



---

## Masters Thesis in Economics

Alexander Michael Peytz

## Dynamic Portfolio Choice Under Various Transaction Costs and Asset Structures

A Dynamic Programming and Probabilistic Machine Learning Approach

Advisor: Bertel Schjerning

Submitted: November 6, 2024

# Dynamic Portfolio Choice Under Various Transaction Costs and Asset Structures<sup>\*</sup>

Masters Thesis  
Alexander Michael Peytz<sup>†</sup>

November 27, 2024

## **Abstract**

This thesis

---

<sup>\*</sup>I thank my supervisor Bertel Schjerning for his guidance and support.

<sup>†</sup>Department of Economics, University of Copenhagen. E-mail: kqw203@alumni.ku.dk.

## Table of Contents

---

<b>1</b>	<b>What has been the contributions in this paper for now (Not in final paper)</b>	<b>5</b>
<b>2</b>	<b>To-Do</b>	<b>5</b>
<b>3</b>	<b>Introduction</b>	<b>6</b>
<b>4</b>	<b>Literature review</b>	<b>7</b>
<b>5</b>	<b>The Dynamic Portfolio Choice Setting</b>	<b>9</b>
5.1	Asset and goods market . . . . .	9
5.2	Asset dynamics . . . . .	9
5.3	Transaction costs and portfolio reallocation . . . . .	10
5.4	Investor preferences and problem . . . . .	10
5.5	Intertemporal portfolio choice without transaction costs . . . . .	11
5.6	The general class of dynamic portfolio choice with transaction costs and intertemporal consumption . . . . .	11
5.7	No Trade Region . . . . .	14
5.8	Base problem: Portfolio with proportional costs and consumption . . . . .	15
5.9	Portfolio with fixed and proportional costs . . . . .	16
5.10	Portfolio with asset specific costs . . . . .	16
<b>6</b>	<b>Numerical implementation details</b>	<b>16</b>
6.1	Numerical integration . . . . .	16
6.1.1	Gauss-Hermite quadrature . . . . .	16
6.1.2	Monte Carlo integration (MC) . . . . .	19
6.1.3	Quasi-Monte Carlo integration (QMC) . . . . .	20
6.1.4	Randomized Quasi-Monte carlo integration (RQMC) . . . . .	22
6.2	Value function approximation . . . . .	22
6.2.1	Gaussian process regressions (GPR) . . . . .	23
6.3	Approximating the No trade region . . . . .	25
6.3.1	Strategic point sampling . . . . .	27
6.3.2	Utilising the NTR approximation for $\delta$ bounds . . . . .	28
6.3.3	Multiple Gaussian Process Regressions . . . . .	29
6.4	Final solution algorithms . . . . .	29
6.5	Computational stack and implementation . . . . .	30
6.5.1	Optimization details . . . . .	31

<b>7 Results</b>	<b>32</b>
7.1 Dynamic Portfolio Choice without consumption . . . . .	32
7.1.1 Verifying the geometric shape of the No-trade Region . . . . .	33
7.1.2 Investigating the No-Trade Region . . . . .	34
7.1.3 Increasing the dimensionality of the model . . . . .	35
7.2 Dynamic Portfolio Choice with consumption . . . . .	36
7.3 Dynamic Portfolio Choice with fixed costs . . . . .	36
<b>Appendices A-F</b>	<b>41</b>

## List of Figures

---

5.1	Example No Trade Region with $k = 2$ risky assets.	15
6.1	Comparison of sample generation for Monte Carlo and Quasi-Monte Carlo	21
6.2	Comparison of sample generation for Monte Carlo and Quasi-Monte Carlo with increased dimensionality	21
6.3	Illustration of the no-trade region (NTR) and the optimal policies outside this.	26
6.4	The designed sampling strategy for state space coverage.	28
7.1	Comparison of No Trade Regions.	33
7.2	Verifying the assumptions of the NTR in 2 dimensions.	34
7.3	No Trade Region for Schober Parameters over Time.	35
7.4	No Trade Region for the iid Parameters with different values of $\tau$ .	35
7.5	Comparison of No Trade Regions.	36
7.6	Comparison of No Trade Regions over time with consumption.	37
7.7	No trade regions with consumption in multiple dimensions.	37
B.1	Uniform grid sampling strategy	42
B.2	Naive random sampling strategy	42

## List of Tables

---

1	Parameters for Examples of Portfolio Problems	32
---	---	----

## Abbreviations

---

<b>DP</b>	dynamic programming
<b>GP</b>	Gaussian process
<b>GPR</b>	Gaussian process regression
<b>LDS</b>	low-discrepancy sequences
<b>MC</b>	Monte Carlo
<b>MPT</b>	randomized quasi-Monte Carlo
<b>MPT</b>	quasi-Monte Carlo
<b>MPT</b>	modern portfolio theory
<b>NTR</b>	no-trade-region
<b>SKIP</b>	Structured Kernel Interpolation for Products

## 1 What has been the contributions in this paper for now (Not in final paper)

---

- Bridging Cai, Judd and Xu (2020) and Gaegauf, Scheidegger and Trojani (2023). By adding the following to the framework of Gaegauf, Scheidegger and Trojani (2023):
  - By adding time period spacing  $\Delta t$ . Adds dimensions from behavioural finance, and continuous finance theory.
  - Adding  $\theta_t$  **maybe** (Not coded!)
  - By adding option pricing (Not coded!). Adds dimensions from derivatives pricing. Complex asset dynamics.
- By adding other transaction cost types. Such as:
  - Fixed transaction costs (Not coded!)
  - Asset specific transaction costs (Not coded!)
- Computationally:
  - Using Kernel interpolation for the GP (Speeding up Kernel Matrix Computations)

## 2 To-Do

---

- Economic theory
  - NTR Theory (Dybvig etc)
  - Option Price theory (Black-Scholes).
  - Transaction cost theory (Fixed, asset specific (Just formulations))
  - Price impact theory **maybe** (See Garlenau etc).
- Portfolio choice models
  - Model with options (After option theory)
  - Model with other transaction costs
- Skrive implementation details
  - Final Algorithms
- Kode:

- Option pricing, new: Exotic options.
- Implement other transaction cost types
- Omskrive
  - Litterature reveiw. Add Liu 2002. Order by: Merton then Finance and mathematics guys. then Computation guys. then Cai Judd then Scheidegger then Me.

### 3 Introduction

---

Dynamic portfolio choice problems consider the optimal portfolio construction over time. These have a general solution in the absence of market frictions. When frictions are introduced, the problem becomes significantly more realistic, as investors face costs when trading assets. However, this increased realism comes at a tradeoff of increased complexity in the problem, as the optimal portfolio construction is no longer trivial to find. In Dynamic Portfolio choice dynamic programming (DP) schemes have been implemented to solve these problems numerically, but the computational complexity of these schemes suffer from the curse of dimensionality in a multitude of ways using multiple grid-based methods. In this regard the work of Gaegauf, Scheidegger and Trojani (2023) is of particular interest, as they develop a computational framework which reduces the need for grid-based methods. While much work has been put to developing a computational framework which reduces the need for grid-based methods, this has not been applied to a broader set of portfolio choice models, and we therefore only have a limited idea of the scope of applicability of these methods.

I therefore extend the framework of Gaegauf, Scheidegger and Trojani (2023), to new asset types and new cost functions, to broaden the scope of models which can be solved using this framework, and to provide a broader understanding of the class of dynamic portfolio choice problems. I analyse the impact of introducing various transaction costs types, such as fixed costs, and asset specific costs, including the propotional transaction costs often seen in the litterature. Furthermore i broaden the investment universe to include multiple asset types, such as stocks, bonds and vanilla options. This paper therefore aims to provide a broader understanding of the class of dynamic portfolio choice problems, utilizing the newest insights in computational methods seen in the litterature.

Furthermore a novel extension to the computational framework is provided, which aims to reduce the computational burden in higher dimensions. The framework suffers in higher dimensions, as the number of grid points increase, but also because the function approximation which leverages Gaussian process (GP) becomes more complex. I introduce Structured Kernel Interpolation for Products (SKIP), which has been shown to increase the efficiency of the GP when dimensionality is increased.

I implement this framework on parametrization analyzed earlier in the litterature, and compare the results to the existing literature. Following this i extend the framework to include options, as seen in Cai, Judd and Xu (2020), and new cost functions, as seen in Dybvig and Pezzo (2020).

## 4 Literature review

---

The purpose of this section is to review relevant literature to help understand the contributions made in this thesis. This review covers modern portfolio theory (MPT), from its foundations and into the 21st century.

Modern theory on portfolio choice can be traced back to the mean-variance framework of Harry Markowitz, who constructed and solved the now well established, static and single period, portfolio optimization problem, Markowitz (1952). This covers the mean-variance framework which is the foundation of MPT, suggesting investors should allocate wealth in order to maximize expected return, while minimizing exposure to risk. Following this, the mean-variance framework has since been extended to a continious time setting, most notably by Robert Merton, who introduced a solution to the intertemporal portfolio choice problem in frictionless markets, Merton (1969), and later adding consumption rules aswell Merton (1971). This solution is known as the Merton point in the asset allocation space, or the Merton portfolio. Mertons closed form solution suggests optimal asset allocations based on the asset return dynamics (mean-variance), and the risk aversion of the investor (preferences). Hence in a continous time setting, the optimal allocation changes if the asset dynamics change.

Multiple extensions have been made to the classical dynamic portfolio choice problem, such as the introduction of transaction costs, adding realistic constraints to the problem, since trading assets incurs costs in the real world, and markets are not frictionless. Zabel (1973) adresses transaction costs with CRRA preferences, but is limited to a discrete time setting, a single risky asset and a small horizon.

Constantinides (1976) and Constantinides (1986) returns to the continious time setting, and find that for multiple preference types, under proportional transaction costs. The investors decision then depends on the the remaining life span, wealth and current allocation. Trading costs create a no-trade-region (NTR), where the optimal reallocation decision for portfolios inside this, is do to nothing, and for portfolios outside this region, the optimal decision is to trade towards the boundary of the NTR. This is a shift from Mertons framework, where constant trading toward the Merton allocation, which is the optimal allocation in the absence of transaction costs, is optimal. Hence transaction costs restrain investors from acting optimally in the classical sense.

Numerical examples only cover the case of one risky asset, with restrictions on the decision space, and results remain qualitative or approximate. Notably Davis and Norman

(1990) derive explicit solutions for the case of a single risky asset. They similarly find that proportional transaction costs lead to a NTR around the Merton point, and provide a solution algorithm for the stochastic control problem. This has later been made more rigorous such as Akian, Menaldi and Sulem (1996) who use a Hamilton-Jacobi-Bellman equation in the  $N$ -dimensional asset space, and provide further insight to the properties of the NTR, however the problem is only solved for the case of  $k = 2$  risky assets with one risk free asset. Further analysis of this has been conducted extensively, e.g see Shreve and Soner (1994), Oksendal and Sulem (2002), Janeček and Shreve (2004), however the asset space is still constrained or solutions remain asymptotic. Muthuraman and Kumar (2006) and Muthuraman and Kumar (2008) tackle a  $D = 3$  risky asset space, and provide a numerical solution to the problem, using a finite differences.

The paper by Cai, Judd and Xu (2013), which is central to this thesis, consider a more general setting, with multiple risky assets and a risk-free asset, and provides a solution algorithm, based on dynamic programming, numerical integration and polynomial approximation, to solve the dynamic problem for up to  $k = 6$  risky assets and thus  $D = 7$  assets in total, and later introduce and solve the problem with novelties, such as stochastic asset parameters or an option on an underlying asset in the portfolio Cai, Judd and Xu (2020). The curse of dimensionality, which haunts the prior methods applied, is somewhat tackled by the use of adaptive sparse grid methods, and sparse quadrature rules by Schober, Valentin and Pflüger (2022).

Gaegau, Scheidegger and Trojani (2023) further reduces the computational burden by using a Gaussian process regression to approximate value functions, and a problem specific point sampling strategy to reduce the number of points in the state space needed to characterize the NTR. Increasing the dimensions of the asset space does still increase the dimensionality of the problem, and the computational burden, however this is at a much lower extent than previous methods.

Beyond the analysis conducted by the authors above, several related avenues of research have been conducted on the dynamic portfolio choice problem. Garleanu and Pedersen (2013) remains an influential paper, which aims to derive optimal closed form portfolio policy, when returns are driven by signals with mean reversion. This provides an insightful analysis of how to trade towards the optimal portfolio, given quadratic transaction costs, within a set scope of serially correlated assets. Dybvig and Pezzo (2020) provides a comprehensive overview on the usage of different transaction cost functions, hedging with futures and security specific costs. Dybvig find that by changing the transaction cost function, the properties of the NTR is altered.<sup>1</sup>

---

<sup>1</sup>Gaegau, Scheidegger and Trojani (2023), also note that their framework is applicable to different transaction cost functions.

## 5 The Dynamic Portfolio Choice Setting

---

This section covers the basics of modern portfolio theory and components of the dynamic portfolio choice problem with transaction costs. This section leans heavily on Cai, Judd and Xu (2020) and Gaegauf, Scheidegger and Trojani (2023), bridging the model from the former, with the framework of the latter.

