RISK DIVERSIFICATION FRAMEWORK IN ALGORITHMIC TRADING

A Thesis
Presented to
The Academic Faculty

by

Jiangchuan Yuan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2014

RISK DIVERSIFICATION FRAMEWORK IN ALGORITHMIC TRADING

Approved by:

Professor Jim Dai, Co-advisor, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Shijie Deng, Co-advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Anton J. Kleywegt
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Ton Dieker
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Jim Xu
College of Computing
Georgia Institute of Technology

Date Approved: 31 March 2014

ACKNOWLEDGEMENTS

I am deeply grateful to all people who have helped and inspired me during my doctoral study in Georgia Tech. This thesis would not have been possible without them.

First of all, I would like to thank my advisors, Dr. Jim Dai and Dr. Shijie Deng. If it were not for them, I would not have chosen Algorithmic Trading as my dissertation topic, a topic that has also become a focus of my current work in J.P.Morgan. I am also thankful that they introduce me to Approximate Dynamic Programming, a interesting and powerful method for solving problems. Their passion, knowledge, and rigorous attitude toward research have deeply influenced me. These are things that I will always carry with me as I continue on in my work. I am also grateful for their understanding, patience and encouragement during the latter part of my graduate studies when I needed to work full time in New York.

I would also like to thank Dr. Anton J. Kleywegt, Dr. Ton Dieker and Dr. Jim Xu for agreeing to serve on my thesis committee and offering their valuable feedback.

The past heads of the graduate program, Drs. Gary Parker and Paul Kvam, and current head, Dr. Alan Erera, along with Pam Morrison in the ISyE graduate studies office performed many miracles for me over the past few years without which I would not have completed this thesis.

I would also like to thank Dr. Warren Powell and Dr. Hugo Simao at Princeton University for being generous with their time and their willingness to help me understand Approximate Dynamic Programming.

I spent the last two years working in the Linear Quantitative Research group at J.P.Morgan on equity algorithmic trading. Discussions with my coworkers in LQR helped me to understand the trading scheduling problems from an industry perspective. I owe my appreciation especially to Daniel Nehren, David Fellah and Shilong Yang, who have taught me a lot about equity trading and market microstructure.

On a more personal note, I am thankful to my friends at Georgia Tech. Dexin Luo, Haiyue Yu and Fangfang Xiao, your friendship and support have been invaluable to me. I have to say, we had some good memories at Georgia Tech.

Finally, to my parents, I thank you for your love and support for giving me the freedom to pursue my dreams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Algorithmic trading provides systematic methods for traders to execute order to minimize execution cost. Execution cost is measured as the difference between the average execution price over the entire trade and some pre-specified benchmark price. Execution cost often involves multiple market dynamic risks, which include price volatility risk, liquidity risk due to short term supply and demand imbalance, execution risk due to uncertainty of order completion and the risk of large deviation between prediction and actual realization of market variables, such as volume, etc.

The major contribution of this thesis is to provide a systematic framework that can be applied to make sequential adaptive decisions in algorithmic trading problems. These problems often involve different market dynamic risks, and the key to solving them is to diversify these risks over sequential decisions, which are spread in time. This general framework solves trading problems in a discrete time setting with a mean-variance objective, which is closely related with the common practice in the broker-dealer industry. One of the major advantages of this framework is that it solves mean-variance problems with different risk aversion factors all at once, which allows the practitioners to easily plot the efficient frontier of execution costs. The other contribution of this thesis is to apply this risk diversification framework to explicitly formulate the joint optimization problem of the trade scheduling and optimal order placement problems, which challenges the common practice in both academia and industry that treat these two problems separately.

To demonstrate the advantages of the risk diversification framework, we apply it to tackle three trading problems in this thesis to illustrate its powerfulness and effectiveness.

The first two problems concern trade scheduling which seeks to minimize the execution cost by splitting a large stock-trading order into a group of separately executed child orders. The two problems differ in how they measure the execution cost. The first problem measures the execution cost against the arrival price. The resulting objective, which is the mean-variance of the execution cost, requires the trader to carefully balance the trade-offs between the liquidity risk of quick trading and the volatility risk of slow trading. The second problem measures the execution cost against the volume weighted average price (VWAP). The resulting objective aims to balance the risk of volume prediction error and the risk due to price volatility. In both cases, the adaptive strategies significantly outperform deterministic strategies.

In trading practice, the trade scheduling decisions are fed into the optimal order placement model (OOPM) to decide how to split each child order into market orders and limit orders. OOPM seeks the optimal trade-off between the liquidity risk (namely, market impact and bid-ask spread) of using market orders and the execution risk of using limit orders. Rather than addressing OOPM alone, our third problem combines the trade scheduling and OOPM, in order to find the optimal trade-off among three market dynamic risks: liquidity risk, volatility risk and execution risk. This unified strategy outperforms both the strategy that only applies market orders and the strategy that treats trading scheduling and OOPM separately.

All three problems share several key components of the proposed risk diversification framework. Specifically, the execution cost is first quantified to incorporate different market dynamic risks. Next, the traders' objectives are modeled as a mean-variance measure of the execution cost over a finite time horizon. Since variance is not a time-consistent measure, the mean-variance problem can not be directly solved through dynamic programming. Instead, it is approached by solving a family of auxiliary linear-quadratic problems whose solutions contain the solutions of the original mean-variance problem. By carefully decomposing the linear-quadratic objective as a sum of cost components, the Bellman equations are derived in all problems. We introduce the linear-quadratic weights as a time-varying state variable which allows us to efficiently solve the whole family of linear-quadratic problems through one round of backward induction.

CHAPTER I

INTRODUCTION

Over the last three decades there has been a significant increase in the use of algorithmic trading in financial markets among institutional investors. Algorithmic trading is the use of electronic platforms to enter trading orders, guided by an algorithm that executes pre-programmed trading instructions whose variables may include timing, price, and quantity of the order. Algorithmic trading not only allows traders to have access to market liquidity in a quick and efficient way, it also provides traders with a systematic approach for achieving pre-specified objectives.

Among the large uses of algorithmic trading is execution trading, where a trader executes a larger order for an investor to buy or sell a stock, according to that investor's pre-specified objectives. In this scenario, the pre-specified objective often revolves around minimizing the execution cost. Execution cost is measured as the difference between the average execution price over the entire trade and some benchmark price specified by the investor. Commonly used benchmark prices include the arrival price (namely, the initial market price of the stock when the instruction for executing an order arrives at the trading desk) and Volume-Weighted-Average-Price (VWAP) of the stock during a specified horizon, such as one day. A positive (or, negative) execution cost represents a loss (or, gain) compared with trading at the benchmark price. For instance, a negative execution cost for a buy order means the order is bought on average cheaper than the benchmark price thus representing a gain for the investor.

Over the course of execution of a large order, the price tends to move up, to the traders disadvantage. This adverse price reaction to the trader's own trading (often called market impact) is a result of limited market liquidity. Instead of trading the given order all at once and incurring a large execution cost, traders often split the large order into sequentially executed smaller pieces. However, this extends the trading duration for completing the order, which incurs the risk of have significantly worse execution price than the arrival price due to price volatility and the prolonged trading horizon (this is termed as the timing risk). For example, if price maintains an upward trend, a buy order may have an average execution price being much higher than the arrival price (i.e. higher execution cost). Therefore, a good execution strategy should maintain a balance between the liquidity risk (including the risk of a high market impact) of fast trading and the timing risk of slow trading. Mathematically, it means both the expectation and the variance of the execution cost should be considered. We will study an adaptive strategy in Chapter 2 that minimizes the weighted sum of the mean and variance of the execution cost using arrival price as a benchmark.

The problem of how to best split a large order (i.e. parent order) into multiple small orders (i.e. child orders) that are spread out in time is called the trade scheduling problem. In Chapter 3, we study the trade scheduling problem with VWAP as the benchmark price, where the goal of the trader is to define an ex-ante strategy which ex-post leads to an average trading price as close as possible to (or even beat) the market VWAP (an example of the market VWAP benchmark is given in the footnote below). The VWAP benchmark encourages traders to spread their orders out over time to avoid the risk of trading at extreme prices. If we can perfectly forsee the market volumes in all the trading periods, then we can schedule child orders to match those volumes precisely, and thereby reduce the execution cost to zero regardless of price evolution[1]. As the market impact of the trader's own trading appear in the calculation

[1] For example, consider a trading problem of buying 1000 shares within 3 minutes. The market volumes for the three minutes are 5000 shares, 2000 shares and 3000 shares. The average prices for the three periods are \$10, \$11, and \$12. The market VWAP is $\frac{5000\times\$10+2000\times\$11+3000\times\$12}{5000+2000+3000} = \$10.8$. If the trader splits 1000 shares sequentially into 500, 200 and 300 shares, and if the trader can achieve the same market average price within each period, the trader's total average execution

of both the market VWAP and the trader's average execution cost, its effects on the execution cost get canceled out when we compute the execution cost. Therefore, market impact plays a less significant role in the VWAP scheduling problem. In reality, we can never perfectly predict the actual market volume. As a result, the optimal adaptive strategy needs to optimize the trade-off between the risk of volume prediction error and the risk due to price volatility.

For both trade scheduling problems just discussed, the adaptive strategies described in Chapter 2 and Chapter 3 significantly outperform strategies whose trajectories are determined at the beginning of the order (deterministic strategies). In the case of the arrival price benchmark, an adaptive strategy may take advantage of price to lock in profit early in the trading horizon and reduce the risk for the remaining trading horizon (for example, for a buy order where price maintains a downward trend, the adaptive strategy will finish trading faster than a deterministic strategy to preserve the negative execution cost he have earned in the early periods). Similarly, the adaptive VWAP strategy may adjust allocation according to intraday volume changes, including both volume spikes and lulls with respect to the historical average volumes (for example, if the market volume is much higher than the predicted volume in earlier periods of trading, the adaptive strategy tends to trade faster than the deterministic strategy to "catch up").

Once the scheduling problem is solved, and the child order size is determined, one needs to specify how each individual order should be placed. This optimal order placement placement model (OOPM) decision involves the choice of an order type (limit order or market order), and order size. If the trader is patient, he can submit a limit order where he can specify the maximum price at which he is willing to buy or

price $\frac{500\times\$10+200\times\$11+300\times\$12}{500+200+300} = \$10.8$ matches the market VWAP. The execution cost is therefore \$10.8-\$10.8=\$0. Note that the price evolution does not play any role here. For example, if the trader's trades have market impacts on prices and drive the market prices higher to \$10.2,\$11.1 and \$12.3. Both the market VWAP and the trader's average execution price will be $\frac{5000\times\$10.2+2000\times\$11.1+3000\times\$12.3}{5000+2000+3000} = \$11.01$.

the minimum price at which he is willing to sell. If his price is inferior to prevailing market quotes (too low for a buy order or too high for a sell order), his limit orders may not be filled. On the other hand, if the trader is impatient and he is willing to trade at whatever price is necessary to complete the order, he can submit the market order to demand all liquidity available to him. For a buy order, OOPM seeks the optimal trade-off between liquidity risk and execution risk. Liquidity risk refers to the higher execution price paid when using market orders, which includes the market impact and the bid-ask spread. Execution risk refers to the uncertainty of order completion when using limit orders.

Currently, both industry and academia treat scheduling and OOPM as two separate problems. Each child order is assumed to be completed within its own time period in the scheduling problem. This is realized in OOPM by submitting all remaining shares as market orders at the end of each time period to guarantee child order completion. As a result of this separation, market orders may be submitted exactly when liquidity is scarce and execution cost is likely to be high. As a quick remedy, rather than submitting remaining shares through a market order, traders sometimes postpone these shares till the next period for better price opportunities. This remedy, however, breaks the assumption in the scheduling problem where all child orders are expected to be completely executed.

To address the lack of communication between scheduling and OOPM, Chapter 4 combines the two problems together and provides a unified approach[2]. Rather than deciding the child order size for each period, the unified approach determines the order sizes of both limit and market orders in each period. The benchmark price used in Chapter 4 is the arrival price. By combining scheduling and OOPM together, the unified approach optimizes the trade-offs among three market dynamic risks:

[2]To our best knowledge, besides [24], there are no papers that address scheduling and OOPM together for the optimal execution problem in the case of a risk-averse trader.

liquidity risk of market orders, execution risk of limit orders and timing risk due to price volatility.

We compare this unified adaptive strategy with two other adaptive strategies. Both of these strategies follow the optimal adaptive trading schedule, which minimizes the mean-variance of the execution cost with respect to the arrival price. The first benchmark strategy submits each child order as a market order. The second benchmark strategy submits both market and limit orders, but it separates trade scheduling and OOPM into two sequential steps, as in the common practice in the broker-dealer industry. Out-of-sample tests based on actual market data reveal that the unified approach consistently outperforms both benchmark strategies under a variety of settings.

The major contribution of this thesis is to provide a systematic framework that can be applied to make sequential adaptive decisions in algorithmic trading problems. In fact, all three problems from Chapter 2 - Chapter 4 are solved based on this framework. These problems often involve different market dynamic risks[3] and the key to solving them is to diversify these risks over sequential decisions, which are spread in time. Unlike traditional portfolio theory where the risk can be managed by diversifying over different investment assets (*"Don't put all your eggs in one basket."*),

[3]The term "risk" in this thesis is more general than just the standard deviation of a random variable. It includes various uncertainties in market dynamics. In this thesis, four types of market dynamic risks are referred:

1. timing risk: risk associated with price volatility;
2. liquidity risk: risk associated with short term supply and demand imbalance, which includes market impact, bid-ask spread, etc. It is often associated with market order submission;
3. execution risk: risk associated with incomplete order transaction, which is often associated with limit order submission;
4. prediction risk: risk associated with the difference between the prediction of a market variable and its actual realization.

Note that these market dynamic risks often play conflicting roles in different problems and an algorithmic trading problem often aims to achieve the optimal trade-offs among these risks. For example, Chapter 2 tries to balance the liquidity risk of fast trading against the timing risk of slow trading. Chapter 3 tries to balance volume prediction risk against timing risk. Chapter 4 considers the best trade-offs among liquidity risk, execution risk and timing risk.

risk management in algorithmic trading stresses diversifying risk over (sequential decision) time (*"It's ok to put all your eggs in one basket, if you do it one at a time."*).

This risk diversification framework can be applied to solve discrete time decision problems with mean-variance objectives:

$$\mathbf{MV}(\kappa)\text{:}\ \ \min \mathbb{E}[I] + \kappa \text{Var}[I].$$

where I is the execution cost and $\kappa > 0$ is the risk aversion factor with larger value of κ implying higher risk aversion. Since variance is not a time-consistent measure, the mean-variance problem $\mathbf{MV}(\kappa)$ can not be solved through dynamic programming. We circumvent this difficulty with a two step process. First, we solve a family of auxiliary linear-quadratic problems

$$\mathbf{LQ}(r_0)\text{:}\ \ \min_{\Pi} \mathbb{E}[r_0 I + I^2] \text{ for } r_0 \in \mathbf{R}$$

through dynamic programming where the linear-quadratic weights r_0 span all real values $\mathbf{R}$[4]; Second, by proving the solutions of the mean-variance problem also solves a particular linear-quadratic problem, the search for the solution of $\mathbf{MV}(\kappa)$ shrinks to the search for the appropriate linear-quadratic weight $r_0^*(\kappa)$. The key of this two steps process is to solve the linear quadratic problems $\mathbf{LQ}(r_0)$ for all possible $r_0 \in \mathbf{R}$.

