left[\phi_k(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m}=\mathbf{S}_{t_m}(i)\right]$, $k=1,\ldots,K$, which are achievable when the basis functions $\{\phi_k(\mathbf{S}_{t_{m+1}})\}_{k=1}^{K}$ are chosen appropriately. The option value can be computed via Equation (2.7).

2.4. CONFIGURATION OF SGBM

There are basically two distinct features between the algorithms of SGBM and SRM:

- The basis functions in SGBM are required to have explicit analytic moments so that there is no error introduced in the last step of the algorithm. For SRM, the basis functions can be chosen freely.
- At each time step, the regression in SRM is done for all paths, while the regression in SGBM is done separately within each bundle. By the bundling technique in SGBM, the global fitting problem reduces to a local fitting problem.

Based on these two points, we will explain how to configure SGBM to make it feasible and robust for different scenarios.

2.4.1. CHOICE OF BASIS FUNCTIONS

The special requirement for the basis functions in SGBM may complicate the application of this pricing algorithm for some involved options. For example, in [57], the powers of the maximum of asset values are chosen as the basis functions for pricing max-on-call options. Since the moments of these basis functions are not analytically available, they need to be approximated by Clark's algorithm [22]. Because of the inaccuracy of this numerical approximation, the duality method is required. This procedure makes the pricing algorithm less tractable.

We find that if the following conditions are satisfied, it is not necessary to choose "max" or "min" functions as the basis functions.

Condition 2.4.1. *The transition probability density function $f(s,\mathbf{S}_s;t,\mathbf{S}_t)$, which denotes the probability density function from state $(s,\mathbf{S}_s)$ to state $(t,\mathbf{S}_t)$, is continuous with respect to $\mathbf{S}_s$.*

Condition 2.4.2. *The option's direct exercise value $g(\mathbf{S}_t)$ is continuous with respect to $\mathbf{S}_t$.*

With these conditions, we can prove the following theorem:

Theorem 2.4.3. *At each exercise time, the option value $V_{t_m}(\mathbf{S}_{t_m})$ can be uniformly approximated by polynomials formed by $\mathbf{S}_{t_m}$.*

Proof. Consider the backward pricing process of Bermudan options. At maturity time we have: $V_T(\mathbf{S}_T) = g(\mathbf{S}_T)$, which is continuous with respect to $\mathbf{S}_T$. This follows directly from Condition 2.4.2. We then use backward induction. Assuming that $V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})$ is continuous with respect to $\mathbf{S}_{t_{m+1}}$, we have:

$$\begin{aligned} Q_{t_m}(\mathbf{S}_{t_m}) &= D_{t_m}\mathbb{E}\left[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m}\right] \\ &= D_{t_m}\int_{\mathbb{R}^d} V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})f(t_m,\mathbf{S}_{t_m};t_{m+1},\mathbf{S}_{t_{m+1}})\mathrm{d}\mathbf{S}_{t_{m+1}} \\ &\approx D_{t_m}\int_{H} V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})f(t_m,\mathbf{S}_{t_m};t_{m+1},\mathbf{S}_{t_{m+1}})\mathrm{d}\mathbf{S}_{t_{m+1}}. \end{aligned}$$

The second equality is from the definition of conditional expectation and assuming that the dimension of $\mathbf{S}_{t_{m+1}}$ is d. The approximation sign is because of truncation of the integral from $\mathbb{R}^d$ to H. Without loss of generality, we assume that H is a compact subspace of $\mathbb{R}^d$.

Since $V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})$ is continuous with respect to $\mathbf{S}_{t_{m+1}}$ on the compact domain H, it is bounded. With Condition 2.4.1, we can prove that $Q_{t_m}(\mathbf{S}_{t_m})$ is continuous with respect to $\mathbf{S}_{t_m}$.

The option price $V_{t_m}(\mathbf{S}_{t_m})$ is constructed by taking the maximum of the continuation value and the direct exercise value:

$$V_{t_m}(\mathbf{S}_{t_m}) = \max(Q_{t_m}(\mathbf{S}_{t_m}), g(\mathbf{S}_{t_m})) \tag{2.9}$$

and both $Q_{t_m}(\mathbf{S}_{t_m})$ and $g(\mathbf{S}_{t_m})$ are continuous with respect to $\mathbf{S}_{t_m}$. So, the option price $V_{t_m}(\mathbf{S}_{t_m})$ is also continuous with respect to $\mathbf{S}_{t_m}$.

We conclude the proof by using the generalized Stone-Weierstrass theorem on the space H. □

Conditions 2.4.1 and 2.4.2 generally hold in option pricing. The continuous transition density functions associated to the commonly implemented models, such as the geometric Brownian motion and the jump-diffusion model, satisfy Condition 2.4.1 directly. The direct exercise value of a call or a put option is continuous with respect to the values of underlying assets. However, since a digital option does not have a continuous payoff function, using the Regress-Later approach to price a Bermudan digital option may lead to a large approximation error.

Theorem 2.4.3 tells us that it is not necessary to include the "max" or "min" of underlying assets as a basis function. We choose here to only use polynomials as the basis functions in SGBM for multi-dimensional problems.

2.4.2. BUNDLING

A good "bundling" technique should make the regression within the bundle easier, or, more precisely, make the regression less biased even though only a few paths are inside the bundle. This gives us a hint for bundling: if we bundle the paths such that paths in one bundle have similar option values, we expect that regression in this bundle would be easier.

The instruction that paths inside one bundle should have similar option values is not directly under our control, since bundling is done at time t_m but the option values considered in regression are from time t_{m+1}. However, the option value at time t_{m+1} should be to some degree related to its intrinsic value at time t_m. For example, considering the max-on-call option, if one path has a large intrinsic value at time t_m, which means that one asset associated to this path has a large value, we expect that the option value at this path at time t_{m+1} would still be large. In other words, since asset values are usually continuous in probability, paths, whose intrinsic values at time t_m are almost identical, are supposed to have similar option values at time t_{m+1}.

"Bundling" is not new in the field of Bermudan option pricing. In [81] Tilley initiated the technique for pricing Bermudan options by Monte Carlo simulation using a simple bundling algorithm, which is however only applicable for a one-dimensional problem. Tilley's bundling algorithm can be described as a two-step method: "reordering" and "partitioning". In the "reordering" step, all paths are sorted according to their asset values. Then in the "partitioning" step, the reordered paths are partitioned into distinct bundles of N_b paths each. The first N_b paths are assigned to the first bundle, the second N_b paths to the second bundle and so on.

In [41] and [3], Tilley's bundling is extended to high-dimensional scenarios. The technique in [41], where multi-dimensional max options are dealt with using bundling, is to first reduce the multi-dimensional bundling problem to one dimension by choosing one single asset as representative for the multi-dimensional function. All paths are then bundled by applying Tilley's algorithm on the one-dimensional data. Within each bundle, a next bundling step is done by choosing another single asset as the new representative and again applying Tilley's algorithm. These newly generated bundles are called the "sub-bundles". The bundling can be done recursively within each sub-bundle until a prescribed number of bundles is reached.

Inspired by bundling in [81], [3] and [41], we also define our bundling algorithm as a two-step method. For reordering the paths in the multi-dimensional case we first transform the multi-dimensional problem to a single-dimensional problem. Mathematically, it is equivalent to mapping the vector $\mathbf{S}_t = (S_t^1, \ldots, S_t^d)$ to a number by specifying a function $R(\cdot)$, such that $R : \mathbb{R}^d \to \mathbb{R}$. In this chapter, we call the variable $R(\mathbf{S}_t)$ the "bundling reference". Sometimes, we need more than one bundling reference as shown in [41]. In that case, we denote the bundling references subsequently as $R_1(\mathbf{S}_t)$, $R_2(\mathbf{S}_t)$ and so on.

In [81] and [41], the bundling is done to make each bundle cover the same number of paths so that we call it "equal-size bundling". This is different from the bundling in the original SGBM in [57], which we call "equal-range bundling"[3]. In this chapter we will perform "equal-size bundling". According to our tests, there is no clear advantage on ac-

[3] If we want to bundle the paths into two parts, the partition point for "equal-size bundling" is the median of the asset prices while that for "equal-range bundling" is the mean of the asset prices.

curacy of either bundling scheme over the other. However, "equal-size bundling" is more robust than "equal-range bundling", because we always keep enough paths within each bundle to support the regression. If we choose the latter, the number of paths within some bundles may be so small that the estimation in those bundles is highly biased. The necessity of having enough paths inside one bundle will be further discussed in the next section.

Our bundling algorithm for the paths with asset values $\{\mathbf{S}_t(i)\}_{i=1}^N$, where $\mathbf{S}_t(i) = (S_t^1(i), \ldots, S_t^d(i))$, can be described as:

Step I: Reordering

(1) Based on the type of option, choose mapping functions $R_1(\cdot), \ldots, R_P(\cdot)$, by which the bundling references can be generated.

(2) Start with bundling reference $R_1(\mathbf{S}_t)$, bundle all paths equally into n_1 partitions following Tilley's bundling. Record the index of the bundle $b_1(i)(b_1(i) \in \{1,2,\ldots,n_1\})$, where the ith ($i \in \{1,2,\ldots,N\}$) path is located in.

(3) With reference $R_2(\mathbf{S}_t)$, divide the paths in a sub-bundle generated in the previous step into n_2 partitions. Again record the index of the bundle $b_2(i)(b_2(i) \in \{1,2,\ldots,n_2\})$, where the ith ($i \in \{1,2,\ldots,N\}$) path is located in.

(3) Repeat the process above with each bundling reference inside a sub-bundle. For the ith path, we get the vector recording its location $(b_1(i), \ldots, b_P(i))$, see Figure 2.2 for an example of recording the location of a single path.

(4) Construct the global bundling reference for the ith path as:

$$R(\mathbf{S}_t(i)) = b_1(i) \cdot N^{P-1} + b_2(i) \cdot N^{P-2} + \cdots + b_{P-1}(i) \cdot N + b_P(i), i = 1, \ldots, N$$

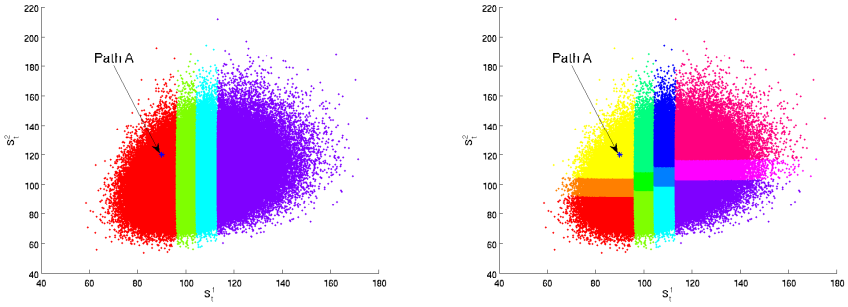
(5) Reorder the paths according to the global bundling reference $R(\mathbf{S}_t)$.

Step II: Partitioning
Partition the sorted paths into $\prod_{p=1}^P n_p$ bundles each of $\frac{N}{\prod_{p=1}^P n_p}$ paths, where $\prod_{p=1}^P n_p$ is an integer factor of N.

The following examples demonstrate that some common bundling schemes fit into our generalized bundling technique.

Example 2.4.4. *For a one-dimensional problem, we choose the bundling reference $R(S_t) = S_t$. So the bundling algorithm covers the simplest one-dimensional case.*

For a basket option of assets $\mathbf{S}_t = (S_t^1, \ldots, S_t^d)$, if we choose bundling references respectively equal to the value of each individual asset, we will get "bundling on the original state space", as termed in [57].



(a) Start bundling according to $R_1(\mathbf{S}_t) = S_t^1$ with $n_1 = 4$. Record the location of Path A as $(1, \cdot)$.

(b) Within each bundle perform sub-bundling according to $R_2(\mathbf{S}_t) = S_t^2$ with $n_2 = 3$. Record the location of Path A as $(1, 3)$.

Figure 2.2: Obtaining the location of Path A with two bundling references.

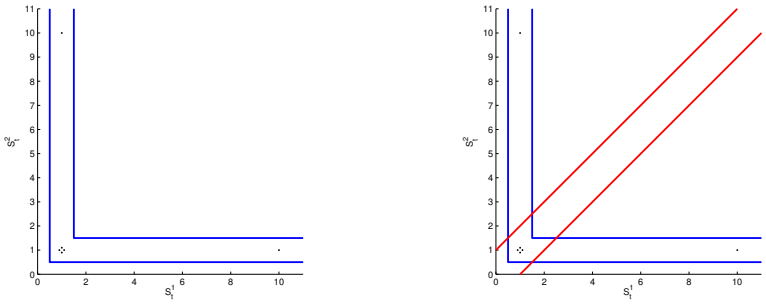
CHOOSING THE BUNDLING REFERENCE

After we have specified the basis functions of polynomial type, the performance of SGBM depends on whether we can choose an accurate bundling reference. For example, when we consider the geometric basket option with underlying assets following multi-dimensional geometric Brownian motion, an accurate bundling reference is the geometric mean of the asset values. This is supported by the fact that the geometric average of (jointly) log-normal random variables is still log-normal. This implies that when dealing with the geometric basket option an optimal bundling reference is the geometric mean of the asset values. Moreover, although there is no representation technique for the arithmetic basket option, our tests suggest that the arithmetic mean of asset values is a preferred bundling reference for arithmetic basket options.

For options whose payoff functions are related to the "max" or "min" of asset values, choosing the intrinsic value alone as the bundling reference is not sufficient, as shown in Example 2.4.5. Inspired by this example, we should separate paths whose option values are related to only one asset, from paths whose option values are affected by each asset. This gives us another bundling reference: the difference between the asset values. In Section 2.8.2, we can see that combining them offers us a much better result than using any of them individually and this combination also outperforms other possible combinations of the bundling references.

Example 2.4.5. *If we consider a two-dimensional put-on-min option with assets* $\mathbf{S}_t = (S_t^1, S_t^2)$ *and strike* $K = 2$*, following the instructions in the previous subsection we choose basis functions as* $[1, S_t^1, S_t^2]$*. Assume that we have six paths respectively with assets* $\mathbf{S}_t(1) = (1, 10)$*,* $\mathbf{S}_t(2) = (10, 1)$*,* $\mathbf{S}_t(3) = (1, 0.9)$*,* $\mathbf{S}_t(4) = (1, 1.1)$*,* $\mathbf{S}_t(5) = (0.9, 1)$ *and* $\mathbf{S}_t(6) = (1.1, 1)$*. Their option values are recorded as* $[1, 1, 1.1, 1, 1.1, 1]$*.*

If we bundle these paths in the same partition based on their intrinsic values (Figure 2.3(a)) , then the approximated option values will be $[1, 1, 1.05, 1.05, 1.05, 1.05]$*. If we introduce one more bundling reference (Figure 2.3(b)) so that the first two points are separated*



(a) One bundling reference (b) Two bundling references

Figure 2.3: Bundling for pricing the two-dimensional put-on-min option.

from the other ones, then the approximated option values for the last four paths will be given by [1.1, 1, 1.1, 1].

2.5. ERROR ANALYSIS: COMPARING SRM AND SGBM

In this section, we will compare the errors of SRM and SGBM when estimating the conditional continuation value $Q_{t_m}(\hat{\mathbf{S}})$, where we denote $\hat{\mathbf{S}}$ as a realization of $\mathbf{S}_{t_m}$. Here we consider the approximation error in one backward pricing step, so the option value $V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})$ at time t_{m+1} is assumed to be known exactly. In the following discussions, we will write $\pi(\mathbf{S}_{t_{m+1}})$ as the density function of $\mathbf{S}_{t_{m+1}}$ conditioned on $\mathbf{S}_{t_m} = \hat{\mathbf{S}}$. With these notations, the analytic continuation value $Q_{t_m}(\hat{\mathbf{S}})$ reads:[4]

$$Q_{t_m}(\hat{\mathbf{S}}) = \mathbb{E}[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})|\mathbf{S}_{t_m} = \hat{\mathbf{S}}] = \mathbb{E}^{\pi}[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})], \tag{2.10}$$

where $\mathbb{E}^{\pi}[\cdot]$ indicates that the expectation is computed with $\pi(\mathbf{S}_{t_{m+1}})$ as the density function of $\mathbf{S}_{t_{m+1}}$.

2.5.1. ERROR IN THE OPTIMAL REGRESSION-BASED APPROACH

Let us start with a trivial problem where we perform sub-simulation to calculate $Q_{t_m}(\hat{\mathbf{S}})$. In the framework of Monte Carlo pricing, we simulate the realizations $\{\hat{\mathbf{S}}_{t_{m+1}}(i)\}_{i=1}^{N}$ with the dynamics associated to the density function $\pi(\mathbf{S}_{t_{m+1}})$[5]. We denote their empirical density function as $\hat{\pi}(\mathbf{S}_{t_{m+1}})$, which can be defined by[6]:

$$\hat{\pi}(\mathbf{S}_{t_{m+1}}) := N^{-1}\sum_{i=1}^{N}\delta(\mathbf{S}_{t_{m+1}}(i) - \mathbf{S}_{t_{m+1}}),$$

[4] For simplicity, we neglect the discounting term D_{t_m}.

[5] Since $\pi(\mathbf{S}_{t_{m+1}})$ is defined by the density function of $\mathbf{S}_{t_{m+1}}$ conditioned on $\mathbf{S}_{t_m} = \hat{\mathbf{S}}$, the simulation of $\mathbf{S}_{t_{m+1}}$ with respect to this density function can be treated as a sub-simulation from the unique state $\mathbf{S}_{t_m} = \hat{\mathbf{S}}$.

[6] In the following sections, other empirical density functions can be defined in a similar fashion.

where $\delta(\cdot)$ indicates a kernel density function. With a suitable simulation technique, we assume that the empirical density $\hat{\pi}(\mathbf{S}_{t_{m+1}})$ resembles its theoretical counterpart $\pi(\mathbf{S}_{t_{m+1}})$ when the number of simulation trajectories goes to infinity.

Since the function $V_{t_{m+1}}(\mathbf{S}_{t_{m+1}})$ is assumed to be known, we find the realizations $\{V_{t_{m+1}}(\hat{\mathbf{S}}_{t_{m+1}}(i))\}_{i=1}^{N}$ of the exact option value. If we estimate $Q_{t_m}(\hat{\mathbf{S}})$ by regression instead of simply by taking the average of $\{V_{t_{m+1}}(\hat{\mathbf{S}}_{t_{m+1}}(i))\}_{i=1}^{N}$, we can regress $\{V_{t_{m+1}}(\hat{\mathbf{S}}_{t_{m+1}}(i))\}_{i=1}^{N}$ on $\{\phi_1(\hat{\mathbf{S}}_{t_{m+1}}(i)) \ , \ \dots \ , \ \phi_K(\hat{\mathbf{S}}_{t_{m+1}}(i))\}_{i=1}^{N}$ and obtain the regression parameters $\{\hat{\alpha}_k\}_{k=1}^{K}$, which minimize the sum of the squared errors of the samples:

$$\sum_{i=1}^{N}\left(V_{t_{m+1}}(\hat{\mathbf{S}}_{t_{m+1}}(i)) - \sum_{k=1}^{K}\hat{\alpha}_k\phi_k(\hat{\mathbf{S}}_{t_{m+1}}(i))\right)^2.$$

Since $\{\hat{\mathbf{S}}_{t_{m+1}}(i)\}_{i=1}^{N}$ follows the empirical distribution of density $\hat{\pi}(\mathbf{S}_{t_{m+1}})$, the regression parameter $\{\hat{\alpha}_k\}_{k=1}^{K}$ also minimizes the mean square error:

$$\mathbb{E}^{\hat{\pi}}\left[\left(V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) - \sum_{k=1}^{K}\hat{\alpha}_k\phi_k(\mathbf{S}_{t_{m+1}})\right)^2\right].$$

If we denote the regression error $\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})$ as

$$\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}}) = V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) - \sum_{k=1}^{K}\hat{\alpha}_k\phi_k(\mathbf{S}_{t_{m+1}}),$$

the least-squares linear regression guarantees:

$$\mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})] = 0. \tag{2.11}$$

The approximated continuation value $\hat{Q}_{t_m}(\hat{\mathbf{S}})$ can be computed as:

$$\begin{aligned}\hat{Q}_{t_m}(\hat{\mathbf{S}}) &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{K}\hat{\alpha}_k\phi_k(\mathbf{S}_{t_{m+1}})\right]\\ &= \mathbb{E}^{\pi}[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) + \epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})]\\ &= Q_{t_m}(\hat{\mathbf{S}}) + \left(\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})]\right).\end{aligned}$$

The first equality is directly from the regression-based approximation scheme. The second equality is valid because we rewrite the approximated option value as the true option value plus regression error, and the last step is supported by Equation (2.11). Since we can simulate a large number of realizations of $\mathbf{S}_{t_{m+1}}$, the empirical distribution function $\hat{\pi}(\mathbf{S}_{t_{m+1}})$ resembles $\pi(\mathbf{S}_{t_{m+1}})$. Moreover, using the Cauchy-Schwarz inequality we have:

$$\begin{aligned}&\left|\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})]\right|\\ =\ &\left|\int_H \epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})\pi(\mathbf{S}_{t_{m+1}})\mathrm{d}\mathbf{S}_{t_{m+1}} - \int_H \epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})\hat{\pi}(\mathbf{S}_{t_{m+1}})\mathrm{d}\mathbf{S}_{t_{m+1}}\right|\\ \leq\ &\left(\int_H \left(\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})\right)^2\mathrm{d}\mathbf{S}_{t_{m+1}}\right)^{\frac{1}{2}}\cdot\left(\int_H\left(\pi(\mathbf{S}_{t_{m+1}}) - \hat{\pi}(\mathbf{S}_{t_{m+1}})\right)^2\mathrm{d}\mathbf{S}_{t_{m+1}}\right)^{\frac{1}{2}},\end{aligned}$$

where the integral domain H is defined in Section 2.4.1 as a truncated subspace of $\mathbb{R}^d$ and we assume that the regression error is bounded on this domain. When the sample size is sufficiently large, $\left|\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}}[\epsilon^{\hat{\pi}}(\mathbf{S}_{t_{m+1}})]\right|$ will be close to 0 and therefore $\hat{Q}_{t_m}(\hat{\mathbf{S}})$ will be an accurate approximation of $Q_{t_m}(\hat{\mathbf{S}})$.

However the process mentioned above is not achievable in a real application, since we cannot afford sub-simulation for every state $(t_m, \mathbf{S}_{t_m})$. As feasible alternatives, we have the cross-path regression methods, for example, SRM and SGBM.

2.5.2. ERROR IN SRM

In the regression step of SRM, we regress the option values $\{V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}(i))\}_{i=1}^{N}$ on the basis functions $[\phi_1(\mathbf{S}_{t_m}(i)), \ldots, \phi_K(\mathbf{S}_{t_m}(i))]_{i=1}^{N}$. Since all paths are generated from the same initial state, we denote the theoretical density function of $\mathbf{S}_{t_{m+1}}$ by $\pi_G(\mathbf{S}_{t_{m+1}})$ and the empirical density function, which is represented by the samples $\{\mathbf{S}_{t_{m+1}}(i)\}_{i=1}^{N}$, by $\hat{\pi}_G(\mathbf{S}_{t_{m+1}})$. Since the regression is done with respect to the samples $\{\mathbf{S}_{t_{m+1}}(i)\}_{i=1}^{N}$, we have

$$\mathbb{E}^{\hat{\pi}_G}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})] = 0. \tag{2.12}$$

After we determine the regression parameters $\{\alpha_k^G\}_{k=1}^{K}$, the approximated continuation value $\hat{Q}_{t_m}^G(\hat{\mathbf{S}})$ can be generated as:

$$\begin{aligned}\hat{Q}_{t_m}^G(\hat{\mathbf{S}}) &= \mathbb{E}^{\pi}\left[\sum_{k=1}^{K}\alpha_k^G\phi_k(\mathbf{S}_{t_m})\right] = \mathbb{E}^{\pi}[V_{t_{m+1}}(\mathbf{S}_{t_{m+1}}) + \epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})] \\ &= Q_{t_m}(\hat{\mathbf{S}}) + \left(\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})]\right) + \left(\mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}_G}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})]\right).\end{aligned}$$

The last equality is found by writing $\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})]$ in the form of a telescopic sum and eliminating the last term based on Equation (2.12).

When there are enough samples, we have $\hat{\pi}_G(\mathbf{S}_{t_{m+1}}) \approx \pi_G(\mathbf{S}_{t_{m+1}})$, which leaves the approximation bias in SRM merely determined by

$$\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\pi_G}[\epsilon^{\hat{\pi}_G}(\mathbf{S}_{t_{m+1}})].$$

Since $\pi(\mathbf{S}_{t_{m+1}})$ stands for the analytic density function of $\mathbf{S}_{t_{m+1}}$ conditioned on $\mathbf{S}_{t_m} = \hat{\mathbf{S}}$ and $\pi_G(\mathbf{S}_{t_{m+1}})$ for the analytic density function of $\mathbf{S}_{t_{m+1}}$ conditioned on $\mathbf{S}_{t_0}$, they are obviously not identical. This makes the path-wise bias in SRM uncontrollable no matter how we change the setup of simulation.

2.5.3. ERROR IN SGBM

In SGBM, we consider the paths originating from the same bundle $\mathscr{B}_{t_m}(b)$, which covers the state $(t_m, \hat{\mathbf{S}})$, and regress the option values $\{V_{t_{m+1}}^{(b)}(\mathbf{S}_{t_{m+1}}^{(b)}(i))\}_{i=1}^{N_B(b)}$ on the basis functions $[\phi_1(\mathbf{S}_{t_{m+1}}^{(b)}(i)), \ldots, \phi_K(\mathbf{S}_{t_{m+1}}^{(b)}(i))]_{i=1}^{N_B(b)}$. Again we denote the theoretical density function of $\mathbf{S}_{t_{m+1}}$, whose previous state $(t_m, \mathbf{S}_{t_m})$ is within the spreading of bundle $\mathscr{B}_{t_m}(b)$, by $\pi_B(\mathbf{S}_{t_{m+1}})$ and the empirical density function of $\{\mathbf{S}_{t_{m+1}}^{(b)}(i)\}_{i=1}^{N_B(b)}$ by $\hat{\pi}_B(\mathbf{S}_{t_{m+1}})$.

With similar arguments as in SRM, we obtain the approximated continuation value $\hat{Q}_{t_m}^B(\hat{\mathbf{S}})$ by SGBM as:

$$\hat{Q}_{t_m}^B(\hat{\mathbf{S}}) = Q_{t_m}(\hat{\mathbf{S}}) + \left(\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})]\right) + \left(\mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})]\right).$$

Different from SRM, the setup in SGBM can help us to control the bias. When we increase the number of bundles, the spreading of any individual bundle will reduce. In the limiting case where the bundle covers only the state $(t_m, \hat{\mathbf{S}})$, we will have $\pi_B(\mathbf{S}_{t_{m+1}}) = \pi(\mathbf{S}_{t_{m+1}})$. However, in a simulation-based approach, if we do not increase the total sample size, increasing the number of bundles will cause a decrease of the number of paths per bundle, which makes the empirical density function $\hat{\pi}_B(\mathbf{S}_{t_{m+1}})$ different from the analytic density function $\pi_B(\mathbf{S}_{t_{m+1}})$.

To summarize, if we regard $\mathbb{E}^{\pi}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})]$ as the "distribution bias" and $\mathbb{E}^{\pi_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})] - \mathbb{E}^{\hat{\pi}_B}[\epsilon^{\hat{\pi}_B}(\mathbf{S}_{t_{m+1}})]$ as the "sample bias", the number of bundles is a "trade-off" between these two types of biases. To make a balance, we should choose the number of bundles neither too small nor too large so that both the biases are controlled.

Based on the analysis above, we can conclude that the path-wise estimation error of regression methods comes from two parts: the regression error and the sample bias. By choosing suitable basis functions, we reduce the impact of the first part. By introducing "bundling", we control the sample bias and also simplify the problem of global regression to that of local regression.

2.6. VARIANCE REDUCTION FOR PATH ESTIMATOR

2.6.1. PATH ESTIMATOR

From the backward pricing algorithm of SGBM, we will get a biased high estimator $\overline{V}_0(\mathbf{S}_0)$ of the initial option value $V_0(\mathbf{S}_0)$. We call this estimator the direct estimator. Once we obtain the regression parameters for any bundle at any time step, the approximated continuation value $\hat{Q}_{t_m}(\mathbf{S}_{t_m})$ of the option at the given state $(t_m, \mathbf{S}_{t_m})$ can be calculated. Relying on this approximation, we can decide either to exercise the option or to hold it at the specified state. Based on this exercise policy and some fresh simulated paths, we can develop a biased low estimator $\underline{V}_0(\mathbf{S}_0)$ of the option value $V_0(\mathbf{S}_0)$. We call this estimator the path estimator. The procedure of calculating the path estimator can be described as:

Step I Simulate a new sample of paths $\{\mathbf{S}_0(i), \ldots, \mathbf{S}_{t_M}(i)\}$, $i = 1, \ldots, N_p$.

Step II Based on the approximation of the continuation value, determine the optimal exercise time $\hat{\tau}(i)$ for the ith path:

$$\hat{\tau}(i) = \min(t_m : h(\mathbf{S}_{t_m}(i)) \geq \hat{Q}_{t_m}(\mathbf{S}_{t_m}(i)), m = 1, \ldots, M). \tag{2.13}$$

Step III Compute the path estimator:

$$\underline{V}_0(\mathbf{S}_0) = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{h(\mathbf{S}_{\hat{\tau}(i)}(i))}{\mathscr{D}_{\hat{\tau}(i)}}.$$

The proof of convergence and the bias of the path estimator are shown in [57].

2.6.2. VARIANCE REDUCTION: CONTROL VARIATES

When estimating the option value via Monte Carlo simulation, we not only desire a precise point estimate but also pursue a reasonable interval estimate, which is constructed

in the form of the point estimate plus-or-minus its standard error multiplied by the confidence factor. A satisfactory simulation-based method should provide us a narrow interval estimate.

Within the framework of the general simulation-based estimation, the standard error of the point estimate is believed to be proportional to the reciprocal of the square root of the sample size. Therefore, to reduce the range of the interval estimate by a factor of 10, the sample size should increase by a factor of 100.

Variance reduction methods offer us an alternative approach to reduce the standard error of the estimation. A commonly used variance reduction method is the control variate method, which has been implemented for American-style option pricing, for example, in [15], [19] and [73].

As the first choice for control variates for pricing Bermudan options one would consider the corresponding European options, whose values can be easily computed. From the perspective of optimal exercise we never exercise a single asset Bermudan call option before its maturity, so using the European option will provide us a zero-variance control. As concluded by Rasmussen in [73], for pricing the Bermudan option "a good control variate" should have the following two properties: it should be highly correlated with the payoff of the option in question and its conditional expectation should be easy to compute.

For simplicity, we restrict the following discussion to using only one control variate. For the generalized control variate method, where multi-controls are involved, we refer the reader to [1].

PATH ESTIMATOR WITH CONTROL VARIATES

To improve in particular the path estimator based on crude Monte Carlo simulation by control variates, the Bermudan option value $\frac{h(\mathbf{S}_{\hat{\tau}(i)}(i))}{\mathscr{D}_{\hat{\tau}(i)}}$ for the ith path will be replaced by:

$$OV_{\hat{\tau}}(i) = \frac{h(\mathbf{S}_{\hat{\tau}(i)}(i))}{\mathscr{D}_{\hat{\tau}(i)}} + \theta(Y(i) - \mathbb{E}[Y]),$$

where $Y(i)$ denotes the control variate for the ith path and $\mathbb{E}[Y]$ is its analytic expectation with filtration $\mathscr{F}_0$. The weighting parameter θ can be chosen freely, since the new estimate $OV_{\hat{\tau}}$ is always unbiased to the original estimate $\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}$. A reasonable choice would be:

$$\theta = \frac{Cov\left(\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}, Y\right)}{Var(Y)},$$

which controls the variance of $OV_{\hat{\tau}}$ to the minimum value:

$$Var\left(\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}\right)(1-\rho^2),$$

where $\rho = Corr\left(\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}, Y\right) = \frac{Cov\left(\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}, Y\right)}{\sqrt{Var\left(\frac{h(\mathbf{S}_{\hat{\tau}})}{\mathscr{D}_{\hat{\tau}}}\right)Var(Y)}}$. In [73], the ratio $\frac{1}{1-\rho^2}$ is called the "speed-up factor", which indicates that utilizing control variates is equivalent to amplifying the sample size in a crude Monte Carlo by a factor of $\frac{1}{1-\rho^2}$.

TRADITIONAL CONTROL VARIATES

As mentioned before, the conditional expectations of the control variates should be easy to calculate. For the Bermudan option with multi-dimensional underlying assets $\mathbf{S}_t = (S_t^1, \ldots, S_t^\delta, \ldots, S_t^d) \in \mathbb{R}^d$, the traditional choice of control variate is the discounted payoff of the European option measured at maturity with the single underlying asset S_t^δ, i.e.

$$Y = \frac{g(S_T^\delta)}{\mathscr{D}_T},$$

and the Monte Carlo estimate with control variate for the ith path is:

$$OV_{\hat{\tau}}(i) = \frac{h(\mathbf{S}_{\hat{\tau}(i)}(i))}{\mathscr{D}_{\hat{\tau}(i)}} + \theta\left(\frac{g(S_T^\delta(i))}{\mathscr{D}_T} - \mathbb{E}\left[\frac{g(S_T^\delta)}{\mathscr{D}_T}\right]\right).$$

Here $\mathbb{E}\left[\frac{g(S_T^\delta)}{\mathscr{D}_T}\right]$ indicates the value of the single asset European option starting at state $(0, S_0^\delta)$ and maturing at time T with the same strike as the discussed Bermudan option.

IMPROVED CONTROL VARIATES

Monte Carlo pricing with the option payoff measured at maturity as control variate is quite cheap, because after the simulation of the paths we get $\{g(S_T^\delta(i))\}_{i=1}^N$ immediately. However, empirical tests indicate that this choice of control variate is not always efficient. An alternative introduced in [73] is to replace $\frac{g(S_T^\delta(i))}{\mathscr{D}_T}$ by $ICV_{\hat{\tau}}(i)$ which is defined as:

$$ICV_{\hat{\tau}}(i) = \frac{1}{\mathscr{D}_{\hat{\tau}(i)}}\mathbb{E}_{\hat{\tau}(i)}\left[\frac{g(S_T^\delta(i))}{\mathscr{D}_{T-\hat{\tau}(i)}}\right].$$

The stopping time $\hat{\tau}(i)$ for the ith path is defined in Equation (2.13). $\mathbb{E}_{\hat{\tau}(i)}[\cdot]$ indicates the conditional expectation with filtration $\mathscr{F}_{\hat{\tau}(i)}$ and $\mathbb{E}_{\hat{\tau}(i)}\left[\frac{g(S_T^\delta(i))}{\mathscr{D}_{T-\hat{\tau}(i)}}\right]$ denotes the single asset European option value associated to the ith path starting at state $(\hat{\tau}(i), S_{\hat{\tau}(i)}^\delta(i))$ and maturing at time T with the same strike as the Bermudan option. The expectation $\mathbb{E}[ICV_{\hat{\tau}}]$ is identical to the single asset European option value $\mathbb{E}\left[\frac{g(S_T^\delta)}{\mathscr{D}_T}\right]$. In [73], Rasmussen shows that this new choice of control variate makes variance reduction more efficient. However, we notice that $\{ICV_{\hat{\tau}}(i)\}_{i=1}^N$ is not directly available any more, because for the ith path it is the discounted single asset European option value measured at time $\hat{\tau}(i)$. For one-dimensional pricing problems, fast pricing algorithms, that help us calculate the path-wise control variates $\{ICV_{\hat{\tau}}(i)\}_{i=1}^N$ efficiently exist. We will implement the COS method introduced in [37] for the one-dimensional pricing.

In our numerical test, we find that in some situations applying improved control variates is much more efficient than increasing the sample size. However, for the geometric basket option and the arithmetic basket option, the effect of using improved control variates is not so obvious that we still prefer to use traditional control variates and increase the sample size for reducing the standard error of estimation.

2.7. MERTON JUMP-DIFFUSION PROCESS

2.7.1. MOTIVATION OF JUMP-DIFFUSION MODEL

Despite the wide use of the geometric Brownian motion to model the movement of asset prices, the almost instantaneous asset price change cannot be captured well. Such rapid price variations are sometimes modeled by a "jump". It is stated in [61] that the jump model behaves better in modeling the leptokurtic feature of the asset price distribution and the empirical phenomenon "volatility smile" in option markets.

Jump-diffusion models essentially contain a Brownian component punctuated by jumps at random intervals. Compared to their counterparts "infinite activity Lévy processes" in jump models, "finite activity jump-diffusion models" are easier to simulate. In this chapter, we will consider an elementary jump model, the Merton jump-diffusion (MJD) model[7], which was introduced in [70].

2.7.2. MODEL FORMULATION

We consider the MJD model with contagious jumps on each asset, i.e. the jumps in the dynamics of each asset arrive following the same Poisson process. Under this model, the d-dimensional asset prices follow:

$$\mathrm{d}S_t^i = S_t^i((r-\delta_i-\lambda_{\text{jump}}\kappa_i)\mathrm{d}t + \mathrm{d}w_t^i + (e^{z^i}-1)\mathrm{d}\Gamma_t), \quad i=1,\dots,d,$$

where $\kappa_i = \mathbb{E}[e^{z^i}-1]$, $\mathrm{d}w_t^i \mathrm{d}w_t^j = \sigma_i\sigma_j\rho_{ij}\mathrm{d}t$, r the risk-free rate, δ_i the dividend rate, σ_i the volatility of diffusion, Γ_t a Poisson process with mean arrival rate λ_{jump}, $\mathbf{z} = [z^1,\dots,z^d]'$ the multivariate normally distributed jumps with mean $\mu^J = [\mu_1^J,\dots,\mu_d^J]'$ and covariance matrix Σ^{J} with elements $\Sigma_{ij}^J = \sigma_i^J\sigma_j^J\rho_{ij}^J$.

The analytic formulas for the dynamics read:

$$S_t^i = S_0^i \exp((r-\delta_i-\lambda_{\text{jump}}\kappa_i)t + w_t^i)\exp\left(\sum_{m=1}^{\text{NP}(t)} z_m^i\right), \quad i=1,\dots,d, \tag{2.14}$$

where $\mathbf{S}_0 = (S_0^1,\dots,S_0^d)$ is the initial state, $\mathbf{w}_t = [w_t^1,\dots,w_t^d]'$ the diffusion component, $\mathbf{z}_m = [z_m^1,\dots,z_m^d]'$ the jump component and $\text{NP}(t)$ the number of Poisson jumps within time interval t with mean arrival rate λ_{jump}. The diffusion component $\mathbf{w}_t$ follows multivariate normal distribution with mean 0 and covariance matrix Σ with elements $\Sigma_{ij} = \sigma_i\sigma_j\rho_{ij}t$ and the jump component $\mathbf{z}_m$ with mean $\mu^J = [\mu_1^J,\dots,\mu_d^J]'$ and covariance matrix Σ^{J} with elements $\Sigma_{ij}^J = \sigma_i^J\sigma_j^J\rho_{ij}^J$.

The log-process $\mathbf{X}_t = (X_t^1,\dots,X_t^d)$, where $X_t^i = \log(S_t^i), i=1,\dots,d$, has analytic form:

$$\mathbf{X}_t = \mathbf{X}_0 + \mu\cdot t + \mathbf{w}_t + \sum_{m=1}^{\text{NP}(t)} \mathbf{z}_m,$$

where $\mu = [r-\delta_1-\lambda_{\text{jump}}\kappa_1 - \frac{\sigma_1^2}{2},\dots,r-\delta_d-\lambda_{\text{jump}}\kappa_d-\frac{\sigma_d^2}{2}]$.

[7] SGBM is also feasible for another well-known jump-diffusion model, the Kou model [61]. Since the only distinction between the MJD model and the Kou model is the distribution of jump sizes, all discussions in this chapter about SGBM can be extended to the Kou model as well.

2.7.3. DIMENSION REDUCTION: GEOMETRIC AVERAGE OF MJD ASSETS

Within the framework of GBM, the dynamics of the geometric average of multi-dimensional assets can be formulated as a one-dimensional problem. This technique is also applicable for the MJD model[8]. Suppose that the equivalent one-dimensional MJD model has analytic formula:

$$\tilde{S}_t = \tilde{S}_0 \exp((r - \tilde{\delta} - \lambda_{\text{jump}}\tilde{\kappa})t + \tilde{w}_t) \exp\left(\sum_{m=1}^{\text{NP}(t)} \tilde{z}_m\right),$$

where $\tilde{w}_t$ is normally distributed with mean 0 and variance $\tilde{\sigma}^2 t$ and $\tilde{z}_m$ normally distributed with mean $\tilde{\mu}^J$ and variance $\tilde{\sigma}^J$. To make it represent the geometric mean of the assets with dynamics shown in Equation (2.14), we need:

$$\tilde{S}_0 = \left(\prod_{i=1}^{d} S_0^i\right)^{\frac{1}{d}}, \qquad \tilde{\mu}^J = \frac{\sum_{i=1}^{d} \mu_i^J}{d},$$

$$\tilde{\sigma}^J = \frac{\sqrt{\sum_{i,j} \sigma_i^J \sigma_j^J \rho_{ij}^J}}{d}, \qquad \tilde{\sigma} = \frac{\sqrt{\sum_{i,j} \sigma_i \sigma_j \rho_{ij}}}{d},$$

$$\tilde{\kappa} = \exp\left(\tilde{\mu}^J + \frac{(\tilde{\sigma}^J)^2}{2}\right) - 1, \quad \tilde{\delta} = \frac{\sum_{i=1}^{d} (\delta_i + \frac{\sigma_i^2}{2} + \lambda_{\text{jump}}\kappa_i)}{d} - \frac{\tilde{\sigma}^2}{2} - \lambda_{\text{jump}}\tilde{\kappa}.$$

Remark 2.7.1. *This dimension reduction technique also works on the geometric basket containing assets following geometric Brownian motion and assets following the MJD model, because geometric Brownian motion can be regarded as an MJD model with zero jump size.*

2.7.4. ANALYTIC MOMENTS OF BASIS FUNCTIONS IN THE MJD MODEL

For the model with dynamics shown in Equation (2.14), we have the conditional expectations of polynomial basis functions in closed form. The conditional moments of the original stock prices, $\mathbb{E}[(S_t^i)^k | S_0^i] (i = 1, \ldots, d; k = 1, 2, \ldots)$, read:

$$\mathbb{E}[(S_t^i)^k | S_0^i] = (S_0^i)^k \exp\left(k\hat{\mu}_i t + \frac{1}{2}k^2\sigma_i^2 t + \lambda_{\text{jump}} t(\exp\left(k\mu_i^J + \frac{1}{2}k^2(\sigma_i^J)^2\right) - 1)\right), \tag{2.15}$$

where

$$\hat{\mu}_i = r - \delta_i - \frac{\sigma_i^2}{2} - \lambda_{\text{jump}}(\exp\left(\mu_i^J + \frac{(\sigma_i^J)^2}{2}\right) - 1).$$

The conditional moments of the geometric mean of the asset prices $\{S_t^i\}_{i=1}^d$ can be calculated by first presenting the dynamics of the geometric mean in one dimension as shown in Section 2.7.3 then using Equation (2.15).

Since there exists no general form of the conditional expectation of the log-stock prices $\mathbb{E}[\log(S_t^i)^k | S_0^i] (i = 1, \ldots, d; k = 1, 2, \ldots)$, we present the first three moments as fol-

[8] The jumps in the dynamics of each asset should however follow the same Poisson process.

lows:

$$\begin{aligned}
\mathbb{E}[\log(S_t^i)|S_0^i] &= \log(S_0^i) + \hat{\mu}_i t + \lambda_{\text{jump}} t \mu_i^J, \\
\mathbb{E}[\log(S_t^i)^2|S_0^i] &= (\log(S_0^i) + \hat{\mu}_i t)^2 + \sigma_i^2 t + 2(\log(S_0^i) + \hat{\mu}_i t)(\mu_i^J \lambda_{\text{jump}} t) \\
&+ (\lambda_{\text{jump}}^2 t^2 + \lambda_{\text{jump}} t)(\mu_i^J)^2 + \lambda_{\text{jump}} t (\sigma_i^J)^2, \\
\mathbb{E}[\log(S_t^i)^3|S_0^i] &= (\log(S_0^i) + \hat{\mu}_i t)^3 + 3(\log(S_0^i) + \hat{\mu}_i t)(\sigma_i^2 t) \\
&+ 3((\log(S_0^i) + \hat{\mu}_i t)^2 + \sigma_i^2 t)(\mu_i^J \lambda_{\text{jump}} t) \\
&+ 3(\log(S_0^i) + \hat{\mu}_i t)((\lambda_{\text{jump}}^2 t^2 + \lambda_{\text{jump}} t)(\mu_i^J)^2 + \lambda_{\text{jump}} t (\sigma_i^J)^2) \\
&+ (\lambda_{\text{jump}}^3 t^3 + 3\lambda_{\text{jump}}^2 t^2 + \lambda_{\text{jump}} t)(\mu_i^J)^3 + 3(\lambda_{\text{jump}}^2 t^2 + \lambda_{\text{jump}} t)(\mu_i^J (\sigma_i^J)^2).
\end{aligned}$$

The conditional expectation of the cross-product term $\mathbb{E}[\log(S_t^i)\log(S_t^j)|S_0^i, S_0^j]$ $(i \neq j)$ reads:

$$\begin{aligned}
\mathbb{E}[\log(S_t^i)\log(S_t^j)|S_0^i, S_0^j] &= (\log(S_0^i) + \hat{\mu}_i t)(\log(S_0^j) + \hat{\mu}_j t) + \sigma_i \sigma_j \rho_{ij} t \\
&+ (\log(S_0^j) + \hat{\mu}_j t)(\mu_i^J \lambda_{\text{jump}} t) + (\log(S_0^i) + \hat{\mu}_i t)(\mu_j^J \lambda_{\text{jump}} t) \\
&+ (\lambda_{\text{jump}}^2 t^2 + \lambda_{\text{jump}} t)\mu_i^J \mu_j^J + \lambda_{\text{jump}} t \sigma_i^J \sigma_j^J \rho_{ij}^J.
\end{aligned}$$

2.8. NUMERICAL EXPERIMENTS

In this section we perform several numerical experiments to test the performance of SGBM for pricing different types of Bermudan options with assets following the MJD process. We compare different choices of bundling references, basis functions and variance reduction approaches for options on multi-dimensional assets.