### 5.1 Asset and goods market

We consider a financial market with  $k$  risky assets and one risk-free asset, making the asset space  $D = 1 + k$  dimensions. The risk-free asset, such as a bond or a bank deposit, yields a constant gross return  $R_f = e^{r\Delta t}$ , where  $r$  is the annual interest rate and  $\Delta t = \frac{T}{N}$  is the length of one investment period.

The  $k$  risky assets can be considered as listed stocks, subject to proportional transaction costs. For each reallocation of wealth in a risky asset, a transaction cost of  $\tau \in [0, 1]$  is incurred as a percentage of the traded amount. The stochastic one-period gross-return vector of the risky assets is denoted as  $\mathbf{R} = (R_1, R_2, \dots, R_k)^\top$ , and the corresponding net-return vector is  $\mathbf{r} = (r_1, r_2, \dots, r_k)^\top$ .

In the goods market, there is a single non-durable consumption good,  $C$ , which is consumed at each time point  $t$ . The fraction of wealth allocated to consumption at time  $t$  is denoted  $c_t$ , the fraction allocated to risky assets is  $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \dots, x_{k,t})^\top$ , and the fraction allocated to the risk-free asset is denoted  $b_t$ . Thus,  $\mathbf{x}_t \in \mathbb{R}^k$  and  $b_t \in \mathbb{R}$ .

### 5.2 Asset dynamics

I follow Cai, Judd and Xu (2013) for the asset dynamics. The total composition of risky assets is assumed to follow a multivariate log-normal distribution:

$$\log(\mathbf{R}) \sim \mathcal{N} \left( \left( \mu - \frac{\sigma^2}{2} \right) \Delta t, (\mathbf{\Lambda} \mathbf{\Sigma} \mathbf{\Lambda}) \Delta t \right), \quad (1)$$

where  $\mu$  is the drift vector,  $\sigma^2$  is a column vector of the variance  $\sigma_i^2$ ,  $\mathbf{\Sigma}$  is the correlation matrix, and  $\mathbf{\Lambda} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k)$  is the diagonal matrix of volatilities. Following Cai, Judd and Xu (2013) we utilize the Cholesky decomposition of the correlation matrix,  $\mathbf{\Sigma} = \mathbf{L} \mathbf{L}^\top$ , where  $\mathbf{L} = (L_{i,j})_{k \times k}$  is a lower triangular matrix. Hence, for each risky asset  $i$ , the log-return is:

$$\log(R_i) = \left( \mu_i - \frac{\sigma_i^2}{2} \right) \Delta t + \sigma_i \sqrt{\Delta t} \sum_{j=1}^i L_{i,j} z_j, \quad (2)$$

where  $z_i$  are independent standard normal random variables.

### 5.3 Transaction costs and portfolio reallocation

Rebalancing incurs proportional transaction costs  $\tau \in [0, 1]$ , which are paid based on the amount bought or sold of each risky asset. Reallocation decisions are made just before  $t_j + \Delta t$ , such that  $\mathbf{x}_t$  is the portfolio of risky assets right before reallocation.  $\delta_{i,t}$  denotes the change in portfolio allocation of asset  $i$ , and  $\delta_{i,t}W_t$  is thus the currency amount traded in asset  $i$ . Hence  $\delta_{i,t} > 0$  implies buying asset  $i$ , and  $\delta_{i,t} < 0$  implies selling asset  $i$ . Proportional transaction costs imply that the cost function associated with rebalancing is:

$$\psi(\delta_{i,t}W_t) = \tau|\delta_{i,t}W_t| \quad (3)$$

I decompose the decision variable  $\delta_{i,t}$ , representing the fraction of wealth used to trade risky asset  $i$ , into buying ( $\delta_{i,t}^+$ ) and selling ( $\delta_{i,t}^-$ ) components to ensure tractability<sup>2</sup>:

$$\delta_{i,t} = \delta_{i,t}^+ - \delta_{i,t}^-, \quad \delta_{i,t}^+, \delta_{i,t}^- \geq 0.$$

The total transaction cost is then given by  $\tau \sum_{i=1}^k (\delta_{i,t}^+ + \delta_{i,t}^-)W_t$ . And the transaction cost function is therefore a function of each trading direction:

$$\psi(\delta_{i,t}^+, \delta_{i,t}^-, W_t) = \tau(\delta_{i,t}^+ + \delta_{i,t}^-)W_t \quad (4)$$

Following the reallocation, the remaining wealth is allocated between the risk-free asset and consumption. Notation of rebalancing is henceforth simplified using vectors to  $\boldsymbol{\delta}_t = \boldsymbol{\delta}_t^+ - \boldsymbol{\delta}_t^-$  with  $\boldsymbol{\delta}_t^+ = (\delta_{1,t}^+, \delta_{2,t}^+, \dots, \delta_{k,t}^+)$ . We have that  $\boldsymbol{\delta}_t$  is the *net change* in the risky positions, and  $\boldsymbol{\delta}_t^+ + \boldsymbol{\delta}_t^-$  is the *cumulative change* in the risky positions.

### 5.4 Investor preferences and problem

The investor operates over a finite horizon of  $T$  years, during which the aim is to maximize expected utility. Following Cai, Judd and Xu (2013), the investment horizon is discretized into  $N$  equally spaced periods, each with a duration of  $\Delta t = \frac{T}{N}$ . At each time point  $t_j$ , for  $j = 0, 1, \dots, N$ , where  $t_0 = 0$  and  $t_N = T$ , the investor has the opportunity to adjust the portfolio allocations right before  $t_j + \Delta t$ . Reallocation is costly, and the investor is subject to proportional transaction costs. If consumption is included the investor may also choose to consume a non-durable good at each time point.

For notational simplicity, I now use  $t$  to denote these time points unless specifically referring to  $t_j$ . The investor's preferences are modeled using a constant relative risk

---

<sup>2</sup>Gaegau, Scheidegger and Trojani (2023) note that this ensures differentiability. This approach is common and found in earlier work such as Akian, Menaldi and Sulem (1996), who likewise note that this ensures that the variable is continuous from origin in the positive real set.

aversion (CRRA) utility function:

$$u(C_t) = \begin{cases} \frac{C_t^{1-\gamma}}{1-\gamma} & \text{if } \gamma \neq 1, \\ \log(C_t) & \text{if } \gamma = 1, \end{cases} \quad (5)$$

where  $C_t$  is consumption and  $c_t$  is the fraction of wealth  $W_t$  spent on consumption at time  $t$ . Hence  $c_t = C_t/W_t$ , and lowercase notation is henceforth used to denote variables as fractions of wealth.  $\gamma$  is the coefficient of relative risk aversion. The objective is to maximize the expected utility of consumption and wealth over the investor's lifetime:

$$\max_{\mathbf{x}_t, b_t, c_t} \mathbb{E} \left[ \sum_{i=0}^{N-1} \beta^i u(C_i) \Delta t + \beta^N u(W_N) \right], \quad (6)$$

where  $\beta$  is the discount factor,  $\mathbf{x}_t$  is the allocation to risky assets,  $b_t$  is the allocation to the risk-free asset, and  $W_t$  is the investor's wealth at time  $t$ .

## 5.5 Intertemporal portfolio choice without transaction costs

When there are no transaction costs (no market frictions) the investor can freely rebalance the portfolio. This reduces the problem to a classic portfolio optimization problem formulated by Merton (1969) and Merton (1971). For a more detailed treatment, see Björk (2019). In this setting, the investor dynamically allocates wealth between  $k$  risky assets and a risk-free asset to maximize utility over a finite horizon  $[0, T]$ .

The investor's wealth  $W_t$  can be allocated between a risk-free asset and  $k$  risky assets. Consumption is a non-durable good that can be purchased at each time point  $t$ .  $r$  is the risk-free rate,  $\boldsymbol{\mu}$  is the vector of expected returns on the risky assets, and  $C_t$  represents consumption at time  $t$ . The investor's preferences follow a constant relative risk aversion (CRRA) utility function.

Without transaction costs, the optimal portfolio allocation, known as the Merton point is:

$$\mathbf{x}_t^* = \frac{1}{\gamma} \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - r), \quad (7)$$

where  $\gamma$  is the coefficient of relative risk aversion, and  $\boldsymbol{\Sigma}$  is the covariance matrix of the risky assets' returns. This provides a time-independent optimal allocation that serves as a benchmark for models incorporating frictions such as transaction costs.

## 5.6 The general class of dynamic portfolio choice with transaction costs and intertemporal consumption

Now consider when transaction costs are present, and the investor can consume a non-durable good at each time point. The solution to the dynamic portfolio choice problem

is no longer given by the closes form solution of the Merton point. Considering the components presented in this section, the class of dynamic portfolio optimization problems, given one risk free asset and  $k$  risky assets, can be formulated by the following Bellman equation, Bellman (1958)<sup>3</sup>:

$$V_t(W_t, \mathbf{x}_t, \theta_t) = \max_{c_t, \boldsymbol{\delta}_t^+, \boldsymbol{\delta}_t^-} \{u(c_t W_t) \Delta t + \beta \mathbb{E}_t [V_{t+\Delta t}(W_{t+\Delta t}, \mathbf{x}_{t+\Delta t}, \theta_{t+\Delta t})]\}, \quad t < T \quad (8)$$

Given some initial level of wealth  $W_0$  and portfolio allocation  $\mathbf{x}_0$ .  $\theta_t$  is a vector of stochastic variables, which the gross one period risk free return, and risky return depends on, i.e  $\mathbf{R}(\theta_t)$  and  $R_f(\theta_t)$ . These could cover the drift  $\mu$ , volatiliy  $\sigma^2$ , correlation of the risky assets  $\Sigma$ , and the risk free return  $r$  or only some of these, dependent on the model. Notice that future wealth and allocations are stochastic, as they depend on the future realization of  $\theta_t$ .

Notice that consumption and reallocation are decision variables, whereas bond holding are not (Explicitly). This is because bond holdings can be determined as the residual wealth, after consumption and reallocation decisions are made:

$$b_t W_t = \left(1 - \mathbf{1}^\top \cdot \mathbf{x}_t\right) W_t - \mathbf{1}^\top \cdot \boldsymbol{\delta}_t W_t - \psi(\boldsymbol{\delta}_t^+, \boldsymbol{\delta}_t^-, W_t) - c_t W_t \quad (9)$$

Where  $\psi(\cdot)$  is the transaction cost function, and  $\mathbf{1}$  is a vector of ones.

The dynamics of the state variables follow Schober, Valentin and Pflüger (2022) and are given by:

$$W_{t+\Delta t} = b_t W_t R_f(\theta_t) + ([\mathbf{x}_t + \boldsymbol{\delta}_t] W_t)^\top \cdot \mathbf{R}(\theta_t) \quad (10)$$

$$\mathbf{x}_{t+\Delta t} = \frac{((\mathbf{x}_t + \boldsymbol{\delta}_t) W_t) \odot \mathbf{R}_t(\theta_t)}{W_{t+\Delta t}} \quad (11)$$

Where  $\odot$  is the elementwise product (Hadamand product). The terminal value function is given by<sup>4</sup>:

$$V_T(W_T, \mathbf{x}_T, \theta_T) = u((W_T - \psi(\mathbf{x}_T W_T)) \cdot (1 - R_f(\theta_T))) \cdot \Delta t \cdot (1 - \beta)^{-1} \quad (12)$$

Which implies that the investor transfers all wealth to the bank account at the terminal period, and consumes out of the interest returns<sup>5</sup>. Finally we note that the optimization

---

<sup>3</sup>This is consolidated model of the base model, and with consumption model, of Cai, Judd and Xu (2020), however the cost function is generalized and correlation of returns is included.

<sup>4</sup>Stemming from the infinite sum of discounted utility of interest payments.

<sup>5</sup>This formulation stems from Cai, Judd and Xu (2013) and assumes that the investor lives forever. Gaegauf, Scheidegger and Trojani (2023) assumes that the investor consumes everything at the terminal period.

problem is subject to the following constraints:

$$\boldsymbol{\delta}_t W_t \geq -\mathbf{x}_t W_t \quad (13)$$

$$b_t W_t \geq 0 \quad (14)$$

$$\mathbf{1}^\top \mathbf{x}_t \leq 1 \quad (15)$$

The first constraint ensures that the investor does not short sell risky assets, The second constraint is also a no shorting constraint and the third is a no-borrowing constraint. Hence This formulation does not consider leveraged investments.

Furhtermore we can note that the rebalancing decision (in each direction), is only feasible in the space:

$$\delta_{i,t}^+ \in [0, 1 - x_{i,t}] \quad (16)$$

$$\delta_{i,t}^- \in [0, x_{i,t}] \quad (17)$$

This is a direct formulation of the constraints, already captured in the equations above. The problem can be simplified by normalizing wrt. wealth, and removing wealth as a state variable, since wealth is seperable from the rest of the state space  $\mathbf{x}_t, \theta_t$  as noted by Cai, Judd and Xu (2013).