One of the major innovation in this thesis is to address this dilemma by including r_0 as a state variable. This simple trick solves $\mathbf{LQ}(r_0)$ for $r_0 \in \mathbf{R}$ all at once through one round of Bellman-backward induction, which greatly increases computational efficiency[5]. More importantly, r_0, along with its evolution:

[4] Note that r_0 also controls risk aversion level with higher value of r_0 implying more risk seeking and lower value of r_0 implying more risk aversion, which has an opposite effect on risk aversion as κ in $\mathbf{MV}(\kappa)$.

[5] the solutions of the original mean-variance problems $\mathbf{MV}(\kappa)$ for all $\kappa > 0$ are also readily available, which allows the trader to conveniently depict the mean-variance scatter plot (i.e. efficient frontier) and choose the appropriate mean-variance trade-off directly from the efficient frontier.

$$r_i = r_0 + 2 \times \text{execution cost up to current time } i$$

play the role of dynamically adjusting risk aversion levels based on previous execution quality. For example, if the execution cost is small up to the current time (i.e. the trader is doing a good job so far), r_i will be small accordingly, which results in a higher risk aversion level for the remainder of the trading. Therefore, the trader will trade more quickly trying to conserve his realized gains and put less capital for the remaining execution. This mechanism of automatically adjusting their own risk-aversion levels is the main reason why adaptive strategies can outperform deterministic strategies.

To the best of our knowledge, there are few existing works in academia that directly address the discrete time order execution problem in a mean-variance framework[6]. Most trading models are formulated and solved in continuous-time and the continuous solution is used to approximate the discretized solutions. However, modeling in discretized time allows much more flexible assumptions. For example, market features such as the lead-lag relationship in price and volume, and autocorrelation in liquidity, cannot be easily modeled in continuous time. On the other hand, we choose variance as a risk measure because the simple mean-variance approach has the practical advantage that risk and reward are expressed as two real variables and then are easily understood and visualized in a two-dimensional plot. It is important to note that practitioners prefer the mean-variance characterization of the risk-reward trade-offs to the more mathematically sophisticated utility function formulations of the risk-reward trade-offs. More importantly, the framework we provide here allows the practitioners to efficiently solve for the optimal strategies for different risk aversion levels all at once, which makes it fairly easy for them to choose an appropriate risk aversion level that fits their objectives with simple visual inspection.

[6] Even models that address the same problem (i.e. discrete time mean-variance trading problem) tend to have overly complicated solution structures, see Section 2.4.3 for details.

From the industry perspective, the risk diversification framework provides a systematic methodology that practitioners can apply to solve complicated trading problems. Most trading models used in industry are highly heuristic. The decisions are often pre-determined in some parametric form to match the desired trading behavior. The unknown parameters are then determined by applying the parametric policy to historical data and choosing the parameters that would have optimized some objectives, such as minimizing risk adjusted execution cost. By pre-specifying a parametric policy space, practitioners may significantly restrict themselves to a limited (or even incorrect) decision space. The risk diversification framework presented here provides a completely different approach. Rather than guessing what the optimal strategy should look like, we stress the importance of first recognizing the major market dynamic risks for the specific problem[7]. In other words, *rather than directly guessing the trading strategy, we want to first understand how the market works.* It is not until we express the execution cost numerically as a function of various market dynamic risks will we be ready to recognize what the state variables are and what the optimal policy should look like[8]. We believe this simple framework can significantly improve risk management in multi-period trading problems for practitioners.

[7]For example, although we include market impact for the scheduling problem with the arrival price benchmark, it is omitted for the scheduling problem with the VWAP benchmark

[8]Note that the choice of the specific state variables determined through this systematic approach may not be very obvious at first glance. For example, although current execution cost is included in the state space for the scheduling problem with the arrival price benchmark, it is not included in the scheduling problem with the VWAP price benchmark. This may not be obvious during heuristic assessment.

CHAPTER II

TRADE SCHEDULING WITH ARRIVAL PRICE BENCHMARK

2.1 Introduction

A trend observed for the past decade in financial markets is the increase in the usage of algorithmic trading. Computer-driven trading not only allows broker-dealers to increase their capacity for execution business, but also allows investors to closely monitor and scrutinize the execution quality of their orders. Most importantly, algorithmic trading provides a systematic approach for traders to minimize execution costs.

When an order is executed in the financial market it tends to move the price to a trader's disadvantage; for instance, driving the price up when the trader is buying. This adverse price reaction to one's trading (namely, "market impact") is more significant when the order size is relatively large compared with average volume or when the asset is illiquid.

One of the most common ways to measure the execution cost caused by market impact is to obtain the "implementation shortfall", which is defined as the difference between the actual cost of the execution and the notional value of the order when it arrives at the trading desk. Implementation shortfall is usually the largest portion of the execution cost[1]. Two important sources contributing to the implementation shortfall include: limited liquidity (market impact) and adverse price movement due to volatility.

[1]The other parts of execution cost includes direct and predictable costs, such as commissions, taxes and exchange fees.

In order to minimize the expected implementation shortfall, traders split up the initial parent order into multiple small slices (i.e. child orders) and trade them sequentially over a given time horizon. However, the extended trading process incurs extra timing (or, volatility) risk, specifically, the risk of order execution price being significantly worse than the arrival price due to price volatility. The more volatile an asset is, the more widely its price fluctuates over the trading horizon, which makes the implementation shortfall more likely to increase with the length of trading horizon.

Volatility risk can be cut to zero if we execute all of the parent order at the arrival time. However this approach often results in maximal market impact. Therefore, a good execution strategy should maintain a balance between the minimization of both market impact and volatility risk. In this chapter, we measure volatility risk through the variance of implementation shortfall. As a result, our objective is to minimize the weighted sum of mean and variance of the implementation shortfall.

Up until recently, the focus of mean-variance execution has been restricted to deterministic (or static) strategies as in [2], which determine the child order sizes beforehand and do not change them as market conditions evolve. However, in practice, traders prefer more general adaptive (or dynamic) strategies, which generate child orders and their sizes at their respective trading times, depending on the latest available information such as price realizations. As we will show later, adaptive strategies can significantly reduce implementation shortfall and improve execution performance, especially for illiquid or less volatile assets. Due to the time inconsistency of the variance operator in the mean-variance criteria we clarify that the mean-variance objective in our problem is fixed at the arrival time and should not be modified during the trading process, which corresponds to Problem 2 in [31].

In this chapter, we consider a discrete-time trading problem over a finite-horizon, where the trading time is divided into multiple periods. Our goal is to derive an adaptive scheduling algorithm that splits the total order into a sequence of child orders, each of which is traded in a single time period. Unlike some trade scheduling problems such as [39], where the objective is to minimize a trader's expected utility, our problem cannot directly apply the traditional dynamic programming approach due to a failure of the iterated-expectations property for mean-variance objectives.

Inspired by [30], we circumvent the problem by introducing a family of auxiliary problems with linear-quadratic objectives and prove that the optimal strategy of the original mean-variance problem is a subset of the solutions for all auxiliary problems. By extending the weight in the linear-quadratic objective as a time changing state variable updated after each child order, we can solve all auxiliary problems with different weights through only one set of backward inductions, which simplifies computation significantly. Moreover, this state variable also keeps a record of realized (partial) implementation shortfall, which contains information regarding to past price realization. The state variable is then used to dictate the size of the next child order, which allows our algorithm to be adaptive to past price evolution.

It turns out that with the mean-variance objective, the optimal adaptive strategy is “”aggressive-in-the-money” as defined in [28], which means that for a buying order, we trade fast (i.e. large child order generated) when the price goes down (i.e. in the money), and slow when the price goes up. This is consistent with [3] and [31], and as explained in [3], the advantage of the adaptive strategy over the deterministic strategy lies in the anti-correlation between current execution and future execution: “after a fortunate price move the investor would try to conserve his realized gains and put less capital at risk in the remainder”.

An accurate capture of market impact is essential for the adaptation of trading strategies in practice. The modeling of illiquid market microstructure is still an ongoing research topic. In this chapter, we use a similar linear market impact model as [2] because it is simple enough for modeling while still maintaining the key properties for implementation shortfall based strategies. Yet, our linear-quadratic approach can be easily extended to more complex market models. For example, we will show that with an additional state variable, the same linear-quadratic approach can be extended to problems with the resilient market impact introduced in [33].

Recently a few papers have emerged on adaptive execution strategies. For example, [39] takes on a continuous-time problem over infinite time horizon with different utility functions. They have shown that the adaptability pattern (i.e. aggressive-in-the-money, passive-in-the-money, or deterministic) of the adaptive strategy is a result of utility function structure. The utility function of our auxiliary linear-quadratic problem falls into the category of increasing absolute risk aversion (IARA), and the adaptive strategy should be AIM, which is consistent with our findings in the numerical examples. [19] uses the Hamilton Jacobi Bellman approach to solve the continuous-time version of the auxiliary linear-quadratic problems.

Our problem is essentially the same as the one in [31], which also resort to dynamic programming to solve an equivalent constrained variance minimization problem. They achieve the backward induction through the decomposition of variance objective function by Law of Total Variance. However, their model structure is more complicated than that of ours: at each time period our decision variable is simply the child order size, while their decision variable contains both the child order size and an additional integrable function over the sample space of next period's price change. The integrable function represents the upper bound of the expected implementation shortfall with respect to the next price change for the remaining periods. As a well-known problem, dynamic programming suffers the notorious "curse of dimensionality" ([10]). Although the state variables in both our model and the one in [31] are of two dimensions, the dimension of the decision variables differ significantly: our model's decision variable is just a scalar while the decision variable in [31] infinite dimensions. In order to solve this dilemma, [31] proposes to approximate the integrable decision function through a step function. For example, a two-level step function may correspond to whether the price goes up or down for the next time period, which essentially restricts the stochastic price process into a binomial tree. While the approximation greatly reduces the decision variable to a three-dimensional one , it also restricts the adaptability of the trading algorithm. On the other hand, the scale of our algorithm's adaptability relies not on the decision variable but the state variable. With a delicate discretization of the two state variables, our strategy adapts to distinguish hundreds of different price-change levels per period rather than simplifying them as just "up" or "down". This helps us to achieve almost full price adaptability with even less computation complexity. Our numerical results have shown that with the same state variable resolution, our strategy performs consistently better than the restricted adaptive strategy in [31], and in some cases, saves more than 40% of the expected implementation shortfall.

The chapter is structured in the following way. Section 2 introduces the market model, defines the mean-variance problem, and provides the deterministic strategy as a benchmark for later comparison. Section 3 introduces the family of auxiliary linear-quadratic problems and provides an efficient algorithm to solve them all at once. It also explores the relationship between original mean-variance problem and auxiliary linear-quadratic problem, which allows us to essentially solve the original mean-variance problem. Section 4 provides the numerical results and presents its advantage over deterministic strategies and restricted adaptive strategies appeared in current literature. Section 5 concludes the chapter. To help readers follow the model, we try to use similar notation as [31] whenever possible.

2.2 The optimal trading problem

2.2.1 Security price dynamics

Consider a trader receiving an execution order of buying X shares of security within a fixed period of time T. For example, we can set $T = 1$ day, which is 6.5 hours for the US equity market. Without loss of generality, we will only consider buying orders in this chapter. The case for selling is completely analogous.

The security price is determined by both our execution and trading from a population of exogenous traders who submit their orders independently. We model the latter part as an arithmetic Brownian motion $S_t, t \in [0, T]$:

$$S_t = S_0 + \sigma S_0 B_t \tag{1}$$

where B_t is the standard Brownian motion, and S_0 is the initial mid quote when the order arrives at the trading desk. An observable proxy for S_t is the market mid quote at time t. σ is the standard deviation for the percentage of price change within one unit of time(such as one day). For example, if a stock's annual volatility is 20%, and its trading horizon is $T = 1$ day, one may take $\sigma = 20\%/\sqrt{252} \approx 0.0125 = 125\text{bps}$ (one bps $= 10^{-4}$). Although traditionally geometric Brownian motion is used to model the stock price, the trading time horizon in our problem is relatively short(varying from a few hours to a few days), and the assumption of arithmetic Brownian motion is not a major divergence from reality. Note that the assumption of zero drift in (1) underlies the price having no momentum or mean-reverting pattern. The independent increment of Brownian motion also implies it exhibits no autocorrelation.

We focus on a discrete-time framework and divide $[0, T]$ into N equally spanned time periods by $t_0 = 0, t_1, ..., t_{N-1}, t_N = T$. At time $t_i(i = 0, ..., N-1)$, the trader decides to buy a child order of y_i shares within period $[t_i, t_{i+1})$. Naturally,

$$\sum_{i=0}^{N-1} y_i = X \text{ and } y_i \geq 0 \text{ for } i = 0, 1, ..., N-1; \tag{2}$$

To insure trading actions are non-anticipating, we also require y_i to depend only on price up to time t_i, i.e.

$$y_i \text{ is adapted to } \mathcal{F}_{t_i} \text{ for } i = 0, 1, ..., N-1. \tag{3}$$

where $\mathcal{F}_t$ is the $\sigma-$field generated by the underlying Brownian motion $\{B_s : 0 \leq s < t\}$ for $t \in [0, T]$. Any $\pi = \{y_0, y_1, ..., y_{N-1}\}$ that satisfies (2) and (3) is called a feasible trading strategy, which constitutes the decision space of our optimization problem.

As the trader liquidates the child orders, he crosses the limit orders on the opposite side and moves the price towards his disadvantage. The price he actually pays deviates from the market price when the child order is submitted. This deviation is proportional to trading volume as well as trading speed and it is modeled as market impact, which consists of temporary impact and permanent impact on market microstructure[2]. Temporary market impact reflects cost of demanding liquidity, while permanent market impact corresponds to the long term price effect of the order, representing value information exposed to other market participants. There are many different models to quantify market impact, such as [1], [2] and [33]. The majority of this chapter will be based on the simple assumption of linear temporary impact[3]. However, the same methodology we introduce here can be applied to more complicated liquidity models. One such model is described in [33] with the addition of permanent and resilient market impact with exponential decay which depend on previous trading history. We illustrate the optimal strategy under this assumption in Appendix A.2.

The linear temporary market impact assumes the the gap between the executed price $\tilde{S}_{t_i}$ and the fundamental price S_{t_i} is proportional to the trading speed during

[2]The explanation for market impact here is based on standard market clearing framework. Some alternative explanations can be found in other literature, such as [12]. For example, it is believed trading orders convey a signal about private information. Therefore, other market participants may follow a particular buy order and believe its trader possessing more information than they do so they take long positions in the stock, which results in up trend price movement.