The one-dimensional MJD model is furnished with three different choices of model parameters, which respectively indicate "common" jump, "intensive" jump and "rare" jump. The multi-dimensional tests are conducted for various options: geometric basket, arithmetic basket, put-on-min and call-on-max. For some scenarios, we get the benchmark value directly from the literature. However, in case of absence of references we generate the benchmark ourselves. For the geometric basket option, the representation discussed in Section 2.7.3 is implemented and the one-dimensional problem is solved by the COS method. For the remaining scenarios, we implement the LSM method [65] to generate the reference values. MATLAB R2011b is used and the computations are performed on Intel(R) Core(TM) i5 3.33 GHz processor with 16GB RAM.

The parameter sets used for the tests are listed in Table 2.1.

2.8.1. SGBM AND TUNING PARAMETERS

We start testing the performance of SGBM on single asset options under geometric Brownian motion dynamics, which gives a general insight how SGBM performs with its tuning parameters. The performance of SGBM is influenced by three parameters[9]:

1. N and N_p: the number of simulated paths,

[9] In fact, the number of basis functions is also a tuning parameter, but large number of basis functions may result in an over-fitting problem in the regression step. We therefore choose the number of basis functions just three or four in the one-dimensional case.

Table 2.1: Parameter settings used in the test.

Set I(a): "common" jump
$S_0 = 40$, $K = 40$, $r = 0.06$, $\delta = 0$, $\sigma = 0.2$, $\lambda_{\text{jump}} = 3$, $\mu^J = -0.2$, $\sigma^J = 0.2$, $T = 1$, $M^{\text{a}} = 20$.
Set I(b): "intensive" jump
$S_0 = 40$, $K = 40$, $r = 0.06$, $\delta = 0$, $\sigma = 0.2$, $\lambda_{\text{jump}} = 8$, $\mu^J = -0.2$, $\sigma^J = 0.2$, $T = 1$, $M = 20$.
Set I(c): "rare" jump
$S_0 = 40$, $K = 40$, $r = 0.06$, $\delta = 0$, $\sigma = 0.2$, $\lambda_{\text{jump}} = 0.1$, $\mu^J = -0.9$, $\sigma^J = 0.45$, $T = 1$, $M = 20$.
Set II:
$\mathbf{S}_0 = [100, 100]'$, $K = 100$, $r = 0.05$, $\delta = 0$, $\sigma = [0.12, 0.15]'$, $\rho_{ij} = 0.3$, $\lambda_{\text{jump}} = 0.6$,
$\mu^J = [-0.1, 0.1]'$, $\sigma^J = [0.17, 0.13]'$, $\rho^J_{ij} = -0.2$, $T = 1$, $M = 8$.
Set III:
$\mathbf{S}_0 = [100, 100]'$, $K = 100$, $r = 0.05$, $\delta = 0.1$, $\sigma = [0.2, 0.2]'$, $\rho_{ij} = 0$, $T = 3$, $M = 9$.
Set IV:
$\mathbf{S}_0 = [100, 100, 100, 100, 100]'$, $K = 100$, $r = 0.05$,
$\delta = 0$, $\sigma = [0.15, 0.15, 0.15, 0.15, 0.15]'$, $\rho_{ij} = 0.3$, $\lambda_{\text{jump}} = 0.5$,
$\mu^J = [-0.3, -0.2, -0.1, 0.1, 0.2]'$, $\sigma^J = [0.1, 0.1, 0.1, 0.1, 0.1]'$, $\rho^J_{ij} = -0.2$, $T = 1$, $M = 8$.

[a] M denotes the number of early exercise opportunities, which are equidistantly distributed in T years.

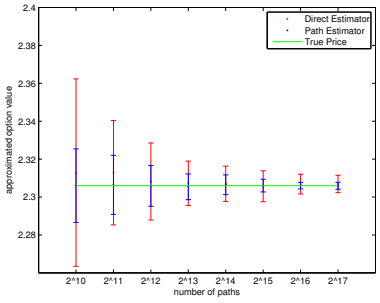
2. n: the number of bundles,

3. M: the number of exercise opportunities.

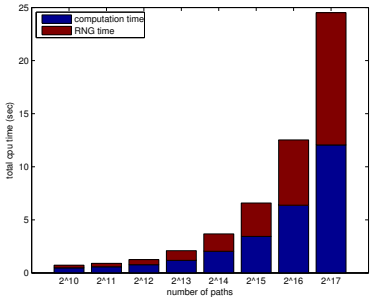
In the following tests, the default setup is: $n = 16, M = 20, N = 2^{17}, N_p = 2^{-2} \cdot N$. The model parameters are chosen from Set I in Table 2.1 without the jump component. We use the improved control variates for variance reduction on the path estimator, and perform three tests respectively by changing n, M and N. The test results are plotted in Figure 2.4. Figure 2.4(a) shows that as the sample size increases, the standard error of the estimators in SGBM decreases by the order $N^{-1/2}$. As shown in Figure 2.4(b) the total computational time increases in order N. When we increase the number of exercise opportunities up to $M = 128$, both the direct and the path estimator of SGBM are satisfactory. If we keep doubling the number of exercise opportunities, the performance of the direct estimator in SGBM decreases while the path estimator remains reliable. The poor performance of the direct estimator in case of many exercise opportunities is mainly caused by the "distribution bias" as explained in Section 2.5. However, in any scenario the path estimator is approving, as in LSM. Figure 2.4(d) displays the trade-off between the "distribution bias" and the "sample bias". When the number of bundles is small, the "distribution bias" is the dominant part that makes the direct estimator highly biased. As the number of bundles increases, SGBM exhibits highly satisfactory performance. However, when the number of bundles increases further, the "sample bias" forms a problem and the direct estimator in SGBM becomes unsatisfactory.

2.8.2. CHOICE OF BUNDLING REFERENCE

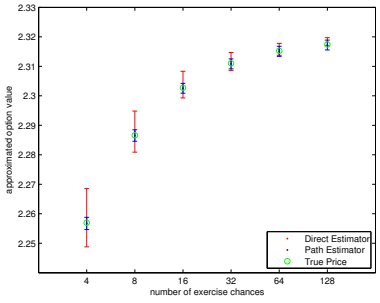
In this section we consider different bundling schemes while testing and focusing on pricing the put-on-min option with assets $\mathbf{S}_t = (S^1_t, S^2_t)$ following the two-dimensional



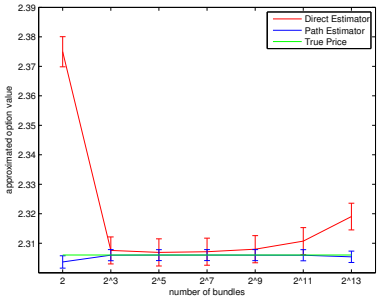
(a) Fix M and n, vary N



(b) Fix M and n, vary N



(c) Fix N and n, vary M



(d) Fix N and M, vary n

Figure 2.4: Performance of SGBM with different tuning parameters. The setup is described in Section 2.8.1. The reference option price is generated by the COS method [37].

MJD model. The model parameters are chosen from Set II in Table 2.1. The basis functions are fixed as: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$. Seven different ways for bundling are included in the test. They are:

1. bundling according to one reference:
   - bundling reference A: $R^{\mathrm{A}}(\mathbf{S}_t) = \min(S_t^1, S_t^2)$,
   - bundling reference B: $R^{\mathrm{B}}(\mathbf{S}_t) = S_t^1$,
   - bundling reference C: $R^{\mathrm{C}}(\mathbf{S}_t) = S_t^1 - S_t^2$.
2. bundling according to two references:
   - bundling reference D: $R_1^{\mathrm{D}}(\mathbf{S}_t) = S_t^1$, $R_2^{\mathrm{D}}(\mathbf{S}_t) = S_t^2$,
   - bundling reference E: $R_1^{\mathrm{E}}(\mathbf{S}_t) = \min(S_t^1, S_t^2)$, $R_2^{\mathrm{E}}(\mathbf{S}_t) = S_t^1 - S_t^2$,
   - bundling reference F: $R_1^{\mathrm{F}}(\mathbf{S}_t) = \min(S_t^1, S_t^2)$, $R_2^{\mathrm{F}}(\mathbf{S}_t) = S_t^1$,
   - bundling reference G: $R_1^{\mathrm{G}}(\mathbf{S}_t) = S_t^1 - S_t^2$, $R_2^{\mathrm{G}}(\mathbf{S}_t) = S_t^1$.

According to Figure 2.5, when pricing the put-on-min options we should not limit ourselves to bundling with a single reference, which is outperformed by any bundling scheme with two references. Among the two-reference bundling schemes, the one involving the intrinsic value of the option and the difference between the asset values is the best choice. For the geometric basket option and the arithmetic basket option, we do not present our test results. However, for both of them bundling simply with the option's intrinsic value yields highly satisfactory results.

2.8.3. CHOICE OF BASIS FUNCTIONS

One aspect influencing the efficiency of regression methods for option pricing is the choice of basis functions. Although we claim that it is sufficient to get convergent results for various option contracts by simply choosing polynomials as basis functions, in some situations we have alternative choices. For example, for the geometric basket option it is recommended in [57] to choose the powers of the geometric mean of asset prices as basis functions. We compare three choices of basis functions for pricing the geometric basket option with assets following the two-dimensional MJD process:

- Basis A: the intrinsic value of option

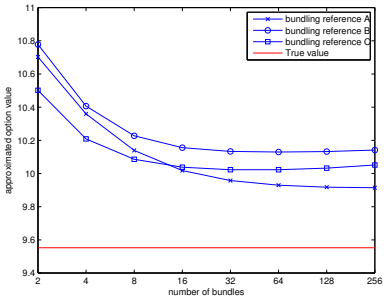
$$1, \sqrt{S_t^1 S_t^2}, S_t^1 S_t^2, \sqrt{S_t^1 S_t^2}^3, (S_t^1 S_t^2)^2.$$

- Basis B: polynomial terms of asset prices

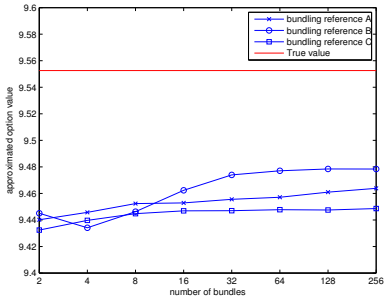
$$1, \log(S_t^1), \log(S_t^1)^2, \log(S_t^2), \log(S_t^2)^2, \log(S_t^1)\log(S_t^2).$$

- Basis C: polynomial terms of asset prices, without the cross-product term

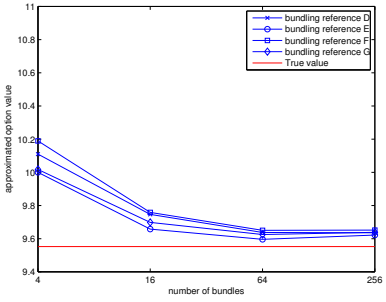
$$1, \log(S_t^1), \log(S_t^1)^2, \log(S_t^2), \log(S_t^2)^2.$$



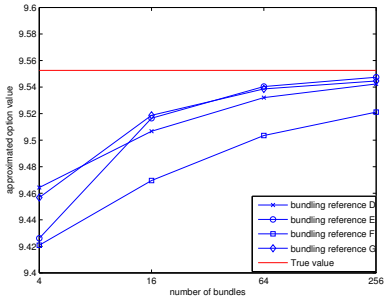
(a) Direct estimator (one bundling reference)



(b) Path estimator (one bundling reference)

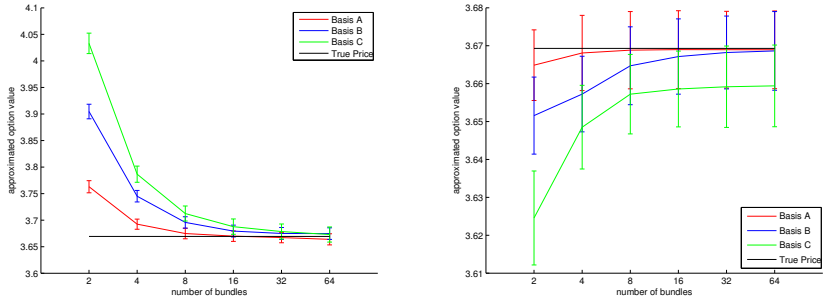


(c) Direct estimator (two bundling reference)



(d) Path estimator (two bundling reference)

Figure 2.5: Comparison of different bundling schemes for pricing two-dimensional put-on-min option. The basis functions are fixed as: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$. When the bundling is done according to two references, the number of bundles with respect to each reference is the square root of the "number of bundles". The sample size for the direct estimator is 2^{17} and the sample size for the path estimator is 2^{18}. The reference option price is collected from [74].



(a) Direct estimator in SGBM

(b) Path estimator in SGBM

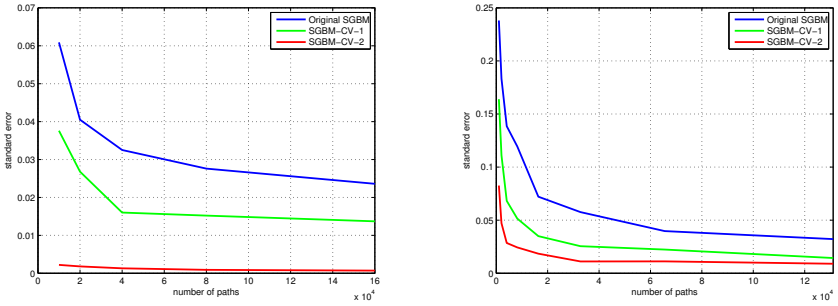
Figure 2.6: Comparing different choices of basis functions in SGBM for pricing two-dimensional geometric basket option. The parameters of the MJD model are chosen from Set II in Table 2.1. The sample size for the direct estimator is 2^{17}. The path estimator with sample size 2^{18} is controlled by traditional control variates.

Figure 2.6 shows that different choices of basis functions in SGBM have an impact on the option price estimates. When the number of bundles is small, Basis A performs best. When the number of bundles is sufficiently large, the final results of SGBM with Basis A and Basis B are very similar. On the other hand, we notice that although Basis C does not appear satisfactory compared to the other two choices, the confidence intervals of the associated direct and the path estimator cover the true option values. For truly high-dimensional problems, including the cross-product terms into basis functions will lead to a quadratic increase of the number of basis functions, which further requires an exponential increase of the sample size to make the regression accurate [47]. In this case, we prefer to form basis functions by the polynomials without cross-product terms. However, ignoring the cross-product terms may lead to approximation errors for some particular types of options.

So, when dealing with low-dimensional problems, we choose the ordinary polynomials as basis functions since their conditional expectations are always available. When the dimensionality of the problem is high, we consider polynomials without cross-product terms, or, if possible, we use polynomials of the option's intrinsic values as the basis functions. In Section 2.8.7 we can see that these choices of basis functions provide us highly satisfactory results.

2.8.4. EFFICIENCY OF USING CONTROL VARIATES

One problem with SGBM [57] is that the standard error of the path estimator is usually larger than that of the direct estimator. Here we test the variance reduction methods introduced in Section 2.6 for reducing the standard error of the path estimator. The test is performed under one-dimensional and two-dimensional MJD models. Figures 2.7(a) and 2.7(b) show that using the control variates helps to reduce the standard error of the path estimator. In the one-dimensional case using the improved control variates is extremely efficient with a speed-up factor around 900, while for the two-dimensional



(a) Using control variates for a put option with asset following a one-dimensional MJD model.

(b) Using control variates for a put-on-min option with assets following a two-dimensional MJD model.

Figure 2.7: Comparison of two variance reduction approaches. The standard errors of the path estimators following three different algorithms are presented. "Original SGBM" means standard SGBM algorithm without control, "SGBM-CV-1" stands for SGBM with the traditional control variates and "SGBM-CV-2" for SGBM with the improved control variates. For all three algorithms, the basis functions, the bundling reference and the number of the paths for the direct estimator are identical. The x-axis indicates the sample size for the path estimator.

model the improved control variate is less efficient with speed-up factor around 14. In both scenarios, applying the traditional control variates provides us a variance reduction with speed-up factor around 3.

The traditional control variates is free of additional cost and therefore we should treat the traditional control variates as an alternative to the improved control variates. As shown in Table 2.2, for geometric and arithmetic basket options, the improved control variates is not effective regarding its cost. In these cases we will use the traditional control variates and increase the sample size for variance reduction.

2.8.5. 1D PROBLEM

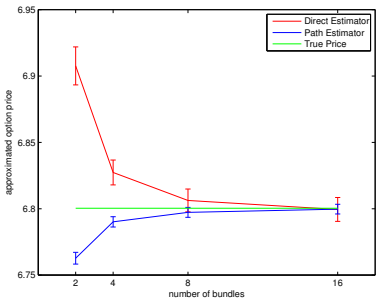
We start systematic testing under the one-dimensional MJD process. Three different types of jumps are considered: common jump, intensive jump and rare jump. Their model parameters are respectively Set I(a), Set I(b) and Set I(c) in Table 2.1. The basis functions in SGBM are chosen as $1, \log(S_t), \log(S_t)^2, \log(S_t)^3$. We choose improved control variate for the one-dimensional case.

In Figure 2.8, we see that the path estimator is always an accurate lower bound estimate to the true option price: its standard error is small and it is consistently smaller than the true option price. The direct estimator also converges to the true option value as the number of bundles grows. The convergence is rapid for the MJD process with common jump and for that with intensive jump. In the rare jump case although the convergence is not satisfactory, the exercise strategy associated to the direct estimator is accurate since the relevant path estimator is very close to the true value. The estimation error of the direct estimator is the curse of rare events. For the MJD model with the rare

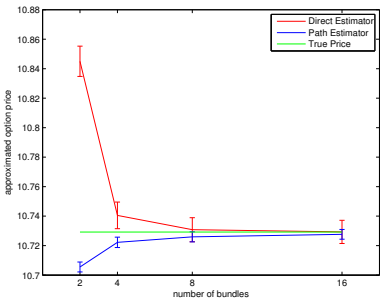
Table 2.2: Comparing two variance reduction approaches in SGBM pricing for different types of two-dimensional options. The sample size for the direct estimator is always fixed as 2^{17}.

	geometric basket		arithmetic basket		put-on-min		
N_p[a]	2^{16}	2^{17}	2^{16}	2^{17}	2^{16}	2^{18}	2^{19}
CV[b]	CV-2	CV-1	CV-2	CV-1	CV-2	CV-1	No
s.e.[c]	0.0123	0.0097	0.0118	0.0107	0.0093	0.0106	0.0151
RNG time[d]	6.2051	11.4671	6.2051	11.4671	6.2051	23.7172	48.8246
CPU time	20.3588	2.9489	20.3984	3.2454	21.8977	5.3324	10.1195
total time	26.5639	14.4160	26.6035	14.7125	28.1028	29.0496	58.9441
speed-up	3.7284	2.1081	2.7834	1.8016	13.8211	3.9224	1

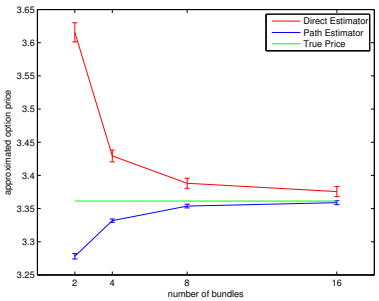
[a] The sample size for the path estimator.
[b] "CV-1" stands for SGBM with the traditional control variates and "CV-2" for SGBM with the improved control variates.
[c] The standard error of the path estimator.
[d] The time for generating the sample of the path estimator. Its unit is seconds.



(a) common jump



(b) intensive jump



(c) rare jump

Figure 2.8: SGBM with the improved control variates is implemented for pricing options with underlying assets following three different one-dimensional MJD processes. The number of the paths for estimating direct estimator is 200000, while the number of the paths for estimating path estimator is only 20000. The reference price is generated using the COS method [37].

Table 2.3: Test of SGBM for two-dimensional max-on-call options.

$\mathbf{S}_0$	Reference upper bound[b] (s.e.)	Reference lower bound (s.e.)	SGBM[a] upper bound[c] (s.e.)	SGBM lower bound (s.e.)	Ref. Value[d]
(90,90)	8.105 (0.086)	8.067 (0.020)	8.075 (0.011)	8.072 (0.008)	8.075
(100,100)	13.906 (0.035)	13.898 (0.023)	13.907 (0.017)	13.897 (0.012)	13.902
(110,110)	21.339 (0.023)	21.338 (0.022)	21.352 (0.022)	21.338 (0.015)	21.345

[a] The sample size for the direct estimator of SGBM is 2^{17} and that for the path estimator is 2^{16}. The single asset European option values measured at the exercise time are used as the control variates for the path estimator. The bundling done in "SGBM" is based on two references: the maximum of assets' prices and the difference between assets' prices. According to each bundling reference, 16 bundles are constructed. This leads to 256 bundles in total. The basis functions are: $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$.
[b] This upper bound is provided in [57], where duality approach is applied to generate this.
[c] This upper bound is just the direct estimator of SGBM.
[d] The reference value is obtained from [57].

jump, the sample distribution of the paths within one bundle is quite likely to be biased to their analytic distribution and consequently SGBM may be inaccurate according to our discussion in Section 2.5.

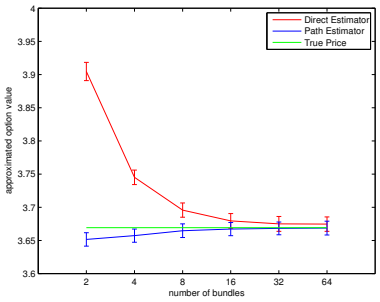
Using improved control variates provides efficient variance reduction in the one-dimensional case. Even though the path estimator has a sample size of only one tenth of the direct estimator, the standard error of the path estimator is smaller. Moreover, even when the number of bundles is small, the path estimator is much closer to the true value than the direct estimator.

2.8.6. 2D PROBLEM

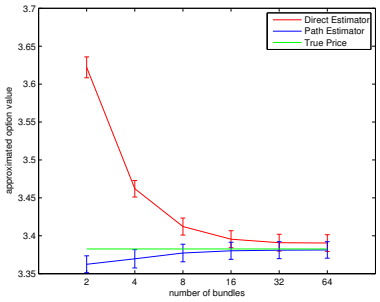
In this section we test SGBM on three different types of two-dimensional options. We fix the basis functions as $1, \log(S_t^1), \log(S_t^2), \log(S_t^1)^2, \log(S_t^2)^2, \log(S_t^1)\log(S_t^2)$ and perform the bundling based on the bundling references recommended earlier: for the geometric basket option and the arithmetic basket option, the paths are bundled according to the option's intrinsic value; for the min option, the paths are bundled based on the option's intrinsic value and the difference between asset prices.

Besides the test on the MJD model, we also consider pricing the max-on-call option with assets following two-dimensional geometric Brownian motion. We compare our results to those of the same test in [57]. The model parameters are presented in Set III of Table 2.1. The test results are shown in Table 2.3. We find that with proper choice of bundling reference SGBM performs highly satisfactorily for pricing max options even though the maximum of the asset values is not included in the basis functions.

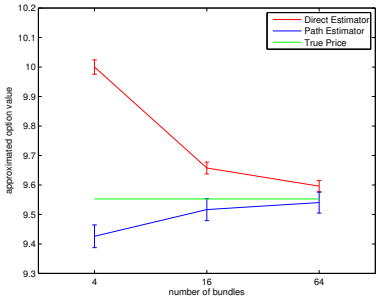
In Table 2.4, we compare the results of SGBM, presented in Figure 2.9, to those of LSM. The parameters of the MJD model are chosen from Set II in Table 2.1. The esti-



(a) geometric basket option



(b) arithmetic basket option



(c) put-on-min option

Figure 2.9: SGBM with control variates for pricing three different types of options with assets following two-dimensional MJD process. The sample size for the direct estimator is always fixed as 2^{17}. For the geometric basket option and the arithmetic basket option, we cast the traditional control variates on the path estimator with sample size 2^{18}. For the put-on-min option, the path estimator with sample size 2^{16} is controlled by the improved control variates. We always consider two controls equal to the European option value of the single asset. The reference option value for the put-on-min option is acquired from [74].

Table 2.4: Comparing SGBM and LSM for pricing three different types of options for two-dimensional MJD model.

	SGBM			LSM[a]		
	DE (s.e.)	PE (s.e.)	CPU time[b] (RNG time[c])	option value (s.e.)	CPU time (RNG time)	Ref. Value
Geometric basket option	3.6747 (0.0055)	3.6686 (0.0053)	8.9260 (34.3233)	3.6682 (0.0063)	2.4796 (46.4791)	3.6693
Arithmetic basket option	3.3904 (0.0056)	3.3812 (0.0055)	8.8582 (34.3233)	3.3813 (0.0066)	1.9655 (46.4791)	3.3825
Put-on-min option	9.5960 (0.0098)	9.5404 (0.0181)	22.0341 (18.1747)	9.5075 (0.0158)	2.2155 (46.4791)	9.5526

[a] The sample size for LSM is 2^{19}.
[b] The time for backward recursive calculation. The unit is second.
[c] The time for simulating paths. The unit is second.

mated option values using 64 bundles are chosen to stand for the reference results of SGBM. We see that for the mean basket options the result of LSM is similar to the path estimator of SGBM. However, for the put-on-min option, SGBM gives a better estimate than LSM.

Remark 2.8.1. *In Figure 2.9(c) we see that the direct estimator of SGBM is not satisfactory. The main cause for this bias is the volatility of the jump size. High volatility of the jump size implies difficulty of having unbiased samples. The impact of the jump intensity is similar but much smaller than that of the volatility of jump size.*

The direct estimator has higher bias than the path estimator. This observation is consistent to the conclusion in [79], as the direct estimator can be viewed as the "value function approximation" and the path estimator as the "stopping time approximation" [79].

2.8.7. 5D PROBLEM

According to our two-dimensional tests, LSM does not perform well at pricing min options with jump assets. For the five-dimensional case, since there is no reliable reference price for the min or max option with jump assets, we restrict our discussion to the geometric and the arithmetic basket options. In Table 2.5, we compare LSM with SGBM for pricing five-dimensional geometric basket options. In-the-money, at-the-money and out-of-the-money options are included and two different types of basis functions are investigated:

- Basis A: $\phi_1(\mathbf{S}_t) = 1$, $\phi_{k+1}(\mathbf{S}_t) = (\prod_{i=1}^{5} S_t^i)^{\frac{k}{5}} \quad k = 1,2,3,4$;
- Basis B: $\phi_1(\mathbf{S}_t) = 1$, $\phi_{2i}(\mathbf{S}_t) = \log(S_t^i)$, $\phi_{2i+1}(\mathbf{S}_t) = \log(S_t^i)^2 \quad i = 1,2,3,4,5$.

In Table 2.5, we can see that Basis A offers slightly better results than Basis B. However, both are close to the reference value.

Table 2.6 contains the results for pricing five-dimensional arithmetic basket options. For this type of option, since the conditional expectation of the power of its intrinsic

Table 2.5: SGBM for pricing five-dimensional geometric basket options. The parameters of the model are chosen from Set IV. The sample sizes for both the direct estimator and the path estimator in SGBM are 2^{17}. The sample size for LSM is 2^{18}. 64 bundles are constructed in SGBM with the intrinsic value of option as the bundling reference.

K	Ref. Value[a]	algorithm	Direct est. (s.e.)	Path est. (s.e.)	Path est. with CV[b] (s.e.)	CPU time[c] (seconds)
90	0.5564	SGBM-A[d]	0.5567(0.0010)	0.5572 (0.0033)	0.5575 (0.0029)	12.4011
		SGBM-B	0.5588(0.0026)	0.5564 (0.0035)	0.5567 (0.0030)	9.3775
		LSM		0.5563 (0.0031)		1.4305
100	3.1231	SGBM-A	3.1233(0.0037)	3.1220 (0.0097)	3.1226 (0.0071)	12.2884
		SGBM-B	3.1228(0.0052)	3.1198 (0.0096)	3.1204 (0.0073)	9.4389
		LSM		3.1238 (0.0063)		1.8754
110	9.8020	SGBM-A	9.8025(0.0075)	9.8014 (0.0108)	9.8018 (0.0101)	12.2220
		SGBM-B	9.8055(0.0080)	9.7986 (0.0104)	9.7990 (0.0102)	9.3486
		LSM		9.8045 (0.0103)		2.4182

[a] The reference price is generated by using the technique introduced in Section 2.7.3 to reduce the high-dimensional problem to one dimension and pricing the one-dimensional option by the COS method.
[b] The traditional control variates are used here.
[c] The computation time includes the time to compute the direct estimator and the path estimator. However, it does not cover the simulation time, which is around 45 seconds.
[d] "SGBM-A" stands for SGBM with Basis A and "SGBM-B" for SGBM with Basis B.

value has a complicated form, we will not consider them as basis functions. According to our test, SGBM with Basis B still performs well with a small difference between the direct and the path estimator, which means that the true option value is located in an interval with sharp bounds.

2.9. CONCLUSION

We utilized the Stochastic Grid Bundling Method (SGBM), which is a hybrid of regression-based and bundling-based Monte Carlo methods, to solve Bermudan option pricing problems. SGBM was compared to the standard regression method and its configuration is thoroughly discussed, including how to choose basis functions for regression and how to partition the bundles. We conducted error analysis on the regression-based pricing methods, especially focusing on the features of SGBM. Traditional and improved control variate methods were introduced for variance reduction in SGBM. Numerical examples on the Merton jump-diffusion (MJD) model were presented for problems up to five dimensions.

Bundling has a significant impact on the accuracy of SGBM. For the arithmetic and geometric basket options it is sufficient to choose the intrinsic value of the option as the bundling reference, but for "min" or "max" options introducing more than one bundling reference is preferred. Control variates work well for reducing the variance of the path estimator in SGBM. In the one-dimensional case, using an improved control variate is highly efficient. When the dimension of problem grows, the cost for implementing the improved control variates increases while its effect decreases. As a result, we favor the traditional control variates in the high-dimensional case.

Table 2.6: SGBM for pricing five-dimensional arithmetic basket options. The parameters of the model are chosen from Set IV. The sample sizes for both the direct estimator and the path estimator in SGBM are 2^{17}. The sample size for LSM is 2^{18}. 64 bundles are constructed in SGBM with the intrinsic value of option as the bundling reference.

K	algorithm	Direct est. (s.e.)	Path est. (s.e.)	Path est. with CV[a] (s.e.)	CPU time[b] (seconds)
90	SGBM-B	0.3516(0.0017)	0.3504 (0.0030)	0.3506 (0.0025)	9.0072
	LSM		0.3508 (0.0024)		0.8488
100	SGBM-B	2.5783(0.0048)	2.5745 (0.0078)	2.5748 (0.0060)	9.3762
	LSM		2.5795 (0.0059)		1.1445
110	SGBM-B	9.4683(0.0082)	9.4604 (0.0101)	9.4606 (0.0100)	9.0477
	LSM		9.4675 (0.0091)		2.4182

[a] The traditional control variates are used here.

[b] The computation time includes the time to compute the direct estimator and the path estimator. However, it does not cover the simulation time, which is around 45 seconds.

We have shown that it is sufficient to choose the basis functions in SGBM as polynomials to get convergent results. The outcome of our tests suggests that sometimes it is not necessary to include all terms as the basis functions and in some situations choosing the basis functions determined by the type of option contract could be more effective. When the dimensionality of the problem is not large, it appears feasible to use SGBM with basis functions of polynomial type. According to our experience, choosing polynomials without cross-product terms as basis functions works even for ten-dimensional problems. However, when the dimensionality of problem surges up, we need alternatives to polynomials as basis functions to release ourselves from the corresponding demand for the huge sample size. As mentioned in [57], polynomials of the intrinsic value of option are promising choices in this scenario.

CHAPTER 3

Multi-period Portfolio Management based on Utility Optimization

We enhance a well-known dynamic portfolio management algorithm, the BGSS algorithm, proposed by Brandt, Goyal, Santa-Clara and Stroud (Review of Financial Studies, 18, 831-873, 2005). We equip this algorithm with the components from the Stochastic Grid Bundling Method (SGBM) for calculating conditional expectations. When solving the first-order conditions for a portfolio optimum, we implement a Taylor series expansion based on a nonlinear decomposition to approximate the utility functions. In the numerical tests, we show that our algorithm is accurate and robust in approximating the optimal investment strategies, which are generated by a new benchmark approach based on the COS method (SIAM J. Sci. Comput., 31, 826-848, 2008).

Keywords: Dynamic portfolio management · Simulation method · Least-square regression · Taylor expansion · Fourier cosine expansion method

3.1. INTRODUCTION

Solving dynamic portfolio management problems has become an interesting topic ever since empirical findings in financial research suggested that asset returns were predictable. When the distributions of the asset returns are time-invariant, [69] and [75] have shown that an investor using a power utility function, who re-balances her portfolio optimally, should choose the same asset allocation at every time point, regardless of the investment horizon[1].

However, if the distributions of the asset returns are time-dependent, for example, when the asset returns follow a vector auto-regression (VAR) model, the optimal asset allocations at intermediate time points are usually not identical. In this case, an investor, who aims to find an optimal asset allocation at initial time, has to first consider all possible asset allocations in subsequent time points, which in turn are also influenced by the initial investment decision. Mathematically, an investor must solve a multivariate optimization problem regarding asset allocations at all portfolio re-balancing opportunities.

Generally it is difficult to solve this multivariate optimization problem directly, and therefore this problem is usually solved by a backward recursion process, where at each time step the investor considers a simplified problem. Basically, there are two stages for

This chapter is based on the article 'Accurate and robust numerical methods for the dynamic portfolio management problem', published in *Computational Economics*, pages: 1–26, 2016 [25].

[1] For other types of utility functions, the investment horizon may have an impact on investment decisions, as discussed for example in [50].

solving this step-wise optimization problem. First, we determine how to formulate this optimization problem. We can either focus on the optimization problem directly or try to solve its corresponding first-order conditions, which usually depend on a preparatory approximation of the optimization problem. Then, the rest of the problem can be treated as a mathematical problem of computing conditional expectations. In [16], the authors work on the first-order conditions and compute the conditional expectations by simulation and cross-path regression. We call the algorithm the BGSS algorithm. In [85], the authors propose an alternative algorithm, the vBB algorithm, where they work on the optimization problem directly via grid-searching but still utilizing simulation and cross-path regression to compute the conditional expectations. They state that the vBB algorithm is more stable than the BGSS algorithm, since the BGSS algorithm essentially relies on an approximation of the utility function. Many other numerical approaches for computing conditional expectations have been considered, for example, in [2], [58] and [42].

We propose improvements for the BGSS and the vBB algorithms, which, respectively, rely on solving the first-order conditions and grid-searching to tackle the optimization problem.

In the original BGSS algorithm, cross-path standard regression is employed for solving first-order conditions, that correspond to the utility function via a Taylor series expansion. Within this framework, we particularly contribute in two aspects. First, we replace the standard regression method by the (local) regression combined with bundling of simulated paths, as employed in Chapter 2. According to our tests, this modification makes the algorithm more stable and robust, and therefore our algorithm performs highly satisfactorily compared to the BGSS and the vBB algorithms, particularly when the investment horizon is long and risk aversion is high. In the process of approximating the utility function, we consider an alternative Taylor expansion to the expansion employed in the original BGSS algorithm. This Taylor expansion was introduced in [42]. This expansion is however not directly compatible with regression-based approaches. With a specific choice of the Taylor expansion center, we can equip our SGBM regression-based portfolio algorithm with this improved Taylor expansion, making the approximations less biased. In short, our enhanced algorithm still constitutes an algorithm based on simulation and cross-path regression. It thus remains possible to extend this algorithm to high-dimensional scenarios without increasing the computational complexity dramatically.

Based on grid-searching, which is the basic idea of the vBB algorithm, we utilize a Fourier cosine series technique [37, 38] to compute the conditional expectations and come up with a benchmark algorithm, the COS portfolio management method. Because this method is not based on simulation, there is no such error present in the corresponding numerical results. In the test cases to follow, reference solutions can therefore be generated via this COS-based algorithm.

The chapter is organized as follows. Section 3.2 gives the mathematical formulation of the investor's problem. In Section 3.2.1, we introduce a special case of the investor's problem where the simulation- and regression-based methods can be applied. In Section 3.3 the SGBM algorithm is briefly described and the alternative Taylor expansion is also discussed. The benchmark algorithm based on the COS method is presented in Sec-

tion 3.4. Following that, we display results of the numerical tests in Section 3.5. A brief discussion of the errors of the simulation-based methods is performed in Section 3.5.6. We conclude in the last section.

3.2. PROBLEM FORMULATION: THE INVESTOR'S PROBLEM

We consider a portfolio consisting of one risk-free asset and d risky assets, which can be traded at M discrete time points, $t \in [0, \Delta t, \ldots, T - \Delta t]$, before terminal time T. The time step Δt is equal to T/M. At each trading time t, an investor decides her trading strategy to maximize the expected value of the utility of her terminal wealth W_T. Formally, the investor's problem is given by

$$V_t(W_t, \mathbf{Z}_t) = \max_{\{\mathbf{x}_s\}_{s=t}^{T-\Delta t}} \mathbb{E}[U(W_T)|W_t, \mathbf{Z}_t], \tag{3.1}$$

subject to the constraints:

$$W_{s+\Delta t} = W_s \cdot (\mathbf{x}_s' \mathbf{R}^e_{s+\Delta t} + R_f), \qquad s = t, t+\Delta t, \ldots, T-\Delta t.$$

Here $\mathbf{x}_s$ denotes the asset allocation of the investor's wealth in risky assets. Vector transposition is denoted by the prime sign. R_f is the return of the risk-free asset, which is assumed to be constant for simplicity, and $\mathbf{R}^e_{s+\Delta t} = [R^{e,1}_{s+\Delta t}, \ldots, R^{e,d}_{s+\Delta t}]$ are the excess returns of the risky assets at time $s+\Delta t$. The function $U(W_T)$ denotes the utility of the investor's terminal wealth. $V_t(W_t, \mathbf{Z}_t)$ is termed the *value function*, which measures the investor's investment opportunities at time t with wealth W_t and market state $\mathbf{Z}_t$. We assume that $\{\mathbf{Z}_t\}_{t=0}^T$ is an $\mathscr{F}_t$-adapted Markov process.

Mathematically, an investor decides her asset allocations $\{\mathbf{x}_s\}_{s=0}^{T-\Delta t}$ at all time steps to maximize $V_0(W_0, \mathbf{Z}_0)$ or, equivalently, $\mathbb{E}[U(W_0 \prod_{s=0}^{T-\Delta t}(\mathbf{x}_s' \mathbf{R}^e_{s+\Delta t} + R_f))|W_0, \mathbf{Z}_0]$.

3.2.1. NUMERICAL APPROACHES TO THE INVESTOR'S PROBLEM

From Equation (3.1), we see that at time t it is impossible for the investor to determine the optimal asset allocation $\mathbf{x}_t$ without knowing optimal asset allocations $\{\mathbf{x}_s\}_{s=t+\Delta t}^{T-\Delta t}$ at future time points. A multivariate optimization problem with respect to all asset allocations $\{\mathbf{x}_s\}_{s=t}^{T-\Delta t}$ may be considered, but due to the complexity of the dynamics of $\mathbf{Z}_t$ it is usually not feasible to solve this problem.

A special case, discussed in [2], [16] and [85], is when the investor has constant relative risk aversion (CRRA), her optimal asset allocation $\mathbf{x}_t$ is independent of her wealth W_t.

The CRRA utility function $U(W_T)$ reads:

$$U(W_T) = \frac{W_T^{1-\gamma}}{1-\gamma}, \quad \gamma \neq 1, \tag{3.2}$$

and

$$U(W_T) = \log(W_T), \quad \gamma = 1, \tag{3.3}$$

where Equation (3.2) is termed the power utility function and Equation (3.3) the log utility function. This utility function is homothetic in wealth, which means that with identical market state $\mathbf{Z}_t$, two investors, one with wealth W_t and the other with wealth 1, will have the same optimal investment strategy at subsequent time points.

With this utility function, the optimization problem with respect to the original value function, $V_t(W_t, \mathbf{Z}_t)$, which depends on two variables W_t and $\mathbf{Z}_t$, reduces to an optimization problem with respect to a simplified value function, $v_t(\mathbf{Z}_t)$:

$$\begin{aligned} v_t(\mathbf{Z}_t) &:= V_t(1, \mathbf{Z}_t) \\ &= \max_{\{x_s\}_{s=t}^{T-\Delta t}} \mathbb{E}[U(\prod_{s=t}^{T-\Delta t} (\mathbf{x}_s' \mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_t]. \end{aligned}$$

3

Value function $v_t(\mathbf{Z}_t)$ can be written as a recursive procedure:

$$\begin{aligned} v_t(\mathbf{Z}_t) &= \max_{\{x_s\}_{s=t}^{T-\Delta t}} \mathbb{E}[U(\prod_{s=t}^{T-\Delta t} (\mathbf{x}_s' \mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_t] \\ &= \max_{\{\mathbf{x}_s\}_{s=t}^{T-\Delta t}} \mathbb{E}[\mathbb{E}[U(\prod_{s=t}^{T-\Delta t} (\mathbf{x}_s' \mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_{t+\Delta t}]|\mathbf{Z}_t] \\ &= \max_{\mathbf{x}_t} \mathbb{E}\left[\max_{\{\mathbf{x}_s\}_{s=t+\Delta t}^{T-\Delta t}} \mathbb{E}[U(\prod_{s=t}^{T-\Delta t} (\mathbf{x}_s' \mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_{t+\Delta t}] \middle| \mathbf{Z}_t \right] \\ &= \max_{\mathbf{x}_t} \mathbb{E}[v_{t+\Delta t}((\mathbf{x}_t' \mathbf{R}_{t+\Delta t}^e + R_f), \mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]. \end{aligned} \tag{3.4}$$

Equation (3.4) is based on the Bellman principle of optimality and dynamic programming [5], which forms the basis for any recursive solution of the dynamic portfolio problem. The principle can be applied since the state vector is assumed to follow a Markov process and, therefore, the optimal asset allocation $\mathbf{x}_t$ only depends upon time and the current state $\mathbf{Z}_t$.

Using the simplified value function and the power utility function with parameter γ, we can solve the investor's problem, in a backward recursion process[2], as follows:

- At time T, we determine the value function as:

$$v_T(\mathbf{Z}_T) = \frac{1}{1-\gamma}, \gamma \neq 1;$$

- At time $T - \Delta t$, the investor considers the optimization problem:

$$\begin{aligned} v_{T-\Delta t}(\mathbf{Z}_{T-\Delta t}) &= \max_{\mathbf{x}_{T-\Delta t}} \mathbb{E}[U(\mathbf{x}_{T-\Delta t}' \mathbf{R}_T^e + R_f)|\mathbf{Z}_{T-\Delta t}] \\ &= \max_{\mathbf{x}_{T-\Delta t}} \mathbb{E}[(\mathbf{x}_{T-\Delta t}' \mathbf{R}_T^e + R_f)^{1-\gamma} v_T(\mathbf{Z}_T)|\mathbf{Z}_{T-\Delta t}]. \end{aligned}$$

  We denote the optimal asset allocation by $\mathbf{x}_{T-\Delta t}^*$, so that:

$$\max_{\mathbf{x}_{T-\Delta t}} \mathbb{E}[U(\mathbf{x}_{T-\Delta t}' \mathbf{R}_T^e + R_f)|\mathbf{Z}_{T-\Delta t}] := \mathbb{E}[U(\mathbf{x}_{T-\Delta t}^{*\prime} \mathbf{R}_T^e + R_f)|\mathbf{Z}_{T-\Delta t}].$$

  Recursively, moving backward in time, the following steps are subsequently performed at times t, $t = T - 2\Delta t, T - 3\Delta t, \ldots, \Delta t, 0$.

[2] Choosing the CRRA utility function is essential for this process to be valid.