This is because portfolio optimality is independent of wealth for CRRA utility function. The Bellman equation is then:

$$v_t(\mathbf{x}_t, \theta_t) = \max_{c_t, \delta_t^+, \delta_t^-} \{u(c_t)\Delta t + \beta \mathbb{E}_t \left[ \pi_{t+\Delta t}^{1-\gamma} v_{t+\Delta t}(\mathbf{x}_{t+\Delta t}, \theta_{t+\Delta t}) \right] \}, \quad t < T \quad (18)$$

The normalized bond holdings are then:

$$b_t = 1 - \mathbf{1}^\top \cdot (\mathbf{x}_t - \boldsymbol{\delta}_t - \psi(\boldsymbol{\delta}_t^+, \boldsymbol{\delta}_t^-)) - c_t \Delta t \quad (19)$$

We see that these are still the residual of the wealth after the rebalancing and consumption decision. Where we formulate the transaction cost function  $\psi(\cdot)$  in terms of the buying and selling components, and using changes to allocations proportional to wealth, instead of the prior formulations, where wealth was a direct input. The dynamics are then:

$$\pi_{t+\Delta t} = b_t R_f(\theta_t) + (\mathbf{x}_t + \boldsymbol{\delta}_t)^\top \cdot \mathbf{R}(\theta_t) \quad (20)$$

$$\mathbf{x}_{t+\Delta t} = \frac{(\mathbf{x}_t + \boldsymbol{\delta}_t) \odot \mathbf{R}_t(\theta_t)}{\pi_{t+\Delta t}} \quad (21)$$

$$W_{t+\Delta t} = \pi_{t+\Delta t} W_t \quad (22)$$

Where we now formulate the problem with regard to the proportional wealth change  $\pi_{t+\Delta t} = \frac{W_{t+\Delta t}}{W_t}$ . The terminal value function is:

$$v_T(\mathbf{x}_T, \theta_T) = u((1 - \psi(\mathbf{x}_T)) \cdot (1 - R_f(\theta_T))) \cdot \Delta t \cdot (1 - \beta)^{-1} \quad (23)$$

The constraints are likewise normalized:

$$\delta_t \geq -\mathbf{x}_t \quad (24)$$

$$b_t \geq 0 \quad (25)$$

$$\mathbf{1}^\top \mathbf{x}_t \leq 1 \quad (26)$$

This class of dynamic portfolio choice problems covers any formulation of the problem, where the transaction cost specification is differentiable, and the utility function allows for separability of wealth and remaining state variables. Later formulations will be based on this class structure, covering the necessary Bellman equation, state dynamics, preferences and transaction costs functions as well as the constraints and any extensions not yet presented.

The non-normalized optimal choices can be obtained by multiplying the normalized choices with the wealth level  $W_t$  at a given time point  $t$ . The NTR is in this framework the set of asset allocations where it is sub-optimal to rebalance the portfolio, and is defined as:

$$\Omega_t = \{\mathbf{x}_t : \delta_t^{+,*}, \delta_t^{-,*} = \mathbf{0}\} \quad (27)$$

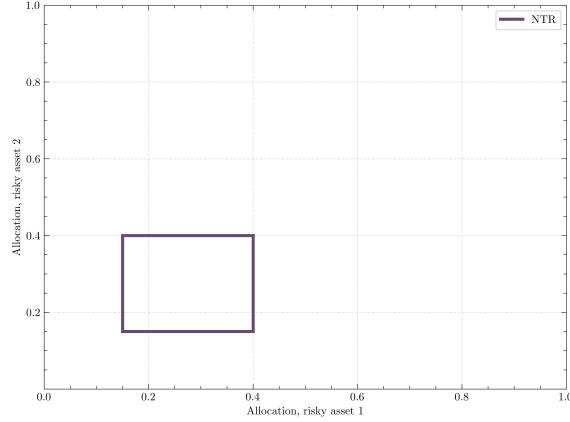
Where  $\delta_t^{+,*}, \delta_t^{-,*}$  are the optimal buying and selling policies at time  $t$ . The next section will cover the NTR in more detail.

## 5.7 No Trade Region

The NTR is a region in the asset space where it is sub-optimal to rebalance the portfolio. Given the parameters of the model the NTR without consumption is defined as in equation (27). If consumption is included, this definition remains the same, but the consumption decision varies within the NTR. Note that the NTR is independent of the wealth level, but only depends on the wealth allocations. The NTR stems from the introduction of transaction costs, and is a connected set.

**SE ANGÅENDE CONVEX HULL:** Kamen (1975), Constantinides (1976, 1979, 1986), Davis and Norman (1990), and Muthuraman and Kumar (2006). Figure 5.1 illustrates an example of a NTR with two risky assets. The square shape of the NTR occurs with proportional transaction costs and independent (i.i.d) risky assets. However the NTR is not always a perfect square, for more on this see (Dybvig and Pezzo 2020).

**Figure 5.1:** Example No Trade Region with  $k = 2$  risky assets.



## 5.8 Base problem: Portfolio with proportional costs and consumption

Considering the class of problems constructed in the prior section, we can now quickly introduce the basic problem formulation. We consider an investor with CRRA utility function. She can invest in one risk free asset and  $k$  risky assets. Trading is subject to proportional transaction costs hence we have the following cost function (in cumulative terms):

$$\psi(\delta_{i,t}^+, \delta_{i,t}^-) = \tau(\delta_{i,t}^+ + \delta_{i,t}^-) \quad (28)$$

We do not assume that returns are dependent on stochastic parameters, but instead are drawn from a distribution with known parameters. Hence we assume  $\theta_t = \theta$  for all  $t$ . That is that we assume a constant return on the risk free asset, hence  $R_f(\theta_t) = R_f$ , and the risky assets follow a multivariate log-normal distribution, with some mean and covariance matrix. We can now formulate the entire problem given the class structure from section 5.6. The terminal value function is given by equation (23). The system is subject to the constraints of equations (24), (25) and (26), as well as a simple constraint on consumption,  $c_t \geq 0$ . We assume that the position in bond holdings is the residual wealth, and they therefore follow the process in (19). The Bellman equation is therefore:

$$v_t(\mathbf{x}_t, \theta_t) = \max_{c_t, \delta_t^+, \delta_t^-} \{ u(c_t) \Delta t + \beta \mathbb{E}_t \left[ \pi_{t+\Delta t}^{1-\gamma} v_{t+\Delta t}(\mathbf{x}_{t+\Delta t}) \right] \}, \quad t < T$$

With same terminal condition as before, where wealth is transferred to the bond account at the terminal period, and consumption is financed by the interest returns.

$$v_T(\mathbf{x}_T) = u((1 - \psi(\mathbf{0}, \mathbf{x}_T)) \cdot (1 - R_f)) \cdot \Delta t \cdot (1 - \beta)^{-1}$$

## 5.9 Portfolio with fixed and proportional costs

## 5.10 Portfolio with asset specific costs

# 6 Numerical implementation details

---

This section covers detail regarding the solution algorithm and numerical implementation. Each method is presented in a separate subsection, and the final solution algorithm is presented in the last subsection, which combines each of the methods. These span points sampling, numerical integration techniques, function approximation methods and solution techniques specific to this class of problems.

## 6.1 Numerical integration

Consider the basic problem with proportional transaction costs, basic risky assets and a risk-free asset and no stochastic parameters. We need to evaluate the expectation of the value function:  $\mathbb{E}[v_{t+\Delta t}(\mathbf{x}_{t+\Delta t})]$ . In order to compute this expectation, we need to evaluate the integral:

$$\mathbb{E}_t \left[ \pi_{t+1}^{1-\gamma} v_{t+1}(x_{t+1}) \right] = \int \pi_{t+1}^{1-\gamma} v_{t+1}(x_{t+1}) f(R_{t+1}), dR_{t+1} \quad (29)$$

where  $f(R_{t+1})$  is the probability density function of the risky asset returns. If we look at the case of stochastic parameters, would need to evaluate the conditional expectation with regard to these as well, given some distributional assumption on the parameters. The integral can be computed using Monte-carlo methods or by using quadrature rules.

### 6.1.1 Gauss-Hermite quadrature

Gaussian quadrature is a numerical integration method based on approximation and interpolation theory. Gaussian quadrature can be used to approximate integrals using the following form, Judd (1998):

$$\int_a^b f(x) w(x) dx \approx \sum_{i=1}^n \omega_i f(x_i), \quad (30)$$

Where  $\omega_i$  are quadrature weights,  $x_i$  are quadrature nodes and  $w(x)$  is a weighting function. This approximation is exact when  $f(x)$  is a polynomial of degree  $2n - 1$  or less. Then we can approximate the integral using  $n$  points  $x_i$  and  $n$  weights  $\omega_i$ . There are many different Gaussian quadrature schemes, with differing intervals  $[a, b]$  and weighting functions  $w(x)$ . We consider the use of a Gauss-Hermite quadrature rule, for a comprehensive review on Gaussian quadrature rules, see Judd (1998). Gauss-Hermite quadrature is used

to approximate integrals of the form:

$$\int_{-\infty}^{\infty} f(x)e^{-x^2} dx \approx \sum_{i=1}^n \omega_i f(x_i) + \frac{n! \sqrt{\pi}}{2^n} \cdot \frac{f^{(2n)}(\zeta)}{(2n)!}, \quad (31)$$

Where  $\zeta \in (-\infty, \infty)$ . If a random variable  $X$  is normally distributed, i.e.  $X \sim \mathcal{N}(\mu, \sigma^2)$ , then we can compute the expectation,  $\mathbb{E}[f(X)]$ , which is given by:

$$\mathbb{E}[f(X)] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} f(x)e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \quad (32)$$

Using a change of variables  $y = \frac{x-\mu}{\sqrt{2}\sigma}$ , then we can rewrite the expectation on the form of the Gauss-Hermite quadrature rule:

$$\mathbb{E}[f(X)] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{\infty} f(\sqrt{2}\sigma y + \mu) e^{-y^2} \sqrt{2}\sigma dy \quad (33)$$

$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-y^2} f(\sqrt{2}\sigma y + \mu) dy \quad (34)$$

$$\approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^n \omega_i f(\sqrt{2}\sigma x_i + \mu) \quad (35)$$

Where  $\omega_i$  are the quadrature weights,  $x_i$  are the quadrature nodes over the interval  $(-\infty, \infty)$ .

When  $X$  is log-normal, i.e  $\log X \sim \mathcal{N}(\mu, \sigma^2)$ , then we can use a variable change once again:  $X = e^Y$  and  $Y \sim \mathcal{N}(\mu, \sigma^2)$ . Then we can rewrite the expectation as:

$$\mathbb{E}[f(X)] = \mathbb{E}[f(e^Y)] \approx \pi^{-\frac{1}{2}} \sum n_{i=1} \omega_i f\left(e^{\sqrt{2}\sigma x_i + \mu}\right) \quad (36)$$

If we want to extend this framework to multiple dimensions we can use product rules as noted by Cai, Judd and Xu (2013). Consider  $Y$  which is multivariate normal, i.e  $Y \sim \mathcal{N}(\mu, \Sigma)$ , where  $\mu$  is the drift vector and  $\Sigma$  is the covariance matrix. Let  $L$  be a lower-triangular matrix such that  $LL^\top = \Sigma$  (Cholesky factorisation). Then we have that:

$$\mathbb{E}\{f(Y)\} = \left((2\pi)^d \det(\Sigma)\right)^{-\frac{1}{2}} \int_{\mathbb{R}^d} f(y) e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)} dy \quad (37)$$

$$= \left((2\pi)^d \det(L)^2\right)^{-\frac{1}{2}} \int_{\mathbb{R}^d} f\left(\sqrt{2}Ly + \mu\right) e^{-\frac{1}{2}y^\top y} dy \quad (38)$$

$$\begin{aligned} &\approx \pi^{-\frac{d}{2}} \sum_{i_1=1}^n \cdots \sum_{i_d=1}^n \omega_{i_1} \cdots \omega_{i_d} f\left(\sqrt{2}L_{1,1}y_{i_1} + \mu_1, \right. \\ &\quad \left. \sqrt{2}(L_{2,1}y_{i_1} + L_{2,2}y_{i_2}) + \mu_2, \dots, \sqrt{2}\left(\sum_{j=1}^d L_{d,j}y_{i_j}\right) + \mu_d\right) \end{aligned} \quad (39)$$

Where  $d$  refers to the number of dimensions,  $n$  is the number of quadrature points,  $\omega_i$  are the quadrature weights and  $y_i$  are the quadrature nodes.  $L_{i,j}$  is the  $i$ th row and  $j$ th column of the Cholesky factorisation matrix  $L$ .  $\det$  is the matrix determinant. We note that the use of product rules suffers from the curse of dimensionality, as the complexity scales exponentially with the number of dimensions. This is because the quadrature points with the product rule, normally use a tensor product grid, which is constructed using the Cartesian product of the quadrature points in each dimension. We can use sparse grid methods to partially tackle this. One common method is the Smolyak method, Smolyak (1963). Smolyaks sparse grid method approximates multidimensional integrals, over dimesion  $d$  while limiting the amount of points used. The method is composed of the following:

1. **Univariate Quadrature Rules:** Each dimension of the integration domain is assigned a univariate quadrature rule, which provides both nodes (quadrature points) and weights for numerical integration in that dimension. The accuracy of each rule is determined by its *level*, denoted by  $i_d$  for each dimension  $d$ . The level determines the number of quadrature points in that dimension, which improves the accuracy of the quadrature rule.
2. **Approximation Level ( $\mu$ ):** The accuracy of the Smolyak sparse grid is controlled by the *approximation level*  $\mu$ . This parameter sets a limit on the sum of levels across all dimensions, controlling the total number of grid points. Higher values of  $\mu$  result in more accurate approximations but increase computational complexity.
3. **Multi-Index and Combination of Levels:** In a  $d$ -dimensional integral, the Smolyak method uses a *multi-index*  $i = (i_1, i_2, \dots, i_d)$  to represent the level of the quadrature rule in each dimension. The multi-index specifies a unique combination of quadrature levels for each dimension, where  $i_d$  denotes the level for dimension  $d$ . To construct a sparse grid, Smolyak's method restricts the sum of these levels using the following condition:

$$d \leq i_1 + i_2 + \dots + i_d \leq d + \mu$$

This constraint on the sum of levels, reduces the number of tensor products. We denote the sum of multi indicies:  $|i| = i_1 + i_2 + \dots + i_d$ .