[3]It can be shown that when the temporary impact factors are assumed constant, the market model with both linear permanent and temporary impact can be restructured as a temporary impact only model.

ith period: $v_i = \frac{y_i}{1/N} = Ny_i$. Therefore, when y_i shares are bought, the executed price is raised up by $\eta_i v_i = N\eta_i y_i$. The impact factor η_i represents the average liquidity level of the underlying asset in period $[t_i, t_{i+1}]$: smaller value of η_i implies higher liquidity. We allow the market impact factor to be a function of trading period to reflects the trader's belief of liquidity profile during the trading day. If price and trading speed use units \$/share and share/time respectively, then impact factor η_i has the unit (\$/share)/(share/time). Furthermore, we assume more selling orders will be attracted to the order book after our trader exhausts the liquidity during period $[t_i, t_{i+1})$, and the price will return completely to the fundamental level $S_{t_{i+1}}$ before the next trading decision. In other words, our execution has no elastic effects on the future price process. In reality, execution price should also incorporate a certain amount of bid-ask spread into the execution price to capture the order placement quality. However, for simplicity, we will omit this part and assume the ith child order will be executed at market mid-quote S_{t_i} plus the temporary market impact $N\eta_i y_i$:

$$\tilde{S}_{t_i} = S_{t_i} + N\eta_i y_i. \tag{4}$$

To calibrate market impact factor η_i, we can simply assume η_i to be a constant η. One may set

$$\eta = (60 \text{ bps}) S_0 / \text{ADV}$$

where ADV is the average daily volume for the security with the unit (share/time). This assumption agrees with the fact that larger orders tend to have larger price impacts, and hence large transaction costs.

2.2.2 Implementation shortfall

The execution cost of a single completed trade is typically the difference between the final average trade price, including commissions, fees and all other costs, and a suitable benchmark price representing a hypothetical perfectly executed trade. The sign is taken so that positive cost represents loss of value: buying for a higher price or selling for a lower price. Some of the costs of trading are direct and predictable, such as broker commissions, taxes and exchange fees. Although these costs can be significant, they are commonly not included in the quantitative analysis of execution cost. Indirect costs include all other sources of price discrepancy, such as limited liquidity (market impact) and price motion due to volatility. In this article, we will assume that the execution cost consists entirely of the indirect costs, and will ignore the direct costs.

When the order size is fixed and the benchmark is the arrival price, i.e. the quoted market price in effect at the time the order was released to the trading desk, the execution cost can be equivalently defined as the difference between final purchase price and the notional value based on arrival price:

$$\begin{aligned} I_N &= \sum_{i=0}^{N-1} \tilde{S}_{t_i} y_i - X S_0 \\ &= \sum_{i=0}^{N-1} N \eta_i y_i^2 + \sum_{i=1}^{N-1} \sigma S_0 (B_{t_i} - B_{t_{i-1}}) x_i \end{aligned} \tag{5}$$

where $x_i = X - (y_0 + y_1 + ... + y_{i-1})$ is the remaining number of shares right before time t_i. I_N is often referred as implementation shortfall ([34]) or slippage. Clearly,

$$\begin{aligned} x_0 &= X, \\ x_{i+1} &= x_i - y_i \quad \text{for} \quad i = 0, 1, ..., N-1. \end{aligned} \tag{6}$$

By convention, we require $x_N = 0$.

Since x_i is $\mathcal{F}_{t_{i-1}}$ measurable, and $\{S_{t_i}\}$ is a martingale, we have $\mathbb{E}[S_{t_i}] = S_0$ and $\mathbb{E}[(S_{t_i} - S_{t_{i-1}})x_i] = 0$. The expected implementation shortfall is

$$\mathbb{E}[I_N] = N\mathbb{E}[\sum_{i=0}^{N-1} \eta_i y_i^2]. \tag{7}$$

The minimal value of (7) is

$$N\frac{X^2}{\sum_{j=0}^{N-1} 1/\eta_j}. \tag{8}$$

with

$$y_i = \frac{X/\eta_i}{\sum_{j=0}^{N-1} 1/\eta_j} \text{ for } i = 0, 1, ..., N-1.$$

When $\eta_i = \eta$ for each period i, $y_i = X/N$, and the minimal expected implementation shortfall is

$$N\eta X^2/N = \eta X^2.$$

This result has been established in [8]. In the following, we will refer to this as the linear strategy. The final average execution price of the linear strategy is the same as the time-weighted average price (TWAP) of the whole trading horizon. If the market volume for each period is constant, TWAP and VWAP (volume weighted average price) are the same. The constant participation rate through all time periods is due to the assumption of fundamental price following arithmetic Brownian motion with zero-drift, which shows no correlation between prices of adjacent periods. Note that when the objective is just minimizing average implementation shortfall, the optimal strategy is deterministic(or non-random). The size of child orders are determined even before the parent order's arrival, and it will remain unchanged during execution process regardless of any price development. In the following sections, we will include a risk factor into the utility function. With risk considered, the optimal trading strategy turns out to be dependent on price realization.

2.2.3 Risk aversion

To simplify notation, we divide the implementation shortfall by the product of the initial notional cost and volatility. Namely,

$$\tilde{I}_N = \frac{I_N}{\sigma X S_0} \tag{9}$$

Thus, $\sigma\tilde{I}$ is the implementation shortfall divided by the total amount of "paper money". Multiplying by 10^4 as $10^4\sigma\tilde{I}$ gives the slippage value in terms of bps, which often appears in the brokerage's execution cost analysis report. Let

$$\begin{aligned}\tilde{y}_i &= y_i/X, \text{ for } i = 0, 1, ..., N-1,\\ \tilde{x}_i &= x_i/X, \text{ for } i = 0, 1, ..., N-1.\end{aligned}$$

In practice, if the minimal unit for trading is assumed to be d shares(such as 1 lot=100 shares), both $\tilde{y}_i$ and $\tilde{x}_i$ will be restricted to values within $\{0, \frac{d}{X}, \frac{2d}{X}, ..., 1\}$.

Constraints (2) and (6) become

$$\sum_{i=0}^{N-1} \tilde{y}_i = 1 \text{ and } \tilde{y}_i \geq 0 \text{ for } i = 0, 1, ..., N-1; \tag{10}$$

$$\tilde{x}_{i+1} = \tilde{x}_i - \tilde{y}_i \text{ for } i = 0, 1, ..., N-1 \tag{11}$$

Based on (5), the scaled implementation shortfall is

$$\tilde{I}_N = \sum_{i=0}^{N-1} N\tilde{\mu}_i\tilde{y}_i^2 + \sum_{i=1}^{N-1}(B_{t_i} - B_{t_{i-1}})\tilde{x}_i \tag{12}$$

where

$$\tilde{\mu}_i = \frac{\eta_i X}{\sigma S_0} = (60\text{bps})(\frac{X}{ADV})\frac{1}{\sigma}. \tag{13}$$

For example, if $X = 1$ million shares, and ADV= 10 million shares, the whole trade accounts for 10% of the average daily volume, and the associated impact factor is $\tilde{\mu}_i = 6\text{bps}/\sigma$. The scale of μ_i can be interpreted this way: regardless of the volatility level, if we trade all the parent order instantaneously at $t = 0$ with a POV(participation of volume) rate of 10%, the implementation shortfall will be 6bps. The idea of using an instantaneous trade's execution cost to estimate the market impact factor originated in the market imbalance model in [28].

If we assume the liquidity level remains the same during the trading horizon, then $\tilde{\mu}_i = \tilde{\mu}$ for $i = 1, 2, ..., N-1$, which is exactly the same nondimensional "market power" parameter defined in [31]. The market power parameter $\tilde{\mu}$ also reveals the relative role of market impact factor η_i and volatility σ in determining the optimal strategy. For example, with all other settings equal, a security with twice the market impact should be traded in the same pattern as a security with half volatility. Both terms in (12) have economic interpretations. The first term corresponds to the transaction costs due to temporary market impact. The second term corresponds to the volatility risk accumulated throughout $[0, T]$ rather than buying all N shares instantaneously at the beginning.

For the rest of the chapter, we will focus on $\tilde{I}_N$ defined in (12) with market power $\tilde{\mu}_i$ defined in (13). We will drop the tilde notation from now on.

With a single goal of minimizing average slippage, the optimal execution strategy (i.e. linear strategy) takes as long as necessary to prevent significant market impact. As (8), its expected implementation shortfall is

$$\begin{aligned} \mathbb{E}[I_{lin}] &= \frac{N}{\sum_{j=0}^{N-1} 1/\mu_j}, \\ Var[I_{lin}] &= \frac{1}{3}\big(1 - \frac{1}{N}\big)\big(1 - \frac{1}{2N}\big). \end{aligned}$$

However, it exposes the order to a much greater timing risk, especially for volatile securities. The more volatile a security is, the more likely its price will move up and its implementation shortfall will increase (for a buy order). Furthermore, the linear strategy sometimes generates a nonintuitive result. As mentioned above, when the market power (or market impact) is assumed to be constant throughout trading horizon, a simple objective of minimizing the average implementation shortfall will lead to a constant speed of accumulating shares over the whole trading horizon, which contradicts the real practice of tradng a liquid security faster than an illiquid one.

This motivates [2] to introduce variance of implementations shortfall as the risk component into the objective function, where the optimal execution problem is treated analogous to the classical portfolio management problem [32], and the efficient frontier of optimal trading execution are constructed.

The trader's urgency preference is reflected through risk aversion factor $\kappa > 0$, with a larger value of κ implying quicker execution intention. Our goal is to minimize the sum of the expectation and variance of the implementation shortfall weighted by κ:

$$\mathbf{MV}(\kappa) : \min_{\Pi} \mathbb{E}[I_N] + \kappa \text{Var}[I_N] \tag{14}$$

where Π stands for all feasible strategies:

$$\begin{aligned} \Pi = \quad \{ \quad & \pi = (y_0, y_1, ..., y_{N-1}) | \forall i, y_i \text{ is a random variable adapted to } \mathcal{F}_{t_i}; \\ & \pi \text{ satisfies (10) and (11)} \}. \end{aligned} \tag{15}$$

Finding the appropriate risk aversion factor is not easy. Brokers sometimes sort the incoming orders into a few different categories such as "urgent", "neutral" or "passive" according to clients' preferences, and map each category with a particular κ value. An alternative approach is to specify a target variance level $v > 0$ while minimizing the expected implementation shortfall:

$$\begin{aligned} \mathbf{E}(v) \quad : \quad & \min_{\Pi} \mathbb{E}[I_N] \\ & \text{s.t. } \text{Var}[I_N] \leq v \end{aligned} \tag{16}$$

or equivalently, to specify an expected slippage level $m > 0$ while minimizing its risk:

$$\begin{aligned} \mathbf{V}(m) \quad : \quad & \min_{\Pi} \text{Var}[I_N] \\ & \text{s.t. } \mathbb{E}[I_N] \leq m \end{aligned} \tag{17}$$

From the theory of Lagrange methods, all three formulations (14) - (17) are equivalent in producing the same set of optimal trading strategies:

Property 1 1. *Considering three dynamic programming problems: (14),(16) and (17). If* $\pi^*(\kappa)$ *is the optimal strategy of* $\boldsymbol{MV}(\kappa)$ *in (14), then* $\pi^*(\kappa)$ *also solves* $\boldsymbol{E}(v)$ *with* $v = Var[I_N|_{\pi^*(\kappa)}]$ *and* $\boldsymbol{V}(m)$ *with* $m = \mathbf{E}[I_N|_{\pi^*(\kappa)}]$.

By varying κ, we plot the optimal trading strategy's efficient frontier(EF):

$$EF = \{(\mathbf{E}[I_N|_{\pi^*(\kappa)}], \text{Var}[I_N|_{\pi^*(\kappa)})] \quad | \quad \kappa > 0 \quad \text{and} \quad \pi^*(\kappa) \text{ solves } \mathbf{MV}(\kappa)\}. \tag{18}$$

Any point (m, v) that falls on the efficient frontier satisfies the following the property: there does not exists a strategy $\pi \in \Pi$ such that both $\mathbf{E}[I_N|_\pi] < m$ and $\text{Var}[I_N|_\pi] < v$. Because of the equivalence shown in **Property 1**, we will mainly focus on $\mathbf{MV}(\kappa)$ in the following analysis.

2.2.4 Deterministic strategies

Before deriving the optimal strategy for $\mathbf{MV}(\kappa)$, let's consider the suboptimal solution within deterministic strategy space $\hat{\Pi} \subset \Pi$:

$$\mathbf{D}(\kappa) : \min_{\hat{\Pi}} \mathbb{E}[I_N] + \kappa \text{Var}[I_N] \tag{19}$$

where

$$\hat{\Pi} = \{\pi = (y_0, y_1, ..., y_{N-1}) | \forall i, y_i \text{ is non-random scalar}; \pi \text{ satisfies (10) and (11)}\}. \tag{20}$$

Compare the definition of Π in (15) and $\hat{\Pi}$ in (20): an adaptive strategy $\pi \in \Pi$ is a discrete stochastic process with filtration $\mathcal{F}_t$ while a deterministic strategy $\hat{\pi} \in \hat{\Pi}$ is just a non-random N dimension real vector.

Given a deterministic strategy $\hat{\pi} = \{y_0, y_1, ..., y_{N-1}\}$, the variance of the implementation shortfall depends only on the variance of the price change:

$$\text{Var}[I_N] = \frac{T}{N} \sum_{i=1}^{N-1} x_i^2, \tag{21}$$

Therefore, the total mean-variance cost is

$$\sum_{i=0}^{N-1} N\mu_i y_i^2 + \frac{\kappa T}{N}\sum_{i=1}^{N-1} x_i^2. \tag{22}$$

When the market power is assumed to be a constant μ, the deterministic strategy that minimizes (22) is the solution to this:

$$\min_{\hat{\Pi}} \sum_{i=0}^{N-1} y_i^2 + \alpha \sum_{i=1}^{N-1} x_i^2 \tag{23}$$

where

$$\alpha = \frac{\kappa T}{N^2 \mu} > 0. \tag{24}$$

(23) is a convex optimization problem with convex constraints, which can be solved with the Lagrange method. We provide a recursive procedure to derive the optimal deterministic strategy $\hat{\pi}^* \in \hat{\Pi}$ in Appendix A.1. In the following, we will use $\hat{\pi}^* \in \hat{\Pi}$ to refer to the optimal deterministic strategy for $\mathbf{D}(\kappa)$. [2] solved the same problem by solving difference equations.