- When the investor's optimal asset allocations, $\{\mathbf{x}_\mathbf{s}^*\}_{s=t+\Delta t}^{T-\Delta t}$, are determined, we can calculate the value function $v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})$ as:

$$v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t}) = \mathbb{E}[U(\prod_{s=t+\Delta t}^{T-\Delta t}(\mathbf{x}_s^{*\prime}\mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_{t+\Delta t}].$$

Then, the value function $v_t(\mathbf{Z}_t)$ reads

$$\begin{aligned} v_t(\mathbf{Z}_t) &= \max_{\{\mathbf{x}_s\}_{s=t}^{T-\Delta t}} \mathbb{E}[U((\mathbf{x}_t'\mathbf{R}_{t+\Delta t}^e + R_f)\prod_{s=t+\Delta t}^{T-\Delta t}(\mathbf{x}_s'\mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_t] \\ &= \max_{\mathbf{x}_t}\mathbb{E}\left[(\mathbf{x}_t'\mathbf{R}_{t+\Delta t}^e + R_f)^{1-\gamma}\max_{\{\mathbf{x}_s\}_{s=t+\Delta t}^{T-\Delta t}}\mathbb{E}[U(\prod_{s=t+\Delta t}^{T-\Delta t}(\mathbf{x}_s'\mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_{t+\Delta t}]\Big|\mathbf{Z}_t\right] \\ &= \max_{\mathbf{x}_t}\mathbb{E}[(\mathbf{x}_t'\mathbf{R}_{t+\Delta t}^e + R_f)^{1-\gamma}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t], \end{aligned} \tag{3.5}$$

where the last equality is valid by using the definition of $v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})$. Value function $v_t(\mathbf{Z}_t)$ can also be written as:

$$\begin{aligned} v_t(\mathbf{Z}_t) &= \max_{\mathbf{x}_t}\mathbb{E}[(\mathbf{x}_t'\mathbf{R}_{t+\Delta t}^e + R_f)^{1-\gamma}\mathbb{E}[U(\prod_{s=t+\Delta t}^{T-\Delta t}(\mathbf{x}_s^{*\prime}\mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_{t+\Delta t}]|\mathbf{Z}_t] \\ &= \max_{\mathbf{x}_t}\mathbb{E}[(\mathbf{x}_t'\mathbf{R}_{t+\Delta t}^e + R_f)^{1-\gamma}U(\prod_{s=t+\Delta t}^{T-\Delta t}(\mathbf{x}_s^{*\prime}\mathbf{R}_{s+\Delta t}^e + R_f))|\mathbf{Z}_t], \end{aligned} \tag{3.6}$$

where the last equality follows from the law of iterated expectations.

Either Equation (3.5) or Equation (3.6) can be employed to evolve the information in the backward recursion. They respectively correspond to the "*value function iteration*" and the "*portfolio weight iteration*", to be discussed in the following subsection. In either case, the optimization problem with respect to $\mathbf{x}_t$ can be solved via numerical techniques.

As mentioned before, there are basically two numerical approaches available for dealing with this problem, one is by grid-searching and the other is by solving the first-order conditions. These techniques are discussed in subsequent sections.

PORTFOLIO WEIGHT ITERATION OR VALUE FUNCTION ITERATION

In the backward recursion process, after either the optimal asset allocations $\{\mathbf{x}_s\}_{s=t+\Delta t}^{T-\Delta t}$ or $\mathbf{x}_{t+\Delta t}$ and $v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})$ have been determined, we need to evolve the information from time step $t+\Delta t$ to time step t to proceed the recursive computation. We can consider either Equation (3.5) or Equation (3.6) for this purpose. The former is termed "portfolio weight iteration" and the latter "value function iteration". In [85] the authors show that more stable results can be obtained by the portfolio weight iteration. They explain their results as follows. In the value function iteration, the value function is a conditional expectation approximated by cross-path regression and approximation errors may accumulate in the backward recursion process. In the portfolio weight iteration, since the portfolio weights are bounded by borrowing and short-sale constraints, the approximation error remains bounded throughout the whole valuation process.

However, if the value function at each intermediate time step can be approximated accurately, the value function iteration should yield similar results as the portfolio weight iteration. In the numerical tests to follow, we will see that our enhanced numerical methods perform highly satisfactory and, in most cases, using the value function iteration produces comparable results as the portfolio weight iteration.

3.3. SOLVING FIRST-ORDER CONDITIONS

When the value function $v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})$ is known, we consider the optimization problem displayed in Equation (3.5).

One approach to obtain the optimal asset allocation $\mathbf{x}_t$ in Equation (3.5) is to solve the first-order conditions for an optimum, i.e.

$$\mathbb{E}[\frac{\partial}{\partial \mathbf{x}_t}((\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t}+R_f)^{1-\gamma}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t}))|\mathbf{Z}_t]=0. \tag{3.7}$$

Since Equation (3.7) is not directly solvable with respect to $\mathbf{x}_t$, in [16] the authors proposed an approach to first approximate the value function $v_t(\mathbf{Z}_t)$ via a Taylor series expansion and then solve the first-order conditions corresponding to the approximated function. Second-order Taylor expansion of the value function is written as[3]:

$$\begin{aligned} v_t(\mathbf{Z}_t) \quad &\approx \quad \max_{\mathbf{x}_t}\Big\{\mathbb{E}[(R_f)^{1-\gamma}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]+\mathbb{E}[(1-\gamma)(R_f)^{-\gamma}\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t] \\ &+ \quad \mathbb{E}[\frac{1}{2}(1-\gamma)(-\gamma)(R_f)^{-1-\gamma}(\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t})^2 v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\Big\}. \end{aligned}$$

The corresponding first-order conditions read:

$$\mathbb{E}[(1-\gamma)(R_f)^{-\gamma}\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]+\mathbb{E}[(1-\gamma)(-\gamma)(R_f)^{-1-\gamma}(\mathbf{R}^e_{t+\Delta t}\mathbf{R}^{e\prime}_{t+\Delta t})v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\mathbf{x}_t=0,$$

and the optimal asset allocation $\mathbf{x}_t^*$, which is assumed to be $\mathbf{Z_t}$-measurable, is given by:

$$\mathbf{x}_t^*=[\mathbb{E}[\gamma\cdot(\mathbf{R}^e_{t+\Delta t}\mathbf{R}^{e\prime}_{t+\Delta t})v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]]^{-1}\cdot\mathbb{E}[R_f\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]. \tag{3.8}$$

Here the conditional expectations can be approximated via simulation and cross-path regression, as done in [16], [65] and [82].

It is mentioned in [16] that solving first-order conditions is quite sensitive to the order of the Taylor expansion of the value function and the results from second-order and fourth-order expansions can be different. If we consider the fourth-order Taylor expansion of the value function $v_t(\mathbf{Z}_t)$, i.e.

$$\begin{aligned} v_t(\mathbf{Z}_t) \quad &\approx \quad \max_{\mathbf{x}_t}\Big\{\mathbb{E}[(R_f)^{1-\gamma}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t] \\ &+ \quad \mathbb{E}[(1-\gamma)(R_f)^{-\gamma}\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t] \\ &+ \quad \mathbb{E}[\frac{1}{2}(1-\gamma)(-\gamma)(R_f)^{-1-\gamma}(\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t})^2 v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t] \\ &+ \quad \mathbb{E}[\frac{1}{6}(1-\gamma)(-\gamma)(-1-\gamma)(R_f)^{-2-\gamma}(\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t})^3 v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t] \\ &+ \quad \mathbb{E}[\frac{1}{24}(1-\gamma)(-\gamma)(-1-\gamma)(-2-\gamma)(R_f)^{-3-\gamma}(\mathbf{x}_t'\mathbf{R}^e_{t+\Delta t})^4 v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\Big\}, \end{aligned}$$

[3] We first consider the Taylor expansion as implemented in [16]. Another Taylor expansion will be introduced in Section 3.3.2.

the optimal asset allocation $\mathbf{x}_t^*$ is defined as an implicit solution of the following equation:

$$\begin{aligned}\mathbf{x}_t^* \approx\ & [\mathbb{E}[\gamma\cdot(\mathbf{R}^e_{t+\Delta t}\mathbf{R}^{e\prime}_{t+\Delta t})\,v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]]^{-1}\cdot\Big\{\mathbb{E}[R_f\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\\ &+\frac{1}{2}\mathbb{E}[\frac{(-\gamma)(-1-\gamma)}{R_f}(\mathbf{x}_t^{*\prime}\mathbf{R}^e_{t+\Delta t})^2\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\\ &+\frac{1}{6}\mathbb{E}[\frac{(-\gamma)(-1-\gamma)(-2-\gamma)}{(R_f)^2}(\mathbf{x}_t^{*\prime}\mathbf{R}^e_{t+\Delta t})^3\mathbf{R}^e_{t+\Delta t}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\Big\}. \qquad (3.9)\end{aligned}$$

This equation can be treated as a fixed point problem, $\mathbf{x} = RH(\mathbf{x})$ with $RH(\cdot)$ denoting the right-hand side in Equation (3.9). This can be solved by an iterative method. To start the iteration, we need an initial guess of the optimal asset allocation. Following the discussion in [16], we can take the solution from the second-order Taylor expansion of the value function as the initial guess $\mathbf{x}_t^0$.

The iteration can be conducted by Newton's method for $RH(\mathbf{x})-\mathbf{x}=0$:

$$\mathbf{x}_t^{l+1} = \mathbf{x}_t^l - \frac{RH(\mathbf{x}_t^l)-\mathbf{x}_t^l}{RH'(\mathbf{x}_t^l)-1}, \quad l=0,1,2,\ldots.$$

We stop the iteration, if either the 2-norm of the distance between two consecutive approximations $\mathbf{x}_t^l$ and $\mathbf{x}_t^{l+1}$ is smaller than a tolerance value ϵ_{TOL} or the number of iterations reaches a predetermined value $l_{\max}$. We take the last iteration $\mathbf{x}_t^{l+1}$ as the final solution of Equation (3.9). In the numerical tests, we choose $\epsilon_{\text{TOL}} = 0.0001$ and $l_{\max} = 30$. Always the tolerance ϵ_{TOL} can be reached, unless stated otherwise.

3.3.1. STOCHASTIC GRID BUNDLING METHOD

The Stochastic Grid Bundling Method (SGBM), introduced in [57], is a powerful regression-based method for calculating conditional expectations in Equations (3.8) and (3.9).

It is shown in [57] that applying SGBM is highly efficient for obtaining the early-exercise boundary when pricing American-style options and the estimated path-wise option value is so accurate that the Greeks can be generated directly. In this chapter, we implement SGBM for the dynamic utility-based portfolio management problem. Similar as [16], we take the second-order Taylor expansion in the description of the algorithm for expositional ease. However, in our numerical experiments, we always employ the fourth-order Taylor expansion. Extension to fourth-order expansion can be achieved with the formulas in Equation (3.9). Our algorithm can be formally described as follows:

Step I: *Simulation.*

Simulate N paths $[\mathbf{R}^e_t(i), \mathbf{Z}_t(i)]_{i=1}^N$, $t = 0, \Delta t, \ldots, T$, and set the value function at terminal time T as:

$$v_T(\mathbf{Z}_T(i)) = \frac{1}{1-\gamma}, \quad i = 1,\ldots,N.$$

The following steps are subsequently performed at times t, $t \le T-\Delta t$.

Step II: *Bundling.*

We bundle the paths at time t into B non-overlapping partitions, $\mathscr{B}_t(1),\ldots,\mathscr{B}_t(B)$. Let each bundle cover a similar number of paths.

Step III: *Regression.*
Assume that there are $N_B(b)$ paths in bundle $\mathscr{B}_t(b)$ and their value functions at time $t+\Delta t$ are $\{v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})\ (i)\}_{i=1}^{N_B(b)}$, or, equivalently, the optimal asset allocations read $\{\mathbf{x}_s^*(i)\}_{i=1}^{N_B(b)}$, $s=t+\Delta t,\ldots,N$, and their excess returns $\{\mathbf{R}_{t+\Delta t}^e(i)\}_{i=1}^{N_B(b)}$ are known. For these paths, we determine bundle-wise regression parameters $\{\alpha_k(b)\}_{k=1}^K$ by regressing the values $\{\gamma\cdot(\mathbf{R}_{t+\Delta t}^e(i)\mathbf{R}_{t+\Delta t}^{e\prime}(i))v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})(i)\}_{i=1}^{N_B(b)}$ on basis functions $[\phi_1(\mathbf{R}_{t+\Delta t}^e(i),\mathbf{Z}_{t+\Delta t}(i)),\ldots,\phi_K(\mathbf{R}_{t+\Delta t}^e(i),\mathbf{Z}_{t+\Delta t}(i))]_{i=1}^{N_B(b)}$, which are constructed using the information at time $t+\Delta t$. In this chapter, we always choose basic polynomials for the basis functions. Following our discussion in the previous chapter, this choice of basis functions should be sufficient in the regression step when the approximated function is continuous.

For any path whose state $\mathbf{Z}_t$ is covered by bundle $\mathscr{B}_t(b)$, the denominator of the right-hand side part in Equation (3.8), $\mathbb{E}[\gamma\cdot(\mathbf{R}_{t+\Delta t}^e\mathbf{R}_{t+\Delta t}^{e\prime})\ v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]$, can be approximated by:

$$\mathbb{E}[\gamma\cdot(\mathbf{R}_{t+\Delta t}^e\mathbf{R}_{t+\Delta t}^{e\prime})\,v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\approx\sum_{k=1}^K\alpha_k(b)\mathbb{E}[\phi_k(\mathbf{R}_{t+\Delta t}^e,\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t].$$

Similarly, $\mathbb{E}[R_f\mathbf{R}_{t+\Delta t}^e v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]$, the numerator of the right-hand side part in Equation (3.8), can be approximated by:

$$\mathbb{E}[R_f\mathbf{R}_{t+\Delta t}^e v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\approx\sum_{k=1}^K\beta_k(b)\mathbb{E}[\phi_k(\mathbf{R}_{t+\Delta t}^e,\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t],$$

where the regression parameters $\{\beta_k(b)\}_{k=1}^K$ are obtained by regressing $\{R_f\mathbf{R}_{t+\Delta t}^e(i)\ v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t}(i))\}_{i=1}^{N_B(b)}$ on the basis functions $[\phi_1(\mathbf{R}_{t+\Delta t}^e(i),\mathbf{Z}_{t+\Delta t}(i)),\ldots,\phi_K(\mathbf{R}_{t+\Delta t}^e(i),\mathbf{Z}_{t+\Delta t}(i))]_{i=1}^{N_B(b)}$.

For any path whose state $\mathbf{Z}_t$ is covered by bundle $\mathscr{B}_t(b)$, the optimal asset allocation is approximated by:

$$\mathbf{x}_t^*\approx\left[\sum_{k=1}^K\alpha_k(b)\mathbb{E}[\phi_k(\mathbf{R}_{t+\Delta t}^e,\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\right]^{-1}\cdot\left[\sum_{k=1}^K\beta_k(b)\mathbb{E}[\phi_k(\mathbf{R}_{t+\Delta t}^e,\mathbf{Z}_{t+\Delta t})|\mathbf{Z}_t]\right].$$

The regression step is repeated for all bundles at each time step, so for each path we find the corresponding optimal asset allocation.

Step IV: *Transition.*
For the i-th path in bundle $\mathscr{B}_t(b)$, we can either apply portfolio weight iteration or value function iteration to transfer the information of the optimal investment strategy from time t to time $t-\Delta t$.

When using the portfolio weight iteration, we just store the optimal asset allocations $\{\mathbf{x}_s\}_{s=t}^{T-\Delta t}$ and write $v_t(\mathbf{Z}_t)$ as $\prod_{s=t}^{T-\Delta t}(\mathbf{x}_s'\mathbf{R}_{s+\Delta t}^e+R_f)^{1-\gamma}/(1-\gamma)$ in the regression step at time $t-\Delta t$.

If we use the value function iteration, the process is slightly more involved. For all paths in bundle $\mathscr{B}_t(b)$, we regress $\{(\mathbf{x}_t'(i)\mathbf{R}_{t+\Delta t}^e(i)+R_f)^{1-\gamma}v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t}(i))\}_{i=1}^{N_B(b)}$ on the following polynomial basis functions $[\hat{\phi}_1(\mathbf{Z}_t(i)),\ldots,\hat{\phi}_K(\mathbf{Z}_t(i))]_{i=1}^{N_B(b)}$ formed by $\{\mathbf{Z}_t(i)\}_{i=1}^{N_B(b)}$ and obtain regression parameters $[\xi_1(b),\ldots,\xi_K(b)]$. The value function is then approximated by:

$$v_t(\mathbf{Z}_t)\approx\sum_{k=1}^K\xi_k(b)\hat{\phi}_k(\mathbf{Z}_t).$$

It should be noted that the standard regression method is implemented in this step. According to our tests, introducing SGBM in this step is not helpful.

3.3.2. TAYLOR EXPANSION BASED ON A NONLINEAR DECOMPOSITION

In both, the BGSS and SGBM[4] algorithms, an essential step before solving the equations for the first-order conditions is to rewrite the value function, $v_t(\mathbf{Z}_t)$, in a Taylor series expansion in which the asset allocation $\mathbf{x}_t$ is separated from the conditional expectations of $\mathbf{R}^e_{t+\Delta t}$. A Pth-order Taylor expansion in SGBM can be written as[5]:

$$(x_t R^e_{t+\Delta t} + R_f)^{1-\gamma} \approx \sum_{p=0}^{P} \frac{g_1^{(p)}(0)}{p!} (x_t R^e_{t+\Delta t})^p, \tag{3.10}$$

where $g_1^{(p)}(0)$ denotes the pth derivative of function $g_1(y)$ when $y = 0$. Function $g_1(y) = (y + R_f)^{1-\gamma}$. So, $g_1(x_t R^e_{t+\Delta t}) = (x_t R^e_{t+\Delta t} + R_f)^{1-\gamma}$.

Since the excess return R^e_t is a nonlinear transformation of the log excess return $\mathbf{r}^e_t$, i.e.

$$R^e_{t+\Delta t} = \exp(r^e_{t+\Delta t}) R_f - R_f,$$

an alternative way to perform a Taylor expansion for $(x_t R^e_{t+\Delta t} + R_f)^{1-\gamma}$ is given by:

$$(x_t R^e_{t+\Delta t} + R_f)^{1-\gamma} \approx \sum_{p=0}^{P} \frac{g_2^{(p)}(0)}{p!} (r^e_{t+\Delta t})^p, \tag{3.11}$$

where function $g_2(z)$ is defined by:

$$g_2(z) = (R_f + x_t(\exp(z) R_f - R_f))^{1-\gamma}.$$

Functions $g_1(x_t R^e_{t+\Delta t})$ and $g_2(r^e_t)$ are both identical to $(x_t R^e_{t+\Delta t} + R_f)^{1-\gamma}$, but different ways of choosing the underlying variable yield different Taylor expansion formulas.

In [43], the authors term the expansion described in Equation (3.10) as "Taylor expansion based on a linear decomposition" and the expansion described in Equation (3.11) as "Taylor expansion based on a nonlinear decomposition". They show that when the centers of Taylor expansions are carefully chosen, the "Taylor expansion based on a nonlinear decomposition" is more accurate than the "Taylor expansion based on a linear decomposition" when approximating the function $(x_t R^e_{t+\Delta t} + R_f)^{1-\gamma}$. We will call these expansions "original Taylor" and "log Taylor" expansions, respectively, in the rest of this chapter. Although the log Taylor expansion has been implemented in [42] for dynamic portfolio management, their choice of expansion center is not compatible with the algorithm discussed here. We deal with this problem by performing a log Taylor expansion around center 0, as displayed in Equation (3.11). We especially choose the Taylor expansion center to be 0, because only in this case is there no term related to x_t inside the

[4] With a little abuse of the term, we also denote our dynamic portfolio management method by "SGBM".
[5] For simplicity, we describe the dynamic portfolio management problem with one risky asset here.

power transformation, for example:

$$\begin{aligned} g_2(0) &= (R_f)^{1-\gamma}, \\ g_2'(0) &= (1-\gamma)(R_f)^{1-\gamma} x_t, \\ g_2''(0) &= (1-\gamma)(R_f)^{1-\gamma}(-\gamma x_t^2 + x_t). \end{aligned}$$

According to the numerical tests in Section 3.5, we find that the log Taylor expansion is indeed a superior choice even when the expansion center is chosen to be 0. The reasoning is that the log excess return r_t^e usually exhibits a distribution similar to the normal distribution. Therefore, a Taylor expansion with respect to this variable, i.e. the so-called "log Taylor" expansion, can yield accurate results with a limited number of expansion terms. The distribution of the excess return R_t^e usually exhibits a fat tailed distribution, which requires more terms in the original Taylor expansion to approximate its moments.

3.4. GRID-SEARCHING METHODS

An alternative technique to solving first-order conditions is based on grid-searching, which is an intuitive idea for solving the optimization problem described in Equation (3.5). In grid-searching, we reduce the optimization problem on the continuous domain to a problem on a discrete domain. For example, if we consider the allocation, x_t, of one risky asset, the original optimization problem is solved on a domain $[0,1]$. By grid-searching, we construct K_M equidistant grid points $\{\frac{k_m}{K_M}\}_{k_m=0}^{K_M}$ and consider the optimization problem on the discrete domain $\text{DM} = \{\frac{k_m}{K_M} |\ k_m = 0, 1, \ldots, K_M\}$. To solve this discrete optimization problem, we test each possible choice of the allocation $x_t^{(k_m)} = \frac{k_m}{K_M}, k_m = 0, \ldots, K_M$ and calculate the corresponding value functions:

$$v_t^{(k_m)}(\mathbf{Z}_t) = \mathbb{E}[(x_t^{(k_m)} R_{t+\Delta t}^e + R_f)^{1-\gamma} v_{t+\Delta t}(\mathbf{Z}_{t+\Delta t}) | \mathbf{Z}_t]. \tag{3.12}$$

We determine the maximum, $v_t^{\max}(\mathbf{Z}_t)$, from $\{v_t^{(k_m)}(\mathbf{Z}_t)\}_{k_m=0}^{K_M}$ and denote its corresponding asset allocation as "the optimal asset allocation".

Although it is mentioned in [84, 85] that the grid-searching method is robust and avoids a number of numerical issues regarding convergence that occur when solving first-order conditions, it should be noted that grid-searching is an expensive numerical approach. The workload of grid-searching grows exponentially as the dimensionality of the problem increases. Moreover, according to our numerical tests in the low-dimensional cases, the vBB algorithm, which employs grid-searching together with simulation, yields more "uncertain" results (larger variance) compared to the other simulation-based algorithms.

However, if we wish to find an accurate reference solution to the dynamic portfolio management problem, grid-searching seems our only choice since solving first-order conditions essentially relies on Taylor approximations of the utility function, whereas grid-searching does not. In the next subsection we will present our benchmark approach based on the idea of grid-searching.

3.4.1. COS PORTFOLIO MANAGEMENT METHOD

In this section, we present a benchmark method, based on the Fourier cosine series expansion (COS) method to calculate the conditional expectations. This method was introduced in [37] for pricing one-dimensional European options and later in [38] for pricing one-dimensional Bermudan and barrier options. In [74], this method was extended to the two-dimensional case. Because the COS method is not based on simulation, it can yield benchmark solutions to the investor's problem, especially in the basic case with one risky asset and one risk-free asset. Following the previous discussions, this basic investor's problem with power utility function is given by:

$$v_{t-\Delta t}(Z_{t-\Delta t}) = \max_{x_{t-\Delta t}} \mathbb{E}[(x_{t-\Delta t}R_t^e + R_f)^{1-\gamma} v_t(Z_t)|Z_{t-\Delta t}] \quad t = 1,\dots,T-\Delta t, \tag{3.13}$$

where the terminal condition reads:

$$v_T(Z_T) = \frac{1}{1-\gamma}, \gamma \neq 1.$$

If we denote the conditional transition density function from state $Z_{t-\Delta t}$ to (R_t^e, Z_t) as $f(R_t^e, Z_t|Z_{t-\Delta t})$, the investor's problem reads:

$$v_{t-\Delta t}(Z_{t-\Delta t}) = \max_{x_{t-\Delta t}} \iint_{\mathbb{R}^2} (x_{t-\Delta t}R_t^e + R_f)^{1-\gamma} v_t(Z_t) f(R_t^e, Z_t|Z_{t-\Delta t}) \mathrm{d}R_t^e \mathrm{d}Z_t, \quad t = 1,\dots,T-\Delta t. \tag{3.14}$$

The COS algorithm for calculating conditional expectations can be described in five steps:

Step I: *Truncate the integration range in Equation (3.14).*

If we assume that the integrand is integrable, we can truncate the integration range from $\mathbb{R}^2$ to $[a_R, b_R] \times [a_Z, b_Z]$ without losing significant accuracy. The approximated value function $\hat{v}_{t-\Delta t}(Z_{t-\Delta t})$ reads:

$$\hat{v}_{t-\Delta t}(Z_{t-\Delta t}) \quad = \quad \max_{x_{t-\Delta t}} \int_{a_Z}^{b_Z} \int_{a_R}^{b_R} (x_{t-\Delta t}R_t^e + R_f)^{1-\gamma} v_t(Z_t) f(R_t^e, Z_t|Z_{t-\Delta t}) \mathrm{d}R_t^e \mathrm{d}Z_t.$$

Remark 3.4.1. *For one variable, for example Z_t, the suggested integration range $[a_Z, b_Z]$ in [38] and [74] is $[\xi_1^Z - L\xi_2^Z, \xi_1^Z + L\xi_2^Z]$, where ξ_1^Z is the mean of Z_t and ξ_2^Z the standard deviation of Z_t. L should be large enough to make the truncation error acceptably low.*

Step II: *Expand the integrand in Fourier cosines.*

If we denote the Fourier cosine expansion of $f(R_t^e, Z_t|Z_{t-\Delta t})$ on $[a_R, b_R] \times [a_Z, b_Z]$ by:

$$\begin{aligned} A_{k_1,k_2}(Z_{t-\Delta t}) \quad := \quad & \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \int_{a_Z}^{b_Z} \int_{a_R}^{b_R} f(R_t^e, Z_t|Z_{t-\Delta t}) \\ & \cos\left(k_1\pi \frac{R_t^e - a_R}{b_R - a_R}\right) \cos\left(k_2\pi \frac{Z_t - a_Z}{b_Z - a_Z}\right) \mathrm{d}R_t^e \mathrm{d}Z_t, \end{aligned}$$

and similarly define the utility coefficients as:

$$\begin{aligned} \mathcal{V}_{k_1,k_2}(t, x_{t-\Delta t}) \quad := \quad & \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \int_{a_Z}^{b_Z} \int_{a_R}^{b_R} (x_{t-\Delta t}R_t^e + R_f)^{1-\gamma} v_t(Z_t) \\ & \cos\left(k_1\pi \frac{R_t^e - a_R}{b_R - a_R}\right) \cos\left(k_2\pi \frac{Z_t - a_Z}{b_Z - a_Z}\right) \mathrm{d}R_t^e \mathrm{d}Z_t, \end{aligned}$$

value function $v_{t-\Delta t}(Z_{t-\Delta t})$ can be approximated by:

$$\hat{v}_{t-\Delta t}(Z_{t-\Delta t}) = \max_{x_{t-\Delta t}} \left\{ \frac{b_Z - a_Z}{2} \frac{b_R - a_R}{2} {\sum_{k_1=0}^{\infty}}' {\sum_{k_2=0}^{\infty}}' A_{k_1,k_2}(Z_{t-\Delta t}) \mathcal{V}_{k_1,k_2}(t, x_{t-\Delta t}) \right\}.$$

The primed sum $\sum'$ means that the first term of the summation has half weight.
Step III: *Truncate the infinite series.*
We truncate the infinite series, as follows:

$$\bar{v}_{t-\Delta t}(Z_{t-\Delta t}) = \max_{x_{t-\Delta t}} \left\{ \frac{b_Z - a_Z}{2} \frac{b_R - a_R}{2} {\sum_{k_1=0}^{N_1-1}}' {\sum_{k_2=0}^{N_2-1}}' A_{k_1,k_2}(Z_{t-\Delta t}) \mathcal{V}_{k_1,k_2}(t, x_{t-\Delta t}) \right\}.$$

Step IV: *Calculate the coefficients* $A_{k_1,k_2}(Z_{t-\Delta t})$.
The coefficients $A_{k_1,k_2}(Z_{t-\Delta t})$ can be approximated by $F_{k_1,k_2}(Z_{t-\Delta t})$, as follows:

$$\begin{aligned} F_{k_1,k_2}(Z_{t-\Delta t}) \quad := \quad & \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \iint_{\mathbb{R}^2} f(R_t^e, Z_t | Z_{t-\Delta t}) \\ & \cos\left(k_1 \pi \frac{R_t^e - a_R}{b_R - a_R}\right) \cos\left(k_2 \pi \frac{Z_t - a_Z}{b_Z - a_Z}\right) \mathrm{d}R_t^e \mathrm{d}Z_t. \end{aligned}$$

Using the following property of cosines: $2\cos(\alpha)\cos(\beta) = \cos(\alpha+\beta) + \cos(\alpha-\beta)$, we can calculate $F_{k_1,k_2}(Z_{t-\Delta t})$ by:

$$F_{k_1,k_2}(Z_{t-\Delta t}) = \frac{F^+_{k_1,k_2}(Z_{t-\Delta t}) + F^-_{k_1,k_2}(Z_{t-\Delta t})}{2},$$

where

$$\begin{aligned} & F^{\pm}_{k_1,k_2}(Z_{t-\Delta t}) \\ &= \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \iint_{\mathbb{R}^2} f(R_t^e, Z_t | Z_{t-\Delta t}) \cos\left(k_1 \pi \frac{R_t^e - a_R}{b_R - a_R} \pm k_2 \pi \frac{Z_t - a_Z}{b_Z - a_Z}\right) \mathrm{d}R_t^e \mathrm{d}Z_t \\ &= \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \Re\left(\iint_{\mathbb{R}^2} f(R_t^e, Z_t | Z_{t-\Delta t}) \exp\left(i k_1 \pi \frac{R_t^e}{b_R - a_R} \pm i k_2 \pi \frac{Z_t}{b_Z - a_Z}\right) \mathrm{d}R_t^e \mathrm{d}Z_t \right. \\ & \left. \cdot \exp\left(-i k_1 \pi \frac{a_R}{b_R - a_R} \mp i k_2 \pi \frac{a_Z}{b_Z - a_Z}\right)\right) \\ &= \frac{2}{b_Z - a_Z} \frac{2}{b_R - a_R} \Re\left(\psi\left(\frac{k_1 \pi}{b_R - a_R}, \pm \frac{k_2 \pi}{b_Z - a_Z} \middle| Z_{t-\Delta t}\right) \cdot \exp\left(-i k_1 \pi \frac{a_R}{b_R - a_R} \mp i k_2 \pi \frac{a_Z}{b_Z - a_Z}\right)\right). \end{aligned}$$

$\Re(\cdot)$ means taking the real part of the input data. $\psi(u_r, u_Z | Z_{t-\Delta t})$ is the *bivariate conditional characteristic function* of (R_t^e, Z_t) given state $Z_{t-\Delta t}$:

$$\psi(u_R, u_Z | Z_{t-\Delta t}) = \iint_{\mathbb{R}^2} \exp(i[u_R, u_Z] \cdot [R_t^e, Z_t]') f(R_t^e, Z_t | Z_{t-\Delta t}) \mathrm{d}R_t^e \mathrm{d}Z_t.$$

For many asset dynamics models this bivariate characteristic function is known in closed form.
Step V: *Calculate the coefficients* $\mathcal{V}_{k_1,k_2}(t, x_{t-\Delta t})$.

The coefficients $\mathcal{V}_{k_1,k_2}(t,x_{t-\Delta t})$ are not directly related to any closed-form expression. However, we can apply numerical integration and the discrete cosine transform (DCT) to approximate $\mathcal{V}_{k_1,k_2}(t,x_{t-\Delta t})$. To do this, we take $Q \geq \max[N_1,N_2]$ grid points in each spatial dimension and define:

$$\begin{aligned} R_t^{n_1} &:= a_R + (n_1+\frac{1}{2})\Delta R_t \quad n_1 = 1,\ldots,Q \\ Z_t^{n_2} &:= a_Z + (n_2+\frac{1}{2})\Delta Z_t \quad n_2 = 1,\ldots,Q \\ \Delta R_t &:= \frac{b_R-a_R}{Q},\ \Delta Z_t := \frac{b_Z-a_Z}{Q}. \end{aligned}$$

The midpoint-rule integration gives us

$$\begin{aligned} \mathcal{V}_{k_1,k_2}(t,x_{t-\Delta t}) &\approx \sum_{n_1=0}^{Q-1}\sum_{n_2=0}^{Q-1}\frac{2}{b_R-a_R}\frac{2}{b_Z-a_Z}(x_{t-\Delta t}R_t^{n_1}+R_f)^{1-\gamma}v_t(Z_t) \\ &\quad \cos\left(k_1\pi\frac{R_t^{n_1}-a_R}{b_R-a_R}\right)\cos\left(k_2\pi\frac{Z_t^{n_2}-a_Z}{b_Z-a_Z}\right)\Delta R_t\Delta Z_t \\ &= \sum_{n_1=0}^{Q-1}\sum_{n_2=0}^{Q-1}\frac{2}{Q}\frac{2}{Q}(x_{t-\Delta t}R_t^{n_1}+R_f)^{1-\gamma}v_t(Z_t^{n_2})\cos\left(k_1\pi\frac{R_t^{n_1}-a_R}{b_R-a_R}\right)\cos\left(k_2\pi\frac{Z_t^{n_2}-a_Z}{b_Z-a_Z}\right). \end{aligned}$$

The equation above can be calculated efficiently via a two-dimensional DCT, for example, with the function *dct2* of MATLAB. Moreover, we can rewrite the sum of multiplications into a multiplication of sums, that is:

$$\begin{aligned} \mathcal{V}_{k_1,k_2}(t,x_{t-\Delta t}) &\approx \frac{2}{Q}\frac{2}{Q}\left(\sum_{n_1=0}^{Q-1}(x_{t-\Delta t}R_t^{n_1}+R_f)^{1-\gamma}\cos\left(k_1\pi\frac{R_t^{n_1}-a_R}{b_R-a_R}\right)\right) \\ &\quad \cdot\left(\sum_{n_2=0}^{Q-1}v_t(Z_t^{n_2})\cos\left(k_2\pi\frac{Z_t^{n_2}-a_Z}{b_Z-a_Z}\right)\right). \end{aligned}$$

Then, the two-dimensional DCT can be replaced by two separate one-dimensional DCTs, which helps reducing the computational time.

For state $Z_{t-\Delta t}$ and asset allocation $x_{t-\Delta t}$, we can calculate the conditional expectation shown in Equation (3.13) by the COS method. To solve the optimization problem with respect to $x_{t-\Delta t}$, we employ grid-searching: we evaluate discretized values of $x_{t-\Delta t} \in \{\frac{k_m}{K_M} | k_m = 0,\ldots,K_M\}$ and find the largest conditional expectation. The backward recursion process can be performed from time $T-\Delta t$ to the initial time.

Within the COS method, we have five parameters to adjust the truncation and discretization errors. These are N_1, N_2, L, Q and K_M. Generally, larger values of these parameters lead to more accurate approximations but also to higher computational load. We use the following default parameter setting:

$$N_1 = 50, N_2 = 100, L = 8, Q = 100, K_M = 200. \tag{3.15}$$

According to our experiments, the COS method provides highly accurate results under this setting. However, when the admissible asset allocation can be chosen from a very

wide range of values, the COS approach, which is based on discrete grid search, may lose its accuracy. In that case, the SGBM method equipped with the log Taylor expansion and a large number of paths will still generate satisfactory solutions and appears favorable.

Remark 3.4.2. *The COS method suffers from the curse of dimensionality. However, this is a problem for any method involving discretization of the state space and grid-searching. In high-dimensional cases, dimension reduction, adaptive discretization, or sparse grids, and grid-searching may be applied.*

Remark 3.4.3. *The computational load of the COS method for a dynamic portfolio management problem is mainly related to the DCT computations, for which the computational complexity at each time step is* $O(N_2 \cdot K_M \cdot Q \cdot \log(Q))$. *Computations at each time step are performed sequentially, but the computations for the value function at each state point are independent, so it should be possible to accelerate the COS method by parallel processing.*

3.5. NUMERICAL EXPERIMENTS

In this section, we test the performance of five methods for generating the optimal dynamic portfolio management strategy. These are:

- "BGSS": the method introduced in [16];
- "vBB": the method introduced in [85];
- "SGBM": SGBM with the original Taylor expansion;
- "SGBM-LT": SGBM with the log Taylor expansion;
- "COS": the COS method.

We impose borrowing and short-sale constraints on the asset allocations, that are therefore restricted between 0 and 1. When we implement the simulation-based algorithms, *we always generate* 2^{14} *paths*. For "SGBM" and "SGBM-LT", which require bundling, *we employ 32 bundles at each time step*. We approximate the utility function by Taylor expansions, up to 4th-order for both the log Taylor expansion and the original Taylor expansion. For "BGSS" and "vBB", we use polynomials of the state variable *up to second-order as the basis functions for the cross-path regression*. For "SGBM" and "SGBM-LT", the polynomials are also second-order but are of both the state variable and the return variable.

To measure the performance of a dynamic portfolio management strategy, we consider the statistic, "annualized certainty equivalent rate", CER. It describes the annualized return rate of a risk-free asset which at terminal time Y (years) yields the same utility of wealth obtained from the dynamic portfolio management strategy. Equivalently, the CER is the risk-free rate that an investor is willing to accept rather than adopting a particular risky portfolio management strategy. Formally the CER is defined by:

$$U(W_0 \cdot (1 + \text{CER})^Y)) = V_0(W_0, \mathbf{Z}_0), \tag{3.16}$$

where the value function $V_0(W_0, \mathbf{Z}_0)$ is defined by Equation (3.1). Generally, a portfolio management strategy with high CER is close to the optimal strategy and can thus be regarded as an accurate solution to the dynamic portfolio management problem.

We perform numerical tests here for a basic test case where the portfolio contains one risky asset and one risk-free asset. We consider the vector auto-regression (VAR) model to describe the dynamics of the log excess return r^e_t of the risky asset and its log dividend yield d_t, that are chosen as the state variables. Quarterly data are generated with the following process, as in [16], [85] and [42]:

$$\begin{bmatrix} r^e_{t+\Delta t} \\ d_{t+\Delta t} \end{bmatrix} = \begin{bmatrix} 0.227 \\ -0.155 \end{bmatrix} + \begin{bmatrix} 0.060 \\ 0.958 \end{bmatrix} d_t + \begin{bmatrix} \epsilon^r_{t+\Delta t} \\ \epsilon^d_{t+\Delta t} \end{bmatrix},$$

where

$$\begin{bmatrix} \epsilon^r_{t+\Delta t} \\ \epsilon^d_{t+\Delta t} \end{bmatrix} \sim N(\mu_\epsilon, \Sigma_\epsilon),\ \mu_\epsilon = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } \Sigma_\epsilon = \begin{bmatrix} 0.0060 & -0.0051 \\ -0.0051 & 0.0049 \end{bmatrix}.$$

In most of the tests, the initial state, d_0, is chosen as the unconditional mean, i.e., $d_0 = -0.155/(1-0.958) = -3.6905$. Only in Section 3.5.4 we will consider three quantiles, the 25%, 50% and 75% quantiles, of the unconditional distribution of state variable respectively as the initial state. The gross return of the risk-free asset is chosen as $R_f = 1.06^{0.25}$ and the excess return R^e_t of the risky asset is $R^e_t = R_f(\exp(r^e_t) - 1)$.

Associated to the 1D-VAR model, the characteristic function, which is essential for the COS portfolio management method, can be formulated as:

$$\begin{aligned} \psi(u_r, u_Z | Z_{t-\Delta t}) \quad = \quad & \exp\left(i \cdot u_r \cdot (0.227 + 0.060 \cdot Z_{t-\Delta t}) + i \cdot u_Z \cdot (-0.155 + 0.958 \cdot Z_{t-\Delta t})\right) \cdot \\ & \exp\left(i\mu_\epsilon{}'[u_r, u_Z]' - \frac{1}{2}[u_r, u_Z]\Sigma_\epsilon[u_r, u_Z]'\right). \end{aligned}$$

Since R^e_t is an injective function of r^e_t, for all equations in Section 3.4 replacing the functions of R^e_t by functions of r^e_t is valid with trivial modification.

In the following tests, the unit of the investment horizon T is the quarter. We set M always equal to T, and therefore the re-balancing time step Δt is always equal to 1.

3.5.1. QUALITY OF THE COS PORTFOLIO MANAGEMENT METHOD

We first check the validity and quality of the COS portfolio management method. For the dynamic portfolio management problem with the 1D-VAR model, we calculate the optimal asset allocations and the corresponding annualized certainty equivalent rates and compare them with the reference values from [42]. As we can see in Table 3.1, in case of different investment horizons and risk aversions, the COS method always provides accurate approximations of the annualized certainty equivalent rates and also highly satisfactory approximations of the optimal initial asset allocations.

As the COS method with the parameter settings in (3.15) and the reference method involve some approximation errors, it is difficult to say whose optimal initial asset allocation is superior. However, since it is known that first-order deviations in the portfolio policy have only second-order welfare effects [23] and the COS method and the reference method yield similar annualized certainty equivalent rates, we consider these as the optimal solutions when comparing with simulation-based methods.

Remark 3.5.1. *We have also tested the performance of the COS portfolio management method with different initial states d_0. For any initial state tested, it generates very similar results as the reference values in [42].*

Remark 3.5.2. *According to Table 3.1, although the COS method may generate initial asset allocations that are different from the reference values, the annualized certainty equivalent rates are always very satisfactory. This implies that it is possible to generate good portfolio management strategies even based on a very coarse control grid discretization.*

Table 3.1: Initial optimal asset allocations and the corresponding annualized certainty equivalent rates of the COS portfolio management method, based on the 1D-VAR model, with reference values from [42].

	Optimal initial asset allocation(%)				Annualized CER(%)			
	$\gamma = 5$		$\gamma = 15$		$\gamma = 5$		$\gamma = 15$	
	COS	Reference	COS	Reference	COS	Reference	COS	Reference
$T = 10$	41.5	42.8	15.0	15.6	7.23	7.22	6.43	6.43
$T = 20$	56.5	56.3	23.0	25.3	7.84	7.84	6.72	6.72
$T = 30$	69.0	66.9	33.0	35.4	8.26	8.26	7.01	7.01
$T = 40$	77.5	76.8	43.0	44.5	8.53	8.53	7.27	7.26

3.5.2. PORTFOLIO MANAGEMENT WITH THE BUY-AND-HOLD STRATEGY

In this section, instead of the dynamic portfolio management problem, in which an investor decides her optimal asset allocations at intermediate times $t = 0, \Delta t, \ldots, T - \Delta t$, we consider a case where the investor decides her optimal asset allocation at time $t = 0$ and holds a fixed amount of assets until terminal time $t = T$. The corresponding value function reads

$$v_0(Z_0) = \max_{x_0} \mathbb{E}[\frac{1}{1-\gamma}(x_0 R^e_{0\to T} + R_{f,0\to T})^{1-\gamma} | Z_0],$$

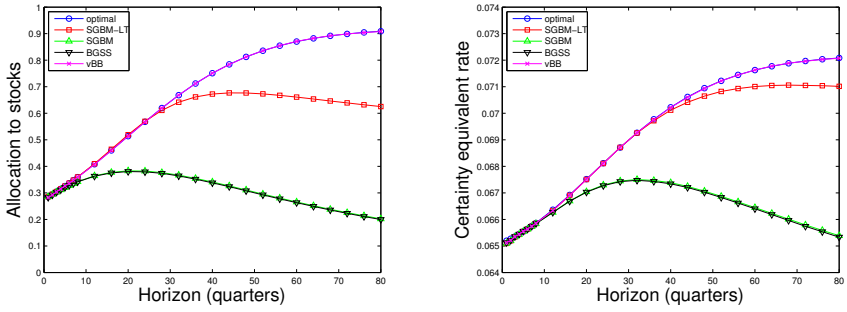
where $R_{f,0\to T} = (R_f)^T, R^e_{0\to T} = R_{f,0\to T} \cdot e^{\sum_{t=\Delta t}^{T} r^e_t} - R_{f,0\to T}$.

This type of problem can be viewed as a static portfolio management problem, for which the aforementioned four simulation-based methods ("SGBM-LT", "SGBM", "BGSS" and "vBB") can be applied. The COS method is utilized to generate benchmarks for the optimal asset allocations and the corresponding annualized certainty equivalent rates.

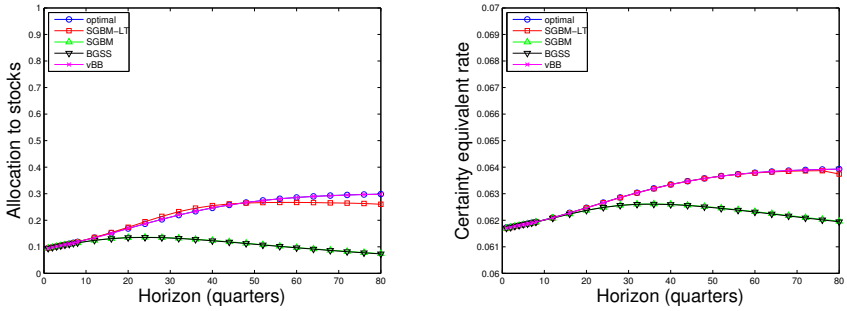
Figure 3.1 shows that "vBB" provides identical results to the optimal ones, since it does not involve Taylor expansion errors. For the other three candidates, in which Taylor expansions are involved, "SGBM-LT" provides the best approximation of the initial asset allocations. When the investment horizon is long, although the asset allocations of "SGBM-LT" are not close to the optimal solutions, their corresponding certainty equivalent rates are similar to the optimal ones. For the other two methods, "SGBM" and "BGSS", the estimates of asset allocations and certainty equivalent rates are acceptable only when the investment horizon is shorter than 10 quarters.