4. **Tensor Product of Univariate Rules:** The Smolyak grid is formed by taking the *tensor product* of univariate quadrature rules that satisfy the multi-index constraint. Each univariate quadrature rule, represented by  $Q_{i_d}$  at level  $i_d$  in dimension  $d$ , is combined across dimensions according to the set of multi-indices  $i$ .

This combination is given by:

$$A(\mu, d) = \sum_{d \leq |i| \leq d+\mu} (-1)^{\mu+d-|i|} \binom{d-1}{\mu+d-1-|i|} \bigotimes_{d=1}^d Q_{i_d}$$

where:

- $Q_{i_d}$  is the univariate quadrature rule at level  $i_d$  in dimension  $d$ ,
- $\bigotimes$  denotes the tensor product, and
- $\binom{d-1}{\mu+d-1-|i|}$  is a combinatorial coefficient that assigns weights to each tensor product, for accurate integration up to the specified approximation level  $\mu$ .

By restricting the multi indices  $i$  with the approximation level  $\mu$ , the Smolyak method reduces the number of points needed for numerical integration in higher dimensions. Tensor grid methods grows exponentially with the number of dimensions  $d$ , the Smolyak grid grows polynomially, Judd et al. (2014), hence it directly combats the curse of dimensionality. For more on this see Smolyak (1963), Judd et al. (2014) and Horneff, Maurer and Schober (2016).

### 6.1.2 Monte Carlo integration (MC)

Monte Carlo integration is a numerical integration method based on *sampling*, as opposed to quadrature rules which are based on interpolation.

The convergence of Monte Carlo integration is generally slower than some quadrature methods; however, its convergence rate is independent of the dimensionality of the integral, making it well-suited for high-dimensional problems. Monte Carlo integration breaks the curse of dimensionality. Monte Carlo (MC) integration is based on random sampling<sup>6</sup> over the domain of the integral, and then computing the sample average of the function to be integrated. Assume we wish to approximate the  $d$ -dimensional integral:

$$I = \int_{\Omega} f(\mathbf{x})g(\mathbf{x})d\mathbf{x} = \mathbb{E}[f(\mathbf{x})], \quad (40)$$

where  $g(\mathbf{x})$  is the probability density function of the random variable  $\mathbf{x}$  over its support  $\Omega$ , we approximate  $I$  as:

$$Q_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{X}_i), \quad (41)$$

where  $\mathbf{X}_i$  are independent samples drawn from  $g(\mathbf{x})$ . The procedure is then:

1. Sample  $N$  points  $\mathbf{x}_1, \dots, \mathbf{x}_N$  from  $g(\mathbf{x})$ .

---

<sup>6</sup>Strictly speaking the samples are not random, but pseudo-random, meaning that deterministic samples are used, which appear random. For more in this see Judd (1998) or Glasserman (2004)

2. Approximate the expectation  $\mathbb{E}[f(\mathbf{x})]$  by the sample average:

$$I \approx Q_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i).$$

The Law of Large Numbers ensures that the sample average converges to the mean as  $N \rightarrow \infty$ :

$$\lim_{N \rightarrow \infty} Q_N = \mathbb{E}[f(\mathbf{x})] = I.$$

And by the Central Limit Theorem, we have:

$$\sqrt{N}(Q_N - I) \xrightarrow{d} N(0, \sigma^2),$$

where  $\sigma^2 = \text{Var}[f(\mathbf{x})]$  does not depend on  $N$  or  $d$ . The standard error of  $Q_N$  is:

$$\sigma_{Q_N} = \frac{\sigma}{\sqrt{N}}.$$

The convergence rate of  $1/\sqrt{N}$  is independent of the dimension.

### 6.1.3 Quasi-Monte Carlo integration (QMC)

Quasi-Monte Carlo integration substitutes the 'random' samples in Monte Carlo integration with specific deterministic sequences such as equidistributed sequences, low-discrepancy sequences (LDS) or Lattice point rules etc. We will focus on the use of low discrepancy sequences. For a comprehensive review of sequences and rules see Judd (1998). LDS are deterministic sequences which cover the domain of the integral more evenly than random samples. Discrepancy is in this case a measure of deviation from perfect uniformity over the domain of the integral. Thus to go from MC in (41) to QMC, we replace the random samples  $\mathbf{X}_i$  with LDS samples. We note that the sampling of the QMC is now dependent on the dimensionality of the integral, as opposed to MC, as the LDS samples have to be drawn with respect to the dimensionality of the integral.

We consider two different types of LDS sequences, the Halton sequence and the Sobol sequence. Both sequences are popular LDS sequences, which are used in quasi-Monte Carlo (MPT) applications, (Glasserman 2004).

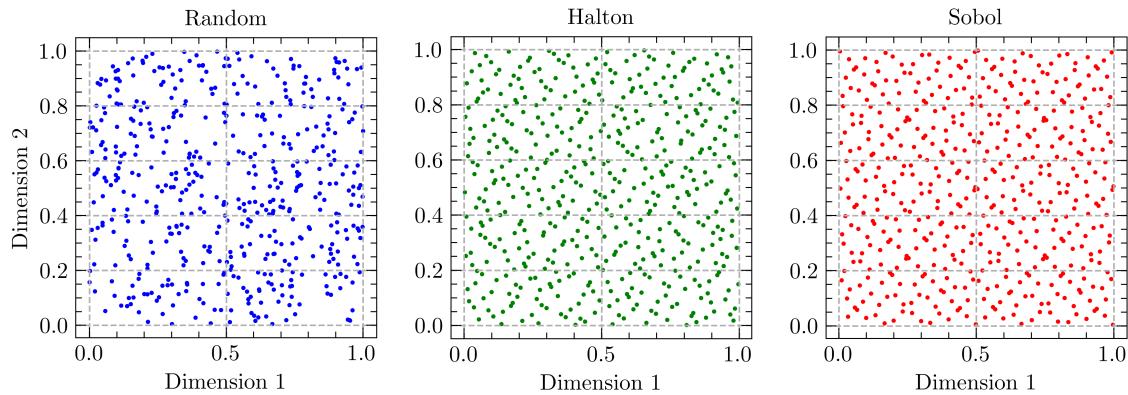
The convergence rate of MPT is:

$$\frac{(\log N)^d}{N} \tag{42}$$

Hence QMC is generally faster than MC, e.g.  $\frac{(\log N)^d}{N} < \frac{1}{\sqrt{N}}$  for large  $N$  and small  $d$ . We note that as dimensionality  $d$  increases, the quality of the Halton sequence decreases, as the dimensions become more correlated, Glasserman (2004). Specifically the Halton sequence will produce diagonal points when projected onto a 2D plane. This is displayed in

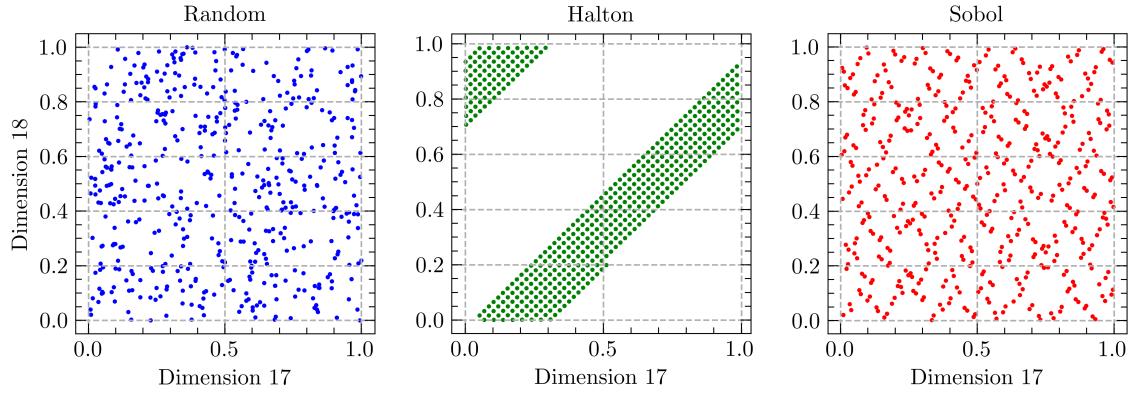
figure 6.1. We therefore prefer the Sobol sequence when the dimensionality is sufficiently high, and as not to complicate matters, also use the Sobol sequence in lower dimensions, when MPT schemes are used. Figures below shows Random samples, Halton samples and Sobol samples in 2d. Second figure shows the same in 18 dimensions. Halton shows that dimension 17 and 18 are correlated.

**Figure 6.1:** Comparison of sample generation for Monte Carlo and Quasi-Monte Carlo



**Note:** Each sequence was generated using  $N = 500$  samples and  $d = 2$  dimensions.

**Figure 6.2:** Comparison of sample generation for Monte Carlo and Quasi-Monte Carlo with increased dimensionality



**Note:** Each sequence was generated using  $N = 500$  samples and  $d = 18$  dimensions.

QMC is generally found to be more efficient than MC, as noted by Glasserman (2004), Judd (1998), and notably Glasserman find that dimensionality has to be quite large before the Monte Carlo method is favorable to the quasi Monte Carlo method. Furthermore Glasserman find that while we generally might assume that  $N$  must be increased a lot

when  $d$  is increased, this is not always the case in classic financial applications, as the integrals employed in these examples can often be approximated by integrals of much lower dimension. QMC therefore performs better than to be expected.

However we note that MPT lacks a straightforward variance estimator, a feature recovered through *randomized QMC*, which will be discussed in the next section.

#### 6.1.4 Randomized Quasi-Monte carlo integration (RQMC)

Randomized quasi-Monte Carlo integration (RQMC) is a combination of MPT and MC integration. We consider the the QMC integral, i.e the equaiton of (41), using an LDS sequence. The point of randomized quasi-Monte Carlo (MPT) is then to introduce randomness to the sequence:  $P_n = \{x_1, \dots, x_n\}$ . We will cover the most simple case, *Random shift* and *Scrambling* methods, however for a comprehensive review of randomization methods see Glasserman (2004). The most simple method of randomizing  $P_n$  is to add a *random shift* to each point in the sequence, using random numbers drawn from a uniform distribution of the same dimensionality as the sequence, wrapped to the interval of  $P_n$ . Hence if  $x_i \in [0, 1]^d$  then we add a random shift  $u_i \bmod 1$ , where  $\bmod 1$  keeps the shift within the interval  $[0, 1)$ . A major disadvantage of the random shift is that it changes the discrepancy properties of the sequence, and hence the quality of the sequence is lost. Scrambled nets is a method of randomization which can be applied to LDS sequences specifically. Scrambling works by applying a sequence of random permutations to the digits in the base- $b$  representation of each coordinate in the LDS. Each digit is permuted based on the values of the digits that came before it. This structure retains the low-discrepancy properties while introducing a controlled level of randomness, which enables the calculation of variance for RQMC estimates. In multi-dimensional settings, this scrambling is applied independently to each coordinate of the sequence, allowing us to estimate variance across the entire space. Scrambling the Sobol sequence has been found to be particularly effective in financial applications, as noted by Hok and Kucherenko (2023). QMC is generally more efficient than MC, and RQMC increases the rate of convergence of QMC and allows for the estimation of variance.

## 6.2 Value function approximation

This section covers the necessary function approximation methods used in the solution algorithm. We will cover the use of Gaussian process regression (GPR) and Bayesian optimization, in order to maximize the value function of the dynamic portfolio allocation problem.

### 6.2.1 Gaussian process regressions (GPR)

A GP is a probabilistic model that defines a distribution over functions used to make predictions based on available data. It is specified by two functions: the mean function and the covariance function, also called the kernel. The mean function,  $m(\mathbf{x})$ , represents the expected value of the function at a given input  $\mathbf{x}$ , and the covariance function,  $k(\mathbf{x}, \mathbf{x}')$ , captures the covariance between function values at different input points  $\mathbf{x}$  and  $\mathbf{x}'$ . In a GP, any finite set of input points  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  within the domain  $\mathbb{R}^d$  results in the function values  $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_N))$  having a joint multivariate Gaussian distribution. This property enables a GP to provide a prior distribution over functions based on the defined mean and covariance.