By combining market power parameters μ(consists of liquidity η and volatility σ) and risk aversion factor κ into one parameter α, we simplify the deterministic strategy's dependence on exogenous setting. Through (13) and (24) we observe that α increases as the market impact factor decreases (i.e. security is more liquid), price volatility increases (i.e. security is more volatile) and the trader's risk aversion increases. All three factors result in a strategy that trades more quickly in the early periods, such that the volatility risk can be minimized. However, trading more quickly earlier incurs a higher market impact cost at the beginning. In the extreme case, when $\alpha = \infty$, the optimal deterministic strategy is to buy all the shares at the beginning: $y_0 = 1, y_i = 0$ for $i > 0$. In this case, the trader exposes himself to no further uncertainties, but he pays the highest liquidity premium for incurring the largest market impact:

$$\mathbb{E}[I_{inst}] = N\mu_0.$$

On the other hand, in the setting of a less risk averse customer or a less volatile or liquid security, slower trading may be preferred. In the extreme case when $\alpha = 0$, the deterministic strategy buys shares at a constant speed: $y_i = \frac{1}{N}$ for $i = 0, 1, ..., N-1$, which is the linear strategy mentioned above. It provides the minimum average implementation shortfall, but exposes the trader to the greatest timing risk.

2.3 *Linear quadratic formulation*

We now expand the decision space from deterministic strategies $\hat{\Pi}$ to adaptive strategies Π. We hope to use dynamic programming to solve the mean-variance problem. However, a direct implementation to minimize the mean-variance of implementation shortfall as in (14) is not amenable since the variance operator does not satisfy the smoothing property, i.e. $\forall 0 \le s \le t$, $\mathrm{Var}[\mathrm{Var}(\cdot|\mathcal{F}_t)|\mathcal{F}_s] \neq \mathrm{Var}(\cdot|\mathcal{F}_s)$.

Inspired by [30], we set up a family of auxiliary problems that can be solved by dynamic programming, and show that the optimal strategy of a certain auxiliary problem also solves the original problem $\mathbf{MV}(\kappa)$.

2.3.1 Auxiliary problem

For $r_0 \in \mathbf{R}$, consider the linear quadratic objective:

$$\mathbf{LQ}(r_0): \ \min_{\Pi} \mathbb{E}[r_0 I_N + I_N^2]. \tag{25}$$

where Π is the same set of feasible strategies as in (15). $\mathbf{LQ}(r_0)$ can be solved by dynamic programming since the expectation operator satisfies smoothing property: $\forall 0 \le s \le t$, $\mathbb{E}[\mathbb{E}(\cdot|\mathcal{F}_t)|\mathcal{F}_s] = \mathbb{E}(\cdot|\mathcal{F}_s)$. The following proposition underlies the relationship between solutions of $\mathbf{MV}(\kappa)$ and $\mathbf{LQ}(r_0)$. It follows a similar proof as in [30].

Proposition 1. *Let*

$$\begin{aligned}
\Pi_{\boldsymbol{MV}}(\kappa) &= \{\pi|\pi \in \Pi \textit{ and } \pi \textit{ is a minimizer of (14)}\}, \\
\Pi_{\boldsymbol{LQ}}(r_0) &= \{\pi|\pi \in \Pi \textit{ and } \pi \textit{ is a minimizer of (25)}\}, \\
\Pi_{\boldsymbol{LQ}} &= \bigcup_{r_0 \in \mathbf{R}} \Pi_{\boldsymbol{LQ}}(r_0),
\end{aligned}$$

then $\Pi_{\boldsymbol{MV}}(\kappa) \subset \Pi_{\boldsymbol{LQ}}$*. More specifically, if* $\pi^*(\kappa) \in \Pi_{\boldsymbol{MV}}(\kappa)$*, then* $\pi^*(\kappa) \in \Pi_{\boldsymbol{LQ}}(r_0^*)$ *where*

$$r_0^* = \frac{1}{\kappa} - 2\mathbb{E}[I_N|_{\pi^*(\kappa)}]. \tag{26}$$

Proof. Define $f(a,b) = a + \kappa b - \kappa a^2$ be a map on $\mathbf{R}^2 \to \mathbf{R}$. The objective function in (14) can be writtten as $\mathbb{E}[I_N] + \kappa \text{Var}[I_N] = f(\mathbb{E}[I_N], \mathbb{E}[I_N^2])$. Assume $\pi^*(\kappa) \in \Pi_{\mathbf{MV}}(\kappa)$, then $\forall \pi \in \Pi$:

$$f(\mathbb{E}[I_N|_\pi], \mathbb{E}[I_N^2|_\pi]) \geq f(\mathbb{E}[I_N|_{\pi^*(\kappa)}], \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]). \tag{27}$$

Let f_a, f_b be the partial derivatives with respect to first and second variable. Since $f(a,b)$ is a concave function of a and b,

$$\begin{aligned}
f(\mathbb{E}[I_N|_\pi], \mathbb{E}[I_N^2|_\pi]) &\leq f(\mathbb{E}[I_N|_{\pi^*(\kappa)}], \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]) \\
&\quad + f_a(\mathbb{E}[I_N|_{\pi^*(\kappa)}], \mathbb{E}[I_N^2|_{\pi^*(\kappa)}])(\mathbb{E}[I_N|_\pi] - \mathbb{E}[I_N|_{\pi^*(\kappa)}]) \\
&\quad + f_b(\mathbb{E}[I_N|_{\pi^*(\kappa)}], \mathbb{E}[I_N^2|_{\pi^*(\kappa)}])(\mathbb{E}[I_N^2|_\pi] - \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]) \\
&= f(\mathbb{E}[I_N|_{\pi^*(\kappa)}], \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]) \\
&\quad + (1 - 2\kappa \mathbb{E}[I_N|_{\pi^*(\kappa)}])(\mathbb{E}[I_N|_\pi] - \mathbb{E}[I_N|_{\pi^*(\kappa)}]) \\
&\quad + \kappa(\mathbb{E}[I_N^2|_\pi] - \mathbb{E}[I_N^2|_{\pi^*(\kappa)}])
\end{aligned} \tag{28}$$

Combine (27) and (28):

$$(1 - 2\kappa \mathbb{E}[I_N|_{\pi^*(\kappa)}])(\mathbb{E}[I_N|_\pi] - \mathbb{E}[I_N|_{\pi^*(\kappa)}]) + \kappa(\mathbb{E}[I^2|_\pi] - \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]) \geq 0. \tag{29}$$

Let

$$r_0^* = \frac{1}{\kappa} - 2\mathbb{E}[I_N|_{\pi^*(\kappa)}], \tag{30}$$

As $\kappa \geq 0$, through (29) we have

$$r_0^* \mathbb{E}[I_N|_\pi] + \mathbb{E}[I_N^2|_\pi] \geq r_0^* \mathbb{E}[I_N|_{\pi^*(\kappa)}] + \mathbb{E}[I_N^2|_{\pi^*(\kappa)}]. \tag{31}$$

Because (31) holds for $\forall \pi \in \Pi$, $\pi^*(\kappa) \in \Pi_{\mathbf{LQ}}(r_0^*)$. Since this is true for $\forall \pi^*(\kappa) \in \Pi_{\mathbf{MV}}(\kappa)$, $\Pi_{\mathbf{MV}}(\kappa) \subset \bigcup_{r_0 \in \mathbf{R}} \Pi_{\mathbf{LQ}}(r_0) = \Pi_{\mathbf{LQ}}$. $\square$

The above **Proposition** shows that by choosing the appropriate r_0^*, we can derive the optimal strategy of the original problem $\mathbf{MV}(\kappa)$ by solving auxiliary problem $\mathbf{LQ}(r_0^*)$. However, according to (26), r_0^* depends on $\pi^*(\kappa) \in \Pi_{\mathbf{LQ}}(r_0^*)$, which would not be available until we know the value of r_0^*. Thus we have a "chicken and egg" dilemma.

One solution is to generate optimal strategies of $\mathbf{LQ}(r_0)$ for all r_0 values on the real line, i.e. attaining the set $\Pi_{\mathbf{LQ}}$. Since $\Pi_{\mathbf{MV}}(\kappa) \subset \Pi_{\mathbf{LQ}}$, an optimal strategy $\pi^*(\kappa) \in \Pi_{\mathbf{MV}}(\kappa)$ that solves $\mathbf{MV}(\kappa)$ should satisfy

$$\pi^*(\kappa) = \arg \min_{\pi \in \Pi_{\mathbf{LQ}}} \{\mathbb{E}[I_N|_\pi] + \kappa \mathrm{Var}[I_N|_\pi]\} \tag{32}$$

The search for $\pi^*(\kappa)$ can be decomposed as two respective optimization steps. First, note that for $r_0 \in \mathbf{R}$, $\mathbf{LQ}(r_0)$ may have multiple solutions which all contribute to $\Pi_{\mathbf{LQ}}(r_0)$. For $r_0 \in \mathbf{R}$, we are only interested in the strategy within $\Pi_{\mathbf{LQ}}(r_0)$ that minimize the mean-variance objective, which we define as a representing strategy $\pi(r_0)$ for the set $\Pi_{\mathbf{LQ}}(r_0)$:

$$\pi(r_0) = \arg \min_{\pi \in \Pi_{\mathbf{LQ}}(r_0)} \{\mathbb{E}[I_N|_\pi] + \kappa \mathrm{Var}[I_N|_\pi]\}. \tag{33}$$

Secondly define $r_0^*(\kappa)$ as

$$r_0^*(\kappa) = \arg \min_{r_0 \in \mathbf{R}} \{\mathbb{E}[I_N|_{\pi(r_0)}] + \kappa \mathrm{Var}[I_N|_{\pi(r_0)}]\}. \tag{34}$$

Naturally the optimal strategy $\pi^*(\kappa) \in \Pi_{\mathbf{MV}}(\kappa)$ satisifes

$$\pi^*(\kappa) = \pi(r_0^*(\kappa)).$$

By representing $\Pi_{\mathbf{LQ}}(r_0)$ through a single element $\pi(r_0) \in \Pi_{\mathbf{LQ}}(r_0)$, we shift our focus from directly searching $\pi^*(\kappa)$ within $\Pi_{\mathbf{LQ}}$ as in (32) to searching $r_0^*(\kappa)$ within $\mathbf{R}$ as in (34).

A procedure similar to the one used in (33)-(34) can be applied to solve $\mathbf{V}(m)$ or $\mathbf{E}(v)$, although the mean-variance minimization operator in the definition of both $\pi(r_0)$ and $r_0^*(\kappa)$ would have to be replaced by constraint optimization operators. Without confusion, we will use the same notations $\pi(r_0)$ and r_0^* as in (33)-(34) for the following (35)-(38). The differences lie in the independent variables risk aversion factor κ, target mean m or target variance v):

Corollary 1. *For* $\mathbb{E}[I_{lin}] \leq m \leq \mathbb{E}[I_{inst}]$*, let*

$$\pi(r_0) = \arg \min_{\pi \in \Pi_{\boldsymbol{LQ}}(r_0)} \{ Var[I_N|_\pi] \quad | \quad \mathbb{E}[I_N|_\pi] \leq m\} \tag{35}$$

and

$$r_0^*(m) = \arg \min_{r_0 \in \mathbf{R}} \{ Var[I_N|_{\pi(r_0)}] \quad | \quad \mathbb{E}[I_N|_{\pi(r_0)}] \leq m\}, \tag{36}$$

then $\pi(r_0^*(m))$ *solves* $\boldsymbol{V}(m)$*.*

For $0 \leq v \leq Var[I_{lin}]$*, let*

$$\pi(r_0) = \arg \min_{\pi \in \Pi_{\boldsymbol{LQ}}(r_0)} \{\mathbb{E}[I_N|_\pi] \quad | \quad Var[I_N|_\pi] \leq v\} \tag{37}$$

and

$$r_0^*(v) = \arg \min_{r_0 \in \mathbf{R}} \{\mathbb{E}[I_N|_{\pi(r_0)}] \quad | \quad Var[I_N|_{\pi(r_0)}] \leq v\}, \tag{38}$$

then $\pi(r_0^*(v))$ *solves* $\boldsymbol{E}(v)$*.*

Proof. We prove $\pi(r_0^*(m))$ solves $\mathbf{V}(m)$, the case for $\mathbf{E}(v)$ is analogous. Define

$$v' = \min_{\pi \in \Pi} [\mathrm{Var}(I_N|_\pi) \quad | \quad \mathbb{E}(I_N|_\pi) \leq m].$$

If $\pi(r_0^*(m))$ does not solve $\mathbf{V}(m)$, then $v' < \mathrm{Var}[I_N|_{\pi(r_0^*(m))}]$. Since $\mathbb{E}I_{lin} \leq m \leq \mathbb{E}I_{inst}$, there exists $\kappa' \in \mathbf{R}$ such that the solution of $\mathbf{MV}(\kappa')$ and $\mathbf{V}(m)$ coincides, i.e. $\exists \pi' \in$

$\Pi_{\mathbf{MV}}(\kappa')$, such that

$$v' = \mathrm{Var}[I_N|_{\pi'}] \quad \text{and} \quad \mathbb{E}[I_N|_{\pi'}] \le m.$$

Due to **Proposition**, $\pi' \in \Pi_{\mathbf{LQ}}$. From the definition of $\pi(r_0)$ in (35) and $r_0^*(m)$ in (36),

$$v' = \mathrm{Var}[I_N|_{\pi'}] \ge \min_{\pi \in \Pi_{\mathbf{LQ}}} \{\mathrm{Var}[I_N|_{\pi}] | \mathbb{E}[I_N|_{\pi}] \le m\} = \mathrm{Var}[I_N|_{\pi(r_0^*(m))}].$$

This leads to a contradiction, and hence $\pi(r_0^*(m))$ does solve $\mathbf{V}(m)$. □

r_0 in $\mathbf{LQ}(r_0)$ represents the asssigned weight ratio between first and second moment of implementation shortfall, with different values of r_0 corresponding to different optimal strategies. However, (34), (36) and (38) all require us to solve $\mathbf{LQ}(r_0)$ for any $r_0 \in \mathbf{R}$. Later for numerical computation, we will solve (34), (36) and (38) through $\mathbf{LQ}(r_0)$ only for a finite number of discretized r_0 values(ranging from 100 to 600 values). Even so, using dynamic programming methods to solve $\mathbf{LQ}(r_0)$ one by one will still be computationally intensive. The following algorithm solves the optimal strategies for $r_0 \in \mathbf{R}$ all at once by extending r_0 as a state variable r_i for $i = 0, ..., N-1$, which greatly increases the computation efficiency. (34), (36) and (38) provide a mechanism to choose the approximate r_0 to derive the optimal strategies for different risk aversion levels κ (or target mean m or variance v). More importantly, the introduction of time varying state variables r_i also allows the trader the flexibility during execution to drift away from predetermined adaptive policies for changed risk version factor κ_i (or target mean m_i or variance v_i) for the remaining trading time periods.