This test indicates that the log Taylor expansion (2^{14} paths, 32 bundles) outperforms the original Taylor expansion for approximating the utility functions. The advantage of

using the log Taylor expansion is obvious when the distribution of the accumulated excess return, $R^e_{0\to T}$, exhibits a fat tail.



(a) optimal allocation $\gamma = 5$ (b) certainty equivalent rate $\gamma = 5$



(c) optimal allocation $\gamma = 15$ (d) certainty equivalent rate $\gamma = 15$

Figure 3.1: For the simulation-based methods, we report the point estimate of the initial asset allocations from 100 runs. The optimal values are generated with the COS method.

3.5.3. DYNAMIC PORTFOLIO MANAGEMENT WITH DIFFERENT INVESTMENT HORIZONS AND RISK AVERSION PARAMETERS

Following the discussion in [85], we consider for the dynamic optimization problem the portfolio weight iteration in the transfer step and compare the four simulation-based methods.

In Table 3.2, we observe that "SGBM-LT", among the four methods, always provides the highest certainty equivalent rates, which implies that the portfolio management strategy generated by "SGBM-LT" is most similar to the optimal one. However, when the investment horizon is long and risk aversion is high, even the results of "SGBM-LT" are not highly satisfactory. In that case, we prefer to solve the dynamic portfolio management problem by the COS portfolio management method. Regarding the simulation-based methods, "SGBM" and "SGBM-LT" are superior to "BGSS" and "vBB", since their corre-

sponding CERs have larger means and smaller standard errors.

Table 3.2: Mean and the standard derivations of the CER from 100 runs, comparing 4 simulation-based methods for dynamic portfolio management for different investment horizons and risk aversion parameters. The portfolio weight iteration is utilized; The COS method serves as the reference.

		annualized certainty equivalent rate (%)								
	γ	BGSS	(s.e.)	vBB	(s.e.)	SGBM	(s.e.)	SGBM -LT	(s.e.)	COS
$T = 40$	5	8.51	(0.02)	8.52	(0.02)	8.55	(0.02)	8.56	(0.02)	8.53
	10	7.61	(0.03)	7.63	(0.04)	7.67	(0.02)	7.77	(0.03)	7.74
	15	7.12	(0.03)	7.08	(0.05)	7.17	(0.02)	7.26	(0.02)	7.27
	20	6.85	(0.03)	6.72	(0.06)	6.90	(0.01)	6.97	(0.02)	6.98
$T = 80$	5	8.79	(0.02)	8.78	(0.02)	8.96	(0.01)	8.97	(0.01)	8.94
	10	7.76	(0.04)	7.63	(0.03)	8.08	(0.02)	8.28	(0.02)	8.29
	15	7.18	(0.05)	7.03	(0.04)	7.47	(0.01)	7.65	(0.02)	7.83
	20	6.86	(0.05)	6.65	(0.06)	7.13	(0.01)	7.28	(0.01)	7.49

Different from the findings in [85] that value function iteration also results in low certainty equivalent rates here. Table 3.3 shows that when using "SGBM" or "SGBM-LT", we can also get satisfactory results by the value function iteration in most test cases. Portfolio weight iteration is significantly better than value function iteration when the risk aversion is large and the investment horizon is long.

3.5.4. INFLUENCE OF VARYING INITIAL STATE

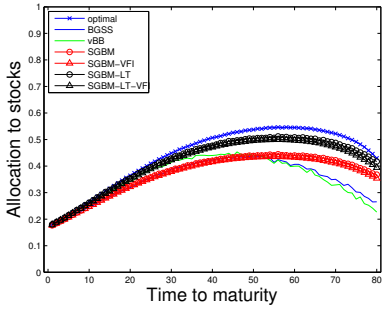
We consider three different initial values, d_0, of the state variable. Each value corresponds to the p-th quantile of the unconditional distribution of d, where p takes values 25, 50 and 75.

Figure 3.2 shows that, for any initial state, "SGBM-LT" performs better than the other three simulation-based algorithms. The intermediate asset allocations generated by "SGBM-LT" are most similar to the optimal ones. At the initial recursion steps, "vBB" also generates similar asset allocations. However, as the backward recursion progresses, the uncertainty in the "vBB" estimates grows and hence the accuracy of "vBB" gets worse.

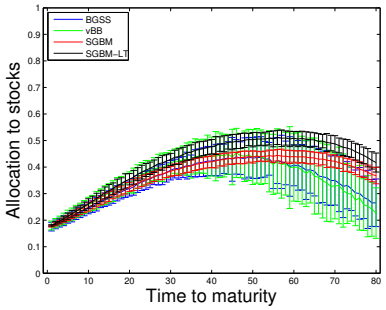
In any case, "SGBM" and "SGBM-LT" yield estimates with low uncertainties. Moreover, we see that "SGBM-VFI" and "SGBM-LT-VFI", in which the value function iteration is considered in the recursion step, respectively, generate very similar results to those of "SGBM" and "SGBM-LT". These are advantages of the new method to calculate conditional expectations.

3.5.5. INFLUENCE OF VARYING MODEL UNCERTAINTY

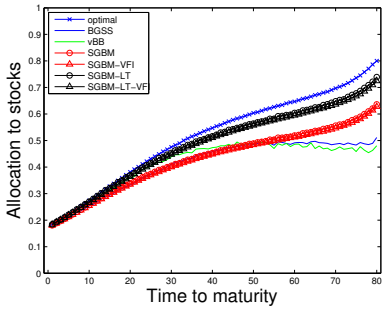
If we consider higher model uncertainty in the 1D-VAR model, the aforementioned methods perform differently. The model uncertainty can be modified by introducing a multiplier MP^2 to the original covariance matrix Σ_ϵ of the white noise vector $[\epsilon^r_{t+\Delta t}, \epsilon^d_{t+\Delta t}]'$, so



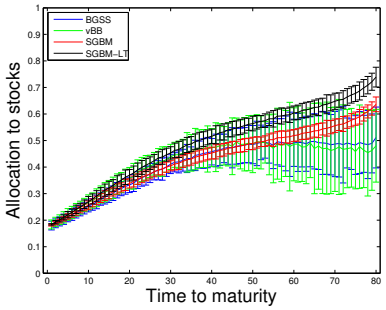
(a) Mean of the average asset allocations, $d_0 = -3.8551$



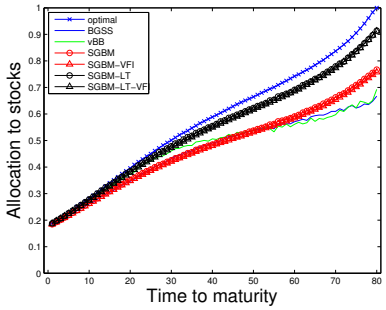
(b) Confidence interval of the average asset allocations, $d_0 = -3.8551$



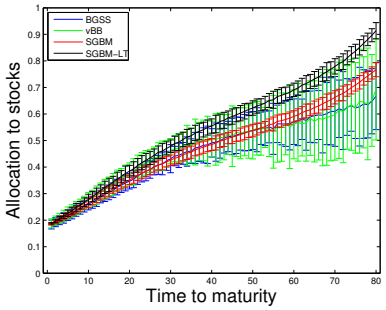
(c) Mean of the average asset allocations, $d_0 = -3.6905$



(d) Confidence interval of the average asset allocations, $d_0 = -3.6905$



(e) Mean of the average asset allocations, $d_0 = -3.5258$



(f) Confidence interval of the average asset allocations, $d_0 = -3.5258$

Figure 3.2: Comparison of simulation-based algorithms for estimating the optimal intermediate asset allocations for different initial states. At each time step, the average asset allocations are computed. For the simulation-based algorithms, the mean and the standard derivation of the average asset allocations are generated from 100 runs. The optimal values are generated by the COS method. Notice that the x-axis denotes "time to maturity". Since we perform recursive programming backward in time, we get accurate solutions when time to maturity is zero, i.e. $t = T$. However, the uncertainty in the asset allocations increases when time to maturity gets larger.

Table 3.3: Mean and the standard derivations of the CER from 100 runs, comparing "SGBM" and "SGBM-LT" with the portfolio weight and the value function iteration; different investment horizons and risk aversion parameters; The COS method serves as the reference.

		annualized certainty equivalent rate (%)								
		SGBM				SGBM-LT				COS
	γ	PWI	(s.e.)	VFI	(s.e.)	PWI	(s.e.)	VFI	(s.e.)	
$T = 40$	5	8.55	(0.02)	8.53	(0.02)	8.56	(0.02)	8.53	(0.02)	8.53
	10	7.67	(0.02)	7.61	(0.03)	7.77	(0.03)	7.70	(0.04)	7.74
	15	7.17	(0.02)	7.13	(0.02)	7.26	(0.02)	7.18	(0.16)	7.27
	20	6.90	(0.01)	6.86	(0.08)	6.97	(0.02)	6.81	(0.39)	6.98
$T = 80$	5	8.96	(0.01)	8.94	(0.02)	8.97	(0.01)	8.94	(0.02)	8.94
	10	8.08	(0.02)	8.02	(0.02)	8.28	(0.02)	8.18	(0.15)	8.29
	15	7.47	(0.01)	7.42	(0.05)	7.65	(0.02)	7.55	(0.11)	7.83
	20	7.13	(0.01)	7.09	(0.04)	7.28	(0.01)	7.16	(0.17)	7.49

that the covariance matrix of the error term will be:

$$\Sigma_\epsilon^M = \mathrm{MP}^2 \cdot \Sigma_\epsilon. \tag{3.17}$$

In this test, with a fixed risk aversion parameter $\gamma = 10$, we change the multiplier MP and the investment horizon and report the certainty equivalent rates corresponding to the different algorithms.

As shown in Table 3.4, when the model uncertainty increases, "vBB" is the most impacted algorithm. "BGSS" performs somewhat better than "vBB" but worse than "SGBM" and "SGBM-LT" as the corresponding certainty equivalent rate is smaller and with higher uncertainty. "SGBM-LT" outperforms "SGBM". The differences are obvious when the model uncertainty is high and the investment horizon is long. The "SGBM-LT" values in the table are obtained with sample size 2^{14}. In any case, "COS" yields the reference results, which are verified by using "SGBM-LT" with a large sample size 2^{18}. In that case, we find, for example, the certainty equivalent rate of "SGBM-LT" has mean value 7.71 and standard error 0.02 when $T = 20$ and MP = 4.

3.5.6. ERRORS OF THE FOUR SIMULATION-BASED METHODS

In this subsection, we would like to briefly summarize the errors encountered within the methods analyzed. If we do not consider errors in the simulation part, the errors of the four simulation-based methods, "vBB", "BGSS", "SGBM" and "SGBM-LT", can be subdivided into three categories:

- *approximation error*, which occurs when we approximate the true value functions by the Taylor series expansion.
- *projection error*, which occurs when we use low-order polynomials to approximate the conditional expectations of the value functions or of the approximated value functions.

Table 3.4: Comparing four methods with various model uncertainties. The risk aversion parameter is fixed as 10. For different investment horizons T in quarters and MP values in Equation (3.17), the table reports the mean and the standard derivation of the certainty equivalent rate from 100 runs. The results from the COS method serve as the reference values.

	annualized certainty equivalent rate (%)									
	MP	BGSS	(s.e.)	vBB	(s.e.)	SGBM	(s.e.)	SGBM -LT	(s.e.)	COS
$T = 10$	1	6.64	(0.03)	6.64	(0.03)	6.65	(0.03)	6.65	(0.03)	6.64
	2	6.65	(0.03)	6.65	(0.03)	6.66	(0.03)	6.66	(0.03)	6.65
	3	6.79	(0.38)	6.59	(0.82)	6.84	(0.05)	6.86	(0.03)	6.85
	4	6.98	(1.05)	0.11	(6.46)	7.11	(0.04)	7.14	(0.04)	7.13
$T = 20$	1	7.06	(0.03)	7.07	(0.04)	7.07	(0.03)	7.10	(0.03)	7.06
	2	7.06	(0.03)	7.01	(0.06)	7.06	(0.03)	7.12	(0.04)	7.07
	3	7.21	(0.31)	6.84	(0.34)	7.27	(0.06)	7.40	(0.05)	7.34
	4	7.20	(0.99)	-0.23	(5.00)	7.54	(0.11)	7.78	(0.07)	7.72

- *regression bias*, which occurs when we use cross-path regression to approximate the conditional expectations.

The *approximation error* does not occur when Taylor series expansions are not involved, for example, in "vBB". However, as we have seen in the numerical tests, "BGSS" and "SGBM" suffer from this source of error in a similar fashion, while "SGBM-LT" appears to suffer less.

The *projection error* is the main source of error in "vBB", where low-order polynomials are implemented to approximate the value functions, which may be high-order functions when the risk aversion is high, see Equation (3.12). For "BGSS", "SGBM" and "SGBM-LT", this is generally not a problem since the object functions, as in Equation (3.9), are at most of fourth-order.

The *regression bias*, which has been discussed in Chapter 2, can be controlled effectively by bundling. The regression bias is high in "vBB" and "BGSS" but relatively low in "SGBM" and "SGBM-LT", that benefit from their bundling technique.

A general description of the error components of the four simulation-based methods is listed in Table 3.5. "SGBM-LT" exhibits a highly satisfactory performance in our tests, since it has relatively small-sized errors in all three aspects. We expect however that when the risk aversion parameter is high and the model volatility is large, even "SGBM-LT" may fail to converge in some cases. In those cases, we propose either to use a large number of paths in the simulation together with more bundles.

3.6. CONCLUSION

In this chapter, we enhanced a popular dynamic portfolio management algorithm, the BGSS algorithm, in two aspects. First, for the computation of the conditional expectations appearing, we replaced the standard regression method by the techniques from the Stochastic Grid Bundling Method, so that the variances of the approximated asset

Table 3.5: Errors of the four simulation-based methods.

	vBB	BGSS	SGBM	SGBM-LT
approximation error	-	high	high	low
projection error	high	low	low	low
regression bias	high	high	low	low

allocations and the corresponding certainty equivalent rates can be reduced. Then, a log Taylor expansion, based on a nonlinear decomposition, was employed in our algorithm. This expansion gives rise to improved results compared to the original ones when approximating the utility function. The resulting SGBM-based portfolio management algorithm results in a lower biased approximation of the optimal asset allocations.

Based on the COS method and the grid-searching technique, we developed the COS portfolio management method for generating reference values, which are quite comparable to the reference values and further serve as the "optimal" solutions in our numerical tests.

In our tests, combining SGBM and the log Taylor expansion yielded superior results to those of other simulation-based algorithms. In all testing cases, "SGBM-LT" shows the higher certainty equivalent rates. When we merely consider introducing the SGBM components in the regression step, the benefits are obvious: the value function iteration and the portfolio weight iteration associated to both "SGBM" and "SGBM-LT" generate quite similar results, which indicate that the approximation errors at each recursion step are small.

Our simulation- and regression-based algorithm "SGBM-LT" can be generalized to higher-dimensional dynamic portfolio management problems. Besides, since our algorithm is robust even in scenarios with high volatility dynamics, it is also possible to focus on models with more complicated dynamics, for example, models with jump components or other time series models. In those cases, we may need some effective bundling technique as proposed in Chapter 2 but in each local domain we may still use low-order polynomials as the basis functions. This helps to retain the robustness of our algorithm.

CHAPTER 4

Multi-period Mean-Variance Portfolio Optimization

"If you live each day as if it was your last, some day you'll most certainly be right."

In this chapter, we propose a simulation-based approach for solving the constrained dynamic mean-variance portfolio management problem. For this dynamic optimization problem, we first consider a sub-optimal strategy, called the multi-stage strategy, which can be utilized in a forward fashion. Then, based on this fast yet sub-optimal strategy, we propose a backward recursive programming approach to improve it. We design the backward recursion algorithm such that the result is guaranteed to converge to a solution, which is at least as good as the one generated by the multi-stage strategy. In our numerical tests, highly satisfactory asset allocations are obtained for dynamic portfolio management problems with realistic constraints on the control variables.

Keywords: Dynamic portfolio management · Mean-variance optimization · Constrained optimization · Simulation method · Least-square regression

4.1. INTRODUCTION

Since Markowitz's pioneering work [68] on a single-period investment model, the mean-variance portfolio optimization problem has become a cornerstone of investment management in both academic and industrial fields. An interesting topic, extending Markowitz's work, is to consider the mean-variance target for a continuous or multi-period optimization problem. Along with introducing dynamic control into the optimization process, constraints on the controls can be included.

In some situations, the constrained dynamic mean-variance optimization problem can be solved analytically. For example, [64] solves this portfolio management problem with no-shorting of stock allowed and [10] solves the problem with bankruptcy prohibition. In [40], the authors investigate the mean-variance problem with a borrowing constraint, where the investor faces a borrowing rate different from the risk-free saving rate. However, all this research is performed in the framework of continuous optimization. In fact, as mentioned in [28], the continuous constrained optimization problem is usually easier than the discrete one. In general, an elegant analytic mean-variance formulation can be derived in case of a complete market, where re-balancing can be performed continuously and there are no constraints on the controls and no jumps in asset dynamics.

This chapter is based on the article 'Multi-period mean-variance portfolio optimization based on Monte Carlo simulation', published in *Journal of Economic Dynamics and Control*, 64(1):23–38, 2016 [26].

If we consider a realistic problem which is designed in an incomplete market, it will be difficult to obtain analytic solutions and thus utilizing computational techniques to calculate numerical solutions is the preferred choice.

The nonlinearity of conditional variance is the main obstacle for solving the dynamic mean-variance optimization problem. In [91] and [63], an embedding technique, by which the mean-variance problem is transformed into a stochastic linear-quadratic (LQ) problem, is introduced. For the linear-quadratic problem, an investor does not need to choose a trade-off parameter between mean and variance. Instead, she decides a final optimization target of her investment. In [91] and [63], it is proved that varying the final investment target traces out the same so-called *time-0 mean-variance efficient frontier*[1] as varying the mean-variance trade-off parameters. To generate the efficient frontier of the mean-variance optimization problem, we can thus solve the LQ problem with different target parameters.

To solve the constrained target-based problem, the Hamilton-Jacobi-Bellman partial differential equation (HJB PDE) is often considered. Accurate results can be generated by solving the HJB PDEs for a one-dimensional scenario. For example, [88] solves the continuous constrained mean-variance optimization problem with various constraints and the risky asset following geometric Brownian motion. The authors of [33] solve a similar problem with the risky asset following jump-diffusion dynamics. In both papers, realistic constraints are cast on the control variables. However, it may be rather expensive to implement the algorithm, which is based on solving the HJB PDE, for a problem with several risky assets. Reducing the dimensionality of such a problem is a potential solution. However, when constraints are introduced, the assumption for establishing the well-known mutual fund theory is not valid and the ratio between the different risky assets is not constant any more. A general multi-dimensional problem can hardly be transformed into a one-dimensional problem.

To deal with the curse of dimensionality, using Monte Carlo simulation constitutes a possible solution. A well-known simulation-based dynamic portfolio management algorithm is proposed in [16] and further enhanced as in the previous chapter. However, this is for a problem where the investor has constant relative risk aversion. For such problems, the investor's optimal asset allocation is only influenced by the dynamics of the risky asset, so dynamic programming can be performed after the (forward) simulation of the risky assets. For an investor with other types of risk aversion, her optimal intermediate decisions usually do not only depend on the dynamics of the risky asset but also on the amount of wealth at that time. The simulation approach proposed in [16] is therefore not feasible for a general investment problem. Solving the constrained dynamic mean-variance problem based on Monte Carlo simulation is the focus of our work in the present and the following chapters.

Our methods depend on transforming the mean-variance problem into the LQ problem, which is a so-called target-based problem. In [4], the investment strategy for solving the LQ problem is named the *pre-commitment strategy*, which however does not guarantee time consistency. As mentioned in [89], a time-consistent strategy can be formulated as a pre-commitment strategy plus time consistent constraints on the asset allocations. Thus, the pre-commitment strategy generally yields an efficient frontier which is

[1] In the following part we use the term "efficient frontier" for short.

superior to the one generated by a time consistent strategy. In this chapter, we will contribute to pre-commitment strategies and propose two solutions for the dynamic mean-variance problem, one is performed in a forward manner and the other in a backward manner.

In the forward approach, we decompose the dynamic optimization problem into several static optimization problems by specifying intermediate investment targets at all re-balancing time steps. A reasonable approximation for the optimal controls can be determined at each single stage and solving the problems at all stages provides us a sub-optimal strategy called the "multi-stage strategy". We prove that the multi-stage strategy is the optimal strategy when there are no constraints on the asset allocations. Although the multi-stage strategy becomes sub-optimal in case of constrained controls, it is straightforward to implement the multi-stage strategy for either high-dimensional problems or problems with complicated constraints. While experimenting, we observe that the multi-stage strategy can yield highly satisfactory results compared to the reference solutions.

The main challenge to perform backward programming for a constrained optimization problem is that the value function at each time step is non-smooth and thus the optimality cannot be computed efficiently by solving the corresponding first order conditions. To tackle this problem, we utilize the idea of differential dynamic programming [56], by which a stochastic control problem is solved by a local optimization strategy, and we come up with a backward recursive approach. Our backward recursive programming algorithm is an iterative method. With special design of the algorithm, we can guarantee that the outcome converges to a solution, which is not worse than the solution generated by the multi-stage strategy. In the backward process, conditional expectations are calculated recursively via cross-path least-squares regression. To make this numerical approach stable, we implement the "bundling" and the "regress-later" techniques, as adopted in Chapters 2 and 3. The idea of "bundling" is highly compatible with the local optimization in differential dynamic programming. The backward recursive programming is initiated with a reasonable guess for the asset allocations, which can be, but is not restricted to, the one generated by the multi-stage strategy. In our tests with the initial allocation generated by the multi-stage strategy, we achieve highly satisfactory results after at most four backward iterations. Like the multi-stage strategy, the backward recursive programming can be performed highly efficiently. In our numerical tests, one iteration of the backward recursive programming only takes a few seconds.

This chapter is organized as follows. In Section 4.2, we introduce the formulation of the dynamic mean-variance problem and the embedding into a stochastic LQ problem. Section 4.3 describes the multi-stage strategy. The optimality of the multi-stage strategy in the unconstrained case is proved in Section 4.3.1. In Section 4.4, the backward dynamic programming method is presented. Section 4.5 displays several realistic constraints for the portfolio management problems and in Section 4.6 numerical tests are performed for both one- and two-dimensional problems. We conclude in Section 4.7.

4.2. PROBLEM FORMULATION

For convenience, we consider a portfolio consisting of one risk-free and one risky asset. To extend the analysis to a problem with more than one risky asset is feasible. We assume

that the re-balancing dates for the portfolio are equidistantly distributed and that the total number of re-balancing opportunities before terminal time T is equal to M. The time step Δt between two re-balancing dates is thus $\frac{T}{M}$, and the portfolio can be traded at times $t \in [0, \Delta t, \ldots, T-\Delta t]$. At each trading time t, an investor decides the trading strategy to maximize the expectation of the terminal wealth and to minimize the investment risk. Formally, the investor's problem is given by

$$J_t(W_t) = \max_{\{x_s\}_{s=t}^{T-\Delta t}} \Big\{ \mathbb{E}[W_T|W_t] - \lambda \cdot \mathrm{Var}[W_T|W_t] \Big\}, \tag{4.1}$$

subject to the wealth restriction:

$$W_{s+\Delta t} = W_s \cdot (x_s R_s^e + R_f) + C \cdot \Delta t, \qquad s = t, t+\Delta, \ldots, T-\Delta t.$$

4

$J_t(W_t)$ is termed the *value function*, which measures the investor's investment opportunities at time t with wealth W_t. x_s denotes the asset allocation of the investor's wealth in the risky asset in the period $[s, s+\Delta t)$. It is assumed that the admissible investment strategy x_t is an $\mathscr{F}_t$-measurable Markov control, i.e. $x_t \in \mathscr{F}_t$. R_f is the return of the risk-free asset in one time step, which is assumed to be constant for simplicity, and R_s^e is the excess return of the risky asset during $[s, s+\Delta t)$. We assume that the excess returns $\{R_t^e\}_{t=0}^{T-\Delta t}$ are statistically independent. $C \cdot \Delta t$ stands for a contribution of the investor in the portfolio during $[s, s+\Delta t)$, and a negative C can be interpreted as a constant withdrawal of the investor from the portfolio. The risk aversion attitude of the investor is denoted by λ, which is the trade-off factor between maximizing the profit and minimizing the risk.

Remark 4.2.1. *For a risky asset with dynamics following geometric Brownian motion or a Levy process, assuming the returns to be statistically independent is valid. For some problems where the asset returns are directly defined, for example, as model-free data, the independence assumption is also correct. For a VAR or GARCH model, this assumption is, however, not satisfied.*

Remark 4.2.2. *When the mean-variance optimization problem proposed by Equation (4.1) is convex, then solving Equation (4.1) is equivalent to determination of the Pareto optimal points, i.e. solving*

$$\min_{\{x_s\}_{s=t}^{T-\Delta t}} \Big\{ \mathrm{Var}[W_T|W_t] \Big\},$$
$$s.t.\ \mathbb{E}[W_T|W_t] \geq d,$$

with a suitable choice of d. However, when the problem is not convex, solving Equation (4.1) generates Pareto optimal points, but not all of them.

The difficulty of solving this mean-variance optimization problem is caused by the nonlinearity of conditional variances, namely $\mathrm{Var}[\mathrm{Var}[W_T|\mathscr{F}_t]|\mathscr{F}_s] \neq \mathrm{Var}[W_T|\mathscr{F}_s], s \leq t$, which makes the well-known dynamic programming valuation approach not applicable. To tackle this problem, the original mean-variance equations can be transformed into another framework as done in [63, 88, 91]. The following theorem supports the transformation.

Theorem 4.2.3. *If $\{x_s^*\}_{s=t}^{T-\Delta t}$ is the optimal control for the problem defined in Equation (4.1), then $\{x_s^*\}_{s=t}^{T-\Delta t}$ is also the optimal control for the following problem:*

$$\min_{\{x_s\}_{s=t}^{T-\Delta t}} \left\{\mathbb{E}[(W_T - \frac{\gamma}{2})^2 | W_t]\right\}, \tag{4.2}$$

where $\gamma = \frac{1}{\lambda} + 2\mathbb{E}_{x^*}[W_T|W_t]$. Here the operator $\mathbb{E}_{x^*}[\cdot]$ denotes the expectation of the investor's terminal wealth if she invests according to the optimal strategy $\{x_s^*\}_{s=t}^{T-\Delta t}$.

Proof. See [63]. □

Based on this theorem, the original mean-variance problem can be embedded into a tractable auxiliary LQ problem. The investment strategy corresponding to this LQ problem is called the pre-commitment strategy. This technique can also be interpreted as transforming the original mean-variance problem into a target-based optimization problem, which has been discussed in [44, 52]. For numerical computation, the pre-commitment optimization problem as shown in Equation (4.2) is usually formulated as an HJB PDE and realistic constraints on either controls or state variables can be correspondingly established as boundary conditions. In this manner, [88] solves the mean-variance problem numerically and derives a solution to the constrained pre-commitment strategy.

However, even for the LQ problem, casting constraints in the numerical approach is in general not trivial. Imposing constraints on the controls will substantially change the formulation of the problem and make it nontrivial to solve the problem efficiently. The reasoning is as follows. For the unconstrained problem, the value function at each time step forms a smooth function and thus the optimality can be obtained by solving the first-order conditions associated to this smooth function. Adding constraints will remove the smoothness of the value function. Derivative-based optimization techniques cannot be applied in this situation and the optimality has to be computed by grid-searching on the whole domain of possible controls, for example, as in [88].

In the following section, we propose a sub-optimal yet highly efficient strategy for the mean-variance portfolio management problem. In this strategy, we avoid dealing with non-smooth value functions even if there are constraints on the controls. It is possible to extend this sub-optimal strategy to high-dimensional problems and to problems with complicated asset dynamics.

4.3. A FORWARD SOLUTION: THE MULTI-STAGE STRATEGY

When we prescribe constraints on the allocations, neither the original mean-variance strategy associated to solving Equation (4.1) nor the pre-commitment strategy associated to solving Equation (4.2) is easy to obtain.

Here we treat the mean-variance optimization problem from a different angle. After writing it into the pre-commitment form, or equivalently into the target-based form, the objective of the reformulated optimization problem is to minimize the difference between the final wealth and a predetermined target. Hence the optimization problem at time t reads:

$$V_t(W_t) := \min_{\{x_s\}_{s=t}^{T-\Delta t}} \left\{\mathbb{E}[(W_T - \frac{\gamma}{2})^2 | W_t]\right\}, \tag{4.3}$$

or, in a recursive fashion,

$$V_t(W_t) = \min_{x_t} \left\{ \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t}) | W_t] \right\}, \tag{4.4}$$

with $V_T(W_T) = (W_T - \frac{\gamma}{2})^2$. Here we denote the value function by $V_t(\cdot)$ instead of $J_t(\cdot)$, which is used in Equation (4.1). As mentioned in Chapter 1, $J_t(\cdot)$ stands for a value fuction which cannot be solved directly by Bellman dynamic programming.

At the state (t, W_t), i.e. time t and wealth W_t, the value function $V_t(W_t)$ depends on all optimal allocations at subsequent time steps. That is why generally this kind of optimization problem has to be solved in a backward recursive fashion via, for example, solving the HJB PDEs or backward stochastic differential equations (BSDEs) with conditions at the terminal time.

Solving this dynamic programming problem numerically in a backward recursive fashion suffers from two problems. First, solving the optimality, especially for constrained cases, at each time step may be difficult or computationally expensive. Secondly, since we use the value function to transmit information between two recursive steps, the error accumulates as the recursion proceeds.

Reflection on these two issues leads us to define a sub-optimal strategy, which does not involve these two types of errors, which we call the multi-stage strategy.

Notice that at the terminal time, our target is $\frac{\gamma}{2}$. Then, at state (t, W_t), in the multi-stage strategy we choose x_t to be:

$$x_t^{*ms} := \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) + C \cdot \Delta t - \Xi_{t+\Delta t} \right)^2 \middle| W_t \right] \right\}, \tag{4.5}$$

where

$$\Xi_t = \frac{\frac{\gamma}{2} - C \cdot \Delta t \cdot \frac{1-(R_f)^{(T-t)/\Delta t}}{1-R_f}}{(R_f)^{(T-t)/\Delta t}}. \tag{4.6}$$

So, here we do not consider the optimality in the future, but perform a single-stage, or static, optimization with respect to a given target value.

Equation (4.6) is straightforward. We set an intermediate target at time t such that once we achieve this target, we can put all the money in the risk-free asset and at the terminal time the wealth reaches the final target. Therefore, this intermediate target is computed by discounting the final target while taking into account the constant contribution.

Considering an intermediate wealth target is not a new idea. In [45], the authors study the target-based optimization problem for a defined-contribution pension plan and make use of intermediate targets mainly for calculating the cumulative losses throughout the management period. In [29], the authors consider an intermediate threshold. Once the portfolio wealth exceeded this threshold, they proved that withdrawing a suitable amount of money from the portfolio will not influence the performance of the portfolio in the sense of mean-variance optimization.

Since our multi-stage method merely depends on solving a single-stage optimization problem at each time point, the problem can be solved in a forward fashion:

- First, we generate the intermediate target values at each time step.

- Then, starting at the initial state we compute the optimal allocation step by step until the terminal time.

If we consider no periodic contributions, i.e. $C = 0$, we can rewrite the optimal allocation for the pre-commitment problem in Equation (4.3) as:

$$x_t^{*pc}(W_t) = \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot \prod_{s=t+\Delta t}^{T-\Delta t} (x_s^{*pc} R_s^e + R_f) - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}, \tag{4.7}$$

where $\{x_s^{*pc}\}_{s=t+\Delta t}^{T-\Delta t}$ denote the optimal allocations at times $s = t+\Delta t, \ldots, T-\Delta t$.

In this scenario the multi-stage optimization problem can be formulated as:

$$x_t^{*ms}(W_t) = \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot \prod_{s=t+\Delta t}^{T-\Delta t} (R_f) - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}. \tag{4.8}$$

We see that the objective of the multi-stage optimization is indeed different from that of the original optimization. Instead of considering the true optimization $\{x_s^{*pc}\}_{s=t+\Delta t}^{T-\Delta t}$, we "specify" $\{x_s^{*ms}\}_{s=t+\Delta t}^{T-\Delta t}$ to be zero. That is why we call the multi-stage solution sub-optimal. However, by sacrificing the possibility to pursue the optimality, we also gain some profits which are discussed in the next section.

Remark 4.3.1. *The multi-stage strategy can be treated as a "greedy strategy" for the stochastic optimization problem [13]. In each stage, we minimize the distance between our wealth and the target of the current stage.*

Remark 4.3.2. *One crucial restriction for constructing the multi-stage strategy is that there should be a risk-free asset in the market. Regarding the wealth-to-income case discussed in [88], we cannot find a risk-free part in the market and therefore cannot derive the intermediate optimization target for our multi-stage approach.*

GAINS AND LOSSES

By the multi-stage strategy, we avoid the backward recursive programming by constructing determined intermediate targets. Since there is no error accumulation in the recursion, the optimization problem solved at the intermediate time step is unbiased[2] to what we designed it to be.

On the other hand, casting constraints on this sub-optimal strategy is trivial. Since at every time step we deal with a quadratic optimization problem in the multi-stage strategy, solving the constrained optimization problem can usually be performed efficiently. In the one-dimensional case, for example, solving the constrained quadratic optimization problem is equivalent to first solving the unconstrained problem to generate the optimal control and then truncating this optimal control by the constraints.

The drawback of the multi-stage strategy is also obvious: instead of tackling the original optimization problem, we work on a tailored one. The optimal control for the sub-optimal problem may differ from that for the original problem, so the mean-variance

[2] In the traditional backward programming approach, since numerical errors cumulate alongside the recursive procedure, biases in the intermediate optimization problems exist.

pair corresponding to this sub-optimal strategy may be located below the optimal efficient frontier. However, in the next section we will prove that in some situations the optimal allocation for the multi-stage strategy is exactly the same as that for the original pre-commitment strategy.

4.3.1. EQUIVALENCE IN THE UNCONSTRAINED CASE

In this section, we restrict ourselves to the situation where there is no periodic contribution, i.e. $C = 0$. Extending the analysis to the case where $C \neq 0$ is however possible. Under the following condition, we can prove that the multi-stage strategy and the pre-commitment strategy are equivalent for generating the optimal asset allocations.

Condition 4.3.3. *The asset allocations at each time step are unconstrained.*

In this case, we can obtain the analytic form of the value function at intermediate time steps for the pre-commitment problem.

Lemma 4.3.4. *For the pre-commitment problem shown in Equation (4.3), the value function $V_t(W_t)$ can be formulated as:*

$$V_t(W_t) = L_t \cdot \left(W_t \cdot (R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2}\right)^2, \tag{4.9}$$

where $L_t = \prod_{s=t}^{T} l_s$ with l_t defined as follows:

$$\begin{aligned} l_t &= 1 - \frac{\mathbb{E}[R_t^e]^2}{\mathbb{E}[(R_t^e)^2]}, \quad t = 0, \Delta t, \ldots, T - \Delta t, \\ l_T &= 1. \end{aligned}$$

Proof. At time step T, the value function is known as:

$$V_T(W_T) = (W_T - \frac{\gamma}{2})^2,$$

which satisfies Equation (4.9). At time step $T - \Delta t$, the value function reads:

$$\begin{aligned} V_{T-\Delta t}(W_{T-\Delta t}) &= \max_{x_{T-\Delta t}} \left\{\mathbb{E}[V_T(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f)) | W_{T-\Delta t}]\right\} \\ &= \max_{x_{T-\Delta t}} \left\{\mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right)^2 \middle| W_{T-\Delta t}\right]\right\}. \end{aligned} \tag{4.10}$$

To obtain the analytic form of $V_{T-\Delta t}(W_{T-\Delta t})$, we first need to determine the optimal asset allocation $x^*_{T-\Delta t}$, which satisfies:

$$x^*_{T-\Delta t} = \arg\min_{x_{T-\Delta t}} \left\{\mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right)^2 \middle| W_{T-\Delta t}\right]\right\}.$$

Solving the first-order conditions of the optimality problem gives us that $x^*_{T-\Delta t}$ is the solution to the following equation:

$$\mathbb{E}\left[\left(W_{T-\Delta t} \cdot (x_{T-\Delta t} R^e_{T-\Delta t} + R_f) - \frac{\gamma}{2}\right) \cdot W_{T-\Delta t} R^e_{T-\Delta t} \middle| W_{T-\Delta t}\right] = 0.$$

So, the optimal allocation $x^*_{T-\Delta t}$ can be calculated as:

$$x^*_{T-\Delta t} = \frac{(\frac{\gamma}{2} - W_{T-\Delta t}R_f)\cdot \mathbb{E}[R^e_{T-\Delta t}]}{W_{T-\Delta t}\cdot \mathbb{E}[(R^e_{T-\Delta t})^2]}. \tag{4.11}$$

Inserting Equation (4.11) into Equation (4.10) yields:

$$\begin{aligned} V_{T-\Delta t}(W_{T-\Delta t}) &= \mathbb{E}\Big[\Big(1 - \frac{\mathbb{E}[R^e_{T-\Delta t}]\cdot R^e_{T-\Delta t}}{\mathbb{E}[(R^e_{T-\Delta t})^2]}\Big)^2\Big]\cdot\Big(W_{T-\Delta t}R_f - \frac{\gamma}{2}\Big)^2 \\ &= \Big(1 - \frac{\mathbb{E}[R^e_{T-\Delta t}]^2}{\mathbb{E}[(R^e_{T-\Delta t})^2]}\Big)\cdot\Big(W_{T-\Delta t}R_f - \frac{\gamma}{2}\Big)^2. \end{aligned}$$

It is clear that the value function at time step $T-\Delta t$ has the same form as in Equation (4.9). For the remaining time steps, we can formulate the value functions by backward induction.

Assume that at time step $t+\Delta t$ we have:

$$V_{t+\Delta t}(W_{t+\Delta t}) = L_{t+\Delta t}\cdot\Big(W_{t+\Delta t}\cdot(R_f)^{(T-t)/\Delta t-1} - \frac{\gamma}{2}\Big)^2.$$

Then at time step t, the value function can be formulated as:

$$\begin{aligned} V_t(W_t) &= \max_{x_t}\Big\{\mathbb{E}[V_{t+\Delta t}(W_t\cdot(x_tR^e_t + R_f))|W_t]\Big\} \\ &= \max_{x_t}\Big\{\mathbb{E}\Big[L_{t+\Delta t}\cdot\Big(W_t\cdot(x_tR^e_t + R_f)\cdot(R_f)^{(T-t)/\Delta t-1} - \frac{\gamma}{2}\Big)^2\Big|W_t\Big]\Big\} \\ &= \mathbb{E}[L_{t+\Delta t}]\cdot\max_{x_t}\Big\{\mathbb{E}\Big[\Big(W_t\cdot(x_tR^e_t + R_f)\cdot(R_f)^{(T-t)/\Delta t-1} - \frac{\gamma}{2}\Big)^2\Big|W_t\Big]\Big\}. \end{aligned}$$

Here the last equality is based on the independence of the excess returns. Solving the first-order condition yields:

$$x^*_t = \frac{(\frac{\gamma}{2} - W_t\cdot(R_f)^{(T-t)/\Delta t})\cdot\mathbb{E}[R^e_t]}{W_t\cdot(R_f)^{(T-t)/\Delta t-1}\cdot\mathbb{E}[(R^e_t)^2]}, \tag{4.12}$$

and the corresponding value function reads:

$$\begin{aligned} V_t(W_t) &= L_{t+\Delta t}\cdot\Big(1 - \frac{\mathbb{E}[R^e_t]^2}{\mathbb{E}[(R^e_t)^2]}\Big)\cdot\Big(W_t\cdot(R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2}\Big)^2 \\ &= L_t\cdot\Big(W_t\cdot(R_f)^{(T-t)/\Delta t} - \frac{\gamma}{2}\Big)^2. \end{aligned}$$

Thus we finalize the proof. □

Remark 4.3.5. *The optimal asset allocation shown in Equation (4.12) is exactly the same as the one proposed in [63] for the one-dimensional case.*

With Lemma 4.3.4, we can prove the equivalence between our multi-stage strategy and the pre-commitment strategy. Formally, the equivalence is shown in the following proposition.

Proposition 4.3.6. *For a mean-variance portfolio management problem, where Condition 4.3.3 is satisfied, the optimal control for the multi-stage strategy and for the pre-commitment strategy are identical. That is, x_t^{*pc} and x_t^{*ms}, as respectively shown in Equations (4.7) and (4.8), are equal at each time step t.*

Proof. For the pre-commitment problem, the optimal control x_t^{*pc} reads:

$$x_t^{*pc} = \arg\min_{x_t} \left\{ \mathbb{E}[V_{t+\Delta t}(W_t \cdot (x_t R_t^e + R_f)) | W_t] \right\}.$$

Using the form of the value function $V_{t+\Delta t}(\cdot)$ as shown in Lemma 4.3.4, we find:

$$x_t^{*pc} = \arg\min_{x_t} \left\{ \mathbb{E}\left[L_{t+\Delta t} \cdot \left(W_t \cdot (x_t R_t^e + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}.$$

Because of the independence of the excess returns, we can treat $L_{t+\Delta t}$ as a constant factor and thus the minimization problem turns out to be:

$$x_t^{*pc} = \arg\min_{x_t} \left\{ \mathbb{E}\left[\left(W_t \cdot (x_t R_t^e + R_f) \cdot (R_f)^{(T-t)/\Delta t - 1} - \frac{\gamma}{2} \right)^2 \middle| W_t \right] \right\}. \tag{4.13}$$

Since the right-hand-sides of Equations (4.8) and (4.13) have the same form, the proof is finished. □

Notice that we only establish the equivalence between the multi-stage problem and the pre-commitment problem when the excess returns are independent and the allocations are unconstrained. If we equip the excess returns with path-dependent dynamics or cast constraints on the allocations, the equivalence may be lost. However, based on numerical results in [90], we know that the efficient frontiers for different investment strategies may differ significantly in the unconstrained case but the differences are relatively smaller when constraints are introduced. In our numerical approach we only apply the multi-stage strategy to generate asset allocations and then the mean-variance pair is calculated by combining the simulated trajectories and the corresponding "sub-optimal" allocations. As spotted in [88], small errors in asset allocations may not influence the accuracy of the mean-variance pair dramatically.

Even though the multi-stage strategy may be not equivalent to the pre-commitment strategy in some situations, it can serve as a sub-optimal solution to the constrained dynamic mean-variance portfolio optimization problem. If some other accurate solutions exist, the corresponding efficient frontiers should be at least above that of the multi-stage strategy. Moreover, for some numerical methods that depend on iteratively updating the asset allocations, a reasonable initial guess can be provided by the multi-stage method, see Section 4.4.

The authors of [78] proposed a similar technique to the multi-stage strategy for the quadratic convex optimization problem. Our research differs in two aspects. First, instead of dealing with one optimization problem which can yield one point on the efficient frontier, we consider a series of optimization problems which generate results to construct the whole efficient frontier. We find that the multi-stage strategy is particularly satisfactory when the investor is highly risk averse. However, in case the investor is less risk averse, the sub-optimal strategy turns out to be problematic. For this, we propose a backward dynamic programming approach, which is different from the forward strategy.

Remark 4.3.7. *As discussed in [29, 34], a semi-self-financing strategy exists which is better than the pre-commitment strategy. In that strategy, a positive amount of money is withdrawn from the portfolio when the wealth in the portfolio is above a determined value. Similarly, the multi-stage strategy can be adjusted in this respect. However, since the improvement achieved by breaking the self-financing is not significant, we will not deal with the semi-self-financing multi-stage strategy in this chapter.*

4.4. BACKWARD RECURSIVE PROGRAMMING

In the preceding sections, we considered the "greedy policy" for the dynamic optimization problem, and therefore the constrained dynamic mean-variance problem can be solved in a forward fashion via Monte Carlo simulation. Except for the unconstrained case, the multi-stage strategy allocation is generally not the optimal solution to the dynamic optimization problem. In order to get the optimal solution, we have to consider a backward dynamic programming solution. In this section, we present an approach to perform backward recursive calculation based on the solution of the multi-stage strategy.

BENEFIT FROM THE CONSTRAINTS

In general, constraints complicate dynamic optimization problems. For an unconstrained optimization problem, the value functions are smooth, so the optimality can be obtained by solving the first-order conditions. When constraints are introduced, the smoothness of the value functions is destroyed and derivative-based optimization is thus no longer feasible. However, if we treat the constraints differently, they can also be "helpful".