We use GPR to estimate the value function in the dynamic portfolio allocation problem, when we are not at the terminal period, i.e.,  $t < T$ , following Gaegau, Scheidegger and Trojani (2023). The GP is formulated by the previously mentioned mean and covariance functions:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (43)$$

The covariance kernel function  $k(\mathbf{x}, \mathbf{x}')$  can be any Mercer kernel, i.e., positive definite (Murphy 2023). Common kernel choices include the Radial Basis Function (RBF) kernel, the Matern kernel, and the Exponential kernel. We employ a Matern kernel, which, depending on the parameter  $\nu$ , can be a generalization of the RBF kernel or the Exponential kernel. This choice follows Gaegau, Scheidegger and Trojani (2023). The Matern kernel is given by:

$$k_{\text{Matern}}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \frac{\sqrt{2\nu} \|\mathbf{x} - \mathbf{x}'\|_2}{\ell} \right) K_\nu \left( \frac{\sqrt{2\nu} \|\mathbf{x} - \mathbf{x}'\|_2}{\ell} \right), \quad (44)$$

where  $\|\cdot\|_2$  is the Euclidean norm,  $\Gamma$  is the gamma function, and  $K_\nu$  is the modified Bessel function. The length scale  $\ell$  and smoothness parameter  $\nu$  are both positive. As  $\nu \rightarrow \infty$ , the Matern kernel converges to the RBF kernel (Gonzalvez et al. 2019). Functions from this class are  $k$ -times differentiable when  $\nu > k$ . When  $\nu = 1/2$ , the Matern kernel corresponds to the Ornstein-Uhlenbeck process (Murphy 2023), which is commonly used in financial applications, such as models of interest rates (Glasserman 2004).

Consider a training dataset  $\{\mathbf{X}, \mathbf{y}\}$  with  $N$  states  $\mathbf{x}_i$  and observed values  $\mathbf{y}$ . We assume that the observations  $\mathbf{y}$  are generated by an unknown function  $f$ , such that

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2),$$

where  $\sigma_\varepsilon^2$  represents the observational noise<sup>7</sup>. The goal is to train a GP on this dataset

---

<sup>7</sup>The noise assumption implies that the GP model does not interpolate the data but rather fits a smooth function. This results in computational costs of  $O(N)$  for the mean prediction and  $O(N^2)$  for the variance

and then use it to predict the value function at a new state  $\mathbf{x}_*$ , yielding a new predicted output  $f_*$ .

The training observations  $\mathbf{y}$  and the predicted noise-free function  $f_*$  have a joint Gaussian distribution:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I} & k(\mathbf{X}, \mathbf{x}_*) \\ k(\mathbf{x}_*, \mathbf{X}) & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \quad (45)$$

Here I have assumed a zero mean function<sup>8</sup>, and the kernel function is the Matern kernel. The posterior distribution of the predicted value function  $f_*$  given the training data is then a multivariate normal (Murphy 2023), with mean:

$$\tilde{\mu}(\mathbf{x}) = k(\mathbf{x}_*, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}]^{-1} \mathbf{y}, \quad (46)$$

And covariance:

$$\tilde{k}(\mathbf{x}_*, \mathbf{x}'_*) = k(\mathbf{x}_*, \mathbf{x}'_*) - k(\mathbf{x}_*, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}]^{-1} k(\mathbf{X}, \mathbf{x}'_*) \quad (47)$$

Therefore in order to predict the value function at a new state  $\mathbf{x}_*$ , we need to compute the mean and covariance. This step is computationally burdensome as we have to compute the four covariance matrices in the joint distribution (45). Afterwards we can compute predictions using the mean function (46) and the covariance function (47) can be used to compute error bands on our predictions.

As noted, training and predicting with a GP is computationally expensive. I will therefore introduce the methods employed to reduce the computational burden of the GP.

We use automatic relevance detection (ARD) which is a modification to the Matern kernel to use a length scale for each dimension,  $\ell_i$ . Dimensions with low impact has a high length scale, and are effectively ignored. Note that this is not the same as Lasso, as these coefficients are not set to 0. We use SKIP to reduce the computational burden of computing the matrices in the joint distribution (45).

**Use Lanczos Variance Estimate (Love) and SKIP to reduce computational burden.**

Here is the documentation in my package [https://docs.gpytorch.ai/en/stable/examples/02\\_Scalable\\_Exact\\_GPs/index.html](https://docs.gpytorch.ai/en/stable/examples/02_Scalable_Exact_GPs/index.html)

And here is a paper on the subject <https://arxiv.org/pdf/1803.06058.pdf>.

---

<sup>8</sup>prediction. For more details, see (Murphy 2023).

<sup>8</sup>Zero mean ... XXXX

### 6.3 Approximating the No trade region

Since i now have introduced methods to approximate the next-period value function  $v_{t+1}$ , and methods for evaluating the expectation  $\mathbb{E}[\cdot]$  over known distributions, we can now approximate the NTR using a DP scheme. In order to do this some assumptions regarding the unknown NTR are formed, these are drawn directly from (Gaegau, Scheidegger and Trojani 2023)

**Assumption 1.** *The NTR is a  $D$ -dimensional convex polytope.*

A polytope is a generalization of a polyhedron (polytope in 2D), which is a geometric object with flat sides and straight edges. The convex polytope is a polytope which bounds a convex set, and can therefore be defined by a convex hull. Hence, any linear combination of points in the NTR or on the boundary of the NTR is also in the NTR. In other words, the NTR is a closed convex set. [Wikipedia reference here?](#).

**Assumption 2.** *The NTR has  $2^D$  vertices.*

This assumption is regarding the shape of the NTR. Note that if the actual NTR has less than  $2^D$  vertices, the approximation will be close to the actual shape, as the approximated vertices will be on top of each other. However if the NTR has more than  $2^D$  vertices, then the approximation will be a simplification of the actual shape. The existing literature finds that the NTR is a  $D$ -dimensional parallelogram, this is formally shown with uncorrelated assets by (Liu 2004), and with correlated assets the same is found by (Cai, Judd and Xu 2013; Dybvig and Pezzo 2020). Hence i believe this sampling scheme to be sufficient, for the case of proportional transaction costs.

(Dybvig and Pezzo 2020) find that the NTR is a circle or ellipse when there are only fixed costs, and when there are asset specific costs the NTR is a hexagon in the 2D case, as one vertex is added per asset. This would suggest other sampling schemes for these cases. For the circular case i would need to sample evenly around the circle (sphere / hypersphere), this problem is well known in mathematics and computer graphics and many methods for this exist, among others lattice point methods. For more on this see for example ([Distributing points on the sphere n.d.](#)) or (Bono, Nicoletti and Ricci-Tersenghi 2024). For the hexagon case, i could add more midpoints between the vertices of the existing sampling scheme.

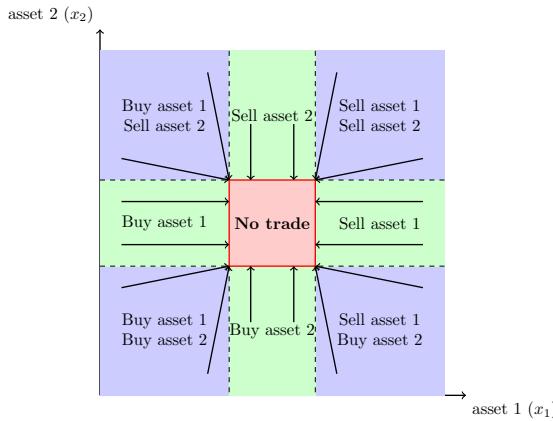
With these two assumptions in place, a strategy for approximating the NTR can be formed with few initial points. Given assumption 2 we can approximate the NTR by using  $2^D$  points, which are the vertices of the NTR, and by assumption 1 we can approximate the NTR by using the convex hull of these points, i.e connecting the vertices by straight lines to form the outer hull.

I can leverage the following intuition from (Gaegau, Scheidegger and Trojani 2023), and from 5.1: For any point outside the NTR, the optimal policy is to trade towards

the boundary of the NTR. Since each point on the boundary of the NTR is optimal, the optimal trading route minimizes the distance, and hence the optimal trading route is a straight line to the boundary of the NTR. If the points are chosen correctly, the optimal trading route will be to a vertex of the NTR. This is seen in the figure below:

If one considers the example in figure 6.3, it can effectively approximate the NTR, by

**Figure 6.3:** Illustration of the no-trade region (NTR) and the optimal policies outside this.



This is a schematic NTR. Blue regions are regions where optimal policy  $\delta$  is to adjust both asset allocations. Green regions are regions where the optimal policy is to hold in one asset and adjust the other. This figure is a recreation of Figure 1. in Gaegau, Scheidegger and Trojani (2023).

sampling a point in each of the blue regions, and then solving the optimization problem to find the vertices. When the NTR is unknown, sampling from the blue regions seem difficult at a first glance. However, it can sample the vertices of each simplex that covers the feasible space, and the midpoints between these. This sampling scheme leads to the following points in the 2-dimensional case:

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0.5 & 0.5 \end{bmatrix}$$

Extensions of this sampling scheme to higher dimensions is trivial. This should effectively cover the feasible space, and allow for approximating the NTR. Note that this sampling scheme only covers NTR with no borrowing, and no short-selling as noted in (Gaegau, Scheidegger and Trojani 2023). If borrowing and short-selling were introduced, we would have to set some bounds on the borrowing and short-selling, and then sample from these bounds. Effectively creating a square (cube / hypercube), around the feasible space, and then sample the vertices of this space.

Having approximated the NTR, we can now use this in the solution algorithm. There are two main ways which the NTR approximation can be leveraged in order to lessen the computational burden of the solution algorithm. These will be covered below.

### 6.3.1 Strategic point sampling

After having approximated the NTR, I need to efficiently approximate the value function in the time step related to the NTR. This is done by sampling points over the entire feasible space, and then solving for the optimal trade route for each point. In order to ensure that the approximation of the value function is of high quality, and that this value function can effectively be used for any point in the state space, we need to ensure that the points are sampled in a strategic manner. This means I need points of a few different types: I need points inside the NTR, and around the NTR in any direction, and various distances to the NTR. This leads to three types of points I need to sample: *Points inside the NTR, points near the kinks of the NTR and points in the general state space, outside the NTR*. An easily implementable solution is to use a naive grid sampling method, such as uniform draws over the feasible state space, or to use a grid-method which evenly covers the feasible state space. However, a simple naive grid method for sampling points over the state-space has a few drawbacks which I need to tackle.

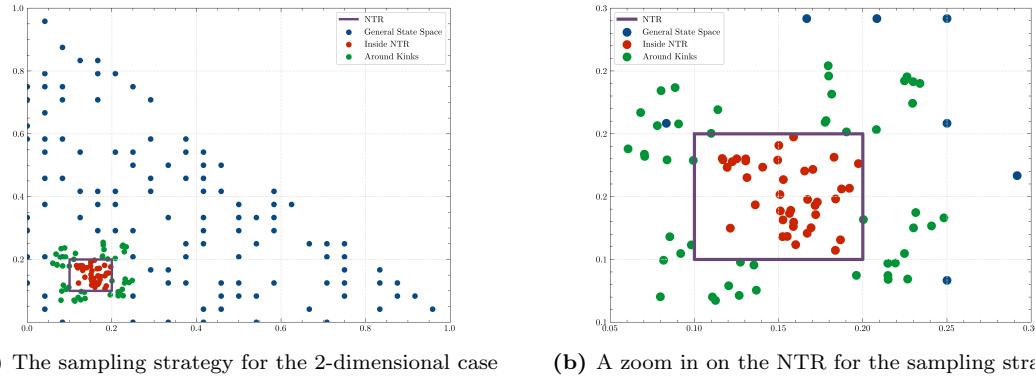
First of all, a naive grid method, such as uniform draws, will not cover the NTR efficiently, especially for small NTRs. I would need a large amount of grid points to be sure that there are multiple points inside the NTR. A pure random grid would likewise need a large amount of points, in order to cover the NTR efficiently, especially in each direction around the NTR. Both of these methods, and a schematic NTR are shown in appendix B.

I therefore instead follow the method of (Gaegau, Scheidegger and Trojani 2023), and sample points in a strategic manner. This scheme consists of the three point types mentioned earlier, with a sampling method for each of these points. Having approximated the NTR, I can effectively sample points in this manner:

1. Sample points outside the NTR in the general state space using a uniform grid. I then remove all points inside the NTR, and sample random grid points until I have enough points.
2. Sample points inside the NTR. For this I consider random draws, as the placement inside the NTR is not of high importance. I just need enough to approximate the value function.
3. Sample points around the NTR kinks. For this I consider each NTR vertex. I then interpolate between adjacent vertices slightly, and extend these outward with random noise draws.