2.3.2 Dynamic programming for auxiliary problem

Given a feasible strategy $\pi = \{y_0, y_1, ..., y_{N-1}\} \in \Pi$, define the first and last i periods' partial implementation shortfalls I_i and $\bar{I}_i$ for $i = 1, 2, ..., N$ as (for notational conformity, assume $B_{t_N} = B_{t_{N-1}}$ and $y_{-1} = 0$):

$$\begin{aligned} I_i &= \sum_{j=0}^{i-1} \left[N\mu_j y_j^2 + (B_{t_{j+1}} - B_{t_j}) x_{j+1} \right] \\ \bar{I}_i &= \sum_{j=N-i}^{N-1} \left[N\mu_j y_j^2 + (B_{t_{j+1}} - B_{t_j}) x_{j+1} \right] \end{aligned} \tag{39}$$

Obviously, $\bar{I}_i + I_{N-i} = \bar{I}_N = I_N, \forall i = 1, 2, ..., N$.

For any given r_0 and π, define r_i as

$$r_i = r_0 + 2I_i \text{ for } i = 1, 2, ..., N-1. \tag{40}$$

For a given r_0, r_i contains information about the realized partial implementation shortfall before time t_i to be $(r_i - r_0)/2$, which is adaptive to price process up to time t_i. To facilitate solving $\mathbf{LQ}(r_0)$ through dynamic programming, we decompose its objective function $r_0\bar{I}_N + \bar{I}_N^2$ to a summation of cost functions at each time period:

$$\begin{aligned}
& r_0\bar{I}_N + \bar{I}_N^2 \\
= \; & r_0(N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})x_1 + \bar{I}_{N-1}) + (N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})x_1 + \bar{I}_{N-1})^2 \\
= \; & r_0(N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})x_1) + (N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})x_1)^2 \\
& + (r_0 + 2N\mu_0 y_0^2 + 2(B_{t_1} - B_{t_0})x_1)\bar{I}_{N-1} + \bar{I}_{N-1}^2 \\
= \; & r_0(N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})(x_0 - y_0)) + (N\mu_0 y_0^2 + (B_{t_1} - B_{t_0})(x_0 - y_0))^2 \\
& + r_1\bar{I}_{N-1} + \bar{I}_{N-1}^2 \\
= \; & \sum_{j=0}^{i-1} \left[r_j(N\mu_j y_j^2 + (B_{t_{j+1}} - B_{t_j})(x_j - y_j)) + (N\mu_j y_j^2 + (B_{t_{j+1}} - B_{t_j})(x_j - y_j))^2 \right] \\
& + r_i\bar{I}_{N-i} + \bar{I}_{N-i}^2 \\
= \; & \sum_{i=0}^{N-1} \left[r_i(N\mu_i y_i^2 + (B_{t_{i+1}} - B_{t_i})(x_i - y_i)) + (N\mu_i y_i^2 + (B_{t_{i+1}} - B_{t_i})(x_i - y_i))^2 \right]
\end{aligned} \tag{41}$$

To preserve the Markovian property in making decision y_i at time t_i, we not only need to know the remaining percentage of shares x_i, but also the realized partial implementation shortfall through r_i. Furthermore, just as r_0 represents the desired ratio of weights between $\bar{I}_N$ and $\bar{I}_N^2$, (41) shows that r_i also works as the (implied) ratio of weights between $\bar{I}_{N-i}$ and $\bar{I}_{N-i}^2$. We set the combinations (x_i, r_i) as our state variables. Combined with the decisions y_i, $\{((x_i, r_i), y_i)|i = 0, 1, ..., N-1\}$ is known as a finite time horizon Markov Decision Process(MDP). The state variables follow the transition functions:

$$\begin{aligned} x_{i+1} &= x_i - y_i \quad \text{for } i = 0, 1, ..., N-1, && (42) \\ r_{i+1} &= r_i + 2(N\mu_i y_i^2 + (B_{t_{i+1}} - B_{t_i})x_i) \quad \text{for } i = 0, 1, ..., N-1. && (43) \end{aligned}$$

Define each item in the sum (41) as the cost function C_i for $i = 0, 1, ..., N-1$:

$$C_i((x, r), y, \Delta S) = r(N\mu_i y^2 + \Delta S(x - y)) + (N\mu_i y^2 + \Delta S(x - y))^2. \quad (44)$$

with the mean

$$\begin{aligned} c_i((x, r), y) &= \mathbb{E}[C_i((x, r), y, (B_{t_{i+1}} - B_{t_i}))] \\ &= rN\mu_i y^2 + (N\mu_i y^2)^2 + \frac{T}{N}(x - y)^2. && (45) \end{aligned}$$

The problem (25) is equivalent to

$$\mathbf{LQ}(r_0){:}\ \min_{\Pi} \mathbb{E} \sum_{i=0}^{N-1} C_i((x_i, r_i), y_i, B_{t_{i+1}} - B_{t_i}). \quad (46)$$

Assume the optimal strategy $\pi^* = \{y_0^*, y_1^*, ..., y_{N-1}^*\}$ for $\mathbf{LQ}(r_0)$ is given, the value functions(also called cost-to-go function) are defined as the expected costs using policy π^* from time t_i onward

$$V_i(x_i, r_i) = \mathbb{E} \sum_{j=i}^{N-1} C_j((x_j, r_j), y_j^*, B_{t_{j+1}} - B_{t_j})$$

A fundamental result of dynamic programming is the Bellman's equation, which describes the recursive relationship of value functions for different time periods and hence provides a backward induction algorithm to compute value functions and optimal strategies:

First, since we need to finish buying all the shares by time T, the optimal action in the last time period is deterministic:

$$y^*_{N-1}(x,r) \quad = \quad x; \tag{47}$$

$$V_{N-1}(x,r) \quad = \quad rN\mu_{N-1}x^2 + (N\mu_{N-1}x^2)^2. \tag{48}$$

Then for $i = N-2, N-3, ..., 0$, the Bellman's equations are:

$$V_i(x,r) = \min_{0\le y\le x}\{c_i((x,r),y) + \mathbb{E}[V_{i+1}(x-y, r+2N\mu_i y^2 + 2(B_{t_{i+1}} - B_{t_i})(x-y))|(x,r)]\} \tag{49}$$

Since the search space for y in (49) is finite, so $\forall i = 0,1,...,N-2, \forall r \in \mathbf{R}$ and $\forall X \in \{0, \frac{d}{X}, \frac{2d}{X}, ..., 1\}$, the optimizer in (49) $y^*_i(X,r)$ exists and is well defined. However, it may not be unique. All possible combination of the sequential decision functions $y^*_i(x,r)$ are combined as Π^*:

$$\begin{aligned}\Pi^*_{\mathbf{LQ}} = \{\{y^*_i(x_i,r_i)\}_{i=0}^{i=N-1} \quad | \quad & y^*_{N-1}(x_{N-1}, r_{N-1}) \text{ satisfies (47) },\\ & \{y^*_i(x_i,r_i)\}_{i=0}^{i=N-2} \text{ minimizes (49)}\\ & \text{for } x_0 = 1, r_0 \in \mathbf{R} \text{ and } (x_i,r_i)_{i=1}^{i=N-1} \text{ satisfies (42),(43)}\}.\end{aligned} \tag{50}$$

$y^*_i(x,r)$ gives the percentage of shares to trade at time $[t_i, t_{i+1})$ when the remaining percentage of shares is x and the realized implementation shortfall is $(r-r_0)/2$ for a given fixed r_0. Note that the realized value of r by time t_i depends on price process by time t_i: $\{S_t|t \in [0,t_i]\}$, so the child order volume percentage $y^*_i(x,r)$ is adaptive to $\mathcal{F}_{t_i}$. Therefore, $\Pi^*_{\mathbf{LQ}}$ consists feasible strategies:

$$\Pi^*_{\mathbf{LQ}} \subset \Pi.$$

Furthermore, for a particular r_0 value such as $r_0 = r'_0$ with $r'_0 \in \mathbf{R}$, we can attain the optimal strategies for $\mathbf{LQ}(r'_0)$ by setting the initial state $r_0 = r'_0$ in $\Pi^*_{\mathbf{LQ}}$:

$$\Pi^*_{\mathbf{LQ}}(r'_0) = \{\pi^*|r_0 = r'_0 \quad \text{and} \quad \pi^* \in \Pi^*_{\mathbf{LQ}}\}. \tag{51}$$

At this moment, $\Pi^*_{\mathbf{LQ}} = \Pi_{\mathbf{LQ}}$ and $\Pi^*_{\mathbf{LQ}}(r'_0) = \Pi_{\mathbf{LQ}}(r'_0)$. However, in the following, we will cut off the state space of r_i for $i = 0, 1, ..., N-1$ from $\mathbf{R}$ to an finite number of values within an interval, the optimal strategy $\Pi^*_{\mathbf{LQ}}$ calculated under this assumption will only be an approximation of the true optimal strategy Π_{LQ}, similarly, $\Pi^*_{\mathbf{LQ}}(r'_0)$ is an approximation for $\Pi_{\mathbf{LQ}}(r'_0)$.

2.3.3 Solution to the original mean-variance problem

Although we have solved $\mathbf{LQ}(r_0)$ for all $r_0 \in \mathbf{R}$, essentially, for $\mathbf{MV}(\kappa)$, what we really care about is only $\mathbf{LQ}(r^*_0(\kappa))$. Therefore, the value functions $V_i(x, r)$ or decision functions $y^*_i(x, r)$ at extreme large or small r_0 values are not necessary. For a given κ, we are only interested in the auxiliary problems $\mathbf{LQ}(r_0)$'s for $r_0 \in [Z_0(\kappa), Z_K(\kappa)]$, which hopefully contains $r^*_0(\kappa)$. Sometimes, without confusion, we may use $[Z_0, Z_K]$ to simplify notation. Furthermore, we require the same interval be applied for $r_i, i = 1, 2, ..., N-1$. In other words, we will calculate Bellman backward induction (49) within the restricted state space $r \in [Z_0, Z_K]$ for $i = 0, 2, ..., N-2$. The cutoff of r_i state space from $\mathbf{R}$ to an interval $[Z_0, Z_K]$ will introduce computational error, especially when r_i is near the boundary points Z_0 and Z_K. So the interval length should also be wide enough to include most possible realizations of $r_i (i = 1, 2, ..., N-1)$ under the unknown optimal strategy $\pi^* \in \Pi^*$ to minimize the cut off error. Based on (40), this process can be separated into approximation for both r^*_0 and an interval for possible I_i values. A simple approach is to estimate these values through simulation of another strategy that approximates π^*. Our experience shows that deterministic strategy $\hat{\pi}$ serves well for this purpose. We test deterministic strategy $\hat{\pi}$ over 10000 sample price paths, and estimate the sample mean of implementation shortfall to be $\hat{I}_N$, and sample minimum and maximum to be $\hat{I}_{min}$ and $\hat{I}_{max}$. Since $\hat{I}_{min}$ and $\hat{I}_{max}$ are based on a finite number of sample paths, we then take an extended interval

$[I_{min}, I_{max}$ with $I_{min} = 1.1\hat{I}_{min}, I_{max} = 1.1\hat{I}_{max}$ as a conservative estimation for all possible values of I_i for $\forall i = 0, 1, ..., N$. Based on (26), we approximate r_0^* through $\hat{r}_0 = \frac{1}{\kappa} - 2\hat{I}_N$. Combine these two parts,

$$\begin{aligned} Z_0(\kappa) &= \hat{r}_0 + I_{min} = \frac{1}{\kappa} - 2\hat{I}_N + I_{min} \\ Z_K(\kappa) &= \hat{r}_0 + I_{max} = \frac{1}{\kappa} - 2\hat{I}_N + I_{max} \end{aligned} \quad (52)$$

So far the auxiliary dynamic programming problem $\mathbf{LQ}(r_0)$ is discrete in time and mixed in state space (discrete in remaining percentage of shares to be traded x_i, but continuous in implied ratio r_i (or realized implementation shortfall)). In order to derive an algorithm that is ready to implement, we also discretize the state space of the implied ratio. This approximation is based on an assumption that if two trades have the same remaining percentages of shares x_i, and similar realized partial slippage I_i, then they should trade similar percentages of shares y_i for the upcoming period. With this assumption, discretization of the state space is legitimate. Note that once the trading decision of y_{i-1} is made at time t_{i-1} and executed during $[t_{i-1}, t_i)$, the state variable r_i will not be updated until t_i. The value of r_i will incorporate the latest price change $S_{t_i} - S_{t_{i-1}}$. Hence it seems that the discretization of r_i's restrict our adaptability of price process $S_{t_i} - S_{t_{i-1}}$ to only finite possibilities. However, since the decision space (i.e. $0 \leq y_i \leq x_i$) is also discretized, when one state variable x_i is fixed, the assumption of close values in the other state r_i generating the same decision y_i makes the restricted price adaptability less significant. We have observed from numerical experiments that for a fixed resolution of x_i's, the performance improvement will flatten out once the resolution of r_i's increase to a certain level. In other words, for a r_i resolution that is dense enough, the optimal decision $y_i^*(x_i, r_i)$ should be the same for the case when r_i is allowed to be continuous within $[Z_0, Z_K]$. Therefore, for high enough r_i resolution, we believe our algorithm can fully achieve price adaptation.

More specifically for $i = 0, 1, ..., N-1$, the variable $x_i \in \{0, \frac{d}{X}, \frac{2d}{X}, ..., 1\}$, where integer d is the minimal shares to trade. Let $J = \lfloor X/d \rfloor$ be the integer part of X/d. In the following we will assume X/d to be an integer. For the state space of r, we discretize $\mathbf{R}$ into $K+2$ equal width subintervals: $(-\infty, Z_0], (Z_0, Z_1], ..., (Z_{K-1}, Z_K], (Z_K, \infty)$, where $[Z_0, Z_K]$ serves as the boundary of r_i values. For each period $i = 0, 1, ..., N-1$, value functions $V_i(\frac{jd}{N}, Z_k)$ are calculated at each grid point $(\frac{jd}{N}, Z_k)$ for $j = 0, 1, ..., J$ and $k = 0, 1, ..., K$. Furthermore, given r_0 fixed, $\{Z_k, k = 0, 1, ..., K\}$ can depend on i as $\{Z_k^i, k = 0, 1, ..., K_i\}$. For example, we can assign a more delicate resolution with smaller intervals near those values where r_i are more likely to take. We believe a good choice of discretization over r_i will improve computation speed and approximation accuracy. However, in this chapter, we use an equally split up discretized state space $\{Z_k = Z_0 + k \cdot \frac{Z_K - Z_0}{K}, k = 0, 1, ..., K\}$ for $\forall r_i$ as $i = 0, 1, ..., N-1$. As mentioned before, due to the cut off and discretizatio of the state space of r_i, $\Pi^*_{\mathbf{LQ}}$ and $\Pi^*_{\mathbf{LQ}}(r'_0)$ defined in (49) and (52) are only approximations for optimal strategy set$\Pi_{\mathbf{LQ}}$ and $\Pi_{\mathbf{LQ}}(r'_0)$.

With the above discretization, the storage space for one strategy

$$\pi^* = \{y^*_0, y^*_1, ..., y^*_{N-2}\} \in \Pi^*_{\mathbf{LQ}} \text{ is roughly } (N-1)(J+1)(K+1).$$

Of course, we also need to store the value functions. But the storage of their full history is not necessary. Instead, only the next period's value functions are needed to conduct Bellman backward induction (49). The computation time, in worst case, is in the order of $O((N-1)(J+1)^2(K+1))$.