Consider an optimization problem at the state (t, W_t):

$$V_t(W_t) = \min_{x_t \in A} \Big\{ \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t}) | W_t] \Big\}, \tag{4.14}$$

where A is the admissible set for the asset allocation x_t. $V_t(\cdot)$ is the value function at time t. When $A \neq \mathscr{R}$, this is a constrained problem, which may not be easy to solve. However, if we consider a special case where $A = \{x_t | x_t = K\}$, i.e. x_t is restricted to be a constant, the constrained problem becomes trivial. In fact, since we know that x_t has to be constant, the "optimal" solution in the admissible set is known immediately.

Using the multi-stage strategy, we obtain ${x_t^*}^{ms}$, which may be a reasonable approximation of x_t^*, which denotes the real optimal allocation. If we construct a truncated admissible control set, $A_\eta = [{x_t^*}^{ms} - \eta, {x_t^*}^{ms} + \eta]$, the solution to the following optimization problem

$$V_t(W_t) = \min_{x_t \in A_\eta} \Big\{ \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t}) | W_t] \Big\},$$

should be the same as that of the problem shown in Equation (4.14). Assuming that the optimal allocations for the state (t, W_t) are in an interval $[{x_t^*}^{ms} - \eta, {x_t^*}^{ms} + \eta]$, the investor's optimal wealth $W_{t+\Delta t}$ should be located in the domain[3]:

$$\Phi_{t+\Delta t} := \{W_{t+\Delta t} | W_{t+\Delta t} = W_t \cdot (x_t \cdot R_t^e + R_f) + C \cdot \Delta t, \quad x_t \in A_\eta\}.$$

[3] This domain is much smaller than the domain obtained without restricting x_t. To avoid unnecessary technicalities, we assume that the wealth process is locally bounded.

We further transform the original optimization problem shown in Equation (4.14) to be:

$$V_t(W_t) = \min_{x_t \in A} \left\{ \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t}) | W_t, W_{t+\Delta t} \in \Phi_{t+\Delta t}] \right\}, \tag{4.15}$$

where an additional condition is introduced into the conditional expectation. Instead of considering the optimization problem on the whole domain of $W_{t+\Delta t}$, we restrict the optimization problem to a finite domain $\Phi_{t+\Delta t}$ and thus establish a local optimization problem. For solving the stochastic optimization problem at state (t, W_t), we focus on the value function $V_{t+\Delta t}(W_{t+\Delta t})$ on a finite interval $\Phi_{t+\Delta t}$.

BENEFIT FROM BUNDLING

By means of the constraints, the original problem in Equation (4.14) is simplified to a truncated problem in Equation (4.15). To solve this truncated problem, we first need to determine the value function $V_{t+\Delta t}(W_{t+\Delta t})$ on domain $\Phi_{t+\Delta t}$. A common simulation-based approach is that we vary x_t around x_t^{*ms} and perform sub-simulation, which is however involved and costly. One way to avoid sub-simulation is by plain Monte Carlo simulation combined with bundling. Using the bundling technique, the domain $\Phi_{t+\Delta t}$ can be approximated by:

$$\hat{\Phi}_{t+\Delta t} = \{W_{t+\Delta t} | W_{t+\Delta t} = \hat{W}_t \cdot (x_t^{*ms} \cdot R_t^e + R_f) + C \cdot \Delta t, \quad \hat{W}_t \in B_\delta\},$$

where $B_\delta = [W_t - \delta, W_t + \delta]$. So, instead of varying x_t, we vary W_t by considering the paths whose states are around (t, W_t). For more details about bundling, we refer the readers to [57] and early chapters of this thesis.

4.4.1. BACKWARD PROGRAMMING ALGORITHM

Now we formally describe the algorithm for the backward programming stage.

- **Step 1: Initiation**:
  Generate an initial guess of optimal asset allocations $\{\tilde{x}_t\}_{t=0}^{T-\Delta t}$ and simulate the paths of optimal wealth values $\{W_t(i)\}_{i=1}^{N}, t = 0, \ldots, T$. At the terminal time T, we have the determined value function $V_T(W_T)$.
  The following three steps are subsequently performed, recursively, backward in time, at $t = T - \Delta t, \ldots, \Delta t, 0$.

- **Step 2: Solution**
  Bundle paths into B partitions, where each bundle contains a similar number of paths and the paths inside a bundle have similar values at time t. Denote the wealth values associated to the paths in the bundle by $\{W_t^b(i)\}_{i=1}^{N_B}$, where N_B is the number of paths in the bundle. Within each bundle, we perform the following procedure.

  - For paths in the bundle, we have the corresponding wealth values $\{W_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ and the continuation values $\{V_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ at time $t+\Delta t$. So, a function $f_{t+\Delta t}^b(\cdot)$, which satisfies $V_{t+\Delta t}^b = f_{t+\Delta t}^b(W_{t+\Delta t}^b)$ on the local domain, can be determined by *regression*.

– For all paths in the bundle, since the value function $f^b_{t+\Delta t}(W^b_{t+\Delta t})$ has been approximated, we solve the optimization problem by calculating the first-order conditions. In this way, we get new asset allocations $\{\hat{x}^b_t(i)\}_{i=1}^{N_B}$.

– Since the wealth values $\{W^b_t(i)\}_{i=1}^{N_B}$ and the allocations $\{\hat{x}^b_t(i)\}_{i=1}^{N_B}$ are known, by *regression* we can also compute the new continuation values $\{\hat{V}^b_t(i)\}_{i=1}^{N_B}$. Here $\hat{V}^b_t(i)$ is the expectation of $V_{t+\Delta t}(W_{t+\Delta t})$ conditional on $W^b_t(i)$ and $\hat{x}^b_t(i)$, that is,

$$\hat{V}^b_t(i) = \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t})|W_t = W^b_t(i), x_t = \hat{x}^b_t(i)].$$

- **Step 3: Update**
  For the paths in a bundle, since we have an old guess $\{\tilde{x}^b_t\}_{i=1}^{N_B}$ for the asset allocations, by *regression* we can also calculate the old continuation values $\{\tilde{V}^b_t(i)\}_{i=1}^{N_B}$. For the i-th path, if $\tilde{V}^b_t(i) > \hat{V}^b_t(i)$, we choose $\hat{x}^b_t(i)$ as the updated allocation. Otherwise we retain the initial allocation. We denote the updated allocations by $\{x^b_t(i)\}_{i=1}^{N_B}$.

- **Step 4: Evolve**
  Once the updated allocations $\{x^b_t(i)\}_{i=1}^{N_B}$ are obtained, again by *regression* we can calculate the "updated" continuation values $\{V^b_t(i)\}_{i=1}^{N_B}$ and proceed with the backward recursion.

In the algorithm, regression refers to the technique of approximating the target function by a truncated basis function expansion, where the expansion coefficients are determined by minimizing the approximation error in the least squares sense. At each time step and inside each bundle, four regression steps are performed. Especially, the last three regression steps are added for calculating value functions. Since the value function is used to evolve information between time steps, an error in calculating them will accumulate due to recursion. In general, we can settle this problem by using a very large number of simulations, which is however expensive. In our numerical approach we always use the "Regress-Later" technique as applied in Chapters 2 and 3.

When we use the regression, polynomials up to order two are considered as basis functions. For the unconstrained problem, the value function is a quadratic function, so this choice of basis functions is sufficient. In the constrained case, although the value function is non-smooth, it is still piecewise quadratic as stated in [40]. Since the regression is performed with respect to paths in the same bundle, it is a local regression, i.e. local polynomial fitting. In all, our algorithm, which adopts second order local polynomial fitting to approximate piecewise quadratic functions, should yield satisfactory results.

After one iteration of the algorithm, we will obtain an "updated" asset allocation at each time step. The algorithm can be performed iteratively. In the remaining part of this section, we will prove that these iterations will lead to a convergent result.

Remark 4.4.1. *In the engineering field, this type of dynamic programming is called differential dynamic programming. For more details, we refer the reader to [56] and [80].*

4.4.2. CONVERGENCE OF THE BACKWARD RECURSIVE PROGRAMMING

For the dynamic backward recursive programming, we have

$$V_t(W_t) = \min_{x_t} \left\{ \mathbb{E}[V_{t+\Delta t}(W_t \cdot (x_t R_t^e + R_f) + C \cdot \Delta t) | W_t] \right\}, \quad t = T - \Delta t, \ldots, 0,$$

where the expectation is taken over the return R_t^e. This recursion can be written as:

$$V_t = \Psi_t V_{t+\Delta t}, \quad t = T - \Delta t, \ldots, 0,$$

where Ψ_t is the Bellman operator, defined as

$$(\Psi_t F)(W_t) = \min_{x_t} \left\{ \mathbb{E}[F(W_t \cdot (x_t R_t^e + R_f) + C \cdot t) | W_t] \right\}, \tag{4.16}$$

for any function $F(\cdot)$.

According to [6], the Bellman operator has the monotonicity property, which is described by the following lemma.

Lemma 4.4.2 (Monotonicity). *For any $F_1 : \mathscr{R} \to \mathscr{R}$ and $F_2 : \mathscr{R} \to \mathscr{R}$,*

$$F_1 \leq F_2 \Rightarrow \Psi_t F_1 \leq \Psi_t F_2,$$

where the inequalities are interpreted pointwise, and Ψ_t is as in (4.16).

We can prove the following proposition:

Proposition 4.4.3. *The backward recursive updating process, as explained in Section 4.4.1, converges. That is: there exists a function $V_0^*(\cdot)$ satisfying*

$$V_0^*(W_0) = \lim_{k \to +\infty} V_0^{(k)}(W_0),$$

where $V_0^{(k)}(W_0)$ denotes the value function at initial state after k iterations of the algorithm in Section 4.4.1.

Proof. The proof is directly based on the design of the backward recursive programming algorithm and the Monotonicity Lemma 4.4.2. Since at each iteration, we compare the previous and the new allocations and retain the one which generates a smaller value function and since the Bellman operator preserves monotonicity, we get:

$$V_0^{(1)}(W_0) \geq V_0^{(2)}(W_0) \geq \cdots \geq V_0^{(k)}(W_0), k = 1, \ldots, k_{\max}.$$

Since the value function is positive, according to the Monotone Convergence Theorem, we can finalize the proof. □

Whereas the convergence can be proved, the convergence rate is not determined. The convergence rate depends on the smoothness of the value functions and the initial guess of the allocations. For example, for a smooth quadratic value function, any non-zero initial guess should give the correct result after one iteration. On the other hand, if we choose zero as the initial guess for all the asset allocations, the solution can never

depart from the stationary point, which is generated by the risk-free investment strategy. In this case, the value function cannot be accurately approximated via the regression technique and thus the backward recursive programming algorithm proposed cannot proceed. In our numerical tests, however, we always achieve satisfactory results using the initial guess of the asset allocations generated by the multi-stage strategy.

On the other hand, the solution of our backward recursive approach is not guaranteed to be the optimal one. We call this algorithm "sub-optimal" because a bias is introduced when we approximate the non-smooth value function by piecewise quadratic polynomials. Although the numerical tests indicate that our algorithm generates highly satisfactory solutions, we expect that when the value function is highly non-smooth the approximation bias in our algorithm will be large and the accuracy of our algorithm may be unsatisfactory. As reported in [87], smooth approximations to a non-smooth value function can be very unreliable and thus a typical *policy iteration* technique may fail to converge. However, since the update step is introduced in our algorithm, we only update value functions and controls when they yield better results. This update step makes our algorithm a bit different from a common policy iteration algorithm but it guarantees that our algorithm will always generate convergent results.

4.5. CONSTRAINTS ON THE ASSET ALLOCATIONS

From the perspective of real-life applications, introducing constraints on the asset allocations is important. For example, when an investor goes bankrupt, she should not be allowed to manage her portfolio any more. Besides, according to regulations for banks, the leverage ratio should be bounded. In this section, we will formulate controls on the asset allocations.

4.5.1. NO BANKRUPTCY CONSTRAINT

In this chapter, the "no bankruptcy" constraint implies that there is zero probability of bankruptcy when the constraint is cast on the asset allocations. To ensure that the allocation at time t does not lead to bankruptcy at time $t+\Delta t$, we need:

$$W_t \cdot (x_t R_t^e + R_f) + C\Delta t \geq 0. \tag{4.17}$$

Note that if there is a "no bankruptcy" constraint, definitely we have $W_t > 0$. Therefore, to guarantee that Equation (4.17) is valid, we need:

$$x_t R_t^e \geq -R_f - \frac{C\Delta t}{W_t}. \tag{4.18}$$

Since $R_t^e = \exp(r_t^e) - R_f$, where r_e is a a random variable, we require

$$-R_f \geq -\frac{R_f}{x_t} - \frac{C\Delta t}{W_t \cdot x_t} \quad \text{and} \quad x_t \geq 0.$$

to ensure Equation (4.18) to be valid. Reformulating these equations, we have:

$$0 \leq x_t \leq 1 + \frac{C\Delta t}{W_t \cdot R_f}. \tag{4.19}$$

The "no bankruptcy" constraint given by Equation (4.19) implies $\lim_{W_t \to 0}(x_t \cdot W_t) = 0$, as given in [88, 89] for the continuous portfolio optimization problem. Our version of the "no bankruptcy" constraint for the multi-period case is stronger than the constraint in [88, 89]. Rather than specifying that wealth should not be invested in the risky asset when the total wealth amount is close to zero, our constraint also indicates that special consideration to the asset allocation should be given even though the wealth amount is far above zero. This is due to the difference between continuous and discrete re-balancing. In the latter case, since we cannot manage the portfolio between two re-balancing opportunities, it is possible that the investor goes bankrupt after an extreme market movement. To avoid this situation, we need to impose more strict constraints on the asset allocations, which is however not necessary for a continuous re-balancing problem.

4.5.2. NO BANKRUPTCY CONSTRAINT WITH $1-2\zeta\%$ CERTAINTY

When the wealth in a portfolio is large, according to the discussion in the last subsection, the upper bound for "no bankruptcy" constraint (4.19) will be close to 1, which is quite rigorous. Using this as the upper bound protects an investor from bankruptcy except in the rare case that the risky asset yields zero return. A possible way to relieve this constraint is to take the possibility of bankruptcy into account.

Assume that the ζ- and the $(1-\zeta)$- quantiles of the excess return R_t^e are respectively $R_t^{e,\zeta}$ and $R_t^{e,1-\zeta}$. Then with certainty $1-2\zeta\%$ and the constraints shown below, we can guarantee that the "no bankruptcy" constraint in Equation (4.17) is valid. The bounds for the asset allocations can be computed as:

$$\frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,1-\zeta}} \leq x_t \leq \frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,\zeta}}. \tag{4.20}$$

Remark 4.5.1. *The constraints proposed by Equation (4.20) correspond to discrete monitoring at re-balancing times. If an investor is restricted by the "no-trading if insolvent" condition, under which she has to liquidate all assets if involvent, continuous monitoring of the portfolio is required.*

4.5.3. BOUNDED LEVERAGE

Equations (4.19) and (4.20) imply that when an investor's wealth is close to zero, the upper bound for the allocation in the risky asset goes to infinity. The "no bankruptcy" constraint does not forbid an investor from gambling when she is almost bankrupt. To avoid this, we can impose constraints on the leverage ratios, for example, by restricting the proportion of investor's wealth in the risky asset to be within $[x_{\min}, x_{\max}]$.

4.6. NUMERICAL EXPERIMENTS

In this section, we test our algorithms by solving several multi-period mean-variance portfolio management problems. We start with a simple case with one risky asset and one risk-free asset in the portfolio. We choose geometric Brownian motion as the dynamics of the risky asset, and assume that the log-return of the risky asset has volatility σ and mean $r_f + \xi \cdot \sigma$. Here r_f is the log-return of the risk-free asset and ξ the market price of risk. Since only the return of the risky asset is stochastic, we call this problem a "1D

Table 4.1: Parameter settings used in the test.

Set I (From [88]):
$r_f = 0.03$, $\xi = 0.33$, $\sigma = 0.15$, $C = 0.1$, $T = 20$(years), $W_0 = 1$, $M^* = 80$.
Set II (From [10]):
$r_f = 0.06$, $\xi = 0.4$, $\sigma = 0.15$, $C = 0$, $T = 1$(year), $W_0 = 1$.
Set III (From [39]):
$r_f = 0.04$, $\xi = 0.4$, $\sigma = 0.15$, $C = 0$, $T = 30$(years), $W_0 = 100$, $M = 30$.

* M denotes the number of early exercise opportunities, which are equidistantly distributed in T years.

problem". We will also consider a "2D problem" with two risky assets and one risk-free asset in the portfolio. We consider only bounded leverage constraints in the 2D problem, and therefore it can be solved highly efficiently.

In the numerical tests, we choose the sample size to be 50000 and the number of bundles to be 20 in the backward recursive programming stage. When we employ the "no bankruptcy" constraint (4.20), we choose parameter ζ to be sufficiently small, $\zeta = 10^{-8}$, which ensures that the undesired event will not happen. Regarding the backward recursive programming, we use a common random seed to generate Monte Carlo paths for one run of the algorithm (including one forward and several backward processes). To ensure that the choice of random seed does not introduce a bias, we consider 20 different random seeds in all tests.

Based on reference results, we choose three different sets of parameters, shown in Table 4.1.

4.6.1. 1D PROBLEM

MULTI-STAGE OPTIMIZATION WITH CONSTRAINTS

We first test the plain multi-stage strategy using the model parameters from Set I in Table 4.1. Different types of constraints are prescribed on the asset allocations. The bounded control is chosen to be $[0, 1.5]$. Figure 4.1 shows that constraints on the controls have a significant influence on the efficient frontiers obtained by the multi-stage strategy.

Without periodic contribution, i.e. $C = 0$, an analytic solution to the "no bankruptcy" case with continuous re-balancing is presented in [10]. For a corresponding test, we choose the model parameters from Set II in Table 4.1. In Figure 4.2, we can see that when the number of re-balancing opportunities is sufficiently large, the efficient frontier generated by the multi-stage strategy is close to the analytic solution, especially when the investment risk is not large.

BACKWARD RECURSIVE PROGRAMMING

In this section, we show the performance of the backward recursive programming stage. Unless indicated differently, the initial asset allocations are generated by the multi-stage strategy. We first consider a continuous re-balancing scenario shown in Figure 4.2, where an analytic solution is available. We observe that when backward recursive programming is implemented, we obtain a better efficient frontier than the one generated by the

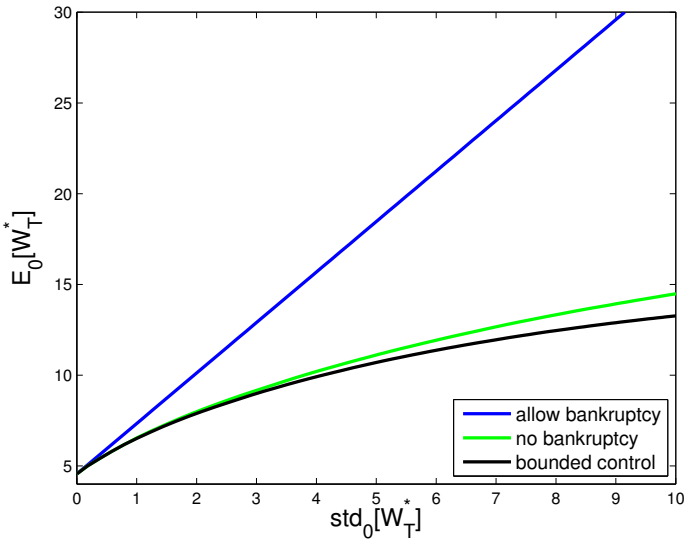
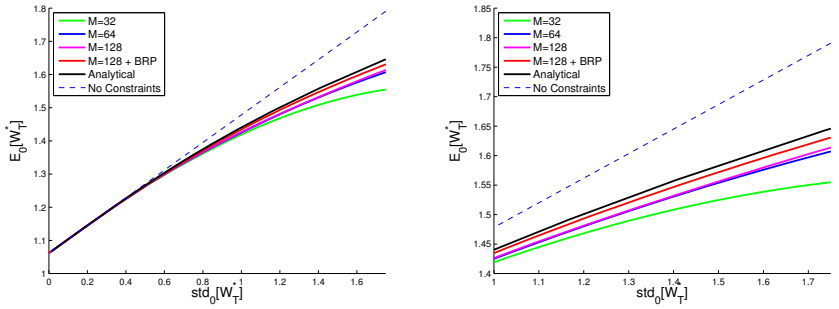


Figure 4.1: Different types of constraints on the management strategies. By varying the target $\gamma \in (9.125, 85.125)$, we trace out the efficient frontiers.



(a) Efficient frontier with no bankruptcy constraint

(b) Efficient frontier with no bankruptcy constraint (zoom in)

Figure 4.2: The "no bankruptcy" case. An analytic solution is available if continuous re-balancing is applied. We check the efficient frontier generated by our algorithm. For backward recursive programming, we use the results obtained after four backward iterations when $M = 128$.

multi-stage strategy. Moreover, we find that the efficient frontier resulting from backward recursive programming is close to the analytic solution.

Subsequently, we perform tests for a multi-period portfolio management case presented in [39] with the parameters from Set III in Table 4.1. We consider bounded constraints $[0, 1.5]$ on the asset allocations.

In Figure 4.3, we present the efficient frontiers generated by the multi-stage approach and by the first four iterations in the backward recursive programming stage. Again we can see that the backward recursive approach generates better results than the multi-stage approach. Even with one iteration of backward programming, the result is already highly satisfactory.

In Table 4.2, we consider two cases for which also reference results are available. We find that in general, even with the same terminal target, the mean-variance pair calculated by the multi-stage strategy is different from the reference value. However, after the backward recursive programming, the result is highly satisfactory. In fact, the backward programming even provides results somewhat superior to the reference values[4]. Moreover, the multi-stage strategy and the backward recursive programming stage cost only seconds.

Table 4.2: Comparison of the results generated by the multi-stage strategy and the backward recursive programming to the reference values in [39].

Target	$\gamma = 1751.94$			$\gamma = 5856.15$		
	$E_0^*(W_T)$ (s.e.)	$\text{Std}_0^*(W_T)$ (s.e.)	CPU time (in seconds)	$E_0^*(W_T)$ (s.e.)	$\text{Std}_0^*(W_T)$ (s.e.)	CPU time (in seconds)
Reference	816.62	142.85		2008.55	969.33	
Multi-stage	823.84	154.37	4.71	2031.65	987.55	4.17
	(0.71)	(1.28)		(4.86)	(2.54)	
One backward iteration	818.83	143.33	9.09	2018.47	969.29	8.23
	(0.70)	(1.30)		(4.73)	(2.58)	
Four backward iterations	817.74	141.40	22.25	2014.90	964.80	20.34
	(0.70)	(1.28)		(4.73)	(2.62)	

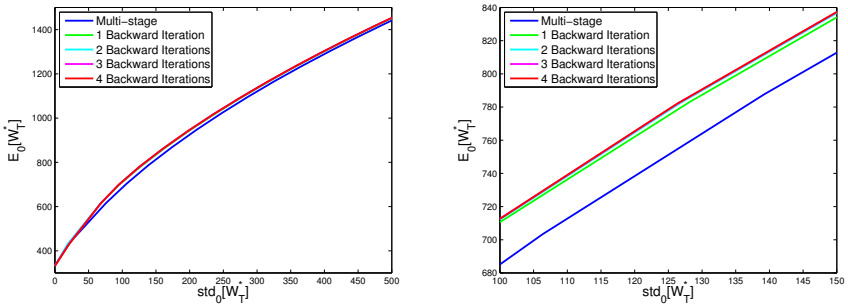
The backward recursive programming technique is robust regarding different choices of initial allocations. As shown in Figure 4.4, even though we fix the initial asset allocations to be constant[5], the backward recursive programming stage still gives us a satisfactory result after some iterations.

4.6.2. 2D PROBLEM WITH BOX CONSTRAINTS

To tackle dynamic portfolio management problems with either the forward or the backward strategy proposed, we essentially need to deal with a constrained convex optimization problem. Some fast numerical solvers exist for this kind of problems in high dimensional scenarios. In this section, we will consider a simple 2D case where box constraints are cast on the asset allocations. In this case, bounded controls are prescribed for the

[4] For generating the efficient frontiers, we implement Monte Carlo simulation while the authors of [39] utilize a PDE approach. As explained in [66], the simulation based approach usually yields slightly better results.

[5] In general this will lead to quite a rough initial approximation of the mean-variance pair.



(a) Efficient frontier with bounded control

(b) Efficient frontier with bounded control (zoom in)

Figure 4.3: Backward recursive calculation using the allocations generated by the multi-stage strategy as guesses for the optimal asset allocations.

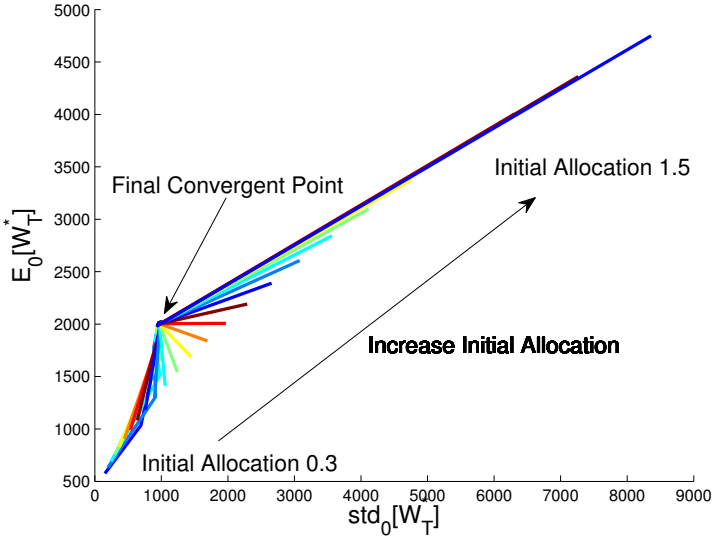


Figure 4.4: A constant initial guess for the asset allocation. The constant varies from 0.3 to 1.5. Generally, this choice will lead to an inaccurate estimate of the mean-variance pair. However, after several (in our tests, at most four) iterations of backward programming, we achieve highly satisfactory results.

allocations of both risky assets. Solving this constrained 2D convex optimization problem is therefore equivalent to solving five simple optimization problems and choosing the best results from them. The reason is that for a quadratic optimization problem with box constraints the optimal solution lies either at the boundary or in the interior of the admissible set. These five optimization problems include one unconstrained 2D problem[6] and four 1D problems with bounded constraints.

We use the parameters displayed in Set III from Table 4.1 and for the other risky asset we use the same market price of risk but a higher volatility, $\sigma_b = 0.4$. The correlation ρ between these two risky assets is fixed at $\rho = 0.4$, unless mentioned otherwise. For both assets, we prescribe bounded constraints $[0, 0.75]$ on their asset allocations.

First, we test the influence of adding another risky asset to the portfolio. Here we simply implement the multi-stage strategy for generating the mean-variance efficient frontier. As shown in Figure 4.5, increasing diversification in the portfolio has a significant impact on the solution of the dynamic optimization problem. When the correlation between the two risky assets is close to -1, an optimal efficient frontier can be obtained. This is intuitive, because a large part of the volatility can be hedged in the case of two negatively correlated risky assets. When their correlation gets larger, the efficient frontier gets worse. However, in most cases two risky assets in the portfolio yield better results than having one risky asset in the portfolio. For example, when we choose the correlation to be 0.4 and the final target to be 5856.15, as used in Section 4.6.1, we obtain $[\mathbb{E}_0[W_T^*], \text{Std}_0[W_T^*]] = [2501.41, 893.87]$ which is significantly better than $[\mathbb{E}_0[W_T^*], \text{Std}_0[W_T^*]] = [2031.65, 987.55]$ as acquired in the 1D case.

In Figure 4.6, we compare the multi-stage strategy and the backward recursive programming approach. The outcome is similar to that observed in the 1D tests. When we implement the backward recursive programming stage, a significant improvement is obtained. For example, when the standard deviation is around 200, an almost 10% higher expected return can be obtained if we consider the backward recursive programming approach rather than the multi-stage approach.

Remark 4.6.1. *In the 2D case, we also observe that satisfactory results can be obtained after several iterations of backward recursive programming even if we start with an inaccurate initial guess of the asset allocations.*

4.7. CONCLUSION

In this chapter, we proposed simulation-based approaches for solving the dynamic mean-variance portfolio management problem. To deal with the nonlinearity of the conditional variance, we used the embedding technique introduced in [63] to transform the mean-variance optimization problem into a linear-quadratic problem, which has a determined final optimization target. To tackle this target-based dynamic optimization problem, also known as "the pre-commitment problem", we proposed two approaches, one in a forward fashion and the other in a backward fashion.

The forward approach, called the "multi-stage strategy", is based on determining an intermediate investment target at each re-balancing time. The intermediate target is

[6] First, we solve the unconstrained 2D problem. Then we penalize the optimal solution when the constraints are not satisfied.

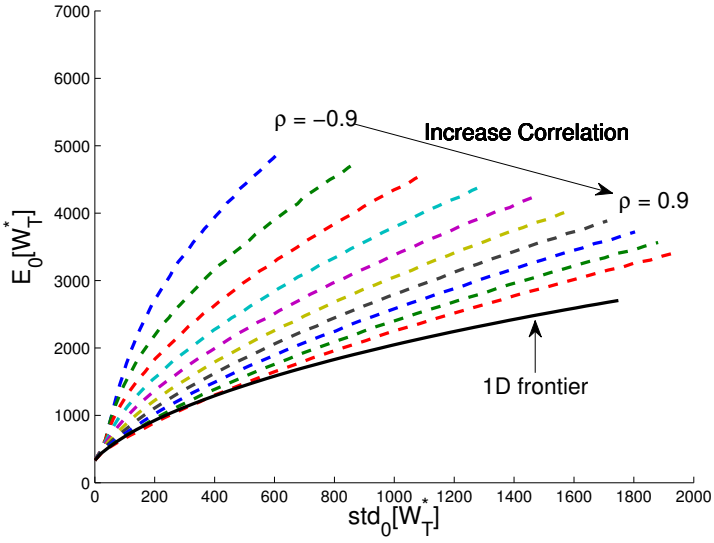
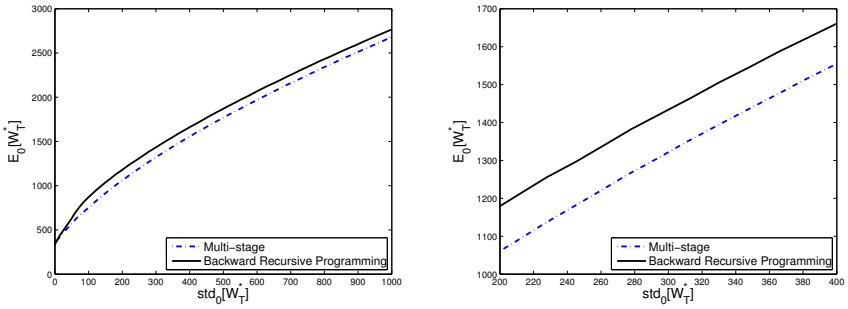


Figure 4.5: Efficient frontiers obtained by investing in two risky assets with different correlations. In each scenario, the two risky assets share the same market price of risk, but one with high volatility and the other with low volatility. Bounded control $[0, 0.75]$ is cast on both assets. For the 1D test case, we employ the risky asset with low volatility in the portfolio and bounded control $[0, 1.5]$.



(a) 2D efficient frontier (b) 2D efficient frontier (zoom in)

Figure 4.6: We compare forward and backward strategies for the 2D dynamic portfolio management problem with box constraints. When considering the backward recursive programming, we use the results obtained after four iterations.

chosen as the amount of wealth, which, if obtained, an investor can invest with a risk-free strategy and still reach the final investment target. Although it is generally believed that backward programming is essential for solving a dynamic optimization problem, we proved that the multi-stage strategy yields optimal controls when no constraints are involved. In the case that there are constraints on the controls, the multi-stage strategy can only yield a sub-optimal solution. However, since it is a forward and thus highly efficient approach, it is always feasible even when the dimensionality of the problem increases.

Although the forward approach is fast and easy-to-implement, in general it is not optimal for a dynamic optimization problem. Therefore, we proposed another simulation-based approach which involves backward recursive programming. The main idea of the backward recursive approach is that we consider local quadratic optimization instead of global optimization. By tailoring the numerical algorithm, the backward recursive programming is guaranteed to yield convergent results after several iterations. In the numerical tests, it was shown that, although the backward approach is also sub-optimal, it always generated better efficient frontiers than the multi-stage strategy.

In the backward approach, we need to calculate conditional expectations associated to each simulated path recursively by least-squares regression. To make this regression-based numerical approach stable, we implemented the "bundling" and the "regress-later" techniques. We found that our backward recursive approach is very robust. Even if it is initiated by an inaccurate guess for the allocation, highly satisfactory results can be obtained after several iterations.

CHAPTER 5

Multi-period Mean-Variance Portfolio Optimization with Time-Consistency Constraint

In this chapter, we establish a link between a time-consistent and a pre-commitment investment strategy. We define an implied investment target, which is implicitly contained in a time-consistent strategy at a given time step and wealth level. By imposing the implied investment target at the initial time step on a time-consistent strategy, we form a hybrid strategy which may generate better mean-variance efficient frontiers than the time-consistent strategy. We extend the numerical algorithm proposed in the previous chapter to solve constrained time-consistent mean-variance optimization problems. Since the time-consistent and the pre-commitment strategies generate different terminal wealth distributions, time-consistency is not always inferior to pre-commitment.

Keywords: Finance · Investment analysis · Decision analysis · Simulation · Time-consistency

5.1. INTRODUCTION

In this chapter, we consider dynamic mean-variance optimization problems. In general the Bellman dynamic programming principle [5] should be applied to this kind of path-dependent optimization problem. Due to the nonlinearity of the variance operator, however, the mean-variance problem cannot be solved in this manner as explained in Chapter 4.

In [91] and [63], the authors introduced an embedding technique, by which the original mean-variance problem was formulated as a tractable linear-quadratic (LQ) problem. Instead of pursuing an optimal balance between profit and risk, an investor then designs an investment strategy to minimize the difference between her wealth and a predetermined investment target. In [4], the optimal strategy for the LQ problem is termed *pre-commitment* strategy. Related research on the pre-commitment mean-variance portfolio optimization problem has been performed in [64], [92], [10], [40] and [28], where realistic constraints were considered in the dynamic optimization process. Moreover, since target-based features of the pre-commitment strategy fit well to pension plan investment, some discussion regarding this strategy arose in the insurance management literature as well, see [52], [44] and [86].

However, as noticed in [4], after an initial time step, a real world investor will be on only one of many possible stochastic paths and an investor with a determined risk

This chapter is based on the article 'On pre-commitment aspects of a time-consistent strategy for a mean-variance investor', published in *Journal of Economic Dynamics and Control*, forthcoming, 2016 [27].

aversion attitude may then have an incentive to depart from the pre-commitment strategy. The authors in [4] emphasize that an optimal dynamic strategy should be a *time-consistent* strategy, i.e. given an optimal investment policy for the complete mean-variance problem, a sub-policy should also constitute the optimal policy for the corresponding sub-problem. As spotted in [89], [90] and [21], introducing the time-consistency condition requires casting additional constraints on the controls and the mean-variance efficient frontier generated by a time-consistent strategy is thus usually lower than the frontier generated by a pre-commitment strategy. In the case of discrete and continuous re-balancing, the author in [32] developed a time-consistent formulation of mean-variance efficiency in a semimartingale framework. Extensions of the time-consistent mean-variance strategy have been made by modeling the risk aversion parameter as a state-dependent variable, see [54], [30], [11] and [31]. The strategy of a dynamic mean-variance investor is then defined to be *time-consistent in efficiency*, which does not necessarily meet the time-consistency conditions but may yield higher mean-variance efficient frontiers. Despite some discussion on the pre-commitment and the time-consistent mean-variance strategies in the operational research literature, the link between these strategies is mainly at the stage of problem formulation: a time-consistent strategy can be obtained by casting a time-consistency constraint on a pre-commitment strategy.

In this chapter, the point-of-departure is to focus on the relation between the two strategies. Since for the pre-commitment as well as for the time-consistent strategy the optimal allocations for a mean-variance investor are available, we can establish a link between the former, which is driven by a target, and the latter, driven by a risk aversion parameter. In the case where a pre-commitment investor adopts the same asset allocations as a time-consistent investor, a one-to-one function between the target and the risk aversion parameter can be established. This function gives information on how the risk aversion attitude of a pre-commitment investor changes and how much wealth a time-consistent investor desires during the dynamic investment process. We call the target for the time-consistent investor *the implied target*.

During an investment period, the implied target of a time-consistent investor varies while the target of a pre-commitment investor remains fixed. This makes the distributions of their potential terminal wealth significantly different. Reflecting on this, we define a hybrid strategy by introducing a fixed target into the time-consistent strategy. This hybrid mean-variance strategy is dual to a mean-variance strategy which minimizes the shortfall to a target. Introduction of a target into the mean-variance formulation essentially makes a strategy *semi-self-financing*, which suggests that the investor can withdraw money from the portfolio in some scenarios. Since a semi-self-financing strategy offers a broader set of admissible controls than a self-financing strategy, as explained in [29] and [34], introducing semi-self-financing into a pre-commitment strategy can improve the mean-variance efficient frontiers.

In principle, any target higher than the risk-free target could be a reasonable target for the hybrid strategy. However, if the target is too high, a hybrid investor will behave like the time-consistent investor. In this chapter, we choose the target to be equal to the *implied target at the initial time step* of a time-consistent investor. With this setting, we find that the mean-variance frontier generated by the hybrid strategy is higher than the one generated by the time-consistent strategy. In the constrained case, the mean-variance

frontier generated by the hybrid strategy is often almost identical to the one generated by the pre-commitment strategy. In this case the hybrid investor behaves more like a pre-commitment than a time-consistent investor.

Since different strategies may generate different terminal wealth distributions, the mean-variance efficient frontier may not tell us all about the performance of a strategy. When considering *partial variance*, for example, we find that the time-consistent strategy is not always inferior to the pre-commitment strategy. If we only consider downside risk, the time-consistent strategy and the pre-commitment strategy perform comparably. Moreover, in terms of the potential to achieve higher returns, the time-consistent strategy may perform even better.

In the previous chapter, we proposed an efficient simulation-based algorithm for solving the pre-commitment mean-variance problem with constraints on the controls. In the unconstrained case, the optimal pre-commitment strategy was shown to be identical to the strategy taken by a "myopic investor", who designs her optimal strategy without taking future optimality into account and therefore simply utilizes a forward strategy. In this chapter, we will extend the numerical algorithm, based on *a forward followed by a backward iteration* proposed in the previous chapter, to the time-consistent problem. Solving the constrained time-consistent mean-variance problem with a simulation-based approach has, to our knowledge, not yet been performed previously. The authors in [4] handle the variance operator by writing it as the difference of two variables being the mean of the square of the variable and the square of the variable's mean. With such modifications in our algorithm, highly satisfactory results for the constrained time-consistent problem can again be obtained. In the unconstrained case the optimal strategy of a time-consistent investor can be duplicated by a myopic investor as well.

This chapter is organized as follows. In Section 5.2, we describe the dynamic mean-variance problem, the pre-commitment and the time-consistent strategies. Section 5.3 has its focus on the time-consistent strategy. We derive an analytic formula for the time-consistent optimal allocation in Section 5.3.1. Reflecting on this, we compare the time-consistent and the pre-commitment strategies in Section 5.3.2. The hybrid strategy, which is constructed by inserting a target in the time-consistent strategy, is introduced in Section 5.3.3. Section 5.3.4 introduces the partial variance for evaluating the mean-variance strategies. Our numerical algorithm is presented in Section 5.4 and following that in Section 5.5 numerical tests are performed. We conclude in the last section.

5.2. PROBLEM FORMULATION

We again consider the multi-period mean-variance optimization problem as formed in Chapter 4. Therefore, we still use the notations as defined in Section 4.2.

In Chapter 4, the original mean-variance problem was embedded into a tractable auxiliary linear-quadratic (LQ) problem. The investment strategy corresponding to this LQ problem is called the pre-commitment strategy. A pre-commitment investor always pursues a pre-determined target and thus invests according to the pre-determined strategy. However, this also implies that the investor does not stay with the initial trade-off parameter λ.

If an investor has a *time-consistent risk aversion*, then the investment strategy is termed the time-consistent strategy as in [4]. Mathematically, the time-consistent strat-

egy can be defined as follows:

Definition 5.2.1 (Time-consistent strategy)**.** *At time t, the time-consistent optimal control $\{x_s^{*tc}\}_{s=t}^{T-\Delta t}$ is defined by the optimal control for Equation (4.1) with an additional time-consistency condition requiring that the subsets $\{x_s^{*tc}\}_{s=\tau}^{T-\Delta t}, \tau = t+\Delta t, t+2\cdot\Delta t, \ldots, T-\Delta t$, also constitute optimal controls for:*

$$J_\tau(W_\tau) = \max_{\{x_s\}_{s=\tau}^{T-\Delta t}} \Big[\mathbb{E}[W_T|W_\tau] - \lambda \cdot \mathrm{Var}[W_T|W_\tau]\Big], \tag{5.1}$$

for $\tau = t+\Delta t, t+2\cdot\Delta t, \ldots, T-\Delta t$.

Similar as in Chapter 4, various constraints can be cast on the asset allocations. For example, the no bankruptcy constraint, which requires that the asset allocation does not lead to bankruptcy of the investor, and the leverage constraint, which forms bounds on the asset allocation.

Remark 5.2.2. *Similar as in Chapter 4, we consider a no-bankruptcy constraint with $1-2\cdot\zeta\%$ certainty. At time t, for an investor with wealth W_t, the constraint on the asset allocation x_t reads:*

$$\frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,1-\zeta}} \le x_t \le \frac{-C\Delta t - W_t \cdot R_f}{W_t \cdot R_t^{e,\zeta}}, \tag{5.2}$$

$R_t^{e,\zeta}$ and $R_t^{e,1-\zeta}$ respectively denote the ζ- and the $(1-\zeta)$- quantiles of the excess return R_t^e.

When we consider the bounded leverage constraint, we set upper and lower bounds for the asset allocation. The bounded leverage constraint reads:

$$x_{\min} \le x_t \le x_{\max}.$$

Remark 5.2.3. *The authors of [11] introduced a state-dependent risk aversion parameter, which can be determined by the amount of wealth in a portfolio. Formally, the state-dependent risk aversion parameter reads:*

$$\Lambda(W_t) = \frac{\theta}{W_t},$$

where θ is a fixed constant.

The state-dependent risk aversion parameter implies that an investor will be less risk averse when she has a large amount of money and more risk averse when she is almost bankrupt. This may be reasonable for a real-world investor, but it also leads to some problems. For example, when an investor is close to bankruptcy, applying the risk-averse strategy indicates that it is hardly possible to get back to sufficient wealth. On the other hand, with a large amount of money, she is less risk-averse and willing to "gamble", which may lead to significant losses. To sum up, although this way of forming the risk aversion parameter seems reasonable, the corresponding investment strategy has "easy-to-lose" and "hard-to-recover" features. Therefore, as shown in [89] and confirmed in our numerical tests, this strategy will generate a lower efficient frontier than the plain time-consistent strategy.

Note that there is no guarantee of Pareto optimality when the risk aversion parameter is not a constant scalar. This explains why in the numerical tests we can get higher mean-variance frontiers even when more strict constraints are cast on the asset allocations of a time-consistent investor with state-dependent risk aversion.

In the following section, we will analytically derive the optimal unconstrained multi-period time-consistent strategy. Besides, reflecting on our findings in Chapter 4 about the pre-commitment strategy, we will make a thorough discussion comparing the time-consistent strategy and the pre-commitment one.

5.3. THE TIME-CONSISTENT STRATEGY

5.3.1. THE OPTIMAL ASSET ALLOCATION

Under the following condition, we can derive an analytic expression for the asset allocations in the time-consistent strategy.

Condition 5.3.1. *The asset allocations at each time step are unconstrained.*

The formulation of the asset allocations for the time-consistent strategy is described in the following proposition. Since it is not easy to derive the value function of the time-consistent strategy (because of the nonlinearity of the variance operator), our proof is based on the essential properties of an "optimal control".

Proposition 5.3.2. *Variable x_t^{*tc}, the optimal asset allocation for the time-consistent mean-variance optimization problem at time t, can be calculated by:*

$$x_t^{*tc} = \frac{\mathbb{E}[R_t^e]}{2\cdot\lambda\cdot W_t\cdot R_f^{(T-t)/\Delta t-1}\cdot \mathrm{Var}[R_t^e]}, \tag{5.3}$$

where $t = 0, \Delta t, \ldots, T-\Delta t$.