The resulting points are plotted in figure 6.4a, and a zoom in on the NTR kinks are shown in figure 6.4b. (Gaegauf, Scheidegger and Trojani 2023) find that especially increased sampling around the kinks, leads to a better approximation of the value function, and  $N > 100$  points leads to sufficient approximations, as most of the approximation error is due to the kinks of the NTR. This choice of sampling scheme furthermore reduces the strain by curse of dimensionality, as grid sampling schemes would increase the number of points exponentially with the number of dimensions. Note that while this scheme still increases the number of points needed with the dimensionality, the oversampling of grid points especially reduce the number of points needed in higher dimensions.

**Figure 6.4:** The designed sampling strategy for state space coverage.



**Note:** Sample consists of  $N = 200$  points, with 122 points in the general state space (55%), 40 points inside the NTR (20%) and 48 points around the NTR kinks (25%).

### 6.3.2 Utilising the NTR approximation for $\delta$ bounds

Having constructed an efficient sampling strategy, i can further leverage the NTR approximation to find bounds on the optimal policy  $\delta$ , for the optimization step for each of these points. For this consider the schematic NTR in figure 6.3. At each point outside the NTR, the optimal policy is to trade towards the boundary of the NTR. This can either mean trading towards a vertice of the NTR or one of the faces. For the blue regions, trading towards a vertice is optimal, and this means that the optimal policy is to reallocate in both risky assets.

In this case, we can set bounds on the optimal policy  $\delta$ , by considering the euclidian distance to the NTR. Hence if i know beforehand, that for asset 1 we need to sell (lower-right blue region), then i can set bounds on  $\delta_1^+$  to 0 and effectively remove this from the optimization problem. I can likewise do this the other way around for the asset 2, which i need to buy more of, and set bounds on  $\delta_2^+$  to 0.

For the green regions in the figure, the optimal policy is to trade towards a face of the

NTR, and this means that the optimal policy is to reallocate in one risky asset and hold the other. I can therefore set bounds on the optimal policy  $\delta_i$  to 0 for the asset which is to be held, and only consider reallocation in the second asset.

This method of setting bounds on the optimal policy  $\delta$  is a way to reduce the computational burden of the optimization problem, and to ensure that the optimization problem is well defined. Furthermore, by knowing that the optimal policy reduces the euclidian distance to the NTR, i can effectively remove policies which would suggest buying and selling the  $i$ th asset.

### 6.3.3 Multiple Gaussian Process Regressions

The final ingredient in the algorithm is the use of multiple GPRs. Since  $i$  now can effectively sample points, and have information on their placement relative to the NTR,  $i$  can leverage this, and estimate two seperate value functions, one inside the NTR and one outside the NTR. This strategy effectively deals with the kinks of the NTR, as this otherwise would pose a problem for any smooth function approximations. I construct one GP for the points inside the NTR, and one for the points outside the NTR, and when  $i$  then evaluate the value function at a point  $v_{t+1}(\mathbf{x}_{t+1})$ ,  $i$  select the appropriate GP to evaluate the value function.

This is done after having optimized over the  $N$  points from the sampling strategy,  $i$  construct two datasets:

$$\mathbf{X}_{t,\text{inside}} = \{\mathbf{x}_{t,i}, \hat{v}_{t,i} \mid \mathbf{x}_{t,i} \in \hat{\Omega}_t\} \quad (48)$$

$$\mathbf{X}_{t,\text{outside}} = \{\mathbf{x}_{t,i}, \hat{v}_{t,i} \mid \mathbf{x}_{t,i} \notin \hat{\Omega}_t\} \quad (49)$$

Then each GP is fit over the dataset, which consists of asset allocations and the corresponding value function output. In the next period,  $t - 1$  (since we iterate backwards),  $i$  can then evalute the next period value functtion  $v_{t+1}(\mathbf{x}_{t+1})$ , by selecting the appropriate GP, and using the predictive mean from (46):

$$\tilde{\mu}(\mathbf{x}_{t+1}) = k(\mathbf{x}_{t+1}, \mathbf{X}_{t+1})[k(\mathbf{X}_{t+1}, \mathbf{X}_{t+1}) + \sigma_\varepsilon^2 \mathbf{I}]^{-1} \hat{\mathbf{v}}_{t+1}, \quad (50)$$

## 6.4 Final solution algorithms

Now that each component regarding the solution algorithm has been covered,  $i$  can now presents the solution algorithms for the dynamic portfolio allocation problem, in pseudo code. Starting at the second to last period, which is the last period where the investment decision is not trivial, the algorithm is as follows: Sample  $2^D$  points to approximate the NTR. Then Approximate the NTR by solving the optimization problem for these points.

Sample  $N$  points in a strategic manner, as described in ???. For each  $x_{i,t} \in X_t$  with  $\{X_t\}_{i=1}^N$ , solve the optimization problem to find the optimal policy  $\delta_i$ .

Construct the datasets  $\mathbf{X}_{t,\text{inside}}$  and  $\mathbf{X}_{t,\text{outside}}$  and fit two GPRs to the datasets  $\mathbf{X}_{t,\text{inside}}$  and  $\mathbf{X}_{t,\text{outside}}$ . The code can be split into two parts, algorithm (A) and algorithm (B). Algorithm A covers approximatin the NTR and algorithm B covers the entire DP scheme. These are drawn from the framework which has been covered, above, created by (Gaegauf, Scheidegger and Trojani 2023).

---

**Algorithm 1.** Approximate the  $t$ -th period NTR in the discrete-time finite-horizon portfolio choice model with proportional transaction costs.

---

**Input** :  $t + 1$  period's value function approximation  $V_{t+1}$ .  
**Result** : Set of approximated NTR vertices:  $\{\hat{\omega}_{i,t}\}_{i=1}^N$ ; Approximated NTR:  $\hat{\Omega}_t$ .  
Sample the set of  $N = 2^D$  points  $\tilde{\mathbf{X}}_t = \{\tilde{\mathbf{x}}_{i,t}\}_{i=1}^N$  using section strategy from Section 6.3.  
**for**  $\tilde{\mathbf{x}}_{i,t} \in \tilde{\mathbf{X}}_t$  :  
    | Obtain policy  $\hat{\delta}_{i,t}$  for  $\tilde{\mathbf{x}}_{i,t}$  by solving the optimization problem using  $V_{t+1}$  as the next period's value function. (Terminal value function in  $t = T - 1$ )  
    | Compute the approximate NTR vertices  $\hat{\omega}_{i,t} = \tilde{\mathbf{x}}_{i,t} + \hat{\delta}_{i,t}$ .  
**end**  
Compute the NTR approximation:  $\hat{\Omega}_t = \{\lambda\hat{\omega}_t \mid \lambda \in (0, 1)^N, \sum_{i=1}^N \lambda_i = 1\}$ .

---



---

**Algorithm 2.** Complete Dynamic programming scheme with Gaussian process regressions and the NTR approximation.

---

**Input** : Terminal value function  $v_T$ ; time horizon  $T$ ; sample size  $N$ .  
**Result** : Set of GP approximations of the value functions  $\{v_{t-1}\}_{t=0}^{T-1}$ ; set of approximated NTRs  $\{\hat{\Omega}_t\}_{t=0}^{T-1}$ , obtained policies  $\{\{\delta\}_{i=1}^{N+2^d}\}_{t=0}^{T-1}$ .  
Set  $\mathcal{V}_T = v_T$ .  
**for**  $t \in [T, \dots, 1]$  :  
    | Approximate NTR  $\hat{\Omega}_{t-1}$  (Alg. 1) using  $\mathcal{V}_T$  as the next period's value function.  
    | Sample  $N$  points  $\mathbf{X}_{t-1} = \{\mathbf{x}_{t-1,i}\}_{i=1}^N$  using the constructed sampling scheme.  
    | **for**  $\mathbf{x}_{i,t-1} \in \mathbf{X}_{t-1}$  :  
        | | Obtain value  $\hat{v}_{i,t-1}$  and policy  $\{\hat{\delta}_{i,t-1}, \hat{c}_{i,t-1}\}$  for  $\mathbf{x}_{i,t-1}$  by solving the optimization problem using  $\mathcal{V}_t$  as the next period's value function.  
    | | **end**  
    | Define the training sets:  
  

$$\mathcal{D}_{\text{in},t-1} = \{(\mathbf{x}_{i,t-1}, \hat{v}_{i,t-1}) \mid \mathbf{x}_{i,t-1} \in \hat{\Omega}_{t-1}\},$$

$$\mathcal{D}_{\text{out},t-1} = \{(\mathbf{x}_{i,t-1}, \hat{v}_{i,t-1}) \mid \mathbf{x}_{i,t-1} \notin \hat{\Omega}_{t-1}\}.$$
  
    | Given  $\mathcal{D}_{\text{in},t-1}$  and  $\mathcal{D}_{\text{out},t-1}$ , approximate  $v_{t-1}$  for inside and outside of the NTR  $\{G_{\text{in},t-1}, G_{\text{out},t-1}\}$  (using the respective datasets) with GPs.  
    | Set  $v_{t-1} = \{G_{\text{in},t-1}, G_{\text{out},t-1}\}$ .  
**end**

---

## 6.5 Computational stack and implementation

The solution algorithm is implemented in Python, and takes advantage of a simple but powerfull computational stack. following (Gaegauf, Scheidegger and Trojani 2023). The economic identities and dynamic where written using the PyTorch package, which is a ma-

chine learning library implemented in Python. This package has an auto-differentiation feature, which allows for easily implmentable gradients for the constrained optimization scheme. Furthermore this package is also directly linked with the `GPyTorch` package, which is a Gaussian process library implemented using PyTorch. The GPRs were implemented using the `GPyTorch` package, which is a Gaussian process library implemented using PyTorch. This package has multiple speedups for GPRs, such as the Lanczos Variance Estimate (LOVE) and the SKIP method, which reduces the computational burden of the GPRs. Furthermore the predictive mean can be computed using black-box matrix-matrix multiplication, which is a speedup for the predictive mean computation, skipping cholesky decompositions for large matrices.

The constrained optimizer i use is the `Cyipopt` package, which is a Python wrapper for the `Ipopt` package, which is a non-linear optimization package. This package is used to solve the optimization problem for each point in the state space, and is used to find the optimal policy  $\delta$  for each point.

The gaussian quadrature grid-points where implemented with the `Tasmanian` package, which is a sparse grid package. This was taken from (Schober, Valentin and Pflüger 2022), who used this package to implement sparse adaptive grids.

Finally i implemented parallelization at two points in the code. Whenever we run the optimization scheme for a point in the state space, we can run these in parallel, as they are independent operations, as long as i do this within the same timepoint  $t$ .

### 6.5.1 Optimization details

When solving the optimization problem, i use a tolerance of  $10^{-7}$ , and 1000 iterations. When approximating the NTR, i solve for each point 8 times, and select the optimal solution among these. Furthermore i multiply the starting point with a decaying factor, in the number of starts, in order to add small variance at each iteration. This is because non-linear optimization problems can be sensitive to the initial starting points.

The initial starting point is chosen within the feasible space at random, when there is no approximated NTR. The random draws are chosen to be feasible given the constraints of the problem. When i later have approximated the NTR, i use the shortest distance towards the NTR as initial guess, and multiply with a decaying factor over the number of starts. For these points i solve the optimization problem 3 times. This is because when i can leverage the knowledge of the NTR, the optimization problem is easier.

For points inside the NTR i likewise guess no trading, knowing this to be optimal a-priori. Small jitter is added to this when using multiple solutions.

## 7 Results

---

For the following results we consider 3 types of parameterizations for the portfolio problem. The first is a simple case where the assets are identically distributed as seen in (Cai, Judd and Xu 2013), the second is a case where the parameters are chosen to match the parameters in (Schober, Valentin and Pflüger 2022) also seen in (Gaegau, Scheidegger and Trojani 2023). This is in order to be able to draw correct comparisons between the results. Furthermore this case, displays assets with slight variation in the mean and a small correlation between the assets, and no asset is dominating the others. The last parameterization is a modification of the first case where the correlation between the assets is larger (correlation coefficient of 0.75).