Assume our eventual goal is to solve $\mathbf{MV}(\kappa)$. For any $\pi \in \Pi_{\mathbf{LQ}}(r_0)$, we can simulate it over sample price paths, and estimate the mean and variance of implementation shortfall : $\mathbb{E}[I_N|_\pi]$ and $\mathrm{Var}[I_N|_\pi]$. The representing strategy $\pi^*(r_0)$ for $\Pi^*_{\mathbf{LQ}}(r_0)$ can then be determined by choosing the strategy that produces the minimal mean variance:

$$\pi^*(r_0) = \arg \min_{\pi \in \Pi^*_{\mathbf{LQ}}(r_0)} \{\mathbb{E}[I_N|_\pi] + \kappa \mathrm{Var}[I_N|_\pi]\}.$$

Compared with (33), $\pi^*(r_0)$ is an approximation for $\pi(r_0)$. As before, we use a two step decomposition approach to solve $\mathbf{MV}(\kappa)$. First, plot the mean-variance pair for all representing strategies, which is defined as Pseudo Efficient Frontier($PEF[Z_0, Z_K]$):

$$PEF[Z_0, Z_K] = \{(m(r_0), v(r_0)) \quad | \quad r_0 \in [Z_0, Z_K]\} \tag{53}$$

where

$$m(r_0) = \mathbb{E}[I_N|_{\pi^*(r_0)}] \quad \text{and} \quad v(r_0) = \mathrm{Var}[I_N|_{\pi^*(r_0)}].$$

Secondly, use $PEF[Z_0, Z_K]$ to search $r_0^*(\kappa)$. Assume $r_0^*(\kappa) \in [Z_0, Z_K]$, then according to (34), $r_0^*(\kappa)$ can be approximated through

$$r_0^*(\kappa) \approx \arg \min_{r_0 \in [Z_0, Z_K]} \{m(r_0) + \kappa v(r_0)\}. \tag{54}$$

and its representing strategy $\pi^*(r_0^*(\kappa))$ solves $\mathbf{MV}(\kappa)$. Numerically, $m(r_0) + \kappa v(r_0)$ is a convex function with respect to $r_0 \in [Z_0, Z_K]$, so the minimum is easy to search. The geometric meaning of $r_0^*(\kappa)$ is then the index on the $PEF[Z_0, Z_K]$ whose projection along the direction of κ is the smallest for all points over the $PEF[Z_0, Z_K]$.

Note that Pseudo Efficient Frontier is not the Efficient Frontier EF in (18), however, disregard the difference in the continuous state space and discretized state space for r_i, a special case of $PEF[-\infty, +\infty]$ can be used to construct EF:

$$\begin{aligned} EF &= \bigcup_{\kappa \geq 0} \{(m, v) \quad | \quad (m, v) \in PEF[-\infty, +\infty] \quad \text{and} \\ m + \kappa v &= \min_{(m', v') \in PEF[-\infty, +\infty]} \{m' + \kappa v'\}\} \end{aligned} \tag{55}$$

However, as we will demonstrate later through numerical results, we believe $PEF[-\infty, +\infty]$ should be a smooth convex curve such that every point on it is non-inferior(as defined in [38]) and corresponds to a solution for the minimization problem $\min_{(m', v') \in PEF[-\infty, +\infty]} \{m' +$

$\kappa v'\}$. In other words, there are no extra points of $PEF[-\infty, +\infty]$ that lies inside of the EF:

$$\begin{aligned} PEF[-\infty, +\infty] &= \bigcup_{\kappa \geq 0} \{(m, v) \quad | \quad (m, v) \in PEF[-\infty, +\infty] \\ &\quad \text{and} \quad m + \kappa v = \min_{(m', v') \in PEF[-\infty, +\infty]} \{m' + \kappa v'\}\} \\ &= EF. \end{aligned}$$

Furthermore, $[Z_0, Z_K]$ defined in (52) is normally wide enough to cover most interesting part of $PEF[-\infty, +\infty]$. Combined with these two parts, the interior part of $PEF[Z_0, Z_K]$(i.e. when r_0 is away from the boundary Z_0 or Z_K) should coincide with the part of EF very well, which is indeed confirmed by our numerical results. This prompts us to name the curve defined in (53) as Pseudo Efficient Frontier, and we believe it can be used as a good approximation for EF.

The adaptive strategy exhibits a patten of "aggressive-in-the money(AIM)" in the sense of [28]. Our numerical result has shown that for $\forall i = 1, 2, ..., N-1$ and $\forall x \in \{0, \frac{d}{X}, \frac{2d}{X}, ..., 1\}$, $y_i^*(x, r_i)$ is a decreasing function of r_i. Given $r_0 = r_0^*(\kappa)$ chosen, r_i is an indication of the level of realized implementation shortfall I_i. Therefore, whenever the past price movement is in favor of the trader's advantage(for a buying problem, smaller I_i implies downward price movement), the adaptive trader tends to trade more aggressively to reduce timing risk by paying more market impact costs. Similarly, when price is moving up and incur a large r_i, adaptive trader tends to be more passive and trade slower, which is reflected in a smaller value of $y_i^*(x, r_i)$. The AIM patten comes from our simple assumption of fundamental stock price following an arithmetic Brownian motion with zero drift. Naturally, when the price process exhibits mean-reverting patten, the adaptive strategy $\pi^*(r_0^*(\kappa))$ tends to perform better than the deterministic strategy $\hat{\pi}(\kappa)$.

2.3.4 Flexibility in adjusting trading urgency

In practice, a precise description of risk aversion factor κ is not an easy task. Normally what traders have is a range of values for potential κ's. $[Z_0(\kappa), Z_K(\kappa)]$ defined in (52) is a robust estimation for the interval that covers $r_0^*(\kappa)$ and most r_i realizations. For example, in Section 4.4, we fix κ and shrink $[Z_0, Z_K]$ into half or shift it from original setting in (52) and we still observe similar PEF compared with original setting (52). More importantly, the resulting adaptive strategy set $\Pi^*_{\mathbf{LQ}}$ based on $[Z_0(\kappa), Z_K(\kappa)]$ also works well with problem $\mathbf{MV}(\kappa')$ for κ' close to κ.

Note that κ has been used twice in both decomposition steps to derive optimal strategy: first, use (52) to construct $PEF[Z_0(\kappa), Z_K(\kappa)]$ which consists the mean-variance pair of all representing strategies $\pi(r_0)$ for $r_0 \in [Z_0, Z_K]$; second, approximate $r_0^*(\kappa)$ through (54). These two steps can be carried out using different κ. Therefore, the robustness of Z_0, Z_K as mentioned above naturally implies that an adaptive strategy based on a specific risk aversion factor κ can also be used to solve $\mathbf{MV}(\kappa')$ for other risk aversion factors κ' different but close to κ. Once $PEF[Z_0(\kappa), Z_K(\kappa)]$ is constructed, approximate $r_0^*(\kappa')$ through

$$r_0^*(\kappa') \approx \arg \min_{r_0 \in [Z_0(\kappa), Z_K(\kappa)]} \{m(r_0) + \kappa' v(r_0)\}. \tag{56}$$

then the representing strategy $\pi^*(r_0^*(\kappa'))$ solves $\mathbf{MV}(\kappa')$. Similarly, $PEF[Z_0(\kappa), Z_K(\kappa)]$ can also be applied to generate optimal strategies for $\mathbf{E}(v')$ and $\mathbf{V}(m')$:

$$r_0^*(v') \approx \arg \min_{r_0 \in [Z_0, Z_K]} \{m(r_0) \mid v(r_0) \le v'\}, \tag{57}$$

$$r_0^*(m') \approx \arg \min_{r_0 \in [Z_0, Z_K]} \{v(r_0) \mid m(r_0) \le m'\}. \tag{58}$$

In order to achieve a good approximation, v' should be close enough to $\mathrm{Var}[I_N|_{\pi^*(r_0^*(\kappa))}]$ and m' close enough to $\mathbb{E}[I_N|_{\pi^*(r_0^*(\kappa))}]$. Examples of such values can be $v = \mathrm{Var}[\hat{I}_N|_{\hat{\pi}^*(\kappa)}]$ and $m = \mathbb{E}[\hat{I}_N|_{\hat{\pi}^*(\kappa)}]$, where $\hat{\pi}^*(\kappa)$ is the optimal deterministic strategy for $\mathbf{D}(\kappa)$.

Therefore, by calculating $PEF[Z_0, Z_K]$ once, we can use it to solve mean variance objective for different risk aversion levels. More importantly, $\Pi^*_{\mathbf{LQ}}$ allows the trader to diverge from the initial trading path and adjusting his risk aversion level during the trading horizon. Since risk aversion factor dominates our trading urgency, by readjusting its level, we can also incorporate trader's view of near term price direction.

For example, suppose the security price goes under in early time periods, and the buying execution incurs a small (and hence beneficial) implementation shortfall before the mid trading period t_i. And a sudden news event breaks out and the trader has a strong belief the stock price will go down even further by the end of trading horizon. Therefore, rather than following the strict optimal policy of AIM by buying a large amount of $y_i^*(x_i, r_i)$ percentage of shares, the trader hopes to wait and trade more slowly. In other words, the trader reassigns a smaller risk aversion factor $\tilde{\kappa}_i < \kappa$ and be willing to take more timing risk while getting gains for speculative downward price movements. Similar as our original problem, where the value of κ will dictate the appropriate choice of r_0^*, a change of risk aversion level will result in a deviation of r_i from its original value. This will be reflected as the trader opt to a $\tilde{r}_i$ larger than r_i, and shifts from action $y_i^*(x_i, r_i)$ to $y_i^*(x_i, \tilde{r}_i)$ with $\tilde{r}_i > r_i$. The immediate trading action $y_i^*(x_i, \tilde{r}_i)$ will be smaller than $y_i^*(x_i, r_i)$.

Both of the above two cases can be easily adapted in our current algorithm. The following **Property** 2 combined with **Property** 1 shows that we can still resort to the original strategy $\Pi^*_{\mathbf{LQ}}$ even though we have re-assigned the risk aversion factor to κ_i from time t_i on:

Property 2 1. *For a given pair of* $(x, r) \in [0, 1] \times \boldsymbol{R}$*, assume*

$$\begin{aligned}\Pi_i \;=\; & \{\pi_i = (y_i, y_{i+1}, ..., y_{N-1}) | \pi \textit{ satisfies } x_i = x, y_j \geq 0 \textit{ for } j = i, i+1, ..., N-1, \\ & \textit{and } (3), (11)\},\end{aligned}$$

then the value function $V_i(x,r)$ of (49) also satisfies

$$\begin{aligned} V_i(x,r) &= \min_{\Pi_i} \mathbb{E} \sum_{j=i}^{N-1} C_j((x_j,r_j),y_j,B_{t_{j+1}}-B_{t_j}) \\ &= \min_{\Pi_i} \mathbb{E}(r\bar{I}_{N-i}+\bar{I}^2_{N-i}) \end{aligned} \tag{59}$$

Furthermore, the optimal strategy of (59) $\pi_i^*=(y_i^*,y_{i+1}^*,...,y_{N-1}^*)$ is embedded in the optimal solution of (49) $\pi^*=\{y_0^*,...,y_i^*,y_{i+1}^*,...,y_{N-1}^*\}\in\Pi^*_{LQ}$ as in (50).

For implementation, if at the ith period, risk aversion factor needs to be reassigned to $\tilde{\kappa}_i$. We can simulate all embedded strategy $\pi_i^*=(y_i^*,y_{i+1}^*,...,y_{N-1}^*)$ from $\forall\pi^*\in\Pi^*_{LQ}$ for the remaining time periods and choose the one that produce the minimal mean-variance as the representing strategy, which then generates the pseudo efficient frontier $PEF_i[Z_0,Z_K]$ for the remaining periods. $r_i^*(\tilde{\kappa}_i)$ can be searched through

$$r_i^*(\tilde{\kappa}_i)=\arg\min_{(m,v)\in PEF_i[Z_0,Z_K]}\{m+\tilde{\kappa}_i v\}.$$

Once the new state variable is available, the decision will then shift from the predetermined $y_i^*(x_i,r_i)$ to $y_i^*(x_i,r_i^*(\tilde{\kappa}_i))$.In this case, it is the $r_i^*(\kappa_i)$ rather than the past partial implementation shortfall $\frac{r_i-r_0}{2}$ that decides y_i for the upcoming period. Similar as (57) and (58), $PEF_i[Z_0,Z_K]$ can also be applied when it is the target mean or variance for the remaining slippage that needs to be adjusted.

2.4 Numerical results

In this section we compute some numerical examples for the purpose of exploring the qualitative properties of adaptive strategies.

2.4.1 π^* and $\pi^*(\kappa)$

Assume we receive a buying order of $X=1$ million shares of a stock to be traded within $T=1$ day from $9:30$ to $16:00$. We split up the whole day into $N=50$ time periods, each lasting 7.8 minutes. The arrival price is $S_0=\$100$, and the daily volatility $\sigma = 125$bps is constant throughout the day. Assume the average daily volume(ADV) is 10 million, then the order accounts for 10% of ADV. If the market impact factor is constant during the day, then according to (13) the market power $\mu = 0.048$. Through (9) we know that σI_N represents the fraction of slippage over notational value XS_0, which is easier to understand for practitioners. Therefore, in this section, sometimes we present statistics in terms of σI_N instead of I_N, although all the computations are still based on I_N scaled in (9). The risk aversion factor in (14) set as $\kappa = 6.4396$ to make sure that the half-life of deterministic strategy is 30 minutes, i.e. deterministic strategy will trade 50% of volume within half an hour. The state space $\{(X_i, r_i)|X_i \in [0,1], r_i \in [Z_0, Z_K]\}$ is discretized with resolution $J = 250$ and $K = 400$ for $\forall i = 1, 2, ..., N-1$. By running deterministic strategy $\mathbf{D}(6.4396)$ on 10000 sample paths, we observe the sample mean, maximum and minimum of slippage are $\mathbb{E}[\hat{I}_N] = 0.2819$, $\hat{I}_{min} = -0.9272$, $\hat{I}_{max} = 2.0628$ respectively. According to (52), the state space of $r_i(i = 0, 1, ..., N-1)$ can be cut as $[Z_0, Z_K] = [-1.4283, 1.8606]$.