Proof. We consider the dynamic programming technique for solving the time-consistent problem. At time $T-\Delta t$, since the terminal condition at time T is known, the time-consistent strategy will certainly generate the asset allocation as shown in Equation (5.3). That is:

$$x_{T-\Delta t}^{*tc} = \frac{\mathbb{E}[R_{T-\Delta t}^e]}{2\cdot\lambda\cdot W_{T-\Delta t}\cdot \mathrm{Var}[R_{T-\Delta t}^e]}. \tag{5.4}$$

Subsequently, we consider the optimization problem at time $T-2\Delta t$, where

$$x_{T-2\Delta t}^{*tc} = \arg\max_{x_{T-2\Delta t}} \Big[\mathbb{E}[W_T|W_{T-2\Delta t}] - \lambda\cdot\mathrm{Var}[W_T|W_{T-2\Delta t}]\Big]. \tag{5.5}$$

Note that the wealth $W_{T-\Delta t}$ and W_T respectively have the form:

$$\begin{aligned} W_{T-\Delta t} &= W_{T-2\Delta t}\cdot(x_{T-2\Delta t}R_{T-2\Delta t}^e + R_f) + C\cdot\Delta t, \\ W_T &= W_{T-\Delta t}\cdot(x_{T-\Delta t}^{*tc}R_{T-\Delta t}^e + R_f) + C\cdot\Delta t. \end{aligned}$$

Allocation $x^{*tc}_{T-\Delta t}$ indicates that the optimal allocation will be chosen at time step $T-\Delta t$. The formulation for the terminal wealth W_T then reads:

$$\begin{aligned} W_T &= \Big(W_{T-2\Delta t}\cdot(x_{T-2\Delta t}R^e_{T-2\Delta t}+R_f)+C\cdot\Delta t\Big)\cdot x^{*tc}_{T-\Delta t}R^e_{T-\Delta t} \\ &+ \Big(W_{T-2\Delta t}\cdot(x_{T-2\Delta t}R^e_{T-2\Delta t}+R_f)+C\cdot\Delta t\Big)\cdot R_f + C\cdot\Delta t \\ &= \frac{\mathbb{E}[R^e_{T-1}]R^e_{T-1}}{2\cdot\lambda\cdot\mathrm{Var}[R^e_{T-1}]} + W_{T-2\Delta t}\cdot(x_{T-2\Delta t}R^e_{T-2\Delta t}+R_f)\cdot R_f + C\cdot\Delta t\cdot R_f + C\cdot\Delta t \\ &= K_{T-2\Delta t} + W_{T-2\Delta t}\cdot R^e_{T-2\Delta t}\cdot R_f\cdot x_{T-2\Delta t}, \end{aligned} \tag{5.6}$$

where the second equation is based on the property of the optimal control as shown in Equation (5.4): $x^{*tc}_{T-\Delta t}$ multiplied by $W_{T-\Delta t}$ will yield a constant. In the last line in Equation (5.6), we use $K_{T-2\Delta t}$ to denote all terms that are not related to $x_{T-2\Delta t}$. Since we perform optimization with respect to control variable $x_{T-2\Delta t}$ and returns at different time steps are independent, knowledge of the details of $K_{T-2\Delta t}$ is not needed when solving our optimization problem.

We insert Equation (5.6) into the mean-variance formulation and obtain:

$$\begin{aligned} &\mathbb{E}[W_T|W_{T-2\Delta t}]-\lambda\cdot\mathrm{Var}[W_T|W_{T-2\Delta t}] \\ =\ & K_{T-2\Delta t} + W_{T-2\Delta t}\cdot\mathbb{E}[R^e_{T-2\Delta t}]\cdot R_f\cdot x_{T-2\Delta t} - \lambda\cdot W^2_{T-2\Delta t}\cdot\mathrm{Var}[R^e_{T-2\Delta t}]\cdot(R_f)^2\cdot x^2_{T-2\Delta t}. \end{aligned}$$

Solving the first-order conditions with respect to $x_{T-2\Delta t}$ gives us:

$$x^{*tc}_{T-2\Delta t} = \frac{\mathbb{E}[R^e_{T-2\Delta t}]}{2\cdot\lambda\cdot W_{T-2\Delta t}\cdot R_f\cdot\mathrm{Var}[R^e_{T-2\Delta t}]}, \tag{5.7}$$

which is identical to the allocation shown in Equation (5.3), when $t=T-2\Delta t$.

The allocations for other time steps can be proved by backward induction in a similar fashion. □

The asset allocations for the time-consistent strategy shown in Equation (5.3) can be explained in a straightforward way. Consider a myopic investor who only manages the portfolio for the coming period and chooses a risk-free policy afterwards. Then, the optimal time-consistent strategy is identical to the strategy taken by this myopic investor. We will discuss this in some more detail in Section 5.4.1. Moreover, if we assume that the risky asset returns at different time steps have a stationary distribution, Equation (5.3) implies that a fixed constant amount of wealth should be invested in the risky asset at all time steps.

When the periodic contribution C is not included in the portfolio, a general solution for an unconstrained time-consistent multi-period portfolio management problem has been established in the literature, for example in Proposition 5 in [4] and in Lemma 3.2 in [32]. However, we notice that introducing periodic contributions does not affect the optimal asset allocations of a time-consistent investor. Meanwhile, as discussed in the previous chapter, a pre-commitment investor takes the contribution amount into account when the investment strategy is determined. Knowledge of the periodic contribution is essential for the pre-commitment strategy.

In the remaining analysis part of this chapter, we restrict ourselves to the situation where there is no periodic contribution, i.e. $C = 0$. However, extending the analysis to the case $C \neq 0$ is possible.

Remark 5.3.3. *As shown in Equation (5.3), the optimal asset allocation in a time-consistent strategy depends on the first and second moments of the excess return rather than on the complete distribution. However, when we use these asset allocations to generate the mean-variance pairs, the distribution of the excess return has an impact. The same holds for the pre-commitment strategy, see [35] for related discussion.*

Remark 5.3.4. *As reported in [90], in the unconstrained case, the time-consistent mean-variance solution is identical to a mean-quadratic-variation optimization, whose objective function reads:*

$$\max_{\{x_s\}_{s=t}^{T-\Delta t}} \mathbb{E}\Big[W_T - \lambda \int_t^T (e^{r(T-s)} \mathrm{d}W_s)^2 \Big| W_t\Big].$$

In other words, a time-consistent investor implicitly cares about the average volatility of the wealth in the whole investment period.

5

5.3.2. PRE-COMMITMENT VERSUS TIME-CONSISTENT STRATEGY

By combining our findings in Section 5.3.1 with those in Section 4.3.1, we can establish a link between the pre-commitment and the time-consistent strategy for the dynamic mean-variance optimization problem. In fact, the pre-commitment strategy can be seen as a strategy consistent with an investment target but not with a risk aversion attitude, while the time-consistent strategy appears as a strategy consistent with a risk aversion attitude but not with an investment target.

According to our discussions in the last chapter, if a pre-commitment investor has an investment target γ, the optimal control at state (t, W_t) in the unconstrained case is equal to:

$$x_t^{*pc} = \frac{\big(\gamma - W_t \cdot (R_f)^{(T-t)/\Delta t}\big)\mathbb{E}[R_t^e]}{W_t \cdot (R_f)^{(T-t)/\Delta t - 1}\mathbb{E}[(R_t^e)^2]}, \tag{5.8}$$

where the target γ uniquely determines the asset allocations when other parameters are fixed and vice versa. When the periodic contribution is taken into account, the unconstrained asset allocation for the pre-commitment strategy reads:

$$x_t^{*pc} = \frac{\left(\gamma - W_t \cdot (R_f)^{(T-t)/\Delta t} - C \cdot \Delta t \cdot \frac{1-R_f^{(T-t)/\Delta t}}{1-R_f}\right)\mathbb{E}[R_t^e]}{W_t \cdot (R_f)^{(T-t)/\Delta t - 1}\mathbb{E}[(R_t^e)^2]},$$

where the periodic contribution obviously has an impact on the pre-commitment strategy.

Suppose that a time-consistent investor has risk aversion λ, at state (t, W_t) the time-consistent optimal asset allocation then reads:

$$x_t^{*tc} = \frac{\mathbb{E}[R_t^e]}{2 \cdot \lambda \cdot W_t \cdot R_f^{(T-t)/\Delta t - 1} \cdot \mathrm{Var}[R_t^e]},$$

where the asset allocation is also uniquely determined by risk aversion parameter λ.

At state (t, W_t), if a pre-commitment investor has the same optimal allocation as x_t^{*tc}, because the allocation x_t^{*pc} and the investment target γ form a one-to-one function, as shown in Equation (5.8), we can calculate an *implied investment target* γ_t^{imp} by solving:

$$\frac{\left(\gamma_t^{\text{imp}} - W_t \cdot (R_f)^{(T-t)/\Delta t}\right)\mathbb{E}[R_t^e]}{W_t \cdot (R_f)^{(T-t)/\Delta t - 1}\mathbb{E}[(R_t^e)^2]} = \frac{\mathbb{E}[R_t^e]}{2 \cdot \lambda \cdot W_t \cdot R_f^{(T-t)/\Delta t - 1} \cdot \text{Var}[R_t^e]}.$$

This gives us:

$$\gamma_t^{\text{imp}} = W_t R_f^{(T-t)/\Delta t} + \frac{\mathbb{E}[(R_t^e)^2]}{2\lambda \text{Var}[R_t^e]}. \tag{5.9}$$

The reasoning for using an "implied investment target" is similar to that for the well-known "implied volatility" in option pricing problems. An option can be priced by an involved model, but the Black-Scholes model is utilized for generating the implied volatility, which reflects the volatility in the market. In our case, the asset allocation corresponds to the "option price" and the time-consistent strategy to the "involved model". The "implied investment target" informs us about the target hidden in the strategy of a time-consistent investor. A similar idea is introduced in [29], where an induced risk aversion is established for a pre-commitment investor.

In general, it is hard to say whether being consistent with a target or being consistent with a risk aversion attitude is best. Fixing a criterion, either the investment target or the risk aversion parameter, essentially means that the other criterion will vary during an investment period. According to Equation (5.9), at time t, for a wealthy investor with a fixed risk aversion λ, the optimal asset allocation indicates that a higher investment target is desired. On the contrary, in the case of lower wealth values, the implied investment target will be lower, which indicates that a time-consistent investor will focus on a lower terminal wealth. In short, a time-consistent investor has a varying target whereas a pre-commitment investor has a fixed target throughout the investment process.

Adopting a target-oriented strategy essentially keeps a pre-commitment investor away from risky strategies and, of course, from potential profits when the amount of wealth is sufficiently large. In a similar situation, a time-consistent investor will act and perform differently, i.e. when the amount of wealth is large, she desires to gain more due to the consistent risk aversion throughout the investment period. This investor is thus never satisfied with a predetermined target.

The differences between the time-consistent and the pre-commitment strategies also lead to differences in the distributions of the corresponding terminal wealth profiles. For a pre-commitment investor, the terminal wealth follows a distribution with a fat left-tail and a large portion of the potential terminal wealth is located very close to the target. For a time-consistent investor, the distribution of terminal wealth is usually rather symmetric. An illustrative plot is presented in Figure 5.1.

Remark 5.3.5. *If we consider the time-consistency in efficiency, as proposed [29], Equation (5.9) indicates that an investor should have a target that is no less than the risk-free target to guarantee that the risk aversion parameter should be non-negative, i.e. the investor never takes risks that lead to loss of money on average.*

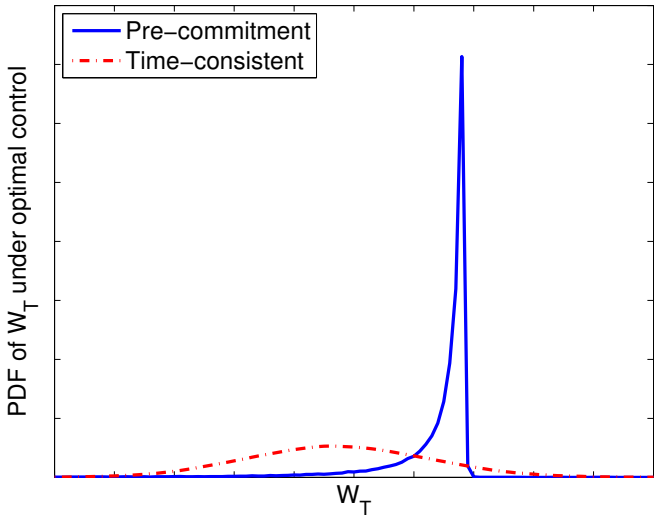


Figure 5.1: Comparison of the probability density functions of the terminal wealth, which is achieved by respectively performing unconstrained pre-commitment and time-consistent strategies. For both strategies, the investment horizon, the initial wealth and the return process are the same. By carefully choosing investment target γ and trade-off parameter λ, we make them generate a terminal wealth with the same variance, but the distributions of the terminal wealth appear to be very different. Meanwhile, the mean return of the time-consistent strategy is of course smaller than that of the pre-commitment strategy.

5.3.3. HYBRID STRATEGY: TIME-CONSISTENT WITH DETERMINED TARGET

Although time-consistency appears to be an important element of a real-life investment strategy, taking time-consistency into account is like casting additional constraints on the asset allocations and a time-consistent strategy therefore generates typically lower efficient frontiers than the pre-commitment strategy [4, 89].

As discussed in Section 5.3.2, an essential difference between the time-consistent and the pre-commitment strategy is the notion of *being satisfied with a target, or not.* In this section we design a *hybrid strategy* which combines features of the time-consistent and the pre-commitment strategy. Our aim is straightforward: we impose intermediate investment targets on a time-consistent strategy and once the target at an intermediate time step is achieved the investor is forced to take a risk-free investment strategy at subsequent time steps. Following this hybrid strategy, a time-consistent investor is satisfied with a target and thus behaves like a pre-commitment investor when the wealth is sufficient. On the other hand, we still assume that the investor has a time-consistent risk aversion when the amount of wealth does not reach the target. Therefore, in this scenario the investor still acts in a time-consistent way.

5

Mathematically, intermediate targets can be introduced to a time-consistent strategy in a similar way as done in [34], where intermediate targets are introduced to a pre-commitment strategy. The definition of the hybrid strategy is as follows.

Definition 5.3.6 (Hybrid strategy)**.** *Assume that a terminal investment target θ_T is given. At time t, the hybrid optimal control $\{x_s^{*hy}\}_{s=t}^{T-\Delta t}$ is defined by the optimal control for*

$$J_t^{hy}(W_t) = \max_{\{x_s\}_{s=t}^{T-\Delta t}} \Big[\mathbb{E}[\min(W_T,\theta_T)|W_t] - \lambda \cdot \mathrm{Var}[\min(W_T,\theta_T)|W_t]\Big],$$

with an additional condition requiring that $\{x_s^{*hy}\}_{s=\tau}^{T-\Delta t}$ also constitutes an optimal control for:

$$J_\tau^{hy}(W_\tau) = \max_{\{x_s\}_{s=\tau}^{T-\Delta t}} \Big[\mathbb{E}[\min(W_T,\theta_T)|W_\tau] - \lambda \cdot \mathrm{Var}[\min(W_T,\theta_T)|W_\tau]\Big],$$

for $\tau = t+\Delta t, t+2\cdot\Delta t, \ldots, T-\Delta t$.

Due to the target-based feature, the optimal asset allocations of a hybrid strategy are known immediately in some situations. If we define intermediate targets $\{\theta_s\}_{s=t}^{T-\Delta t}$ by

$$\theta_s = \frac{\theta_T}{(R_f)^{(T-s)/\Delta t}}, \quad s = t, t+\Delta t, \ldots, T-\Delta t,$$

the optimal asset allocation $\{x_s^{*hy}\}_{s=t}^{T-\Delta t}$ for the hybrid strategy satisfies:

$$x_s^{*hy} = 0, \quad \text{if } W_s > \theta_s, \tag{5.10}$$

for $s = t, t+\Delta t, \ldots, T-\Delta t$.

Since the hybrid strategy can be interpreted as imposing time-consistent constraints on a pre-commitment strategy when the wealth level is low, the hybrid strategy typically generates lower efficient frontiers than the pre-commitment strategy. However, as the hybrid strategy can also be regarded as one that removes the time-consistency requirements in the time-consistent strategy when the wealth level is high, the hybrid strategy should generally generate better efficient frontiers than the time-consistent strategy.

A suitable target? Introducing an investment target for a pre-commitment strategy is not new, see, for example, [29] and [34]. Prescribing an investment target in a time-consistent strategy has however, to our knowledge, not yet been pursued. In this setting, the question is how to define a suitable target θ_T. When the investment target is very high, it will be difficult to reach it and the hybrid strategy will just be the same as the plain time-consistent strategy. When the investment target is low, for example the risk-free target, the investor will always follow the risk-free strategy.

In this chapter, we propose the following approach to generate an investment target in the case where returns of risky assets are identically distributed[1]. We introduce a tuning parameter k in the *implied investment target* at time $t = 0$ and form the terminal target by:

$$\theta_T = W_0 R_f^{T/\Delta t} + k \cdot \frac{\mathbb{E}[(R_0^e)^2]}{2\lambda \text{Var}[R_0^e]}. \tag{5.11}$$

When $k = 1$, Equation (5.11) generates the implied investment target at the initial time step. For larger k-values, it will be increasingly difficult to reach the investment target and the hybrid strategy will be similar to a time-consistent strategy. In the numerical section, we will also numerically analyze the impact of different k-values for the targets in the hybrid strategy. We found that generating the target using $k = 1$ improves the efficient frontiers of the time-consistent strategy. In some cases, the efficient frontier generated by this hybrid strategy is almost identical to the one generated by the pre-commitment strategy. Introducing "targets" in this fashion makes the hybrid investor resemble more a pre-commitment than a time-consistent investor. However, if the predetermined target is too high for a hybrid investor, she will just follow the time-consistent strategy.

Minimizing mean-variance of shortfall The value function of a hybrid strategy reads:

$$J_t^{\text{hy}}(W_t) = \max_{\{x_s\}_{s=t}^{T-\Delta t}} \Big[\mathbb{E}[F(W_T)|W_t] - \lambda \cdot \text{Var}[F(W_T)|W_t] \Big], \tag{5.12}$$

where function $F(u)$ is defined by:

$$F(u) = \min(u, \theta_T).$$

Since we have

$$\min(W_T, \theta_T) = \min(W_T - \theta_T, 0) + \theta_T = -\max(\theta_T - W_T, 0) + \theta_T,$$

and if we define $\hat{F}(u) = \max(\theta_T - u, 0)$, the optimal control for value function (5.12) can be calculated in a duality form:

$$\begin{aligned} & \arg\max_{\{x_s\}_{s=t}^{T-\Delta t}} \Big[\mathbb{E}[F(W_T)|W_t] - \lambda \cdot \text{Var}[F(W_T)|W_t] \Big] \\ = \quad & \arg\min_{\{x_s\}_{s=t}^{T-\Delta t}} \Big[\mathbb{E}[\hat{F}(W_T)|W_t] + \lambda \cdot \text{Var}[\hat{F}(W_T)|W_t] \Big]. \end{aligned}$$

[1] When the returns at different time steps are not identically distributed, our proposed way to generate the target may be questionable and the conclusions regarding the choice of k in Equation (5.11) may be incorrect.

The duality form stands for an optimal control of minimizing the shortfall, which is defined by $\max(\theta_T - W_T, 0)$. A similar way of formulating the problem with the shortfall is proposed in [33].

5.3.4. EVALUATION WITH PARTIAL VARIANCE

Being satisfied with a target may be a double-edged sword. A pre-commitment investor is not exposed to the risk of losing money when she is sufficiently rich but also gives up the chance to gain higher profits. As mentioned in Section 5.3.2, this makes time-consistent and pre-commitment strategies different in terms of the distributions of the terminal wealth. Although it is reported in [89] that the pre-commitment strategy always generates higher mean-variance efficient frontiers than the time-consistent strategy, a portfolio manager may worry about the composition of the variance. All paths that are either lower or higher than the expectation contribute to the variance, but only the lower part is undesired since it reflects downside risk, i.e. potential losses. In this section, we discuss *partial variance* as another measure to evaluate the performance of a portfolio management strategy.

Partial Variance For a random variable L, one way to measure the downside risk is to use the Lower Partial Variance (LPV), defined by:

$$\text{LPV}(L) = \mathbb{E}[(\min(L - \mathbb{E}[L], 0))^2].$$

Meanwhile, we can also define the Upper Partial Variance (UPV) by:

$$\text{UPV}(L) = \mathbb{E}[(\max(L - \mathbb{E}[L], 0))^2].$$

Notice that the variance of L is composed of LPV and UPV:

$$\text{var}(L) = \text{LPV}(L) + \text{UPV}(L).$$

5.4. A SIMULATION-BASED ALGORITHM

In this section, we propose a simulation-based numerical algorithm for generating the time-consistent mean-variance portfolio optimization solution. Our algorithm consists of two steps: a forward approximation and a backward programming iteration.

5.4.1. FORWARD ITERATION: THE MULTI-STAGE STRATEGY

In Section 5.3.1 we mentioned that in the unconstrained case a time-consistent investor chooses the same asset allocations as a myopic investor. We use the name *multi-stage* investor here, because the investor solves the problem in several separate stages. The optimal asset allocation for a multi-stage investor at state (t, W_t), i.e. time t and wealth W_t, reads:

$$\begin{aligned} x_t^{*ms} = \arg\max_{x_t} \Big[& \mathbb{E}[W_t \cdot (x_t R_t^e + R_f) \prod_{s=t+\Delta t}^{T-\Delta t} (R_f) | W_t] \\ & - \lambda \cdot \text{Var}[W_t \cdot (x_t R_t^e + R_f) \prod_{s=t+\Delta t}^{T-\Delta t} (R_f) | W_t] \Big]. \end{aligned} \tag{5.13}$$

Solving the static mean-variance optimization problem as shown in Equation (5.13) *in a forward fashion* yields the optimal control for the multi-stage strategy, which reads:

$$x_t^{*ms} = \frac{\mathbb{E}[R_t^e]}{2 \cdot \lambda \cdot W_t \cdot R_f^{(T-t)/\Delta t - 1} \cdot \mathrm{Var}[R_t^e]}. \tag{5.14}$$

If we consider the value function of the time-consistent strategy, then the optimal control at time t can be defined by:

$$\begin{aligned} x_t^{*tc} \quad = \quad & \arg\max_{x_t} \Big[\mathbb{E}[W_t \cdot (x_t R_t^e + R_f) \prod_{s=t+\Delta t}^{T-\Delta t} (x_s^{*tc} R_s^e + R_f) | W_t] \\ & - \lambda \cdot \mathrm{Var}[W_t \cdot (x_t R_t^e + R_f) \prod_{s=t+\Delta t}^{T-\Delta t} (x_s^{*tc} R_s^e + R_f) | W_t] \Big], \end{aligned} \tag{5.15}$$

where $\{x_s^{*tc}\}_{s=t+\Delta t}^{T-\Delta t}$ denote the future optimal controls in the time-consistent strategy.

As shown in Equations (5.13) and (5.15), the value function of the multi-stage strategy is clearly different from the value function of the time-consistent strategy. However, the optimal controls in Equations (5.3) and (5.14) for these two strategies are identical in the unconstrained case, i.e. $x_t^{*tc} = x_t^{*ms}$ for any t.

Similar as in the last chapter, we consider the multi-stage strategy as a sub-optimal solution for a constrained optimization problem.

The multi-stage strategy can be derived in a forward fashion. As time proceeds, new information becomes available and a new optimal multi-stage solution is obtained for the new state. Therefore, the multi-stage strategy is very efficient for generating a sub-optimal solution for a constrained time-consistent problem and it constitutes a forward solution in our algorithm.

Remark 5.4.1. *In order to generate a forward solution for a hybrid strategy, we can combine the multi-stage strategy with the constraint as in Equation (5.10). However, even in the unconstrained case, this only constitutes a sub-optimal solution for the hybrid case.*

5.4.2. BACKWARD RECURSIVE PROGRAMMING ITERATION

By using the multi-stage strategy, we can solve the time-consistent problem in a forward manner by Monte Carlo simulation. However, in a general situation, the multi-stage strategy is usually not the optimal solution to the dynamic optimization problem. In order to find the optimal solution, we need to consider a backward dynamic programming solution as in Chapter 4. In this section, we present an approach to perform a backward recursive iteration based on the solution of the multi-stage forward iteration.

Decomposing the Variance Operator As mentioned, a problem for the efficient solution of the dynamic mean-variance optimization problem is caused by the nonlinearity of the variance operator. To deal with this problem, as in [4] and [89], we consider the conditional mean of the terminal wealth and the conditional mean of the terminal wealth square, respectively, as two separate value functions. Then, at time t, we have:

$$U_t(W_t) = \mathbb{E}^*[U_T(W_T)|W_t], \quad V_t(W_t) = \mathbb{E}^*[V_T(W_T)|W_t],$$

with terminal conditions

$$U_T(W_T) = W_T, \;\; V_T(W_T) = W_T^2. \tag{5.16}$$

Here, $\mathbb{E}^*[U_T(W_T)|W_t]$ and $\mathbb{E}^*[V_T(W_T)|W_t]$ indicate the expectations of the terminal value functions when the time-consistent optimal control is cast on the portfolio in period $[t,T]$.

Remark 5.4.2. *When we implement this algorithm for generating hybrid strategies, we only need to modify the terminal conditions. In that case, the terminal conditions* $U_T^{hy}(W_T)$ *and* $V_T^{hy}(W_T)$ *read:*

$$U_T^{\text{hy}}(W_T) = \min(W_T, \theta_T), \;\; V_T^{\text{hy}}(W_T) = \Big(\min(W_T, \theta_T)\Big)^2.$$

The value function $J_t(W_t)$ as shown in Equation (4.1) can then be rewritten as:

$$\begin{aligned}
J_t(W_t) &= \max_{\{x_s\}_{s=t}^{T-\Delta t}} \Big[\mathbb{E}[W_T|W_t] - \lambda \cdot \text{Var}[W_T|W_t]\Big] \\
&= \mathbb{E}^*[W_T|W_t] - \lambda \cdot \Big(\mathbb{E}^*[W_T^2|W_t] - (\mathbb{E}^*[W_T|W_t])^2\Big) \\
&= \mathbb{E}^*[\mathbb{E}^*[W_T|W_{t+\Delta t}]|W_t] \\
&\quad -\lambda \cdot \Big(\mathbb{E}^*[\mathbb{E}^*[W_T^2|W_{t+\Delta t}]|W_t] - (\mathbb{E}^*[\mathbb{E}^*[W_T|W_{t+\Delta t}]|W_t])^2\Big) \\
&= \mathbb{E}^*[U_{t+\Delta t}(W_{t+\Delta t})|W_t] \\
&\quad -\lambda \cdot \Big(\mathbb{E}^*[V_{t+\Delta t}(W_{t+\Delta t})|W_t] - (\mathbb{E}^*[U_{t+\Delta t}(W_{t+\Delta t})|W_t])^2\Big) \\
&= \max_{x_t} \Big[\mathbb{E}[U_{t+\Delta t}(W_{t+\Delta t})|W_t] \\
&\quad -\lambda \cdot \Big(\mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t})|W_t] - (\mathbb{E}[U_{t+\Delta t}(W_{t+\Delta t})|W_t])^2\Big)\Big].
\end{aligned}$$

Assume that we have obtained the value functions $U_{t+\Delta t}(W_{t+\Delta t})$ and $V_{t+\Delta t}(W_{t+\Delta t})$, then the optimization at time t can be regarded as a static optimization problem. By combining the optimal control x_t^* and the future value functions $U_{t+\Delta t}(W_{t+\Delta t})$ and $V_{t+\Delta t}(W_{t+\Delta t})$, we can calculate value functions $U_t(W_t)$ and $V_t(W_t)$ and proceed with the backward recursive programming iteration.

Backward Programming Algorithm The basic idea of backward recursive programming is to turn a globally non-smooth optimization problem into several locally smooth optimization problems. In Chapter 4, the pre-commitment mean-variance optimization problem was solved with a similar algorithm. The main difference here is that we have two value functions $U_t(W_t)$ and $V_t(W_t)$ in the backward programming iteration instead of a single value function. The algorithm for solving the time-consistent mean-variance problem is defined as follows:

- **Step 1: Initiation**:
  Generate an initial guess, for example using the multi-stage strategy, of optimal asset allocations $\{\tilde{x}_t\}_{t=0}^{T-\Delta t}$ and simulate paths of optimal wealth values $\{W_t(i)\}_{i=1}^N$, $t =$

$0,\dots,T$. At terminal time T, we have the determined functions $U_T(W_T)$ and $V_T(W_T)$. The following three steps are subsequently performed, recursively, backward in time, at $t = T-\Delta t,\dots,\Delta t,0$.

- **Step 2: Solution**
  Bundle paths into B partitions. In practice, this can be performed by first reordering the paths based on their associated wealth values and then partitioning the reordered paths into bundles such that each bundle contains a similar number of paths and the paths inside a bundle have similar wealth levels at time t.

  Denote the wealth values associated to the paths in the bundle by $\{W_t^b(i)\}_{i=1}^{N_B}$, with N_B the number of paths in the bundle. Within each bundle, we perform the following steps:

  – For the paths in the bundle, we have the wealth values $\{W_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ and continuation values $\{U_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ and $\{V_{t+\Delta t}^b(i)\}_{i=1}^{N_B}$ at time $t+\Delta t$. Functions $f_{t+\Delta t}^b(\cdot)$ and $g_{t+\Delta t}^b(\cdot)$, that locally satisfy $U_{t+\Delta t}^b = f_{t+\Delta t}^b(W_{t+\Delta t}^b)$ and $V_{t+\Delta t}^b = g_{t+\Delta t}^b(W_{t+\Delta t}^b)$, respectively, can be determined by *regression*

  – For all paths in the bundle, with $f_{t+\Delta t}^b(W_{t+\Delta t}^b)$ and $g_{t+\Delta t}^b(W_{t+\Delta t}^b)$ determined, we solve the optimization problem

  $$\max_{x_t}\left[\mathbb{E}[f_{t+\Delta t}^b(W_{t+\Delta t})|W_t] - \lambda\cdot\left(\mathbb{E}[g_{t+\Delta t}^b(W_{t+\Delta t})|W_t] - (\mathbb{E}[f_{t+\Delta t}^b(W_{t+\Delta t})|W_t])^2\right)\right]$$

  by solving the first-order conditions. In this way, we get new asset allocations $\{\hat{x}_t^b(i)\}_{i=1}^{N_B}$.

  – Since the wealth values $\{W_t^b(i)\}_{i=1}^{N_B}$ and allocations $\{\hat{x}_t^b(i)\}_{i=1}^{N_B}$ are now known, by *regression* we can also compute new continuation values $\{\hat{U}_t^b(i)\}_{i=1}^{N_B}$ and $\{\hat{V}_t^b(i)\}_{i=1}^{N_B}$,

  $$\begin{aligned}\hat{U}_t^b(i) &:= \mathbb{E}[U_{t+\Delta t}(W_{t+\Delta t})|W_t = W_t^b(i), x_t = \hat{x}_t^b(i)],\\ \hat{V}_t^b(i) &:= \mathbb{E}[V_{t+\Delta t}(W_{t+\Delta t})|W_t = W_t^b(i), x_t = \hat{x}_t^b(i)].\end{aligned}$$

  By combining them, we obtain the value function $\{\hat{J}_t^b(i)\}_{i=1}^{N_B}$.

- **Step 3: Update**
  For the paths in a bundle, based on a previous approximation, $\{\tilde{x}_t^b\}_{i=1}^{N_B}$, for the asset allocations, by *regression* we can calculate the corresponding continuation values $\{\tilde{J}_t^b(i)\}_{i=1}^{N_B}$. If, for the i-th path, $\tilde{J}_t^b(i) < \hat{J}_t^b(i)$, we choose $\hat{x}_t^b(i)$ as the updated allocation. Otherwise, we retain the previous allocation. We denote the updated allocations by $\{x_t^b(i)\}_{i=1}^{N_B}$.

- **Step 4: Evolve**
  Once the updated allocations $\{x_t^b(i)\}_{i=1}^{N_B}$ are obtained, again by *regression* we calculate the "updated" $\{U_t^b(i)\}_{i=1}^{N_B}$ and $\{V_t^b(i)\}_{i=1}^{N_B}$ values, and the backward recursion proceeds.

After one backward recursive calculation, we have updated the asset allocations at each time step. To further improve the asset allocations, we can apply the backward procedure iteratively. After several iterations, we will obtain highly satisfactory results.

The purpose of performing bundling is to reduce the bias caused by solving the first-order conditions for a non-smooth optimization problem (due to the constraints). In the unconstrained case, starting with any forward solution, we will end up with a satisfactory result after one backward iteration. Therefore, in this scenario, bundling is not required. When the value functions are highly non-smooth, a large number of bundles is required. According to our experience, 20 bundles are in general sufficient for a variety of problems.

In the algorithm, at each time step and inside each bundle, four regression steps are performed. Three of them are employed for calculating the value function. Since the value function is used to evolve information between time steps, an approximation error will accumulate due to recursion. In general, we can deal with this problem by using a large number of simulations, which is however expensive. In our numerical approach we use the so-called "regress-later technique", as applied in Chapters 2, 3 and 4.

When we perform the (local) regression, polynomials up to order two are considered as basis functions, that is, in the local bundle, $U_t(W_t)$ and $V_t(W_t)$ are approximated by:

$$\begin{aligned} U_t(W_t) \approx \quad & f_t^b(W_t) = \quad \alpha_t^b(2)W_t^2 + \alpha_t^b(1)W_t + \alpha_t^b(0), \\ V_t(W_t) \approx \quad & g_t^b(W_t) = \quad \beta_t^b(2)W_t^2 + \beta_t^b(1)W_t + \beta_t^b(0). \end{aligned}$$

Here $\{\alpha_t^b(i)\}_{i=0}^2$ and $\{\beta_t^b(i)\}_{i=0}^2$ denote the regression parameters inside the bundle.

We choose polynomials up to order two as the basis functions for two reasons. The first is that regression with respect to polynomials up to order two should be sufficient for approximating value functions in small-sized, local domains. The other reason is due to the use of the "regress-later technique" in our algorithm. When we implement this technique, one condition is that the conditional expectations of the basis functions should be known analytically, see [57] and [24]. If we choose other basis functions, the conditional expectations may not be directly available and therefore the accuracy of the "regress-later technique" may be affected negatively.

Since the budget constraint reads $W_t = W_{t-\Delta t}\cdot(x_{t-\Delta t}R^e_{t-\Delta t}+R_f)$ and we have $\mathbb{E}[U_t(W_t)|W_{t-\Delta t}]^2$ in the formulation of $J_{t-\Delta t}(W_{t-\Delta t})$, value function $J_{t-\Delta t}(W_{t-\Delta t})$ in the local domain will become a fourth-order function of $x_{t-\Delta t}$. To solve the optimality with respect to $x_{t-\Delta t}$, we implement the Newton-Raphson method with the solution from the previous iteration as the initial guess. In our numerical tests, the Newton-Raphson method always achieves a convergent solution within three iterations. However, the readers should notice that other optimization methods, for example the gradient ascent method, may also be applied here.

Remark 5.4.3. *In the unconstrained case, it is sufficient to parameterize $U_t(W_t)$ by first-order polynomials. However, according to our experiments, in the constrained case approximating $U_t(W_t)$ by second-order polynomials yields more stable and satisfactory results.*

Remark 5.4.4. *The forward and backward numerical iterations described in Sections 5.4.1 and 5.4.2 can also be used for the time-consistent problem with a state-dependent*

risk aversion, as in Remark 5.2.3. The only difference is that, instead of keeping a fixed risk aversion parameter, we choose a state-dependent risk aversion parameter for the value functions and calculate the corresponding optimal strategy. Since this state-dependent time-consistent strategy is only valid when the wealth is positive, which implies a constrained scenario, the forward strategy can only generate a rough approximation for the asset allocations. However, after using the backward recursive updating iterations, we obtain satisfactory results.

5.5. NUMERICAL TESTS

In this section, we utilize our algorithm for solving time-consistent multi-period mean-variance portfolio management problems. We will first check the convergence of the backward recursive programming step in our algorithm. Then, our numerical algorithm will be tested on time-consistent problems with different constraints. Following that, we will perform a test with the hybrid strategy proposed in Section 5.3.3, analyzing the choice of targets. In the last test, the performance of different mean-variance portfolio management strategies will be compared. In the numerical section, we use the following short notations: "TC" for a time-consistent strategy, "PC" for a pre-commitment strategy, "hybrid" for the hybrid strategy proposed in Section 5.3.3 and "TCSD" for a time-consistent strategy with state-dependent risk aversion.

5.5.1. SETUP OF NUMERICAL TESTS

We perform the tests in the case of one risky asset and one risk-free asset in the portfolio. We choose geometric Brownian motion as the dynamics of the risky asset. We assume that the log-returns of the risky asset are governed by volatility σ and mean $r_f + \xi \cdot \sigma$. Here r_f is the log-return of the risk-free asset and ξ the market price of risk.

In our numerical tests, we choose the sample size to be 50000 and the number of bundles to be 20 for the backward recursive iteration. In case the forward solutions are not optimal, we always present the solutions obtained after three backward recursive iterations. We use a common random seed to generate Monte Carlo paths for one run of the algorithm (including one forward and several backward processes). To ensure that the choice of random seed does not introduce a bias, we consider 20 different random seeds in all tests.

When we take the "no bankruptcy" constraint, Equation (5.2), into account, we choose parameter ζ to be sufficiently small, $\zeta = 10^{-8}$, which ensures that the undesired event will happen with a very low probability. When we consider the "bounded leverage" constraint, we always take $[x_{\min}, x_{\max}] = [0, 1.5]$.

The model parameters used in our tests are shown in Table 5.1. With this setup for the tests, both the forward approximation step and one backward iteration in our algorithm can be performed within a few seconds.

Remark 5.5.1. *We also performed tests using the Merton jump-diffusion model [70] with rare and negative jumps as the asset dynamics. However, when we compare the performance of different strategies, we do not get different conclusions from using geometric Brownian motion. Therefore, we do not report the results of testing the jump problem in this chapter. For more details about how the efficient frontier may change when jumps*

Table 5.1: Parameter setting (from [89]) used in the tests.

Risk-free rate r_f	.03
Market price of risk ξ	0.33
Volatility σ	0.15
Periodic contribution C	0.1
Investment duration T (years)	20
Rebalancing opportunities M	40
Initial wealth W_0	1

come in, we refer the readers to [33] and [34].

5.5.2. NUMERICAL RESULTS

Convergence of Backward Recursive Iteration We first analyze numerically the convergence of the backward recursive iterations. We consider two test cases with bounded leverage constraints on the asset allocations. In the first case, we choose the multi-stage strategy as the initial solution and perform a backward recursive iteration. Although the multi-stage strategy does not generate the optimal solution, it should constitute a reasonable initial guess to start the backward iteration. In the second case, we construct the initial solution by constant asset allocations. In general, this kind of initial solution is questionable and the backward iteration process is essential for accurate constrained solutions.

Table 5.2: The convergence of the backward recursive iterations. The multi-stage strategy is chosen as the initial solution.

Risk aversion	$\lambda = 0.05$		$\lambda = 0.25$	
	$E_0^*(W_T)$ (s.e.)	$\mathrm{Std}_0^*(W_T)$ (s.e.)	$E_0^*(W_T)$ (s.e.)	$\mathrm{Std}_0^*(W_T)$ (s.e.)
Forward iteration	13.17	9.60	8.49	2.87
	(0.04)	(0.04)	(0.01)	(0.01)
After one backward iteration	12.89	9.03	8.30	2.76
	(0.04)	(0.04)	(0.01)	(0.01)
After two backward iterations	12.87	8.97	8.28	2.75
	(0.04)	(0.04)	(0.01)	(0.01)
After three backward iterations	12.87	8.97	8.28	2.75
	(0.04)	(0.04)	(0.01)	(0.01)

According to Table 5.2, the solution obtained after the backward iteration is different from the forward solution. The most significant improvement is achieved by the first backward iteration. After the second backward iteration, the convergent solution is obtained.

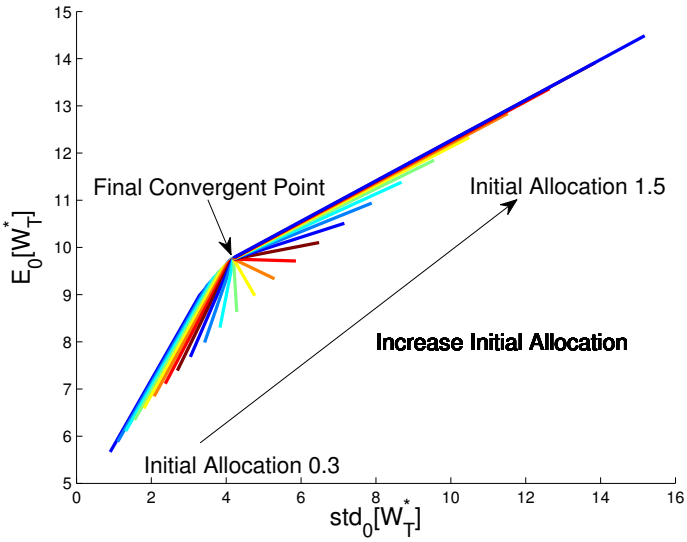


Figure 5.2: Risk aversion $\lambda = 0.15$, constant asset allocations, varying from 0.3 to 1.5, as the (inaccurate) initial solution. After three backward recursive iterations, convergent results are obtained.

Figure 5.2 shows that, starting with inferior (constant) initial guesses, we can reach convergent results after three backward iterations. This is an indication that the backward iteration is highly efficient and robust in this test case.

Time Consistent Strategy with Constraints We consider the time-consistent strategy with different constraints. The efficient frontiers are shown in Figure 5.3. In the case of a bounded leverage constraint, we have as a reference value $[\text{std}_0^*[W_T], \mathbb{E}_0^*[W_T]] = [8.175, 12.661]$ [89] for the continuous time-consistent re-balancing problem. With our algorithm, if we set the re-balancing opportunities to $M = 240$, we obtain the mean-variance pair $[\text{std}_0^*[W_T], \mathbb{E}_0^*[W_T]] = [8.175, 12.582]$. Our algorithm for the time-consistent problem thus performs highly satisfactory.

Hybrid Strategy with Different Investment Targets As mentioned in Section 5.3.3, the choice of investment target has a significant impact on the efficient frontiers generated by the hybrid strategy. In this test we consider different choices of the value k in Equation (5.11), generating targets in the hybrid strategy.

As shown in Figure 5.4, when $k \approx 1$, the corresponding efficient frontiers generated by the hybrid strategy are close to the one generated by the pre-commitment strategy for an optimization problem with bounded leverage constraint. When k increases, the efficient frontiers generated by the hybrid strategy resemble the one generated by the plain time-consistent strategy. In the following numerical tests, when we compare the hybrid strategy to other mean-variance strategies, we will form the target with $k = 1$.

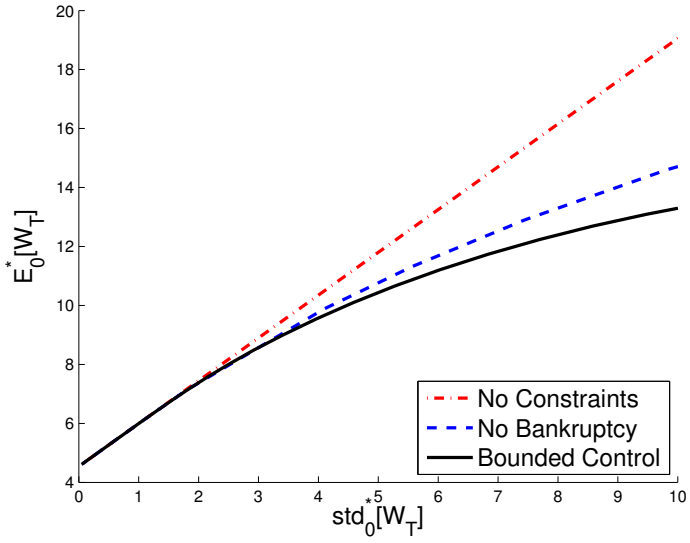
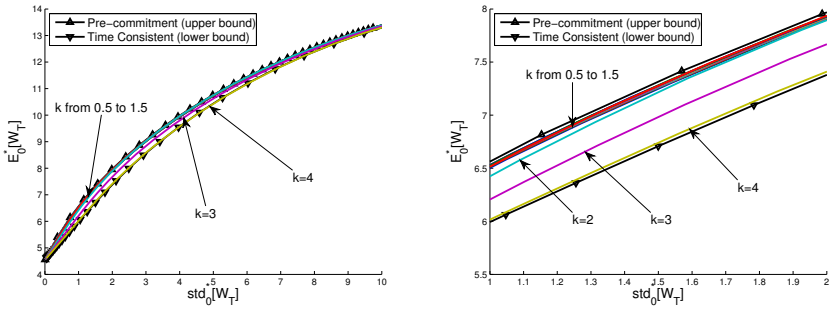


Figure 5.3: Casting different constraints on the time-consistent strategy.

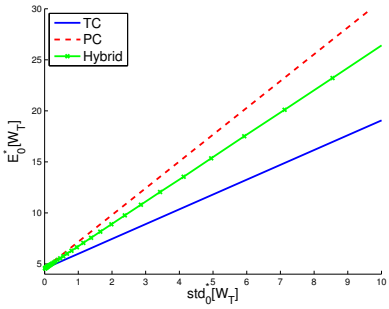


(a) Hybrid strategy, different targets

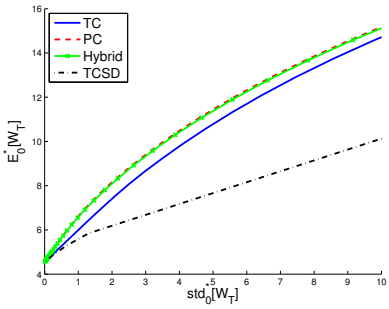
(b) Zoom of left-side figure

Figure 5.4: The influence of choosing different targets for the hybrid strategy. Bounded leverage constraint $[x_{\min}, x_{\max}] = [0, 1.5]$ is imposed on the asset allocations.