**Table 1:** Parameters for Examples of Portfolio Problems

	i.i.d Assets	Schober Parameters	High Correlation
$T$	6	6	6
$k$	3	5	3
$\gamma$	3.0	3.5	3.0
$\tau$	0.5%	0.5%	0.5%
$\beta$	0.97	0.97	0.97
$r$	3%	4%	3%
$\mu^\top$	(0.07, 0.07)	$\mu_{\text{Schober}}$	(0.07, 0.07)
$\Sigma$	$\begin{bmatrix} 0.04 & 0.00 \\ 0.00 & 0.04 \end{bmatrix}$	$\Sigma_{\text{Schober}}$	$\begin{bmatrix} 0.04 & 0.03 \\ 0.03 & 0.04 \end{bmatrix}$

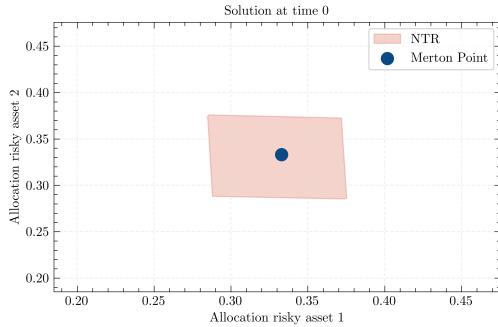
$$\mu_{\text{Schober}}^\top = [0.0572 \ 0.0638 \ 0.07 \ 0.0764 \ 0.0828]$$

$$\Sigma_{\text{Schober}} = \begin{bmatrix} 0.0256 & 0.00576 & 0.00288 & 0.00176 & 0.00096 \\ 0.00576 & 0.0324 & 0.0090432 & 0.010692 & 0.01296 \\ 0.00288 & 0.0090432 & 0.04 & 0.0132 & 0.0168 \\ 0.00176 & 0.010692 & 0.0132 & 0.0484 & 0.02112 \\ 0.00096 & 0.01296 & 0.0168 & 0.02112 & 0.0576 \end{bmatrix}$$

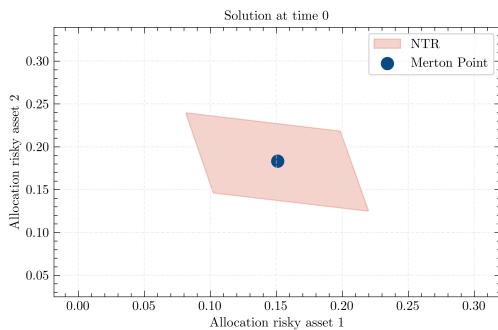
### 7.1 Dynamic Portfolio Choice without consumption

I first consider the base model with proportional transaction costs and no consumption. In the absence of consumption, the optimal portfolio is the merton point, which we plot in every figure. I plot the No-trade region at time point 0 (initial time point) for each of the parameterizations in 7.1. When using the Schober parameters we select the  $d$  first elements of the mean vector, and truncate the covariance matrix to a  $d \times d$  matrix, depending on the number of assets  $d$  in the model. I note that for each of the

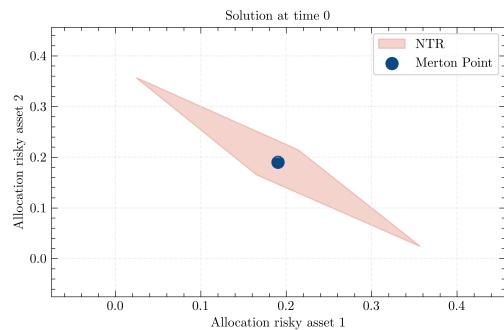
**Figure 7.1:** Comparison of No Trade Regions.



(a) No Trade Region for Independent Identically Distributed Assets.



(b) No Trade Region for Schober Parameters.



(c) No Trade Region for High Correlation.

parameterizations the No-Trade region is a rectangle or parallelogram. For the case of identical and independent assets, the No-Trade region is a perfect square, whereas for the Schober parameters and the high correlation case, the No-Trade region is a parallelogram. This is due to the correlation between the assets. When some correlation is present, the No-Trade region is skewed, since some allocations which would be optimal in the absence of correlation are no longer optimal.

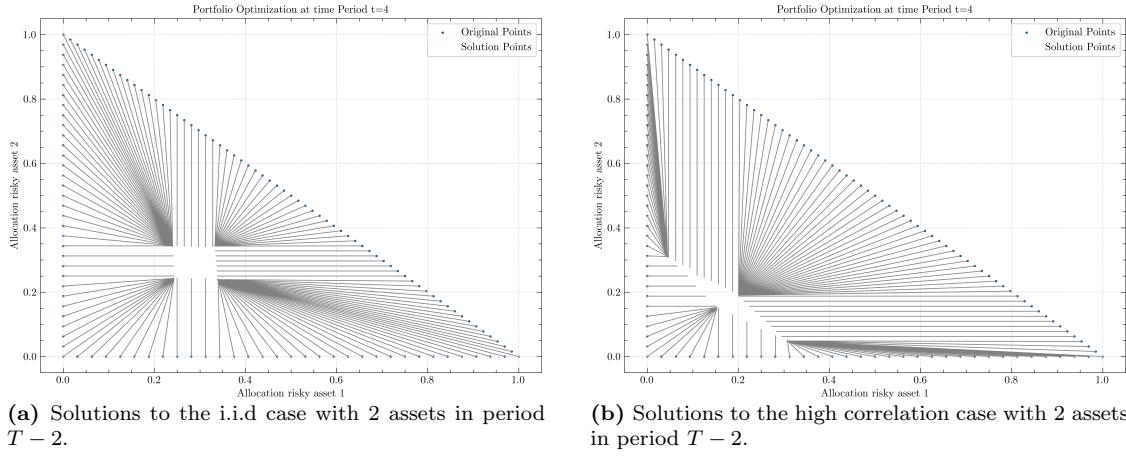
### 7.1.1 Verifying the geometric shape of the No-trade Region

Since much of the procedure for solving this problem, and approximating the NTR, leverages the a-priori assumptions regarding the geometric shape of the NTR, i first want to verify that the NTR indeed has 4 vertices for the 2d case and is a convex hull.

In order to do this, i do a small modification to the solution algorithm proposed earlier. Instead of computing vertices using  $2^d$  predetermined points, i will instead sample a larger set of points, ( $2^7 = 128$ ) covering the boundaries of the feasible state space. For each of these points i then solve the optimization problem, and plot the solution, from allocations  $\mathbf{x}_t$  and their solutions to the problem  $\hat{\omega}_t$ . I do this by using my original sample scheme, and adding mid-points between points, which either sum to 1.0 or have 0.0 as allocation for one of the assets. I consider the i.i.d case and the high correlation

case, with  $\tau_{\text{au}} = 1\%$ . I have increased the costs slightly in order to increase the size of the NTR. This is to ensure that points also converge towards the faces and not only the vertices. This is akin to the green regions in Figure 6.3. Otherwise I would need more points. I plot the solutions for next to last period with investment decisions  $T - 2$ . The solutions are plotted below.

**Figure 7.2:** Verifying the assumptions of the NTR in 2 dimensions.



I find that the assumptions regarding the NTR are indeed correct in the two dimensional examples I have constructed. Furthermore this verifies that the assumptions also hold for correlated asset, which was only postulated by (Liu 2004). Furthermore these plots also nicely confirm that the optimization process as a whole works as intended. Further verification in higher dimensions are not considered. First of all (Liu 2004) confirms this formally in larger dimensions, for the case of uncorrelated assets, and the intuition regarding the NTR does not change when dimensionality is increased.

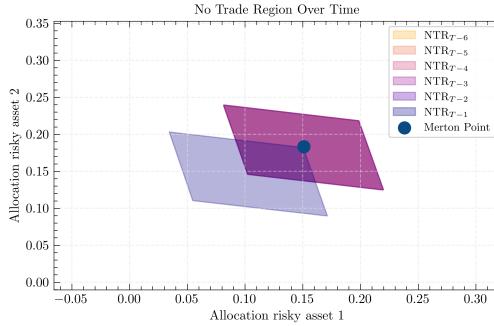
### 7.1.2 Investigating the No-Trade Region

We now look at the No-Trade region for the base model with proportional transaction costs and no consumption in more detail. Specifically we look at how the region behaves over the entire investment horizon  $[0, T]$ , and how the region changes with different transaction cost levels. We choose to look at the model with the Schober parameters, as this is a mixture of the other two parameterizations.

I note that at the last time point  $t = T - 1$  the NTR moves away from the Merton point towards the origin, and the Merton point is now the upper right corner of the NTR. For all other time periods the NTR is the same, and the Merton point is in the center.

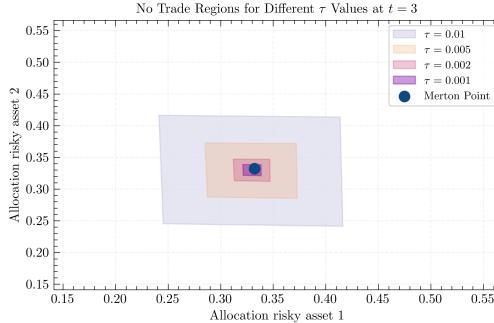
I now investigate how the No-Trade region changes with different transaction cost levels. I do this for the i.i.d. parameters, and plot the NTR for different values of  $\tau$  in

**Figure 7.3:** No Trade Region for Schober Parameters over Time.



The No-Trade region is plotted for the Schober parameters over the entire investment horizon  $[0, T - 1]$ . For time points  $t \in [0, T - 2]$  the NTRs overlap.

**Figure 7.4:** No Trade Region for the iid Parameters with different values of  $\tau$ .



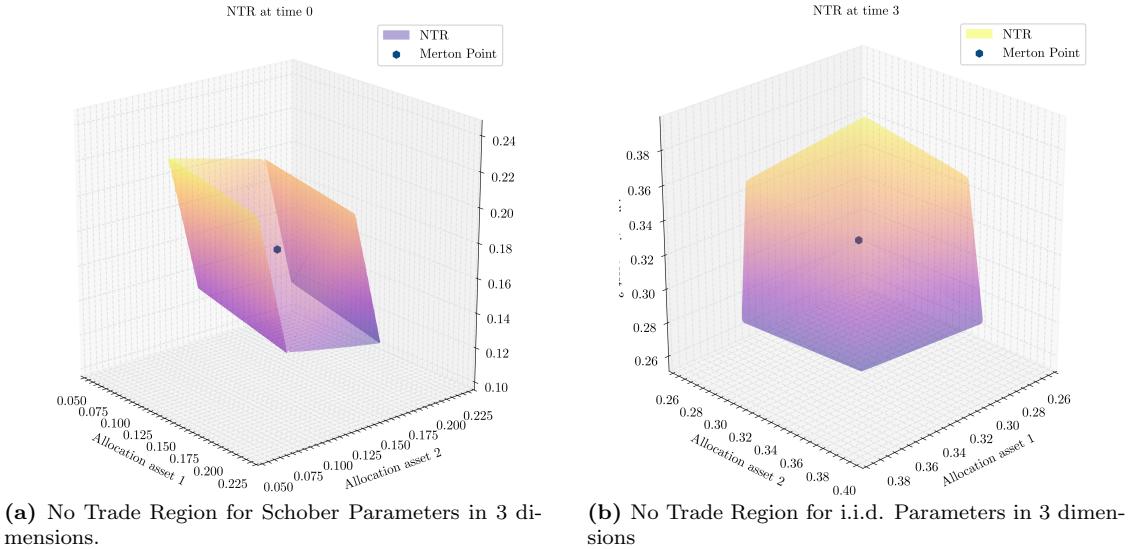
**Figure 7.4.** When the transaction costs are increased, the NTR increases as well and vice-versa. I note that for low enough transaction costs, the NTR shrinks towards the Merton point. However when transaction costs are low enough, the Merton point is not in the exact center, which might signify that at low enough values, some numerical instabilities from the minimizer, and function approximation using GPR might be present.

### 7.1.3 Increasing the dimensionality of the model

We now increase the dimensionality of the model to  $d = 3$  and look at the No-Trade region for the Schober parameters and for the i.i.d parameters.

Note that the i.i.d NTR looks like a skewed cube, whereas this was a perfect square in the 2 dimensional case. Looking that the points forming the convex hull that is the NTR, it is clear that the NTR is restricted by the no-borrowing constraint, since one of the border points, which would otherwise form the perfect cube, would outside the feasible space if this was possible, and is then projected into the feasible space. Hence when the risky returns outweigh the risk-free return, to such a degree that the merton point moves

**Figure 7.5:** Comparison of No Trade Regions.



towards the boundary of the feasible space, cube like shapes are no longer possible. In the 2 dimension case, this is akin to the NTR being close to the budget line, and the NTR would then form a triangle.

This is clear when compared to the Schober parameters, where the merton point is in the center of the NTR, and the NTR is a skewed cube. The merton point in this case suggest lower portfolio allocations to the risky assets, and hence the NTR is not restricted by the no-trading and no-borrowing constraints.

## 7.2 Dynamic Portfolio Choice with consumption

I now consider the base model with proportional transaction costs which now includes consumption of a non-durable good. This adds an extra decision variable which needs to be solved for, and consumption now adds immediate utility to the investor, in each period.

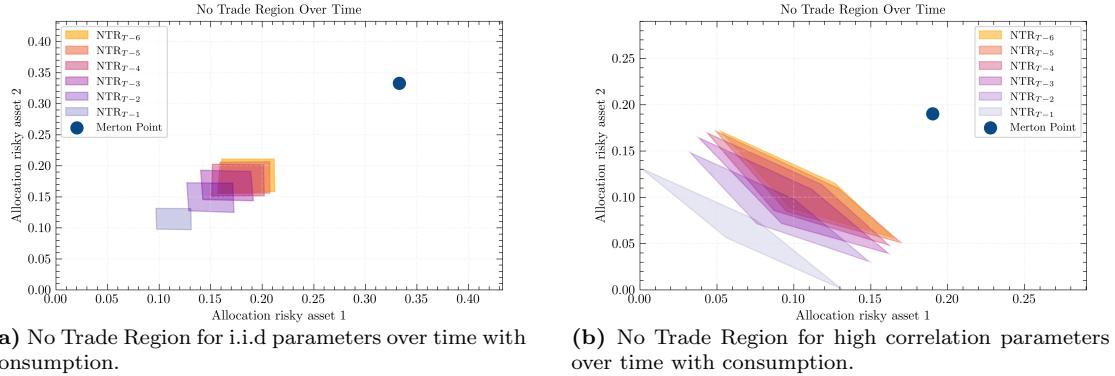
Note that when consumption is included, the NTR no longer encapsulates the Merton point at any time point. Furthermore the NTR now moves over time, towards the origin, as opposed to the case without consumption, where the NTR was static for all time points except the next to last period (last period with trading decisions).