Following steps listed in (47)-(49), we obtain the value function $V_i(x, r)$ and the optimal action $y_i^*(x, r)$ for each discrete state grid point (x, r) and $i = 0, 1, ..., N-1$. Particularly, we compute expectation over normal random variable $B_{t_{i+1}} - B_{t_i}$ in (49) through Gaussian quadrature ([37]). Although Gaussian Hermite quadrature seems more appropriate to compute the expectation for normal random variable as the integrated function contains exponential function, it does not work very well in our case as the discretized value functions V_i are not smooth enough. Instead, we cut the integration domain from $(-\infty, +\infty)$ to an interval within ± 7 standard deviation, and divide it into three sub intervals, and apply degree 4 Gaussian Legendre quadrature separately:

$$
\begin{aligned}
& \mathbb{E}[V_{i+1}(x-y, r+2N\mu_i y^2 + 2(B_{t_{i+1}} - B_{t_i})(x-y))|(x,r)] \\
= & \int_{-\infty}^{+\infty} \sqrt{\frac{N}{2\pi}} e^{-\frac{z^2 N}{2}} V_{i+1}(x-y, r+2N\mu_i y^2 + 2z(x-y)) dz \\
\approx & \int_{-\frac{7}{\sqrt{N}}}^{\frac{7}{\sqrt{N}}} \sqrt{\frac{N}{2\pi}} e^{-\frac{z^2 N}{2}} V_{i+1}(x-y, r+2N\mu_i y^2 + 2z(x-y)) dz \\
= & \int_{-\frac{7}{\sqrt{N}}}^{-\frac{3}{\sqrt{N}}} \cdots + \int_{-\frac{3}{\sqrt{N}}}^{\frac{3}{\sqrt{N}}} \cdots + \int_{\frac{3}{\sqrt{N}}}^{\frac{7}{\sqrt{N}}} \cdots
\end{aligned}
$$

where

$$
\begin{aligned}
\int_a^b \cdots = & \int_a^b \sqrt{\frac{N}{2\pi}} e^{-\frac{z^2 N}{2}} V_{i+1}(x-y, r+2N\mu_i y^2 + 2z(x-y)) dz \\
\approx & \frac{b-a}{2} \sum_{i=1}^{4} w_i \sqrt{\frac{N}{2\pi}} e^{-\frac{(\frac{b-a}{2} z_i + \frac{a+b}{2})^2 N}{2}} V_{i+1}(x-y, \\
& r + 2N\mu_i y^2 + 2(\frac{b-a}{2} z_i + \frac{a+b}{2})(x-y))
\end{aligned}
$$

with w_i and z_i be the ith standard(i.e. integral over $[-1,1]$) weights and abscissas for degree 4 Gaussian Legendre quadrature:

$$
z_1 = 0.3400, z_2 = -z_1, z_3 = 0.8611, z_4 = -z_3;
$$

$$
w_1 = w_2 = 0.6521, w_3 = w_4 = 0.3479.
$$

In total, to compute each execution in (49), we need to visit the value functions at only 12 states of the next time period, which is quite effective computationally.

The computation of value function $V_i(x,r)$ or decision function $y_i^*(x,r)$ requires us to solve a minimization problem within (49) which may not be a convex function. There may be multiple optimizers y_i^*'s that minimize (49). This is the reason the optimal solution of auxiliary problem $\Pi^*_{\mathbf{LQ}}$ in (50) is a set of all possible optimal strategies. However, we find out that numerically we seldom have more than one minimizer y_i^* for (49), and even as it happens, we will opt to use the minimizer y_i^* that has the largest value. With this arrangement, for each state pair (x_i, r_i), $y_i^*(x_i, r_i)$ is a scalar, and $\Pi^*_{\mathbf{LQ}}$ will only have one element $\Pi^*_{\mathbf{LQ}} = \pi^*$. Furthermore, for the minimization problem within the Bellman equation (49), rather than searching y_i within $[0, x_i]$ exclusively, we resort to deterministic decision $\hat{y}_i$ to achieve a smaller search range such as $[0, min(x_i, 4\hat{y}_i)]$, which often result in a significant reduction in computation time. This works particularly well when the market power μ_i or the risk aversion factors κ are small, when adaptive decision $y_i^*(x_i, r_i)$ tends to lie in a close neighborhood of deterministic decision $\hat{y}_i$. Therefore, by searching around deterministic strategy $\hat{y}_i$ we can significantly decrease computation time.

One sample value function for $i = 1$: $V_1(x_1, r_1)$ is presented in Figure 1. Figure 2 shows the corresponding decision function $y_1^*(x_1, r_1)$. $y_i^*(x_1, r_1)$ is a non-increasing function with respect to r_1 for fixed x_1 values. This means an advantageous earlier price movement(i.e. $B_{t_1} - B_{t_0} < 0$ for a buying problem, which leads a smaller value of r_1) will allow the trader to be more aggressive in the next periods trading(i.e. y_1^* to be larger), which corresponds to the aggressive-in-the-money property of mean-variance objective.

We no longer need to pick a representing strategy as $\Pi^*_{\mathbf{LQ}}(r_0)$ has only one element $\Pi^*_{\mathbf{LQ}}(r_0) = \pi^*(r_0)$. Figure 3 shows the adaptive strategy $\pi^*(-0.990)$ for two sample price paths. Depending on different price evolution, adaptive strategy may finish trading in just 9 periods for down sample, or take all 50 periods for up sample.

The $PEF[Z_0, Z_K]$ can be simply constructed through simulation of the single optimal strategy $\pi^*(r_0)$ for different initial state $(1, r_0)$ for $r_0 \in [Z_0, Z_K]$. Although we restrict Π^* into one single element, numerical results have proven it is a highly robust strategy.

By simulating $\pi^*(r_0)$ for $r_0 \in [Z_0, Z_K]$ over 10000 common price paths, we can construct $PEF[Z_0, Z_K]$ consisting of mean-variance pair. Figure 4 plots the $PEF[Z_0, Z_K]$ with $Z_0 = -1.4283$ and $Z_K = 1.8606$, and presents how it can be used to determine appropriate r_0^* to solve either $\mathbf{MV}(\kappa)$ and $\mathbf{E}(v)$. The performance of the resulting strategy is further illustrated in Table 1 for four different problems: $\mathbf{MV}(6.4396)$, $\mathbf{V}(0.0353)$, $\mathbf{D}(6.4396)$ and $\mathbf{MV}(0)$. Note the solution of $\mathbf{MV}(0)$ is just linear(or VWAP) strategy. The value $v = 0.0353$ is chosen to match the sample variance of deterministic strategy for $\mathbf{D}(6.4396)$: $\text{Var}[\hat{I}_N] = 0.0353$. The initial trading percentage $y_0 = 8.40\%$ of $\mathbf{V}(0.0353)$ is much smaller than deterministic counterpart $\hat{y}_0 = 20.63\%$. By trading less in the beginning, adaptive strategy accumulates necessary leeway for later adjustment.

Although the adaptive strategy is based on a price model with zero drift, in reality, small short term momentum is commonly observed among equity prices. When the drift is small, adaptive strategy may still perform better than the deterministic strategy. Figure 5 plots the mean-variance difference of these two strategies tested on price modeled as arithmetic Brownian motion with daily drift θ:

$$S_t = S_0 + \sigma S_0(\theta t + B_t),$$

where $\theta = 1$ corresponds to mean price increasing 100% during $T = 1$ day. In our current setting, as long as the daily drift does not exceed 37.10%, adaptive strategy always outperform deterministic strategy. Furthermore, as shown in Figure 5, adaptive strategy for a buying order performs much better in the bear market than the bull market. This is because adaptive strategy tends to trade less at the earlier periods to remain adaptation flexibility for later trading. When the downward trend is continued, buying more in the latter periods results in smaller implementation shortfall than deterministic strategy. Therefore, the drift boundary for misspecification leans towards the negative side for our buying order, and positive side for the selling order.

2.4.2 Risk aversion factor κ and market power μ

By varying risk aversion factor κ for $\mathbf{MV}(\kappa)$ with fixed market power $\mu = 0.096$, we can plot the efficient frontier(EF) of adaptive strategy consisting of sample mean and variance of implementation shortfall $I_N|_{\pi^*(r_0^*(\kappa))}$ as a function of κ. Note that once κ has changed, so is $[Z_0(\kappa), Z_K(\kappa)]$, $\pi^*(\kappa)$ and $PEF[Z_0(\kappa), Z_K(\kappa)]$. In other words, at this moment, each pseudo efficient frontier $PEF[Z_0(\kappa), Z_K(\kappa)]$ is used to only solve one single problem $\mathbf{MV}(\kappa)$. Figure 6 compares the efficient frontier of adaptive and deterministic strategy for $\kappa \in [0.0336, 120.0750]$. For risk neutral trader with $\kappa = 0$, adaptive, deterministic and VWAP strategy all coincide. As the trader gets more and more risk adverse, the advantage of adaptive strategy gets more prominent.

On the other hand, when κ is fixed, different levels of market power μ reflect various liquidity and volatility conditions based on (13). When market power is small, adaptive strategy performs similarly as determinist strategy. In other words, when the underlying security is very liquid or volatile, the adaptive strategy tends to be price insensitive. [31] has proved that under certain conditions, the adaptive strategy and deterministic strategy coincides for $\mu = 0$.

To give the reader some general idea, we compared adaptive and deterministic strategy under different combinations of κ and μ in Table 2. Note that adaptive strategy is based upon discretized action space $\{0, \frac{d}{X}, \frac{2d}{X}, ..., 1\}$ while deterministic strategy is based upon continuous space $[0, 1]$, it is possible deterministic strategy may outperform adaptive one when market power μ is small and resolution of x_i K is not large enough, which explains the performance for the senario when deterministic half life =1.5hrs and market power corresponds to 5% of ADV. In addition, for each senario, we also list the drift boundary within which adaptive strategy will outperform deterministic strategy in terms of mean-variance sum. We can see that adaptive strategy achieves the most advantage when the asset is illiquid, volatile and the trader is risk averse.

2.4.3 Full price adaptability

As mentioned before, the main advantage of our model over the adaptive strategy in recent [31] is our full price adaptability. While the state variables in both our model and [31]'s are of two dimension and both take remaining unexecuted percentage x_i as one state variable, the difference lies in the second variable. We use state r_i to record realized partial implementation shortfall $I_i = \frac{r_i - r_0}{2}$, while [31]'s state c_i is the limit of expected unrealized implementation shortfall for the remaining periods, i.e. $c_i \geq \mathbb{E}[I_{N-i}]$. In terms of decision variable, both use percentage of shares y_i to be traded in the next period, but [31] requires an extra integrable function $z_i(\xi) \in L^1(\Omega, \mathbf{R})$ of the price change $\xi = B_{t_{i+1}} - B_{t_i} \in \Omega$. By the end of period i of t_i, the price change $B_{t_{i+1}} - B_{t_i}$ will be observed, and the value of function z_i will be set as the new state variable $c_{i+1} = z_i(B_{t_{i+1}} - B_{t_i})$, which dictates decision y_{i+1} for the next time period. To simplify computation, [31] restricts the decision z_i to the space of step functions rather than measurable functions, such as $z_i(\xi) = z_i^+$ for $\xi \geq 0$ and $z_i(\xi) = z_i^-$ for $\xi < 0$. As mentioned in [31], "the adaptability of our trading strategy is restricted to reacting to the observation whether the stock price goes up or down during each of N trading periods." Since z_i determines next period's state variable, a two level step function restricts optimal strategy's decision paths $\{y_i\}_{i=0}^{i=N-1}$ to at most 2^{N-1} possibilities regardless of the resolution of x_i. On the other hand, our model naturally incorporates $\xi = B_{t_{i+1}} - B_{t_i} \in \Omega$ into the next period's state variable r_{i+1}, which takes $K+1$ possible values. Therefore, with comparable resolution of x_i, the total possibilities of our trading paths $\{y_i\}_{i=0}^{i=N-1}$ is in the order of $O((K+1)^{N-1})$. Table 3 compares the numerical performance between our adaptive strategy and the one proposed in [31]. We set the same state vector resolution as in [31], i.e. $J = 250, K = 100$ to make sure the comparison based soely upon two strategies's different price adaptability. We can see that our strategy performs at least as good as [31], and in some cases, we can decrease the average implementation shortfall more than 40% compared with their performance.

2.4.4 $PEF[Z_0(\kappa), Z_K(\kappa)]$ vs. EF

Since the state space of $r_i(\forall i)$ has been trunked as $[Z_0, Z_K]$, a natural question is whether the algorithm is robust with regard to different choices of $[Z_0, Z_K]$. In Figure 7, we implement two kinds of tests, by either shrinking the interval length or by shifting the interval from original setting in (52). The numerical details are listed in Table 4, we can see that they all produce close performance with the original setting (52), which implies (52) as a robust choice for optimal strategy computation.

Our goal in this section is to illustrate through numerical results the legitimacy of simplification for solving $\mathbf{MV}(\kappa')$ for different risk aversion factors κ' based on a single pseudo efficient frontier $PEF[Z_0(\kappa), Z_K(\kappa)]$. An underlying assumption here is that $PEF[Z_0(\kappa), Z_K(\kappa)]$ can work as a good approximation of part of EF that covers $\{(\mathbf{E}[I_N|_{\pi^*(\kappa')}], \mathrm{Var}[I_N|_{\pi^*(\kappa')}])\}$ for κ' close with κ. Note that besides state discretization, there are mainly three layers of approximation of EF through $PEF[Z_0(\kappa), Z_K(\kappa)]$:

1. using $[Z_0(\kappa), Z_K(\kappa)]$ to approximate $\mathbf{R}$ as the state space for r_i for $i = 1, 2, ..., N-1$, this introduce cut off error;

Table 1: Comparison of different strategies

problem	r_0^*	$\mathbb{E}[I_N]$	$\mathrm{Var}[I_N]$	$\mathbb{E}+12.88\mathrm{Var}$	$\mathbb{E}[\sigma I_N]$	$Std[\sigma I_N]$	y_0
MV(6.4396)	-0.4499	0.2991	0.0155	0.3992	37.39	15.58	14.00%
V(0.0353)	-0.0990	0.2137	0.0353	0.4412	26.72	23.50	11.60%
D(6.4396)		0.2819	0.0353	0.5094	35.23	23.50	20.63%
MV(0)		0.0490	0.3309	2.1797	6.13	71.90	2.00%

2. when more than one minimizer are available within Bellman equation (49), picking the minimizer y_i^* with the largest value, which introduced sampling error of $PEF[Z_0(\kappa), Z_K(\kappa)]$ as it is constructed by a particular strategy within $\Pi^*_{\mathbf{LQ}}(r_0)$ for $r_0 \in [Z_0(\kappa), Z_K(\kappa)]$ rather than the representing strategy $\pi^*(r_0)$ for $r_0 \in [Z_0(\kappa), Z_K(\kappa)]$ as defined in (53).

3. assuming every point on the ultimately ideal pseudo efficient frontier $PEF[-\infty, +\infty]$ is non-inferior, i.e. $PEF[-\infty, +\infty] = EF$.