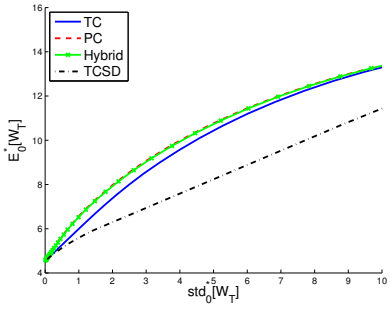
Comparing Different Mean-Variance Strategies First, we compare the mean-variance efficient frontiers for different strategies. We find that in all scenarios the pre-commitment strategy generates the highest efficient frontier, and meanwhile the time-consistent strategy with state-dependent risk aversion as in Remark 5.2.3 gives the lowest frontier. The hybrid strategy generates similar efficient frontiers as the pre-commitment strategy, especially when constraints are imposed. Even though the time-consistent strategy generates lower efficient frontiers, the difference between the time-consistent and the pre-commitment frontiers is not significant.



(a) No constraint



(b) No bankruptcy constraint



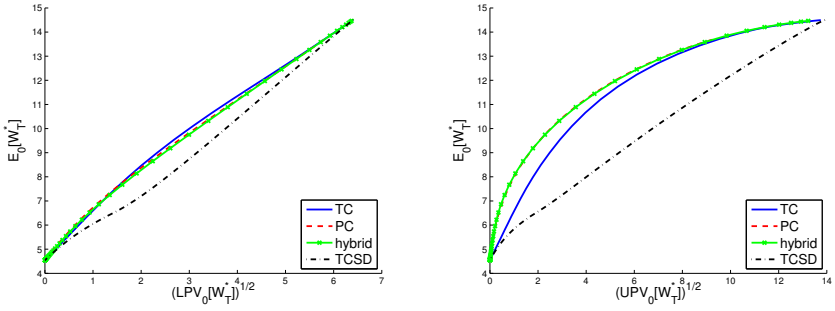
(c) Bounded control

Figure 5.5: Comparing the mean-variance frontiers of different strategies.

Figure 5.6 shows that if we consider the mean-LPV frontier, it is difficult to say whether the pre-commitment strategy is superior to the time-consistent strategy. The time-consistent strategy usually generates a lower mean-variance efficient frontier,but the main reason is that it yields a lower mean-UPV frontier. The UPV indicates the potential for an investor to achieve more wealth than the expectation, so a high UPV is not a problem. The time-consistent strategy with state-dependent risk aversion has the lowest mean-LPV frontier.

By examining how the risk is distributed, we find that the time-consistent strategy may be a better choice than the pre-commitment strategy for multi-period mean-variance

optimization. In order to achieve the same level of mean return, the time-consistent strategy is not more risky than the pre-commitment strategy. On the other hand, the time-consistent strategy is more likely to generate a higher amount of wealth than expected and thus exhibits a higher potential than the pre-commitment strategy.



(a) Lower Partial Variance

(b) Upper Partial Variance

Figure 5.6: For different strategies, we compare the Lower Partial Variance and the Upper Partial Variance. Bounded leverage constraint $[x_{\min}, x_{\max}] = [0, 1.5]$ is imposed on the asset allocations.

5.6. CONCLUSION

In this chapter, we focused on the time-consistent mean-variance optimal asset allocation problem, and compared the results to those obtained by the so-called pre-commitment strategy.

In our analysis, we found that in the unconstrained case the risk aversion parameter of a time-consistent investor and the optimal asset allocation form a one-to-one function. At a given time step, the optimal asset allocation is uniquely determined by the risk aversion attitude. In Chapter 4 we have reported similarly that, at a given time step, the optimal asset allocation of a pre-commitment investor is uniquely determined by the investment target. Based on this, we established a link between the time-consistent and the pre-commitment strategies. We assumed that at a given time step the time-consistent investor had the same asset allocation as the pre-commitment investor and a one-to-one relation between the risk aversion attitude of the time-consistent investor and the investment target of the pre-commitment investor was obtained. For a time-consistent investor, we called this function the "implied investment target".

We find that the (implied) investment target of a time-consistent investor and that of a pre-commitment investor vary in different ways through the dynamic management process. The investment target of a time-consistent investor varies over time, while a pre-commitment investor always has a fixed target. We defined a hybrid strategy by introducing a fixed target into the time-consistent strategy. In our test cases, when we chose the "implied investment target" at the initial time step as the target, the hybrid strategy generates a better mean-variance efficient frontier than the time-consistent strat-

egy. In the constrained case, the efficient frontier generated by this hybrid strategy is very similar to that generated by the pre-commitment strategy.

We also extended the numerical simulation-based algorithm, which was proposed in Chapter 4 for the pre-commitment mean-variance problem, to solving time-consistent mean-variance problems. With some modifications of the algorithm, we could utilize it to achieve highly satisfactory results for the time-consistent problem as well as for the hybrid problem. Our proposed algorithm appears to be robust and very efficient.

"Focusing on a fixed target" often gives completely different distributions of the terminal wealth from time-consistent and pre-commitment strategies. The time-consistent strategy often yields a lower mean-variance frontier, but the main reason is that the time-consistent strategy has a lower mean-UPV frontier (which does not indicate more risk). If we consider the mean-LPV frontier, the time-consistent, the pre-commitment and the hybrid strategies exhibit similar performance. Reflecting on this, we conclude that the time-consistent strategy is not always inferior to the pre-commitment strategy. Requiring time-consistency on a mean-variance strategy is reasonable from the perspective of real-life applications. Meanwhile, it does not introduce additional downside risk to the strategy.

CHAPTER 6

Robust Mean-Variance Portfolio Optimization

We consider robust pre-commitment and time-consistent mean-variance optimal asset allocation strategies, that are required to perform well also in a worst-case scenario. We show that worst-case scenarios for both strategies can be found by solving a specific equation at each time step. In the unconstrained asset allocation case, the robust pre-commitment as well as the time-consistent strategy are identical to the corresponding robust myopic strategies. In the experiments, the robustness of pre-commitment and time-consistent strategies is studied in detail. Our analysis and numerical results indicate that the time-consistent allocation strategy is more robust when possible incorrect assumptions regarding the future asset development are modeled and taken into account. In some situations, the time-consistent strategy can even generate higher efficient frontiers than the pre-commitment strategy (which is counter-intuitive), because the time-consistency restriction appears to protect an investor in such a situation.

Keywords: Robust optimization · Mean-variance optimal asset allocation · Pre-commitment strategy · Time-consistent strategy · Model prediction error

6.1. INTRODUCTION

Mean-variance optimization, which is based on two criteria, is popular with practitioners because it has a clear and very informative target function, which explicitly contains a profit term, a risk term and the trade-off between these. From the perspective of academic research, this mean-variance framework forms the basis for many interesting research directions. One potential way to generalize Markowitz's mean-variance strategy [68] is to take dynamic optimal asset allocation into consideration.

Solving a dynamic mean-variance optimization problem is not a trivial task, however. Due to the nonlinearity of the variance operator, the dynamic mean-variance optimization problem cannot be solved directly via the Bellman dynamic programming principle [5]. To tackle this issue, two possible directions are recommended in the literature. One is based on placing the dynamic mean-variance problem into a dynamic linear-quadratic (LQ) optimization context, using an embedding technique [63, 91], and the other direction is to impose a time-consistency restriction, which also serves as a condition, at all intermediate time steps [4]. See the previous two chapters.

The work in the previous two chapters is based on the assumption that the market evolves exactly as the model prescribes. This may be questionable in reality, since we

This chapter is based on the article 'On robust multi-period mean-variance porfolio optimization', submitted for publication, 2016.

can only estimate model parameters from historical data. As mentioned in [8], [12] and [17], designing an investment strategy based on historical data may lead to significant losses. One possible way to tackle this problem is to take *model uncertainty* into account and to consider *robust variants* of the optimal asset allocation problem. It is suggested in [71] to solve the mean-variance portfolio selection problem using statistically robust estimates. Noticing that accurately estimating returns may be a difficult task, the authors of [62] recommend to replace the original uncertain return process with a tractable one. Another common way to introduce model uncertainty is to consider an approach, in which the corresponding optimal strategy is required to perform well even in a so-called 'worst-case scenario'. In [83], a worst-case static mean-variance problem is transformed into a saddle-point problem and is solved using an interior-point algorithm. In [51], the authors implement a scenario tree to represent stochastic aspects and introduce uncertainty into a multi-period mean-variance portfolio problem. For a general discussion on robust optimization, we refer to [7], where a static robust mean-variance optimization is discussed as a special case. For more pointers to aspects of robust portfolio problems, we refer to a review paper [60] and the references therein.

To our knowledge, no research paper has addressed robust pre-commitment and time-consistent optimization problems. This chapter is meant to fill this gap.

We start our work by analyzing the robust pre-commitment and time-consistent asset allocation problems. Following [36] and [55], we consider an independent structure for parameter uncertainty, which makes the Bellman backward programming principle feasible within the robust dynamic optimization context. Without any constraints on the asset allocations, analytic solutions can be derived. We show that the worst-case scenarios are generated by solving a specific equation at each time step for both the pre-commitment and the time-consistent strategies. The optimal robust pre-commitment and time-consistent strategies are identical to the corresponding robust myopic strategies, where an investor derives the optimal allocation for one upcoming time period without taking future optimal allocations into account. Robustness can be introduced into the pre-commitment and the time-consistent strategies without drastically increasing the computational complexity.

The robustness of the pre-commitment and the time-consistent strategies is examined in particular when model errors occur, meaning that the assumptions on the behavior of the stochastic asset process do not reflect accurately the actually observed asset path (in the future). Considering the efficient frontiers, we find that the time-consistent strategy appears more robust than the pre-commitment strategy.

In the numerical section, we test the robustness of both strategies using the algorithms proposed in Chapters 4 and 5, that are feasible for both unconstrained and constrained optimization problems. We show that in the case of an unexpectedly poor market the time-consistent strategy can be superior to the pre-commitment strategy. In such a situation, the constrained pre-commitment strategy even yields a higher frontier than the unconstrained pre-commitment strategy, since the constraints on the allocations act as a form of "protection" when the model prediction is inaccurate. In the two-dimensional case, the influence of inaccurate correlation prediction between the risky assets is analyzed. The unconstrained pre-commitment strategy appears vulnerable to such prediction inaccuracies, whereas the time-consistent strategy appears robust.

The chapter is structured as follows. In Section 6.2, we describe the pre-commitment and the time-consistent strategies and their robust counterparts. Analysis for both strategies is performed in Section 6.3, where the robustness of the time-consistent strategy is also studied. Numerical results are presented in Section 6.4. We conclude in the last section.

6.2. PROBLEM FORMULATION

6.2.1. MULTI-PERIOD MEAN-VARIANCE PORTFOLIO

We consider a portfolio consisting of $n+1$ assets, one risk-free and n risky. We assume that the portfolio can be traded at discrete opportunities[1], $t \in [0, \Delta t, \dots, T-\Delta t]$, before terminal time T. At the initial time $t_0 = 0$, an investor decides a trading strategy to maximize the expectation of the terminal wealth and to minimize the investment risk. Formally, the investor's problem is given by

$$\hat{J}_0(W_0) = \max_{\{\hat{\mathbf{x}}_t\}_{t=0}^{T-\Delta t}} \Big\{ \mathbb{E}[W_T|W_0] - \lambda \cdot \mathrm{Var}[W_T|W_0] \Big\}, \tag{6.1}$$

with $\hat{J}$ the value function, subject to the wealth restriction:

$$W_{t+\Delta t} = W_t \cdot (\hat{\mathbf{x}}_t' \mathbf{R}_t^e + R_f), \qquad t = 0, \Delta, \dots, T-\Delta t.$$

We use notations with hat here, because we will reserve the plain notations for the formulation of the robust optimization problem, which forms the main part of this chapter. As in the previous chapters $\hat{\mathbf{x}}_t = [\hat{x}_t(1), \hat{x}_t(2), \dots, \hat{x}_t(n)]'$ denotes the asset allocations of the investor's wealth in the risky assets in the time period $[t, t+\Delta t)$. The prime sign denotes the vector transpose. The admissible investment strategy $\hat{\mathbf{x}}_t$ is assumed to be adapted. The risk aversion attitude of the investor is denoted by λ, which is a trade-off factor between maximizing the profit and minimizing the risk. R_f is the return of the risk-free asset in one time step, which is assumed to be constant for simplicity, and $\mathbf{R}_t^e = [R_t^e(1), R_t^e(2), \dots, R_t^e(n)]'$ denotes the vector of excess returns of the risky assets during $[t, t+\Delta t)$. We assume that the excess returns $\{\mathbf{R}_t^e\}_{t=0}^{T-\Delta t}$ are sequentially independent. At each time point t, $\mathbf{R}_t^e$ is supposed to follow a distribution with determined parameters. Extending the problem to a situation where the distribution parameters are uncertain constitutes the robust counterpart of a dynamic mean-variance problem, which we will elaborate on in Section 6.2.2.

The difficulty of solving the dynamic mean-variance problem is caused by the non-linearity of conditional variances, i.e. $\mathrm{Var}[\mathrm{Var}[W_T|\mathscr{F}_t]|\mathscr{F}_s] \neq \mathrm{Var}[W_T|\mathscr{F}_s], s \leq t$, which makes the well-known dynamic programming valuation approach [5] not directly applicable. To tackle this problem, there are basically two viable approaches: one is to use an embedding technique and replace the dynamic mean-variance problem by a dynamic quadratic optimization problem [63, 91], and the other is to introduce a time-consistency restriction as an additional condition into the backward programming approach [4, 89].

[1] The re-balancing times are equidistantly distributed and the total number of re-balancing opportunities before terminal time T equals M. The time step Δt between two re-balancing days is $\frac{T}{M}$.

Following the first path, we can formulate the dynamic quadratic problem as in Equation (6.2):

$$\hat{V}_0(W_0) = \min_{\{\hat{\mathbf{x}}_t\}_{t=0}^{T-\Delta t}} \Big\{ \mathbb{E}[(W_T - \gamma)^2 | W_0] \Big\}, \tag{6.2}$$

where we use $\hat{V}$ to denote the value function. By assigning different values to the parameter γ and solving the corresponding problems, we can trace out points on an efficient frontier. The authors of [63] proved that this efficient frontier is the same as the one obtained by solving Equation (6.1) with the trade-off parameter λ taking different values. An advantage of considering dynamic quadratic problem (6.2) is that the Bellman dynamic programming principle can be applied and the problem can therefore be solved in a backward recursive fashion. Since parameter γ in (6.2) acts as an investment target in the dynamic quadratic problem, this kind of optimization problem is also termed *target-based optimization* by [52] and [44].

It is mentioned in [4] that prescribing a determined target for an investor will cause a time-inconsistency. To solve this problem, they suggest to take a time-consistency restriction into account, which forms the other path for solving the dynamic mean-variance problem. Since a time-consistency restriction can also be treated as a condition in the dynamic programming framework, the dynamic mean-variance problem with time-consistency conditions can also be solved in a backward recursive manner. The authors of [4] call the strategy obtained by following the first path the *pre-commitment strategy* and the one achieved by following the second path the *time-consistent strategy*. For a dynamic mean-variance optimization problem, we formally define these two strategies as follows.

Definition 6.2.1 (Pre-commitment strategy)**.** *The pre-commitment strategy* $\{\hat{\mathbf{x}}_t^{*pc}\}_{t=0}^{T-\Delta t}$ *is defined by the optimal control for Equation (6.2).*

Definition 6.2.2 (Time-consistent strategy)**.** *The time-consistent strategy* $\{\hat{\mathbf{x}}_t^{*tc}\}_{t=0}^{T-\Delta t}$ *is defined by the optimal control for Equation (6.1) with an additional time-consistency condition requiring that the subsets* $\{\hat{\mathbf{x}}_t^{*tc}\}_{t=\tau}^{T-\Delta t}$, $\tau = \Delta t, 2\cdot\Delta t, \ldots, T-\Delta t$, *also constitute optimal controls for:*

$$\hat{J}_\tau^{tc}(W_\tau) = \max_{\{\hat{\mathbf{x}}_t\}_{t=\tau}^{T-\Delta t}} \Big\{ \mathbb{E}[W_T | W_\tau] - \lambda \cdot \mathrm{Var}[W_T | W_\tau] \Big\}, \tag{6.3}$$

for $\tau = \Delta t, 2\cdot\Delta t, \ldots, T-\Delta t$.

6.2.2. THE ROBUST COUNTERPART

In the discussion above, we assume that the excess returns $\mathbf{R}_t^e$ of the risky assets follow a distribution of determined parameters. However, this assumption may not be realistic. Since we can only assess the returns of risky assets by means of historical data, the estimated distribution parameters may be biased and may not necessarily reflect the dynamics of the risky assets in the future. Deriving an investment strategy based on these estimated parameters can lead to significant losses as pointed out by [8] and [12]. To make the investment strategy more reliable, an estimation error in the parameters can be taken into account. To this end, we consider here the *robust counterpart* to a dynamic

mean-variance optimization problem. In that case, an investor has a rival, nature, that gives rise to difficulties in the optimization process. For the mean-variance problem, we assume that the rival specifies a mean vector and a covariance matrix of the excess returns $\mathbf{R}_t^e$ of the risky assets at time $t \in \{0, \Delta t, \ldots, T - \Delta t\}$, that are respectively denoted by $\mathbf{u}_t$ and Σ_t.

We do not assume any special structure on the uncertainty sets of $\mathbf{u}_t$ and Σ_t except that they are bounded and non-empty, so the uncertainty set can be an ellipsoidal set as discussed in [49] or it can be a separable set as considered in [53]. Besides, we assume that $(\mathbf{u}_t, \Sigma_t)$, $t = 0, \Delta t, \ldots, T - \Delta t$, is not stationary. If we consider the robust optimization problem as a game of two players, the investor and nature, this latter assumption implies that nature does not necessarily choose the same adverse strategy at each time step. This leads to a time-varying uncertainty model as termed by [36].

In order to make the Bellman programming principle feasible for the robust dynamic optimization problem, we prescribe the *rectangularity assumption* as proposed by [55]. In this chapter, the rectangularity assumption[2] is defined by:

Assumption 6.2.3 (Rectangularity)**.** *The choice of $\mathbf{u}_t$ and Σ_t at time t, does not restrict the choice of $\mathbf{u}_s$ and Σ_s at time $s \in \{0, \Delta t, \ldots, T - \Delta t\} \backslash \{t\}$*

As mentioned by [55], since the sources of uncertainty in different time periods are typically independent of each other, the rectangularity assumption, which is also an independence assumption, is appropriate for finite horizon Markovian problems in most cases. However, if we would consider a stochastic volatility asset model or a time series asset model, the rectangularity assumption does not hold any more.

For the pre-commitment and the time-consistent strategies as formed in Section 6.2.1, we establish their robust counterparts as follows:

Definition 6.2.4 (Robust pre-commitment strategy)**.** *The robust pre-commitment strategy $\{\mathbf{x}_t^{*\,pc}\}_{t=0}^{T-\Delta t}$ is defined by the optimal control for the following dynamic programming problem:*

$$V_t(W_t) = \min_{\mathbf{x}_t} \max_{\mathbf{u}_t, \Sigma_t} \Big\{ \mathbb{E}[V_{t+\Delta t}(W_t \cdot (\mathbf{x}_t' \mathbf{R}_t^e + R_f)) | W_t] \Big\}, \quad t = 0, \Delta t, \ldots, T - \Delta t, \tag{6.4}$$

with the terminal condition $V_T(W_T) = (W_T - \gamma)^2$.

Definition 6.2.5 (Robust time-consistent strategy)**.** *The robust time-consistent strategy $\{\mathbf{x}_t^{*\,tc}\}_{t=0}^{T-\Delta t}$ is defined by the optimal control for*

$$J_0(W_0) = \max_{\{\mathbf{x}_t\}_{t=0}^{T-\Delta t}} \min_{\{\mathbf{u}_t, \Sigma_t\}_{t=0}^{T-\Delta t}} \Big\{ \mathbb{E}[W_T | W_0] - \lambda \cdot \mathrm{Var}[W_T | W_0] \Big\}, \tag{6.5}$$

with an additional time-consistency condition requiring that the subsets $\{\mathbf{x}_t^{*\,tc}\}_{t=\tau}^{T-\Delta t}$, $\tau = \Delta t, 2 \cdot \Delta t, \ldots, T - \Delta t$, also constitute optimal solutions for:

$$J_\tau^{tc}(W_\tau) = \max_{\{\mathbf{x}_t\}_{t=\tau}^{T-\Delta t}} \min_{\{\mathbf{u}_t, \Sigma_t\}_{t=\tau}^{T-\Delta t}} \Big\{ \mathbb{E}[W_T | W_\tau] - \lambda \cdot \mathrm{Var}[W_T | W_\tau] \Big\}, \tag{6.6}$$

for $\tau = \Delta t, 2 \cdot \Delta t, \ldots, T - \Delta t$.

[2] For a more general definition, we refer the readers to [55].

In order to meet the duality condition as proposed in [53] for a min-max mean-variance optimization problem, we require the asset allocations to be *loosely bounded*, i.e. the allocation $\mathbf{x}_t$ at each time t satisfies $-M \leq \mathbf{x}_t \leq M$ for a large positive number M, where the inequality sign is in element-wise sense. We use the term "loosely bounded" to emphasize that this restriction does not have an impact on the choice of the optimal asset allocations since the positive M can be chosen sufficiently large. Therefore, if we assume the asset allocations to be loosely bounded, the optimal control performed by an investor can still be obtained by solving the first order condition.

Remark 6.2.6. *It should be emphasized that we form the robust pre-commitment strategy by directly imposing parameter uncertainty on a plain pre-commitment strategy as defined in Section 6.2.1. Without the robustness requirement, the equivalence between the pre-commitment strategy and the optimal dynamic mean-variance strategy has been established in [63]. However, it is not yet clear whether the robust pre-commitment strategy is equivalent to the robust dynamic mean-variance strategy. In this chapter we will consider the robust pre-commitment strategy as described in Definition 6.2.4.*

6.3. ANALYSIS IN THE UNCONSTRAINED CASE

Within the framework presented in the last section, we can derive an analytic solution for the optimal robust pre-commitment and the optimal robust time-consistent strategy by the Bellman dynamic programming principle. In the former case, we consider the value function iteration in our proof, while in the latter case we make use of an essential property of a time-consistent control. Meanwhile, we also generate the adverse choices taken by nature at each time step. Similar as our findings in Chapters 4 and 5, we observe that, for either a pre-commitment or a time-consistent investor, the robust dynamic mean-variance strategy is the same as a corresponding robust myopic strategy.

In our derivation, we assume that the expectation of the excess return of any risky asset is larger than the return of the risk-free asset.Also we make the assumption that the covariance of the excess returns of the risky assets is positive definite.

6.3.1. ROBUST PRE-COMMITMENT STRATEGY

We first consider the robust pre-commitment strategy, which has been formed in a recursive setting as in Definition 6.2.4. The optimal control for the robust pre-commitment strategy can be described by the following proposition.

Proposition 6.3.1. *For the robust pre-commitment optimization problem as in Definition 6.2.4, an investor at time t with wealth W_t has the following optimal control:*

$$\mathbf{x}_t^{*pc}(W_t) = \frac{\gamma - W_t R_f^{(T-t)/\Delta t}}{W_t R_f^{(T-t)/\Delta t - 1}} \cdot (\Sigma_t^* + \mathbf{u}_t^* \cdot \mathbf{u}_t^{*\prime})^{-1} \cdot \mathbf{u}_t^*, \quad t = 0, \Delta t, \ldots, T - \Delta t, \tag{6.7}$$

with the parameters $\{\mathbf{u}_t^*, \Sigma_t^*\}$ for the worst-case scenario solving a minimization problem:

$$\{\mathbf{u}_t^*, \Sigma_t^*\} = \arg\min_{\mathbf{u}_t, \Sigma_t} \left\{ \mathbf{u}_t' \Sigma_t^{-1} \mathbf{u}_t \right\}. \tag{6.8}$$

Proof. At time step T, the value function is known as:

$$V_T(W_T) = (W_T - \gamma)^2.$$

At time step $T-\Delta t$, the value function can be calculated by:

$$\begin{aligned} V_{T-\Delta t}(W_{T-\Delta t}) &= \min_{\mathbf{x}_{T-\Delta t}} \max_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \left\{ \mathbb{E}\Big[V_T\Big(W_{T-\Delta t} \cdot (\mathbf{x}'_{T-\Delta t} \mathbf{R}^e_{T-\Delta t} + R_f) \Big) \Big| W_{T-\Delta t} \Big] \right\} \\ &= \min_{\mathbf{x}_{T-\Delta t}} \max_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \left\{ \mathbb{E}\Big[\Big(W_{T-\Delta t} \cdot (\mathbf{x}'_{T-\Delta t} \mathbf{R}^e_{T-\Delta t} + R_f) - \gamma \Big)^2 \Big| W_{T-\Delta t} \Big] \right\}. \end{aligned}$$

Since the asset allocations are assumed to be loosely bounded and the feasible sets for $\mathbf{u}_{T-\Delta t}$ and $\Sigma_{T-\Delta t}$ are bounded and non-empty, based on Lemma 2.3 in [53], perfect duality holds, i.e. changing the order of minimization and maximization does not influence the value of the value function. Therefore, we have:

$$V_{T-\Delta t}(W_{T-\Delta t}) = \max_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \min_{\mathbf{x}_{T-\Delta t}} \left\{ \mathbb{E}\Big[\Big(W_{T-\Delta t} \cdot (\mathbf{x}'_{T-\Delta t} \mathbf{R}^e_{T-\Delta t} + R_f) - \gamma \Big)^2 \Big| W_{T-\Delta t} \Big] \right\}.$$

We define a new function $F_{T-\Delta t}(\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t})$ by:

$$F_{T-\Delta t}(\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t}) := \min_{\mathbf{x}_{T-\Delta t}} \left\{ \mathbb{E}\Big[\Big(W_{T-\Delta t} \cdot (\mathbf{x}'_{T-\Delta t} \mathbf{R}^e_{T-\Delta t} + R_f) - \gamma \Big)^2 \Big| W_{T-\Delta t} \Big] \right\}, \quad (6.9)$$

where $\mathbf{u}_{T-\Delta t}$ and $\Sigma_{T-\Delta t}$ influence the distribution of $\mathbf{R}^e_{T-\Delta t}$. The value function $V_{T-\Delta t}(W_{T-\Delta t})$ can then be written as:

$$V_{T-\Delta t}(W_{T-\Delta t}) = \max_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \Big\{ F_{T-\Delta t}(\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t}) \Big\}.$$

For a given set of parameters $\{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t}\}$, the optimization problem with respect to $\mathbf{x}_{T-\Delta t}$ as shown in Equation (6.9) constitutes a smooth and convex optimization problem. Therefore, by solving the first-order-conditions for the optimality, we get:

$$\mathbf{x}^*_{T-\Delta t}(\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t}) = \frac{\gamma - W_{T-\Delta t} R_f}{W_{T-\Delta t}} \cdot (\Sigma_{T-\Delta t} + \mathbf{u}_{T-\Delta t} \mathbf{u}'_{T-\Delta t})^{-1} \cdot \mathbf{u}_{T-\Delta t}. \quad (6.10)$$

Inserting the optimal control in Equation (6.10) into Equation (6.9) gives us:

$$F_{T-\Delta t}(\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}, W_{T-\Delta t}) = (\gamma - W_{T-\Delta t} R_f)^2 \cdot \Big(1 - \mathbf{u}'_{T-\Delta t} \cdot (\Sigma_{T-\Delta t} + \mathbf{u}_{T-\Delta t} \mathbf{u}'_{T-\Delta t})^{-1} \cdot \mathbf{u}_{T-\Delta t} \Big)$$

and

$$V_{T-\Delta t}(W_{T-\Delta t}) = \max_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \Big\{ (\gamma - W_{T-\Delta t} R_f)^2 \cdot \Big(1 - \mathbf{u}'_{T-\Delta t} \cdot (\Sigma_{T-\Delta t} + \mathbf{u}_{T-\Delta t} \mathbf{u}'_{T-\Delta t})^{-1} \cdot \mathbf{u}_{T-\Delta t} \Big) \Big\}.$$

Therefore, the optimal adverse policy taken by nature should solve the minimization problem:

$$\{\mathbf{u}^*_{T-\Delta t}, \Sigma^*_{T-\Delta t}\} = \arg \min_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \Big\{ \mathbf{u}'_{T-\Delta t} \cdot (\Sigma_{T-\Delta t} + \mathbf{u}_{T-\Delta t} \mathbf{u}'_{T-\Delta t})^{-1} \cdot \mathbf{u}_{T-\Delta t} \Big\}.$$

By the Sherman-Morrison formula [77], we can simplify the optimization target and the optimal adverse policy should satisfy:

$$\{\mathbf{u}^*_{T-\Delta t}, \Sigma^*_{T-\Delta t}\} = \arg\min_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \left\{\mathbf{u}'_{T-\Delta t}\Sigma^{-1}_{T-\Delta t}\mathbf{u}_{T-\Delta t}\right\}.$$

So, the proposition is verified at time $T-\Delta t$ and the value function at time $T-\Delta t$ reads:

$$V_{T-\Delta t}(W_{T-\Delta t}) = (\gamma - W_{T-\Delta t}R_f)^2 \cdot \left(1 - \mathbf{u}^{*\prime}_{T-\Delta t} \cdot (\Sigma^*_{T-\Delta t} + \mathbf{u}^*_{T-\Delta t}\mathbf{u}^{*\prime}_{T-\Delta t})^{-1} \cdot \mathbf{u}^*_{T-\Delta t}\right).$$

Since the policy of nature satisfies the rectangularity assumption, the factor $1 - \mathbf{u}^{*\prime}{}_{T-\Delta t} \cdot (\Sigma^*_{T-\Delta t} + \mathbf{u}^*{}_{T-\Delta t}\mathbf{u}^{*\prime}{}_{T-\Delta t})^{-1} \cdot \mathbf{u}^*{}_{T-\Delta t}$ will not influence the optimality at time $T-2\Delta t$. Therefore, we can finalize the proof by mathematical induction. □

In our proof, in the inner optimization, i.e. the optimization with respect to the asset allocations, we solve the first-order-conditions to obtain the optimality. This is feasible since the asset allocations are assumed to be loosely bounded. In the outer optimization, i.e. choosing the optimal adverse policy, we do not explicitly solve the problem. However, since the rectangularity assumption holds, this will not affect the backward programming process.

6

6.3.2. ROBUST TIME-CONSISTENT STRATEGY

Here, we consider the optimal control for the time-consistent strategy. Different from the derivation for the robust pre-commitment strategy, we do not utilize the value function iteration, which will be complicated in the time-consistent case. Instead, by benefiting from the special structure of a time-consistent optimal control, we focus on generating the optimal controls directly. A similar approach is considered in Chapter 5. Our findings in the robust time-consistent case can be described by the following proposition.

Proposition 6.3.2. *For the robust time-consistent optimization problem in Definition 6.2.5, an investor at time t with wealth W_t has the following optimal control:*

$$\mathbf{x}^{*tc}_t(W_t) = \frac{\Sigma^{*-1}_t \mathbf{u}^*_t}{2\lambda W_t R_f^{(T-t)/\Delta t - 1}}, \quad t = 0, \Delta t, \ldots, T-\Delta t, \tag{6.11}$$

with the parameters $\{\mathbf{u}^*_t, \Sigma^*_t\}$ for the worst-case scenario solving the minimization problem:

$$\{\mathbf{u}^*_t, \Sigma^*_t\} = \arg\min_{\mathbf{u}_t, \Sigma_t} \left\{\mathbf{u}'_t\Sigma^{-1}_t\mathbf{u}_t\right\}. \tag{6.12}$$

Proof. It is not difficult to prove that the proposition is correct at time $T-\Delta t$:

$$\mathbf{x}^{*tc}_{T-\Delta t}(W_{T-\Delta t}) = \frac{\Sigma^{*-1}_{T-\Delta t}\mathbf{u}^*_{T-\Delta t}}{2\lambda W_{T-\Delta t}}, \tag{6.13}$$

and the optimal adverse policy reads:

$$\{\mathbf{u}^*_{T-\Delta t}, \Sigma^*_{T-\Delta t}\} = \arg\min_{\mathbf{u}_{T-\Delta t}, \Sigma_{T-\Delta t}} \left\{\mathbf{u}'_{T-\Delta t}\Sigma^{-1}_{T-\Delta t}\mathbf{u}_{T-\Delta t}\right\}.$$

Assume that at time $T-2\Delta t$ an investor has wealth $W_{T-2\Delta t}$, then, after performing some control at time $T-2\Delta t$, the corresponding terminal wealth is given by:

$$W_T = W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot ((\mathbf{x}^{*tc}_{T-\Delta t})' \cdot \mathbf{R}^{e*}_{T-\Delta t} + R_f),$$

where $\mathbf{x}^{*tc}_{T-\Delta t}$ and $\mathbf{R}^{e*}_{T-\Delta t}$ indicate that the investor will take future optimality into account while designing the optimal control at time $T-2\Delta t$. Further we have:

$$\begin{aligned} W_T &= W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot ((\mathbf{x}^{*tc}_{T-\Delta t})' \cdot \mathbf{R}^{e*}_{T-\Delta t} + R_f) \\ &= \frac{1}{2\lambda}\mathbf{u}^{*\prime}_{T-\Delta t}\Sigma^{*-1}_{T-\Delta t}\mathbf{R}^{e*}_{T-\Delta t} + W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot R_f \\ &= K_{T-\Delta t} + W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot R_f, \end{aligned}$$

the second equality is valid since the optimal control is as in Equation (6.13), which indicates that multiplying the wealth at time $T-\Delta t$ with the optimal time-consistent control should yield a determined number. In the last line, we define the factor $K_{T-\Delta t}$, which only contains elements from time step $T-\Delta t$. Since we have the rectangularity assumption and the excess returns are assumed to be sequentially independent, the optimal control at time $T-2\Delta t$ can be obtained by ignoring factor $K_{T-\Delta t}$ and by solving:

$$\begin{aligned} \max_{\mathbf{x}_{T-2\Delta t}} \min_{\mathbf{u}_{T-2\Delta t}, \Sigma_{T-2\Delta t}} \Big\{ &\mathbb{E}[W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot R_f | W_{T-2\Delta t}] - \\ &\lambda \cdot \mathrm{Var}[W_{T-2\Delta t} \cdot (\mathbf{x}'_{T-2\Delta t}\mathbf{R}^e_{T-2\Delta t} + R_f) \cdot R_f | W_{T-2\Delta t}] \Big\}. \end{aligned}$$

By solving this static max-min optimization problem, we can verify the proposition at time $T-2\Delta t$. At the remaining time steps, we can prove the proposition by mathematical induction. The key point in the proof is that, after taking the structure of an optimal control into account, the robust time-consistent problem shares the same optimal control as a robust myopic problem, which can be solved elegantly. □

For the time-consistent case, we do not give details about how the two-layer optimization problem can be solved, because the basic machinery is the same as in the pre-commitment case. We find that the robust pre-commitment and the robust time-consistent strategies share some common features. We will elaborate on them in Section 6.3.3.

ROBUST MEAN-VARIANCE EFFICIENCY

In the real world, it is impossible to determine the mean $\mathbf{u}_t$ and the covariance matrix Σ_t for the future excess return at time step t, since only one realization can be observed. Usually, the *basic strategy* is that we specify the excess returns at different time steps to follow a stationary distribution with pre-determined mean $\tilde{\mathbf{u}}$ and covariance matrix $\tilde{\Sigma}$, i.e. we assume $\mathbf{u}_t = \tilde{\mathbf{u}}$ and $\Sigma_t = \tilde{\Sigma}$, for $t = 0, \Delta t, \ldots, T-\Delta t$. Assuming the excess returns to be stationary may not be very restrictive, but prescribing the mean and the covariance may be questionable. For example, if in reality the risky assets follow a stationary distribution with a mean value $\hat{\mathbf{u}}$ and covariance matrix $\hat{\Sigma}$, that are significantly different from $\tilde{\mathbf{u}}$ and $\tilde{\Sigma}$, then the corresponding optimal control on the portfolio will certainly be different from the control resulting from the basic strategy. Reflecting on this, the question

is: how much different is a portfolio managed under the basic strategy from a portfolio managed under an *optimal strategy* (where we mean by optimal strategy, the case where accurate asset information is used in the model)? As implied by Proposition 6.3.3, in some situations the basic strategy can generate the same mean-variance efficient frontier as an optimal strategy.

Proposition 6.3.3. *For two investors, Investor A assuming that the excess returns of risky assets have mean* $\tilde{\mathbf{u}}$ *and covariance* $\tilde{\Sigma}$ *and Investor B assuming the mean to be* $\hat{\mathbf{u}}$ *and the covariance* $\hat{\Sigma}$*, their time-consistent strategies generate portfolios with the same Sharpe ratio [76], if*

- *(Condition 1) their strategies are performed in the same market,*
- *(Condition 2)* $\tilde{\Sigma}^{-1}\tilde{\mathbf{u}} = K \cdot \hat{\Sigma}^{-1}\hat{\mathbf{u}}$*, where* K *is a constant.*

Proof. The proof is straightforward. We first generate the optimal time-consistent controls of these two investors and then cast the controls on the portfolios starting with the same amount of wealth.

Following a similar derivation as in the proof of Proposition 6.3.2, we obtain the optimal asset allocations for Investors A and B, as:

$$\mathbf{x}_t^A(W_t) = \frac{\tilde{\Sigma}^{-1}\tilde{\mathbf{u}}}{2\lambda W_t R_f^{(T-t)/\Delta t-1}}, \quad \mathbf{x}_t^B(W_t) = \frac{\hat{\Sigma}^{-1}\hat{\mathbf{u}}}{2\lambda W_t R_f^{(T-t)/\Delta t-1}},$$

at time points $t = 0, \Delta t, \ldots, T - \Delta t$.

For Investor A, imposing the optimal control on a portfolio starting with initial wealth W_0 will generate terminal wealth:

$$W_T^A = W_0 R_f^{T/\Delta t} + \frac{\tilde{\Sigma}^{-1}\tilde{\mathbf{u}}}{2\lambda} \cdot (\mathbf{R}_0^e + \mathbf{R}_{\Delta t}^e + \cdots + \mathbf{R}_{T-\Delta t}^e). \tag{6.14}$$

For Investor B, the terminal wealth can be written in a similar fashion as:

$$W_T^B = W_0 R_f^{T/\Delta t} + \frac{\hat{\Sigma}^{-1}\hat{\mathbf{u}}}{2\lambda} \cdot (\mathbf{R}_0^e + \mathbf{R}_{\Delta t}^e + \cdots + \mathbf{R}_{T-\Delta t}^e). \tag{6.15}$$

Notice that the excess returns $\{\mathbf{R}_t^e\}_{t=0}^{T-\Delta t}$ are random numbers, that indicate the movement of the risky assets in the market. Since we assume that two investors perform their strategies in the same market, the same notations for the excess returns are used in Equations (6.14) and (6.15).

If we choose the risk-free portfolio as the benchmark portfolio, the Sharpe ratio of the portfolio managed by Investor A is given by:

$$\begin{aligned} S^A &= \frac{\mathbb{E}[W_T^A] - W_0 R_f^{T/\Delta t}}{\sqrt{\mathrm{Var}[W_T^A]}} \\ &= \frac{\mathbb{E}[(\tilde{\Sigma}^{-1}\tilde{\mathbf{u}}) \cdot (\mathbf{R}_0^e + \mathbf{R}_{\Delta t}^e + \cdots + \mathbf{R}_{T-\Delta t}^e)]}{\sqrt{\mathrm{Var}[(\tilde{\Sigma}^{-1}\tilde{\mathbf{u}}) \cdot (\mathbf{R}_0^e + \mathbf{R}_{\Delta t}^e + \cdots + \mathbf{R}_{T-\Delta t}^e)]}}. \end{aligned}$$

In a similar way, we can calculate the Sharpe ratio of the portfolio managed by Investor B. Since we have the assumption that $\tilde{\Sigma}^{-1}\tilde{\mathbf{u}} = K \cdot \hat{\Sigma}^{-1}\hat{\mathbf{u}}$, we obtain $S^A = S^B$, i.e. the portfolios managed by Investor A and Investor B share the same Sharpe ratio. □

As shown in Equations (6.14) and (6.15), since two investors have different views on the market, the amounts of their terminal wealth are different. However, when their insights on the market satisfy a special condition, Proposition 6.3.3 implies that their portfolios will share the same return-to-risk ratio. It also means that the mean-variance efficient frontier generated by Investor A with a basic guess of the market will be identical to that generated by Investor B with expert knowledge.

In the one-dimensional case, based on Proposition 6.3.3, we find a stronger conclusion as described in the following corollary.

Corollary 6.3.4. *For a time-consistent investor managing a portfolio with one risky asset and one risk-free asset, no matter what is the estimation of the return of the risky asset, the strategy is always mean-variance efficient.*

Proof. The proof process is similar to that for Proposition 6.3.3. The key point in the proof is that in the one-dimensional case Condition 2, which is required to establish Proposition 6.3.3, is always valid. □

According to the results in Chapter 5 , when there is only one risky asset in the portfolio and its returns have a stationary distribution, a time-consistent investor always invests a constant amount of money in this risky asset. Even if the return distribution is mispredicted, the time-consistent investor will still invest in the risky asset with a constant amount. This makes the portfolio end up on the efficient frontier but at a place different from expected.

6.3.3. SOME REFLECTIONS

Based on the derivations in Sections 6.3.1 and 6.3.2, we give some insights respectively on the adverse policy of nature and on the optimal policy of an investor in a dynamic mean-variance optimization framework.

On the optimal adverse policy According to Equations (6.8) and (6.12), at a given time step the optimal policy of nature solves the same optimization problem for either the pre-commitment or the time-consistent problem. In the case of one risky asset, the optimization problem is intuitive: the expectation of the excess return is assumed to be as low as possible and the variance of the excess return as high as possible.

Although the policy of nature may be strictly constrained, we notice that constraints on the policy of nature do not influence the smoothness of the value function in the pre-commitment case and we assume that neither the smoothness of the value function in the time-consistent case is affected. At a given time step, the policy of nature is independent of the amount of wealth held by the investor.

Moreover, if we assume that at each time step the feasible sets for nature are identical, we see that the optimal adverse policy is to take a stationary strategy although it is not required to perform in this manner.

On the optimal policy of an investor At each time step, for either a robust pre-commitment or a robust time-consistent investor, the optimal asset allocations can be obtained by solving the corresponding robust myopic problems, where a myopic investor is assumed to perform an optimal control only for a next time period and to adopt the risk-free strategy afterwards. For a non-robust pre-commitment or a time-consistent strategy, similar results have been found respectively in Chapters 4 and 5. One application of this is that, in the unconstrained case, the optimal control policy of an investor can be generated efficiently in a forward fashion.

A recently published paper by [72] also deals with the robust dynamic mean-variance problem. In that model setting, the author also finds that the robust dynamic strategy is identical to a myopic strategy. Our discussion differs from this in two aspects. First, [72] requires investment risk to be bounded at each time step and is therefore neither related to the pre-commitment nor to the time-consistent case discussed in this chapter. Secondly, as a consequence of choosing different model settings, our optimal asset allocations are not same. The optimal choice of nature is included in deriving our optimal strategy, while the optimal strategy in [72] does not take this into account.

As discussed in Chapter 5, when the nominal model is correct, the pre-commitment strategy usually generates a higher mean-variance efficient frontier than the time-consistent strategy, since the time-consistent strategy is restricted by the time consistency constraint. However, our derivations in Section 6.3.2 suggest that, when the market does not perform as expected, *a time-consistent strategy appears to generate more robust efficient frontiers* and thus has the potential to yield higher efficient frontiers than a pre-commitment strategy. In a special case where a portfolio consists of one risky asset and one risk-free asset and the risky asset's return has a stationary distribution, any time-consistent mean-variance strategy is guaranteed to generate the same mean-variance efficient frontier as a robust time-consistent strategy.

Of course, all our derivations in Sections 6.3.1 and 6.3.2 are based on the assumption that the asset allocations are loosely bounded. If we impose strict constraints on the policy of an investor, the value functions will be non-smooth and our derivation will be violated. In that case, the optimal strategy of an investor is not the same as an optimal myopic strategy any more, but we can implement numerical methods to generate constrained solutions (as is done in the numerical section to follow).

Remark 6.3.5. *As mentioned in Chapter 5, when there is periodic money withdrawal from (or injection in) the portfolio, a pre-commitment strategy will be very sensitive to the amount of withdrawal. When the amount is not as expected, a time-consistent strategy may also generate a higher efficient frontier than a pre-commitment strategy.*

6.4. NUMERICAL EXPERIMENTS

In the previous sections, we discussed the optimal allocations and the worst-case scenarios for the robust mean-variance optimization problem. A robust mean-variance strategy suggests that an investor should take the worst-case scenario into account and adopt a conservative strategy. In this chapter, we assume that the uncertainty sets of parameters at different time steps are identical. Our proof then implies that the worst-case market, which solves Equation (6.8) or (6.12) for either an unconstrained pre-commitment or

a time-consistent investor, can be generated by the same model parameters at each time step. In the constrained case, we make a conjecture that this choice of model parameters still yields the worst-case scenario. Since the purpose of constructing the worst-cast scenario is to challenge an investor to achieve good performance of the management strategy, we believe that, even if this conjecture is not correct, this choice of model parameters has a significant influence on the performance of a portfolio. With these criteria for choosing model parameters, we can compare the unconstrained and the constrained cases in the same framework.