This behaviour is consistent in higher dimensions.

## 7.3 Dynamic Portfolio Choice with fixed costs

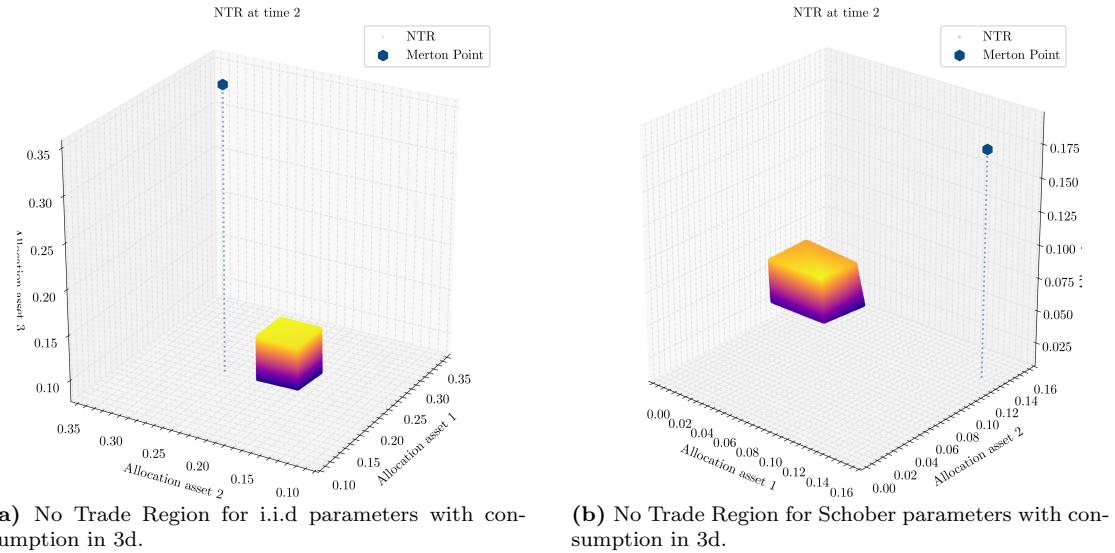
I now consider the base model with fixed transaction costs, and no consumption. From (Dybvig and Pezzo 2020) i know that the NTR is no longer rectangular when we only consider fixed costs, but instead circular with the merton point in the middle. This

**Figure 7.6:** Comparison of No Trade Regions over time with consumption.



The No-Trade regions are plotted over the entire investment horizon  $[0, T - 1]$ .

**Figure 7.7:** No trade regions with consumption in multiple dimensions.



The No-Trade regions are plotted at time  $t = 2$ .

poses a problem for my current sampling scheme, which leverages our predetermined knowledge of the geometric shape of the NTR. As I noted in Section 6.3.1, in order to effectively sample points for the NTR approximation, I now need to sample points, such that when they hit the NTR these points are evenly distributed on the sphere, in order to approximate the NTR correctly.

For the case without consumption this is feasible, but for the case with consumption, the NTR moves over time, and locating its center a priori is not possible.

## References

---

- Akian, Marianne, Menaldi, José Luis and Sulem, Agnès (1996). “On an Investment-Consumption Model with Transaction Costs”. In: *SIAM Journal on Control and Optimization* Vol. 34, No. 1, pp. 329–364. DOI: 10.1137/S0363012993247159.
- Bellman, Richard (1958). “Dynamic programming and stochastic control processes”. In: *Information and Control* Vol. 1, No. 3, pp. 228–239. ISSN: 0019-9958. DOI: [https://doi.org/10.1016/S0019-9958\(58\)80003-0](https://doi.org/10.1016/S0019-9958(58)80003-0).
- Björk, Tomas (2019). *Arbitrage theory in continuous time*. 4. ed. Oxford Univ. Press. ISBN: 9780198851615. URL: [http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+505893878&sourceid=fbw\\_bibsonomy](http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+505893878&sourceid=fbw_bibsonomy).
- Bono, Luca Maria Del, Nicoletti, Flavio and Ricci-Tersenghi, Federico (2024). *The most uniform distribution of points on the sphere*. arXiv: 2407.01503 [cond-mat.stat-mech]. URL: <https://arxiv.org/abs/2407.01503>.
- Cai, Yongyang, Judd, Kenneth L. and Xu, Rong (2013). *Numerical Solution of Dynamic Portfolio Optimization with Transaction Costs*. Working Paper 18709. National Bureau of Economic Research. DOI: 10.3386/w18709.
- Cai, Yongyang, Judd, Kenneth L. and Xu, Rong (2020). *Numerical Solution of Dynamic Portfolio Optimization with Transaction Costs*. Tech. rep. arXiv: 2003.01809 [q-fin.PM]. URL: <https://arxiv.org/abs/2003.01809>.
- Constantinides, George M. (1976). “Note—Optimal Portfolio Revision with Proportional Transaction Costs: Extension to Hara Utility Functions and Exogenous Deterministic Income”. In: *Management Science* Vol. 22, No. 8, pp. 921–923.
- Constantinides, George M. (1986). “Capital Market Equilibrium with Transaction Costs”. In: *Journal of Political Economy* Vol. 94, No. 4, pp. 842–862. ISSN: 00223808, 1537534X. URL: <http://www.jstor.org/stable/1833205>.
- Davis, M. H. A. and Norman, A. R. (1990). “Portfolio Selection with Transaction Costs”. In: *Mathematics of Operations Research* Vol. 15, No. 4, pp. 676–713. DOI: 10.1287/moor.15.4.676.
- Distributing points on the sphere* (n.d.). <https://www.unsw.edu.au/science/our-schools/mathematics-our-school/spotlight-on-our-people/history-school/glimpses-mathematics-and-statistics/distributing-points-sphere>. Accessed: 2024-11-22.
- Dybvig, Philip H and Pezzo, Luca (2020). “Mean-variance portfolio rebalancing with transaction costs”. In: Available at SSRN 3373329.
- Gaegau, Luca, Scheidegger, Simon and Trojani, Fabio (2023). *A Comprehensive Machine Learning Framework for Dynamic Portfolio Choice With Transaction Costs*. Tech. rep. 23-114. Swiss Finance Institute, p. 70. URL: <https://ssrn.com/abstract=4543794>.

- Garleanu, Nicolae and Pedersen, Lasse Heje (2013). "Dynamic Trading with Predictable Returns and Transaction Costs". In: *The Journal of Finance* Vol. 68, No. 6, pp. 2309–2340.
- Glasserman, Paul (2004). *Monte Carlo methods in financial engineering*. New York, NY, USA: Springer. ISBN: 0387004513 9780387004518.
- Gonzalvez, Joan et al. (2019). *Financial Applications of Gaussian Processes and Bayesian Optimization*. Working Paper WP-80-2019. Document for the exclusive attention of professional clients, investment services providers and any other professional of the financial industry. Amundi Quantitative Research.
- Hok, J. and Kucherenko, S. (2023). *The importance of being scrambled: supercharged Quasi Monte Carlo*. arXiv: 2210.16548 [q-fin.CP]. URL: <https://arxiv.org/abs/2210.16548>.
- Horneff, Vanya, Maurer, Raimond and Schober, Peter (2016). "Efficient parallel solution methods for dynamic portfolio choice models in discrete time". In: *Available at SSRN 2665031*.
- Janeček, Karel and Shreve, Steven E. (2004). "Asymptotic analysis for optimal investment and consumption with transaction costs". In: *Finance and Stochastics* Vol. 8, No. 2, pp. 181–206. DOI: 10.1007/s00780-003-0113-4.
- Judd, Kenneth L. (1998). *Numerical Methods in Economics*. Vol. 1. MIT Press Books 0262100711. The MIT Press. ISBN: ARRAY(0x5a5be060). URL: <https://ideas.repec.org/b/mtp/titles/0262100711.html>.
- Judd, Kenneth L et al. (2014). "Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain". In: *Journal of Economic Dynamics and Control* Vol. 44, pp. 92–123.
- Liu, Hong (2004). "Optimal Consumption and Investment with Transaction Costs and Multiple Risky Assets". In: *The Journal of Finance* Vol. 59, No. 1, pp. 289–338. ISSN: 00221082, 15406261. URL: <http://www.jstor.org/stable/3694897> (visited on 11/25/2024).
- Markowitz, Harry (1952). "Portfolio Selection". In: *The Journal of Finance* Vol. 7, No. 1, pp. 77–91. ISSN: 00221082, 15406261. URL: <http://www.jstor.org/stable/2975974>.
- Merton, Robert C. (1969). "Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case". In: *The Review of Economics and Statistics*, pp. 247–257.
- Merton, Robert C. (1971). "Optimum Consumption and Portfolio Rules in a Continuous-Time Model". In: *Journal of Economic Theory* Vol. 3, No. 4, pp. 373–413. DOI: 10.1016/0022-0531(71)90038-X.
- Murphy, Kevin P. (2023). *Probabilistic Machine Learning: Advanced Topics*. MIT Press. URL: <http://probml.github.io/book2>.
- Muthuraman, Kumar and Kumar, Sunil (2006). "Multidimensional Portfolio Optimization with Proportional Transaction Costs". In: *Mathematical Finance: An International Journal* Vol. 16, No. 1, pp. 1–22. DOI: 10.1002/mafi.20031.

- tional Journal of Mathematics, Statistics and Financial Economics* Vol. 16, No. 2, pp. 301–335.
- Muthuraman, Kumar and Kumar, Sunil (2008). “Solving Free-boundary Problems with Applications in Finance”. In: *Foundations and Trends® in Stochastic Systems* Vol. 1, No. 4, pp. 259–341. doi: 10.1561/0900000006.
- Oksendal, Bernt and Sulem, Agnès (2002). “Optimal Consumption and Portfolio with Both Fixed and Proportional Transaction Costs”. In: *SIAM Journal on Control and Optimization* Vol. 40, No. 6, pp. 1765–1790. doi: 10.1137/S0363012900376013.
- Schober, Peter, Valentin, Johannes and Pflüger, Dirk (2022). “Solving High-Dimensional Dynamic Portfolio Choice Models with Hierarchical B-Splines on Sparse Grids”. In: *Computational Economics* Vol. 59, No. 1, pp. 185–224. doi: 10.1007/s10614-021-10118-9.
- Shreve, S. E. and Soner, H. M. (1994). “Optimal Investment and Consumption with Transaction Costs”. In: *The Annals of Applied Probability* Vol. 4, No. 3, pp. 609–692. doi: 10.1214/aoap/1177004966.
- Smolyak, S. A. (1963). “Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions”. In: *Doklady Akademii Nauk SSSR* Vol. 148. English translation: *Soviet Mathematics Doklady*, Vol. 4, pp. 240–243, pp. 1042–1043.
- Zabel, Edward (1973). “Consumer Choice, Portfolio Decisions, and Transaction Costs”. In: *Econometrica* Vol. 41, No. 2, pp. 321–335. URL: <http://www.jstor.org/stable/1913492>.

# Appendices

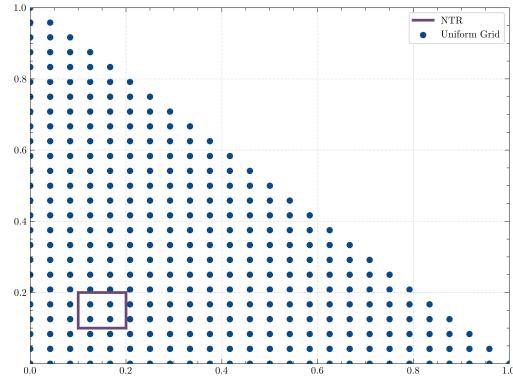
A

---

## B Other sampling strategies

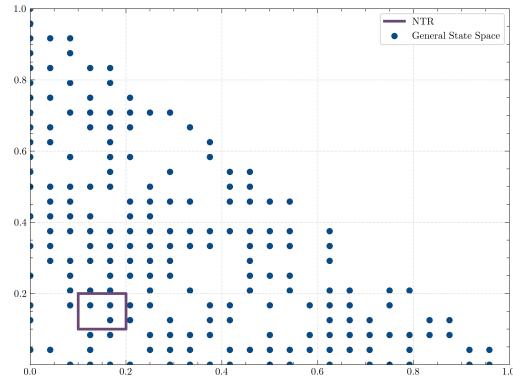
---

**Figure B.1:** Uniform grid sampling strategy



**Note:** Sample consists of  $N = 200$  points.

**Figure B.2:** Naive random sampling strategy



**Note:** Sample consists of  $N = 200$  points.