Different from [51] and [62], we do not consider an uncertainty parameter set in our tests, since a fixed point in the uncertainty set can already give us the worst-case scenario (see Propositions 6.3.1 and 6.3.2). We perform numerical tests to examine the robustness of the proposed mean-variance strategies. We assume that a model error is present in our test cases. Two scenarios are considered, one is the model scenario and the other the real-world observed scenario. We derive the asset allocations from the model scenario, adopt those allocations and analyze their impact for the observed real-world scenario. When the model scenario appears conservative, the corresponding strategy can be seen as a robust strategy.

Numerical Algorithm We utilize the numerical method proposed in Chapters 4 and 5 to solve the pre-commitment and the time-consistent problems, respectively. This numerical method consists of two phases, a sub-optimal solution is first generated in the forward phase and subsequently updating is performed in the backward phase to improve the solution. The backward phase is only necessary for the constrained case. After iterating the forward-backward process for several times, we obtain highly satisfactory results. In our tests, we always choose the myopic strategy as the initial guess and report the result obtained after three backward iterations.

Since we wish to check how a strategy performs if the observed real-world market does not appear to be as expected by the model assumptions, our simulation-based optimization algorithms are slightly adjusted. In the forward phase, we generate paths by using the dynamics of the observed real market; in the backward phase, we update the path-wise controls only based on the model information.

6.4.1. 1D PROBLEM

Test Setup We first perform our numerical tests in the one-dimensional case, where the portfolio contains one risky asset and one risk-free asset. We choose geometric Brownian motion as the dynamics of the risky asset and assume that the log-returns of the risky asset are governed by volatility σ and mean $r_f + \xi \cdot \sigma$. Here r_f is the log-return of the risk-free asset and ξ is the market price of risk. When we consider constrained optimization scenarios, we impose a bounded leverage constraint $x \in [x_{\min}, x_{\max}]$ on the portfolio allocations. The values of the parameters are presented in Table 6.1.

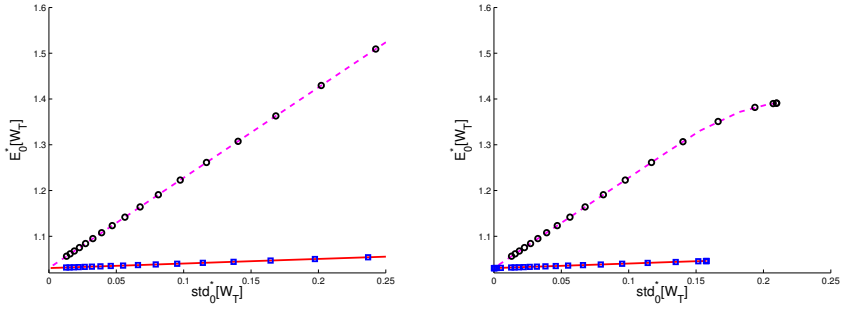
In the following tests, σ and ξ represent the parameters used in the model whereas σ_{real} and ξ_{real} are the parameters in the observed real-world market; the difference between these values we call the model error.

We design numerical experiments with the following questions in mind:

- Is a time-consistent strategy sensitive to model errors?

Table 6.1: Parameter setting.

Risk-free rate r_f	0.03
Investment duration T (year)	1
Re-balancing opportunities M	12
Initial wealth W_0	1
Leverage constraint $[x_{\min}, x_{\max}]$	$[0,1]$



(a) Unconstrained case (b) With leverage constraint

Figure 6.1: By choosing different pairs of (ξ, ξ_{real}), we obtain four efficient frontiers generated by the time-consistent strategy. They are respectively represented by: the red line $(\xi, \xi_{\text{real}}) = (0.1, 0.1)$, the magenta dashed line $(\xi, \xi_{\text{real}}) = (0.1, 2)$, the squares $(\xi, \xi_{\text{real}}) = (1, 0.1)$ and the circles $(\xi, \xi_{\text{real}}) = (1, 2)$.

- When an unexpectedly poor market is encountered, how are efficient frontiers, generated by a pre-commitment or a time-consistent strategy, affected?

In order to answer these questions, various choices for (σ, ξ) and $(\sigma_{\text{real}}, \xi_{\text{real}})$ are made in the following tests.

Robust Mean-Variance Efficiency of Time-Consistent Policy We first check whether the time-consistent strategy is sensitive to model errors (i.e. real-world parameters $\neq$ model parameters). We assume that the volatility of the real-world market is the same as that indicated by the model, $\sigma_{\text{real}} = \sigma = 0.15$. We consider two model settings, $\xi = 0.1$ and $\xi = 1$, and two market settings, $\xi_{\text{real}} = 0.1$ and $\xi_{\text{real}} = 2$. When we choose $\xi = 0.1$, it yields a robust strategy since the management strategy is generated in the worst-case scenario. The choices of ξ_{real} can be explained as follows: $\xi_{\text{real}} = 0.1$ means that the worst-case scenario indeed appears and $\xi_{\text{real}} = 2$ indicates a good market where the risky asset yields a high return.

As shown in Figure 6.1, in the unconstrained as well as the constrained situation, the efficient frontiers generated by the time-consistent strategy are not sensitive to the model errors. The locations of the efficient frontiers mainly depend on the real-world

market parameters. When the real-world market is booming, the time-consistent efficient frontiers are high. When the market is poor, the time-consistent frontiers are low.

Unexpectedly Poor Market In this numerical test, we check the performance of the pre-commitment and the time-consistent strategies when the market is not as expected. We consider two scenarios, one with fixed mean $(r_f + \sigma \cdot \xi)$ and unexpected volatility [3] of the risky asset return and the other with fixed volatility and unexpected mean of the risky asset return. In these two scenarios, the parameters are chosen as in Table 6.2.

Table 6.2: Parameters for modeling an unexpectedly poor market.

Set I: $\sigma = 0.1, \xi = 0.33, \sigma_{\text{real}} = 0.5, \xi_{\text{real}} = 0.066.$
Set II: $\sigma = 0.15, \xi = 1, \sigma_{\text{real}} = 0.15, \xi_{\text{real}} = 0.1.$

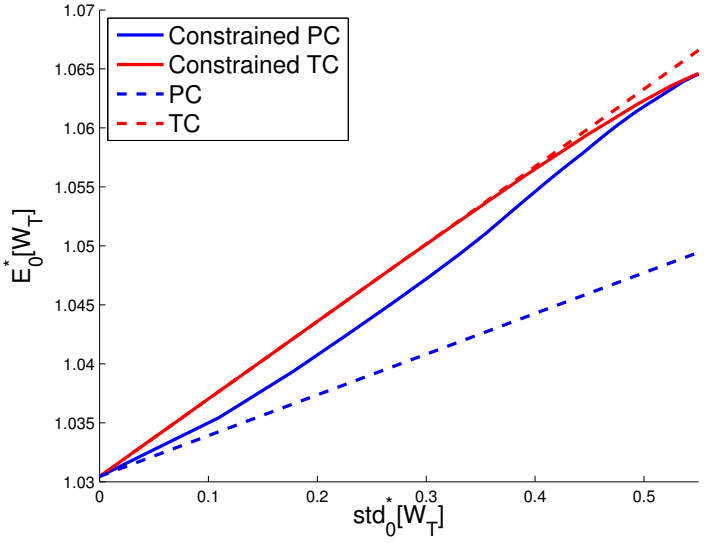
As seen in our first experiment, the time-consistent strategy is not sensitive to the model prediction and yields efficient frontiers that can also be achieved by a time-consistent investor with correct market information. When a bounded leverage constraint is introduced, the time-consistent frontier tends to be somewhat lower than in the unconstrained case.

A surprising finding is that, when the model prediction is inconsistent with the real market, the pre-commitment strategy may generate lower efficient frontiers than a time-consistent strategy. According to our findings in Chapter 5, if the market moves according to the model prediction, a pre-commitment strategy generates a higher frontier than a time-consistent strategy. This is due to the fact that time-consistency can be regarded as a constraint on a pre-commitment strategy. However, in case of an unexpectedly poor market, the time-consistency constraint may *protect an investor*, while a pre-commitment investor may suffer from the poor market. As presented in Figure 6.2, in both situations, the unconstrained pre-commitment strategy generates the lowest efficient frontier. When the constraint is introduced into the pre-commitment strategy, the efficient frontier gets higher than in the unconstrained case. This is not difficult to understand. When a model yields the correct prediction, introducing a constraint forms a restriction on a portfolio; when a model generates an incorrect prediction, the constraint acts as a "protection". Therefore, when market movement is not as anticipated, the constrained pre-commitment strategy may perform better than its unconstrained counterpart.

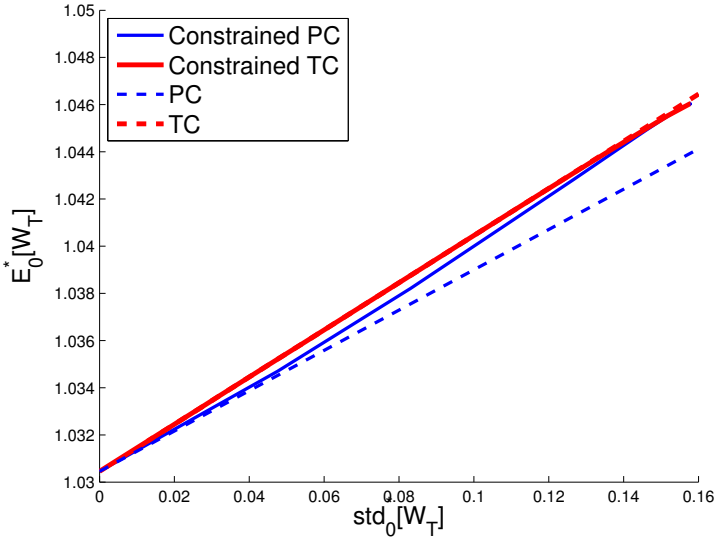
We also see that the constrained pre-commitment and time-consistent frontiers coincide at their right ends. When an investor pursues a very high return, the maximal possible allocation will be set at almost all time steps. Therefore, in this situation, a constrained pre-commitment strategy will be similar to a constrained time-consistent strategy.

Remark 6.4.1. *When we consider an unexpectedly booming market where* $\sigma_{\text{real}} = \sigma$ *and* $\xi_{\text{real}} > \xi$*, the pre-commitment strategy generates a higher efficient frontier than the time-*

[3] In case of unexpected volatility, we choose the volatility of the real market to be 50%, which is similar to the scenario happening after the 2008 financial crisis in the American market [67].



(a) Unexpected volatility



(b) Unexpected mean

Figure 6.2: Comparing the performance of a pre-commitment and a time-consistent strategy when the market is unexpectedly poor. In the unexpected volatility case, we choose parameters from Set I in Table 6.2. In the unexpected mean case, parameters from Set II are used.

consistent strategy. In this scenario, introducing constraints makes both the pre-commitment and the time-consistent efficient frontiers lower, as expected.

Although the time-consistent strategy can generate a robust efficient frontier, it does not mean that the time-consistent investor, who designs a strategy with expected return $\mathbb{E}_0^*[W_T] = d$ in mind, will achieve this amount of return in an unexpected market. When the market is worse than expected, the pre-commitment investor will attempt to reach the predetermined target by taking more risk. Meanwhile, the time-consistent investor may just be satisfied with a lower mean return associated with less risk. In this case, it is interesting to examine both strategies by checking the probability of getting less than, say, 90% of the predetermined return d. This shortfall probability also reflects the robustness of an investment strategy.

As shown in Figure 6.3, when the market is as expected in the model, both the time-consistent strategy and the pre-commitment strategy lead to low shortfall probabilities. When an unexpectedly poor market occurs, the shortfall probabilities increase. It is not easy to say which strategy is more robust with respect to the shortfall probability criterion. According to Figure 6.3(b), the time-consistent strategy has a higher probability of generating the terminal wealth lower than 90% of the desired level. However, if we consider the probability of getting the terminal wealth lower than 60% of the desired target, the time-consistent strategy appears to be less risky as shown in Figure 6.3(c).

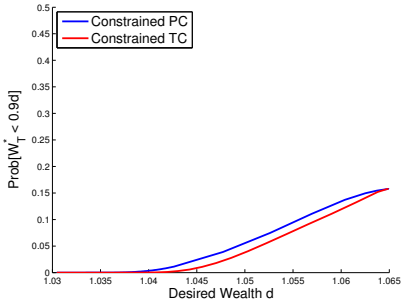
6.4.2. 2D PROBLEM WITH UNEXPECTED CORRELATION

An advantage of using the simulation-based numerical algorithms proposed in Chapters 4 and 5 is that they can be generalized to higher-dimensional scenarios. In this part, we consider a portfolio with two risky assets and one risk-free asset. In terms of model uncertainty, we consider a scenario where the correlation between two risky assets is not as predicted. The parameters for risky assets A and B are respectively shown in Table 6.3, where ρ denotes the correlation between the two risky assets in the model and ρ_{real} denotes the observed real-world correlation in the market. In the constrained case, we consider bounded leverage constraints $[x_{\min}, x_{\max}] = [0, 0.5]$ on both risky assets. For the other parameters, we set them as in Table 6.1.

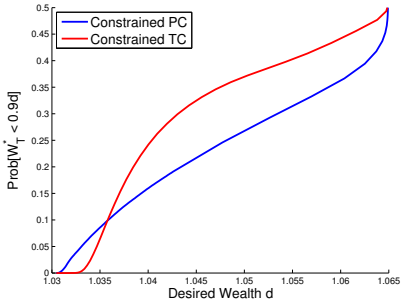
Table 6.3: Parameters for two risky assets A and B.

$\sigma_A = 0.2, \xi_A = 0.5, \sigma_B = 0.4, \xi_B = 0.5$
$\rho = -0.9, \rho_{\text{real}} = 0.9.$

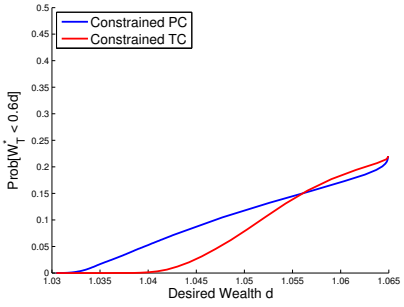
In the unconstrained case, we first check what kind of results a correlation prediction error can bring. We compare the efficient frontiers generated by the pre-commitment and time-consistent strategies, when the correlation is inaccurately predicted, as shown in Table 6.3, as well as the frontiers generated by both strategies when the observed real-world correlation is predicted accurately ($\rho = \rho_{real}$). As shown in Figure 6.4, when accurate information is available, the pre-commitment frontier is slightly higher than the time-consistent frontier. However, when the model correlation is not accurate, the pre-commitment strategy degrades, while the time-consistent strategy does not change significantly in terms of mean-variance efficiency. The time-consistent strategy with inac-



(a) In an expected market



(b) In an unexpectedly poor market



(c) In an unexpectedly poor market

Figure 6.3: The shortfall probability of the pre-commitment and the time-consistent strategies in the constrained case. The x-axis displays the desired target wealth and the y-axis the shortfall probability. For "an unexpectedly poor market", we use the parameters from Set I in Table 6.2. For "an expected market", the same model parameters are used and the real-world market parameters are assumed to be identical to the model parameters.

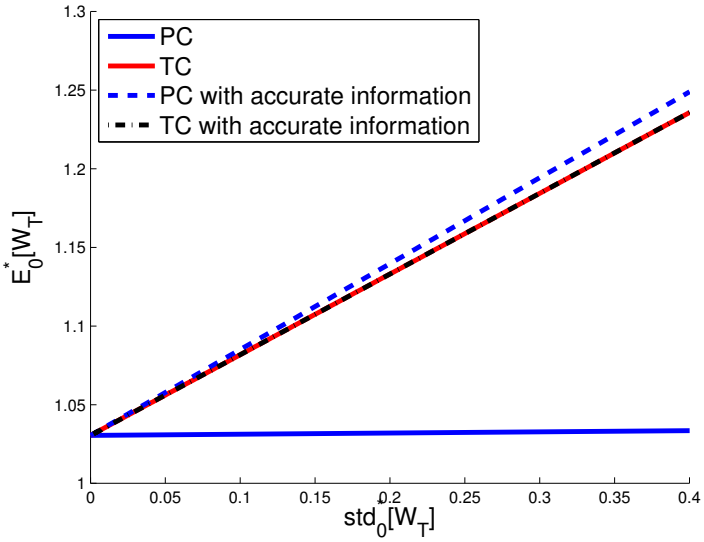


Figure 6.4: Comparison of the efficient frontiers generated by the pre-commitment and the time-consistent strategies in cases with or without accurate information of the correlation between risky assets.

6

curate information generates almost the same frontier as the one with accurate information. When we consider a test case with a longer investment horizon or fewer rebalancing opportunities, the difference between these two time-consistent frontiers will be more pronounced. However, in general, the time-consistent strategy is more robust than the pre-commitment strategy in terms of an inaccurate prediction of the assets correlation.

In Figure 6.5, we show the frontiers generated by the pre-commitment and the time-consistent strategies in the unconstrained and the constrained cases, when the asset correlation is not correctly predicted by the model. When the constraints are introduced on the allocations, the pre-commitment frontier increases and the time-consistent frontier decreases. In our test setting, when the mean return is not large, the time-consistent frontier is higher than its pre-commitment counterpart.

However, please note that when the misprediction of the correlation is not very significant, for example $[\rho, \rho_{\text{real}}] = [-0.2, 0.2]$, the constrained pre-commitment strategy still generates higher frontiers. This is as expected, since the pre-commitment strategy should generate higher frontiers when the market information is known exactly.

In Figure 6.6, we show the asset allocations of both strategies over time. We choose a scenario where both strategies generate mean returns that are equal to 1.13. As shown in Figure 6.5, this is approximately the point where the time-consistent and the pre-commitment frontiers cross. The presented allocations are the average values of the allocations on all simulated paths. Although both strategies generate returns with the same mean and also similar variances, their allocations are significantly different. By

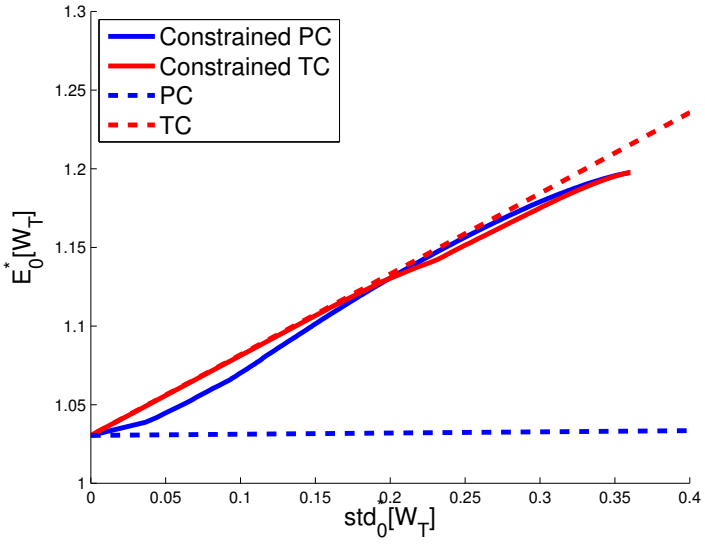


Figure 6.5: Comparison of the efficient frontiers when the correlations between risky assets are inaccurately predicted.

adopting the pre-commitment strategy, an investor assigns more money to the risky assets initially and shifts to risk-free asset allocations at the end of the investment period. This is due to the fact that this investor has a target in mind and close to the target she may not take risk to achieve higher wealth levels. For a time-consistent investor, the optimal asset allocations are quite different. Since the time-consistent investor is not satisfied with a target, the strategy does not reduce to a risk-free strategy. Initially, a time-consistent investor is more risk-averse than a pre-commitment investor; however, at the end, the time-consistent investor appears to be more risk-seeking.

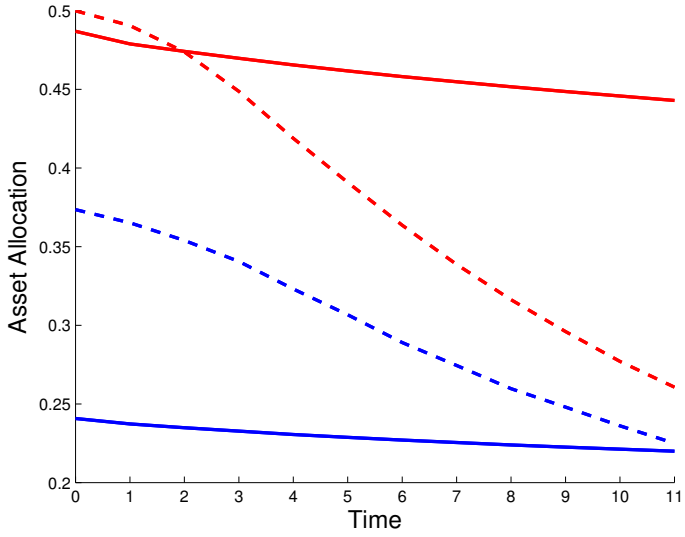


Figure 6.6: Comparison of asset allocations in the two-dimensional case. The red lines represent the allocations in asset A and the blue lines the allocations in asset B, which exhibits higher volatility than asset A. The straight lines denote the allocations for the time-consistent strategy and the dashed lines for the pre-commitment strategy.

6.5. CONCLUSION

In this chapter, we considered the robust pre-commitment and the robust time-consistent mean-variance optimization problems. In the *unconstrained case*, a specific equation for determining the worst-case scenario was derived for the robust pre-commitment and the robust time-consistent strategies. At a given time step, the optimal allocations generated by both strategies are then identical to their myopic counterparts, where an investor derives the optimal allocation for one upcoming time period, assuming that a risk-free strategy will be taken in the future.

The robustness of the pre-commitment and the time-consistent strategies is checked. Our analysis and the corresponding numerical experiments suggest that a time-consistent strategy appears to be more robust in terms of model errors. When an unexpectedly poor market is encountered, the time-consistent strategy may generate higher efficient frontiers than the pre-commitment strategy. Introducing constraints into the robust pre-commitment strategy can even increase the frontiers, since the constraints may serve as a "protection". In the two-dimensional case, the influence of inaccurately predicting the correlation between risky asset returns was examined. Again we found that the pre-commitment strategy may be vulnerable to such prediction errors and constraints on asset allocations can increase a pre-commitment frontier. Meanwhile, the time-consistent strategy still performed in a robust way. We checked the asset allocations of both strategies when they generate similar mean-variance pairs, and found that a pre-commitment investor prefers to bear more risk at the beginning of an investment period while a time-consistent investor appears to be more risk-seeking than a pre-commitment investor at the end of the period.

CHAPTER 7

Conclusions and Outlook

7.1. Conclusions

In this thesis, we have solved various types of multi-period stochastic control problems via numerical approaches. Our methods are based on Monte Carlo simulation and least-squares regression. In order to make our methods robust and accurate, we learned from the Stochastic Grid Bundling Method proposed in [57] and adopted the "regress-later" and "bundling" techniques when calculating conditional expectations. In all the test scenarios, our methods generate highly satisfactory results.

In Chapter 2, we considered a Bermudan option pricing problem with Merton jump-diffusion asset dynamics. To successfully solve this problem required us to determine the optimal exercise policy of the option accurately. We investigated how to choose basis functions in the least-squares regression process and how to set up bundling to achieve good quality results. Based on these discussions, we established a uniform way to configure our numerical methods: we always choose polynomials as basis functions and perform "equal-size bundling". This constituted the guidelines for configuring the "regress-later" and "bundling" techniques in this thesis. We also performed error analysis on our method and the standard regression method.

In Chapter 3, we enhanced a well-known dynamic portfolio management algorithm, the BGSS algorithm, proposed in [16]. We improved the simulation-based BGSS algorithm with the "regress-later" and "bundling" techniques. We found that the modified algorithm could generate robust results. Since both the BGSS algorithm and the algorithm proposed by us rely on a Taylor expansion to approximate value functions, we also introduced an idea, which is similar to [42], for increasing the accuracy of the Taylor expansion. With the modified Taylor expansion, highly accurate results can be obtained especially for a portfolio management problem with a long time horizon. Reflecting on the grid-searching idea in [85], we utilized a Fourier cosine series technique [37, 38] to compute the conditional expectations and came up with a benchmark algorithm. In most of the test cases, combining the modified simulation-based algorithm with the modified Taylor expansion yielded the most approving results, that were confirmed by using our benchmark algorithm.

In Chapters 2 and 3, the optimal controls depended on the state variables, which can be simulated independently without introducing biases. This implies that, when solving the Bermudan option pricing problem and the multi-period utility-based portfolio optimization problem, we can first finalize the Monte Carlo simulation and then calculate the optimal controls. However, this is not the case for either the pre-commitment or the time-consistent mean-variance portfolio optimization problem, respectively discussed in Chapters 4 and 5. The main difficulty encountered in these optimization problems

was that the state variables, by which we determined the optimal controls, evolve in turn depending on the controls cast by us. Therefore, we cannot simulate the state variables without taking the controls into account. We proposed a forward-backward simulation-based algorithm for solving these problems. In the forward step, we simulated the state variables with a rough guess of the optimal controls. In the backward step, we solved optimality locally, computed the value functions and, if necessary, updated the controls used in the forward step. After iterating these forward-backward process for several times, satisfactory results could be obtained. Moreover, for either the pre-commitment or the time-consistent mean-variance problem, we proved that the unconstrained optimal strategy is equivalent to a myopic strategy, by which an investor designs the optimal control for one coming period assuming that the risk-free strategy will be taken afterwards. In the constrained case, this myopic strategy serves as a reasonable guess of the optimal strategy and can be used to initiate the forward-backward algorithm.

In Chapter 6, we considered robust pre-commitment and time-consistent mean-variance strategies, that are required to perform well in a worst-case scenario. We showed that worst-case scenarios for both strategies can be found by solving a specific equation at each time step. We compared the efficient frontiers respectively generated by the pre-commitment and the time-consistent strategies when a model prediction error arises. Different from our findings in Chapter 5, we noticed that the time-consistent strategy may generate a higher mean-variance efficient frontier than a pre-commitment strategy when an unexpectedly poor market occurs. According to our numerical tests, in the 2D case the time-consistent strategy also appears to be more robust when the correlation of asset returns is mispredicted.

7.2. OUTLOOK

We suppose that the "regress-later" and "bundling" techniques can be used to improve most of the simulation-based numerical algorithms, that utilize cross-path least-squares regression to calculate conditional expectations. According to our experience, the improved numerical methods typically generate highly accurate and robust results. Improving currently existing numerical methods in this fashion may extend their limits and give rise to some new findings.

We chose geometric Brownian motion to model asset dynamics when we solved the mean-variance optimization problems. Some other models may also be considered, for example the vector auto-regression model as discussed in Chapter 2, to depict the asset dynamics. Using involved models to describe asset dynamics will essentially increase the dimensionality of the state variables. However, with a suitable bundling scheme, our simulation-based algorithm should still generate satisfactory results.

When we considered the portfolio optimization problems, we always solved the first-order conditions to obtain the optimality. Compared to grid-searching, solving first-order conditions is usually more efficient. However, we also noticed that solving the first-order conditions can be difficult for some problems, for example the mean-partial-variance optimization and the mean-CVaR optimization. For these problems. finding an efficient way to obtain the optimality can be a very challenging task. Nevertheless, the forward-backward algorithm may still form the basic framework for solving these problems.

When a model prediction error is encountered, the pre-commitment mean-variance strategy may suffer. A potential way to improve the pre-commitment efficient frontier is to take model uncertainty into account and perform machine learning techniques when new data become available. In the mean-variance portfolio optimization, the asset allocations are determined by the mean and the covariance of asset returns. The filtering approaches, like Kalman filters, especially the ensemble Kalman filter, may be helpful in this scenario.

References

[1] Søren Asmussen and Peter W Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57. Springer, 2007.

[2] Nicholas Barberis. Investing for the long run when returns are predictable. *The Journal of Finance*, 55(1):225–264, 2000.

[3] Jérôme Barraquand and Didier Martineau. Numerical valuation of high dimensional multivariate American securities. *Journal of Financial and Quantitative Analysis*, 30(03):383–405, 1995.

[4] Suleyman Basak and Georgy Chabakauri. Dynamic mean-variance asset allocation. *Review of Financial Studies*, 23(8):2970–3016, 2010.

[5] Richard Bellman. *Dynamic Programming*. Princeton Univ Press, 1957.

[6] Dimitri P Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific Belmont, MA, 1995.

[7] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[8] Michael J Best and Robert R Grauer. Sensitivity analysis for mean-variance portfolio problems. *Management Science*, 37(8):980–989, 1991.

[9] Eric Beutner, Janina Schweizer, and Antoon Pelsser. Fast convergence of Regress-Later estimates in least squares Monte Carlo. *arXiv preprint arXiv:1309.5274*, 2013.

[10] Tomasz R Bielecki, Hanqing Jin, Stanley R Pliska, and Xunyu Zhou. Continuous-time mean-variance portfolio selection with bankruptcy prohibition. *Mathematical Finance*, 15(2):213–244, 2005.

[11] Tomas Björk, Agatha Murgoci, and Xunyu Zhou. Mean-variance portfolio optimization with state-dependent risk aversion. *Mathematical Finance*, 24(1):1–24, 2014.

[12] Fischer Black and Robert Litterman. Global portfolio optimization. *Financial Analysts Journal*, 48(5):28–43, 1992.

[13] Stephen Boyd, Mark Mueller, Brendan O'Donoghue, and Yang Wang. Performance bounds and suboptimal policies for multi-period investment. *Foundations and Trends in Optimization*, 1(1):1–69, 2013.

[14] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge Univ Press, 2004.

[15] Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8):1267–1321, 1997.

[16] Michael W Brandt, Amit Goyal, Pedro Santa-Clara, and Jonathan R Stroud. A simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, 18(3):831–873, 2005.

[17] Mark Britten-Jones. The sampling error in estimates of mean-variance efficient portfolio weights. *The Journal of Finance*, 54(2):655–671, 1999.

[18] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8):1323–1352, 1997.

[19] Mark Broadie and Paul Glasserman. A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance*, 7:35–72, 2004.

[20] Jacques F Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: Mathematics and Economics*, 19(1):19–30, 1996.

[21] Zhi-ping Chen, Gang Li, and Ju-e Guo. Optimal investment policy in the time consistent mean–variance formulation. *Insurance: Mathematics and Economics*, 52(2):145–156, 2013.

[22] Charles E Clark. The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162, 1961.

[23] John H Cochrane. The sensitivity of tests of the intertemporal allocation of consumption to near-rational alternatives, 1989.

[24] Fei Cong and Cornelis W Oosterlee. Pricing Bermudan options under Merton jump-diffusion asset dynamics. *International Journal of Computer Mathematics*, 92(12):2406–2432, 2015.

[25] Fei Cong and Cornelis W Oosterlee. Accurate and robust numerical methods for the dynamic portfolio management problem. *Computational Economics*, pages 1–26, 2016.

[26] Fei Cong and Cornelis W Oosterlee. Multi-period mean-variance portfolio optimization based on Monte-Carlo simulation. *Journal of Economic Dynamics and Control*, 64(1):23–38, 2016.

[27] Fei Cong and Cornelis W Oosterlee. On pre-commitment aspects of a time-consistent strategy for a mean-variance investor. *Journal of Economic Dynamics and Control*, 70(1):178–193, 2016.

[28] Xiangyu Cui, Jianjun Gao, Xun Li, and Duan Li. Optimal multi-period mean-variance policy under no-shorting constraint. *European Journal of Operational Research*, 234(2):459–468, 2014.

[29] Xiangyu Cui, Duan Li, Shouyang Wang, and Shushang Zhu. Better than dynamic mean-variance: Time inconsistency and free cash flow stream. *Mathematical Finance*, 22(2):346–378, 2012.

[30] Xiangyu Cui, Xun Li, Duan Li, and Yun Shi. Time consistent behavior portfolio policy for dynamic mean-variance formulation. *Available at SSRN 2480299*, 2014.

[31] Xiangyu Cui, Lu Xu, and Yan Zeng. Continuous time mean-variance portfolio optimization with piecewise state-dependent risk aversion. *Optimization Letters*, pages 1–11, 2015.

[32] Christoph Czichowsky. Time-consistent mean-variance portfolio selection in discrete and continuous time. *Finance and Stochastics*, 17(2):227–271, 2013.

[33] Duy-Minh Dang and Peter A Forsyth. Continuous time mean-variance optimal portfolio allocation under jump diffusion: An numerical impulse control approach. *Numerical Methods for Partial Differential Equations*, 30(2):664–698, 2014.

[34] Duy-Minh Dang and Peter A Forsyth. Better than pre-commitment mean-variance portfolio allocation strategies: a semi-self-financing Hamilton–Jacobi–Bellman equation approach. *European Journal of Operational Research*, 250(3):827–841, 2016.

[35] Duy-Minh Dang, Peter A Forsyth, and Ken Vetzal. The 4% strategy revisited: A pre-commitment optimal mean-variance approach in wealth management. *Quantitative Finance*, pages 1–17, 2016.

[36] Laurent El Ghaoui and A Nilim. Robust solutions to Markov decision problems with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[37] Fang Fang and Cornelis W Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848, 2008.

[38] Fang Fang and Cornelis W Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.

[39] Peter A Forsyth and Kenneth R Vetzal. Long term asset allocation for the patient investor. *Working paper*, 2014.

[40] Chenpeng Fu, Ali Lari-Lavassani, and Xun Li. Dynamic mean-variance portfolio selection with borrowing constraint. *European Journal of Operational Research*, 200(1):312–319, 2010.

[41] Michael C Fu, Scott B Laprise, Dilip B Madan, Yi Su, and Rongwen Wu. Pricing American options: A comparison of Monte Carlo simulation approaches. *Journal of Computational Finance*, 4(3):39–88, 2001.

[42] Lorenzo Garlappi and Georgios Skoulakis. Numerical solutions to dynamic portfolio problems: The case for value function iteration using Taylor approximation. *Computational Economics*, 33(2):193–207, 2009.

[43] Lorenzo Garlappi and Georgios Skoulakis. Taylor series approximations to expected utility and optimal portfolio choice. *Mathematics and Financial Economics*, 5(2):121–156, 2011.

[44] Russell Gerrard, Steven Haberman, and Elena Vigna. Optimal investment choices post-retirement in a defined contribution pension scheme. *Insurance: Mathematics and Economics*, 35(2):321–342, 2004.

[45] Russell Gerrard, Steven Haberman, and Elena Vigna. The management of decumulation risks in a defined contribution pension plan. *North American Actuarial Journal*, 10(1):84–110, 2006.

[46] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer, 2004.

[47] Paul Glasserman and Bin Yu. Number of paths versus number of basis functions in American option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004.

[48] Paul Glasserman and Bin Yu. Simulation for American options: regression now or regression later? In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226. Springer, 2004.

[49] Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

[50] Christian Gollier and Richard J Zeckhauser. Horizon length and portfolio risk. *Journal of Risk and Uncertainty*, 24(3):195–212, 2002.

[51] Nalan Gülpınar and Berç Rustem. Worst-case robust decisions for multi-period mean–variance portfolio optimization. *European Journal of Operational Research*, 183(3):981–1000, 2007.

[52] Steven Haberman and Elena Vigna. Optimal investment strategies and risk measures in defined contribution pension schemes. *Insurance: Mathematics and Economics*, 31(1):35–69, 2002.

[53] Bjarni V Halldórsson and Reha H Tütüncü. An interior-point method for a class of saddle-point problems. *Journal of Optimization Theory and Applications*, 116(3):559–590, 2003.

[54] Ying Hu, Hanqing Jin, and Xun Yu Zhou. Time-inconsistent stochastic linear–quadratic control. *SIAM Journal on Control and Optimization*, 50(3):1548–1572, 2012.

[55] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

[56] David H Jacobson and David Q Mayne. *Differential Dynamic Programming*. American Elsevier Pub. Co., 1970.

[57] Shashi Jain and Cornelis W Oosterlee. The Stochastic Grid Bundling Method: Efficient pricing of Bermudan options and their Greeks. *Applied Mathematics and Computation*, 269(1):412–431, 2015.

[58] Eric Jondeau and Michael Rockinger. Optimal portfolio allocation under higher moments. *European Financial Management*, 12(1):29–55, 2006.

[59] Patrik Karlsson, Shashi Jain, and Cornelis W Oosterlee. Counterparty credit exposures for interest rate derivatives using the stochastic grid bundling method. *Available at SSRN 2538173*, 2014.

[60] Jang Ho Kim, Woo Chang Kim, and Frank J Fabozzi. Recent developments in robust portfolios with a worst-case approach. *Journal of Optimization Theory and Applications*, 161(1):103–121, 2014.

[61] Steven G Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

[62] Daniel Kuhn, Panos Parpas, Berç Rustem, and Raquel Fonseca. Dynamic mean-variance portfolio analysis under model risk. *Journal of Computational Finance*, 12(91115):7, 2009.

[63] Duan Li and Wan-Lung Ng. Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 10(3):387–406, 2000.

[64] Xun Li, Xunyu Zhou, and Andrew EB Lim. Dynamic mean-variance portfolio selection with no-shorting constraints. *SIAM Journal on Control and Optimization*, 40(5):1540–1555, 2002.

[65] Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

[66] Kai Ma and Peter A Forsyth. Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation under stochastic volatility. *Journal of Computational Finance*, pages 1–37, 2016.

[67] Kiran Manda. *Stock market volatility during the 2008 financial crisis*. PhD thesis, Stern School of Business, New York University, 2010.

[68] Harry M Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[69] Robert C Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, pages 247–257, 1969.

[70] Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1):125–144, 1976.

[71] Cédric Perret-Gentil and Maria-Pia Victoria-Feser. Robust mean-variance portfolio selection. *Available at SSRN 721509*, 2005.

[72] Mustafa Ç Pınar. On robust mean-variance portfolios. *Optimization*, pages 1–10, 2016.

[73] Nicki S Rasmussen. Control variates for Monte Carlo valuation of American options. *Journal of Computational Finance*, 9(1):83–118, 2005.

[74] Marjon J Ruijter and Cornelis W Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.

[75] Paul A Samuelson. Lifetime portfolio selection by dynamic stochastic programming. *The Review of Economics and Statistics*, pages 239–246, 1969.

[76] William F Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

[77] Jack Sherman and Winifred J Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[78] Joëlle Skaf and Stephen Boyd. Multi-period portfolio optimization with constraints and transaction costs. Technical report, Standard University, 2008.

[79] Lars Stentoft. Value function approximation or stopping time approximation: a comparison of two recent numerical methods for American option pricing using simulation and regression. *Journal of Computational Finance*, 18(1):1–56, 2014.

[80] Yuval Tassa, Tom Erez, and William D Smart. Receding horizon differential dynamic programming. In *Advances in Neural Information Processing Systems*, pages 1465–1472, 2008.

[81] James A Tilley. Valuing American options in a path simulation model. *Transactions of the Society of Actuaries*, 45(83):55–67, 1993.

[82] John N Tsitsiklis and Benjamin Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001.

[83] Reha H Tütüncü and M Koenig. Robust asset allocation. *Annals of Operations Research*, 132(1-4):157–187, 2004.

[84] Jules H van Binsbergen and Michael W Brandt. Optimal asset allocation in asset liability management. Technical report, National Bureau of Economic Research, 2007.

[85] Jules H van Binsbergen and Michael W Brandt. Solving dynamic portfolio choice problems by recursing on optimized portfolio weights or on the value function? *Computational Economics*, 29(3-4):355–367, 2007.

[86] Elena Vigna. On efficiency of mean-variance based portfolio selection in defined contribution pension schemes. *Quantitative Finance*, 14(2):237–258, 2014.

[87] J Wang and Peter A Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 46(3):1580–1601, 2008.

[88] Jian Wang and Peter A Forsyth. Numerical solution of the Hamilton-Jacobi-Bellman formulation for continuous time mean variance asset allocation. *Journal of Economic Dynamics and Control*, 34(2):207–230, 2010.

[89] Jian Wang and Peter A Forsyth. Continuous time mean variance asset allocation: A time-consistent strategy. *European Journal of Operational Research*, 209(2):184–201, 2011.

[90] Jian Wang and Peter A Forsyth. Comparison of mean variance like strategies for optimal asset allocation problems. *International Journal of Theoretical and Applied Finance*, 15(02):1250014, 2012.

[91] Xunyu Zhou and Duan Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics & Optimization*, 42(1):19–33, 2000.

[92] Shu-Shang Zhu, Duan Li, and Shou-Yang Wang. Risk control over bankruptcy in dynamic portfolio selection: A generalized mean-variance formulation. *IEEE Transactions on Automatic Control*, 49(3):447–457, 2004.

Curriculum Vitæ

Fei CONG

22-08-1989 Born in Xi'an, China.

EDUCATION

2007–2011 Bachelor of Science
Major: Applied Mathematics
Minor: Statistics
Shanghai University of Finance and Economics, Shanghai, China

2011–2013 Master Applied Mathematics
Delft University of Technology, Delft, the Netherlands

2013–Now PhD Applied Mathematics
Delft University of Technology, Delft, the Netherlands
Thesis: Stochastic Optimal Control Based on Monte Carlo Simulation and Least-Squares Regression
Promotor: Prof. dr. ir. C. W. Oosterlee

List of Publications

5. **F. Cong** and **C. W. Oosterlee**, On robust multi-period mean-variance porfolio optimization, submitted for publication, 2016.

4. **F. Cong** and **C. W. Oosterlee**, On pre-commitment aspects of a time-consistent strategy for a mean-variance investor, *Journal of Economic Dynamics and Control*, 70(1):178–193, 2016.

3. **F. Cong** and **C. W. Oosterlee**, Multi-period mean-variance portfolio optimization based on Monte-Carlo simulation, *Journal of Economic Dynamics and Control*, 64(1):23–38, 2016.

2. **F. Cong** and **C. W. Oosterlee**, Accurate and robust numerical methods for the dynamic portfolio management problem, Computational Economics, pages 1–26, 2016.

1. **F. Cong** and **C. W. Oosterlee**, Pricing Bermudan options under Merton jump-diffusion asset dynamics, International Journal of Computer Mathematics, 92(12):2406–2432, 2015.

List of Attended Conferences

Presentations:

3. Financial Mathematics Winter School, Lunteren, the Netherlands, February 2016.
2. International Conference of Computational Finance, Greenwich, Britain, December 2015.
1. International Conference of Industrial and Applied Mathematics, Beijing, China, August 2015.

Posters:

2. Workshop on Models and Numerics in Financial Mathematics, Leiden, the Netherlands, May 2015.
1. Actuarial and Financial Mathematics Conference, Brussels, Belgium, February 2015.

Acknowledgement

First and foremost, my sincere appreciation goes to my supervisor and promotor Kees Oosterlee. Without his leading and support, I could not have finished my PhD project. Kees always keeps active communication with me. When I am either depressed or excited by my research outcome, he gives me feedback at the earliest time, encouraging me to go further.

I would like to thank all the committee members for carefully reading my dissertation. Special thanks go to Prof. Forsyth and Prof. Schumacher for their detailed comments.

The financial engineering research group formed by the students of Kees is like a big family. Everyone can share knowledge with and get feedback from others. I appreciate the discussions with Alvaro, Andrea, Anton, Ki Wai, Lech, Marjon, Qian, Shashi, Yanbin and Zaza. I need to thank Nada especially for helping me to arrange the work related affairs every time when I was at CWI.

Working at the numerical analysis group in Delft gave me a lot of pleasure. Many thanks to all former and current group members: Abdul, Baljinnyam, Berna, Behrouz, Daniël, Deborah, Dennis, Domenico, Duncan, Edwin, Elwin, Fahim, Faraz, Fei, Fons, Fred, Gabriela, Guido, Guus, Jiao, Jing, Joanna, Jok, Joost, Kees(× 2), Lisa, Luis, Manuel, Martin, Martijn, Matthias, Menel, Mohamed, Neil, Peiyao, Reinaldo, Rohit, Thea, Valia, Virginia, Xiaozhou, Xin, Yue and Zaza. I enjoy every Tuesday when my admirable office mates Peter and Jos come. Their passion for research cheers me up. Special thanks to Zaza for helping me to translate the summary and the propositions into Dutch and for fetching hundreds of cups of "heet water" with me. I value the time with my former office mate Edwin, who spurred me to practice mine-sweeping, a very good game for mathematicians. I would like to thank Thea in particular as well for sharing the information regarding the graduation procedure and the thesis publication. I appreciate the course Stochastic Simulation given by Ludolf Meester and his patient discussion with me when I wrote my first article. Many thanks to Kees Lemmens for teaching the course Scientific Programming, for offering me a desktop for free and for inspiring me to install a Linux system on it by myself (during the process I learnt a lot about Linux).

Finally, I would like to express my gratitude to my families. My parents brought me up and taught me a lot. Now, as a dad, I know that taking care of a child is far more difficult than I imaged. Not to mention dealing with a stubborn person like me. I would like to thank my wife, Xi, for constantly believing in me. Meeting Xi is the luckiest moment in my life. To memorize this, I use your name (\Xi) in the luckiest finding point of my research. At last. thank you, Ella, for coming to us. Hope that you will be happy forever.

Fei Cong
Delft, November 2016