Even with all those approximation error, the two curves still coincide with each other closely as presented in Figure 8. The robustness of $[Z_0(\kappa), Z_K(\kappa)]$ decreases the first error. We believe the second error is small due to the uniqueness of optimizer in (49), and the third error is small due to the non-inferiority of points on $PEF[-\infty, +\infty]$. Therefore, numerically we can just use $PEF[Z_0(\kappa), Z_K(\kappa)]$ to approximately solve $\mathbf{MV}(\kappa')$, $\mathbf{E}(v')$ and $\mathbf{V}(m')$ for different levels of κ', v' and m' that are close to κ, $\mathrm{Var}[I_N|_{\pi^*(r_0^*(\kappa))}]$ or $\mathbf{E}[I_N|_{\pi^*(r_0^*(\kappa))}]$ respectively. r_0^* can be determined through (57)-(56). Figure 9 goes further to give a robust range of v' values such that the solution of $\mathbf{E}(v')$ through $PEF[Z_0(0.3266), Z_K(0.3266)]$ is close to the optimal strategy.

Table 2: Numerical performance for varying risk aversion factor and market power

	% of ADV	5%		10%		20%	
half life	(bps)	$\mathbb{E}[\sigma I_N]$	$Stdev[\sigma I_N]$	$\mathbb{E}[\sigma I_N]$	$Stdev[\sigma I_N]$	$\mathbb{E}[\sigma I_N]$	$Stdev[\sigma I_N]$
0.5hrs	adaptive	15.70	23.50	26.72	23.50	42.59	23.50
	static	17.62	23.50	35.23	23.50	70.47	23.50
	drift bound	$[-100.00\%, 26.86\%]$		$[-100.00\%, 37.10\%]$		$[-100.00\%, 43.46\%]$	
1.0hrs	adaptive	7.54	38.33	14.47	38.357	26.11	38.35
	static	7.58	38.35	15.17	38.35	30.31	38.35
	drift bound	$[-100.00\%, 3.68\%]$		$[-100.00\%, 11.69\%]$		$[-100.00\%, 22.39\%]$	
1.5hrs	adaptive	4.93	48.56	9.73	48.69	18.93	48.56
	static	4.92	48.56	9.84	48.56	19.68	48.56
	drift bound			$[-100.00\%, 3.09\%]$		$[-100.00\%, 11.43\%]$	

Table 3: Comparison with adaptive strategy in [31]

case	1		2		3		4	
κ	1.62		5.2875		30		108.75	
	$\mathbb{E}[I_N]$	$\text{Var}[I_N]$	$\mathbb{E}[I_N]$	$\text{Var}[I_N]$	$\mathbb{E}[I_N]$	$\text{Var}[I_N]$	$\mathbb{E}[I_N]$	$\text{Var}[I_N]$
Linear	0.1531	0.3309	0.1531	0.3309	0.1531	0.3309	0.1531	0.3309
Deterministic	0.2567	0.1439	0.4537	0.0771	1.0716	0.0274	1.9898	0.0109
Adaptive	0.2331	0.1435	0.3271	0.0771	0.4893	0.0274	0.6373	0.0109
	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{lin}]}$	$\frac{\text{Var}[I_N]}{\text{Var}[I_{lin}]}$	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{lin}]}$	$\frac{\text{Var}[I_N]}{\text{Var}[I_{lin}]}$	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{lin}]}$	$\frac{\text{Var}[I_N]}{\text{Var}[I_{lin}]}$	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{lin}]}$	$\frac{\text{Var}[I_N]}{\text{Var}[I_{lin}]}$
Deterministic	1.68	0.43	2.96	0.23	7.00	0.08	13.00	0.03
Adaptive	1.52	0.43	2.14	0.23	3.20	0.08	4.16	0.03
[31]'s Adaptive	1.52		2.27		3.92		7.09	
improvement %	0		5.73%		18.37%		41.33%	
impv of $\mathbb{E}[\sigma I_N]$	0		2.49		13.78		56.06	

Table 4: Different settings for state space r_i for $i = 0, 1, ..., N-1$

	state space setting			$r_0^*(4.29)$	$y_0(r_0^*)$	$\mathbb{E}[I_N]$	$\text{Var}[I_N]$	$\mathbb{E}[\sigma I_N]$
	Z_0	Z_K	$Z_K - Z_0$					
original	$\hat{r}_0 + I_{min}$	$\hat{r}_0 + I_{max}$	3.2890	-0.0990	0.1160	0.2137	0.0353	26.72
	-1.4283	1.8606						
half	$\hat{r}_0 + \frac{1}{2}I_{min}$	$\hat{r}_0 + \frac{1}{2}I_{max}$	1.6445	-0.1044	0.1120	0.2142	0.0354	26.78
	-0.9184	0.7261						
left shift	$\hat{r}_0 + \frac{3}{2}I_{min}$	$\hat{r}_0 + \frac{1}{2}I_{max}$	2.6644	-0.1075	0.1120	0.2145	0.0353	26.81
	-1.9383	0.7261						
right shift	$\hat{r}_0 + \frac{1}{2}I_{min}$	$\hat{r}_0 + \frac{3}{2}I_{max}$	3.9136	-0.0993	0.1160	0.2138	0.0353	26.72
	-0.9184	2.9952						

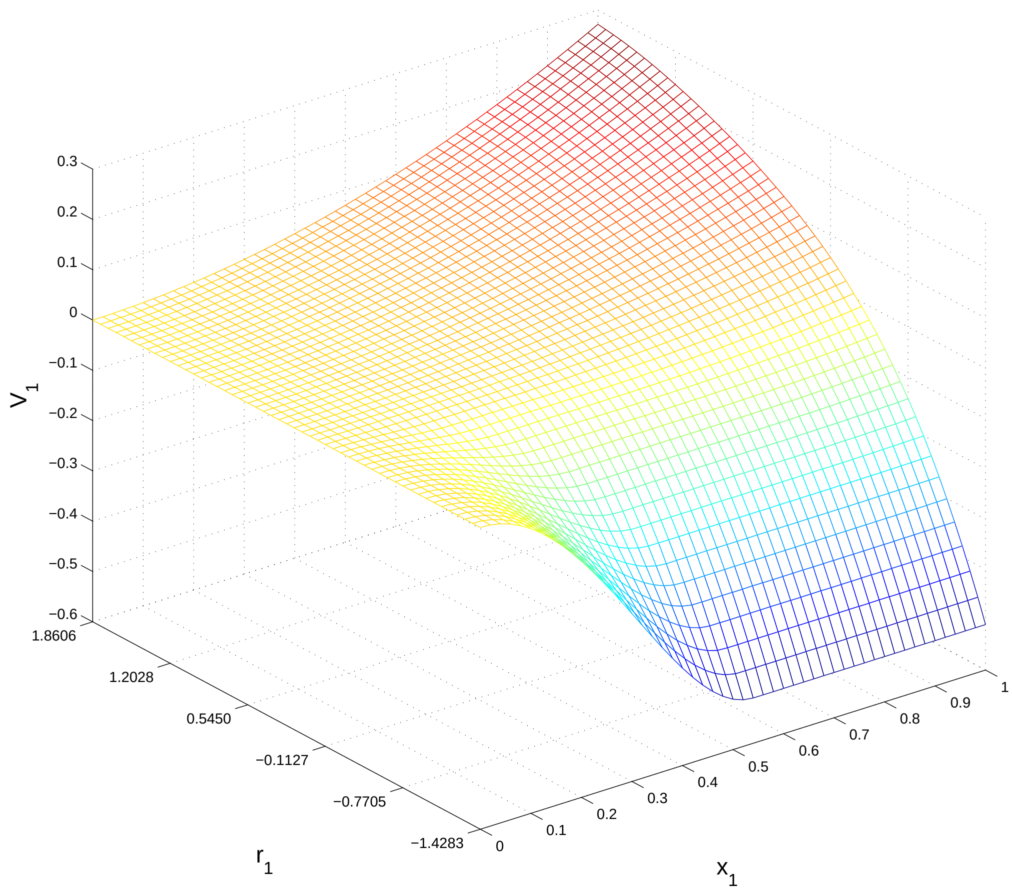

Figure 1: Value function $V_1(x_1, r_1)$ on $[0, 1] \times [-1.4283, 1.8606]$: in general, $V_i(x_i, r_i)$ is a non-decreasing function with respect to r_i for fixed x_i.

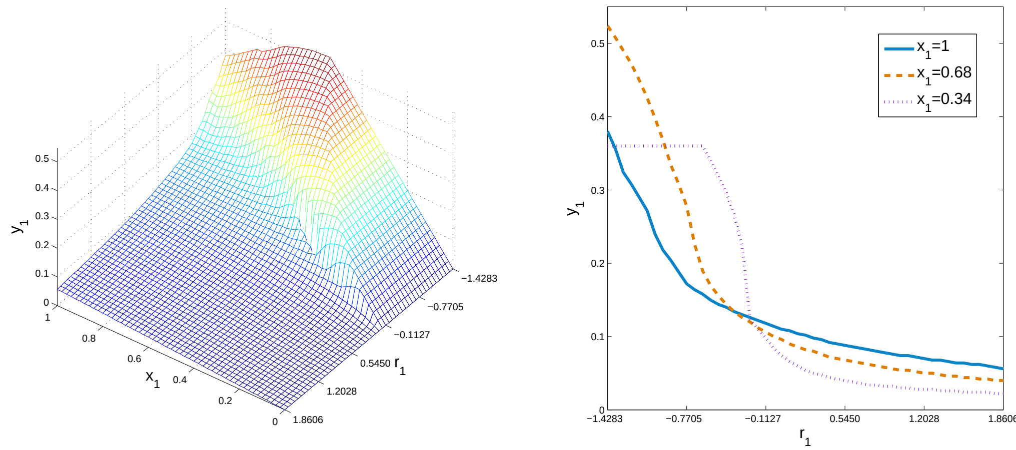

Figure 2: Optimal decision function $y_1(x_1, r_1)$ on $[0, 1] \times [-1.4283, 1.8606]$ with the figure on the right showing three particular choice of x_1: $1, 0.68$ and 0.34. $y_1^*(x_1, r_1)$ is a non-increasing function with respect to r_1 for fixed x_1, which implies the adaptive strategy is aggressive-in-the-money. However, $y_1^*(x_1, r_1)$ may not be a monotone function with respect to x_1, such as when $r_1 = -0.5$.

2.5 Conclusion

This chapter solves the optimal execution problem which requires to balance a trade-off between liquidity risk of fast trading and timing risk of slow trading. The objective is to minimize the weighted sum of the mean and variance of the implementation shortfall. Since the mean-variance problem can not be directly solved through dynamic programming, we circumvent the problem with a two step process: first solve a family of auxiliary linear-quadratic problems with different linear-quadratic weights through dynamic programming, second find the solution for a particular linear quadratic problem that also solves the original mean-variance problem. One of the major contribution of this chapter is to provide a simple framework to combine these two steps in a seamlessly way. Specifically, by incorporating the linear-quadratic weights as a state variable, we can solves all linear-quadratic problems with one single backward induction. This greatly simplifies the modeling and computation complexity. As a result, our algorithm improves the execution performance significantly

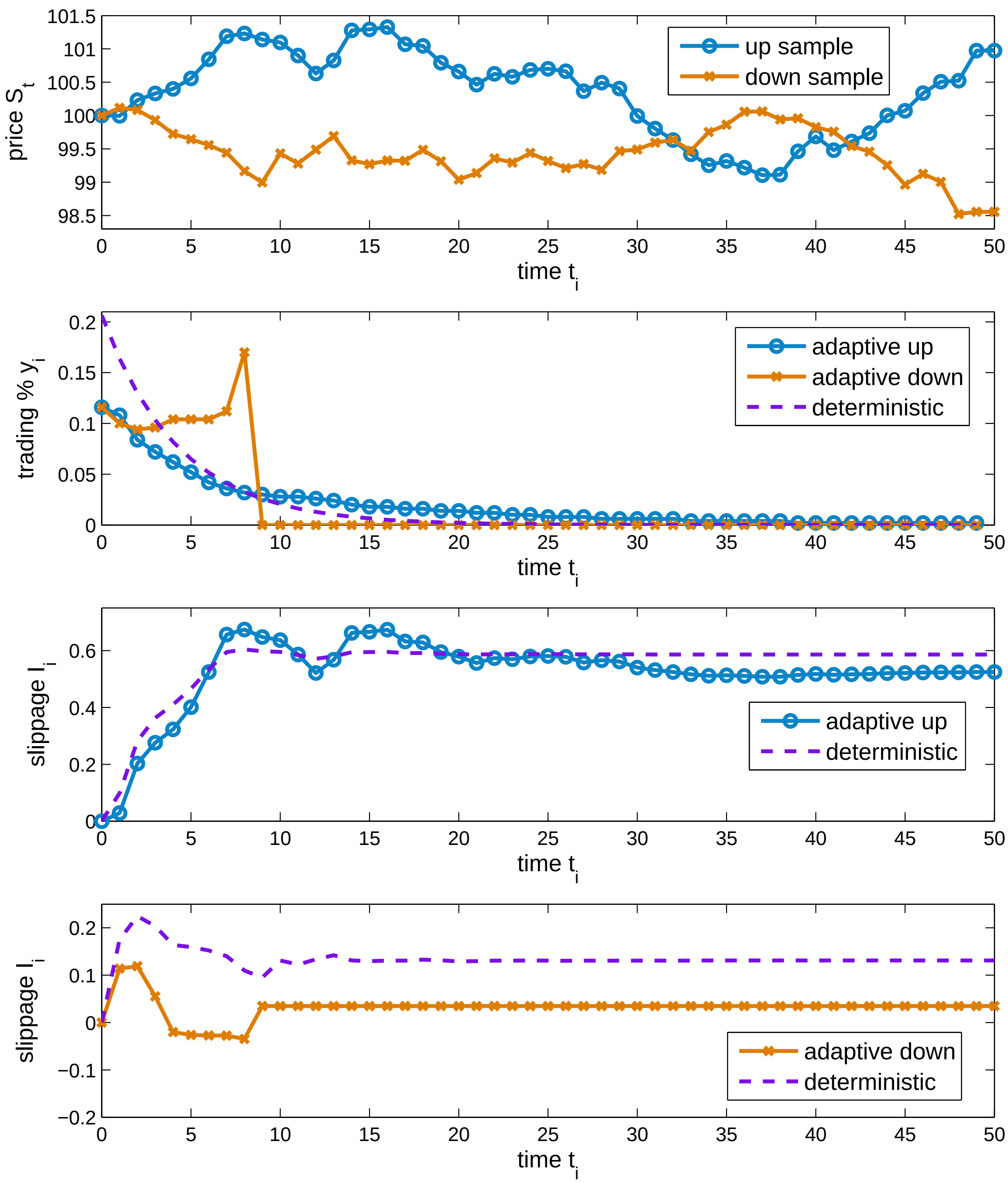

Figure 3: First two figures show two sample price paths S_{t_i} and the trading percentage y_i. At each t_i, the trading decision y_i will not be determined until S_{t_i} has been observed. Unlike deterministic strategy, adaptive strategy buy more shares when the price goes down, and vice versa. The last two figures compare the implementation shortfall for adaptive and deterministic strategy for both sample paths. In both of these particular cases, adaptive strategy perform smaller implementation shortfall than static strategy.

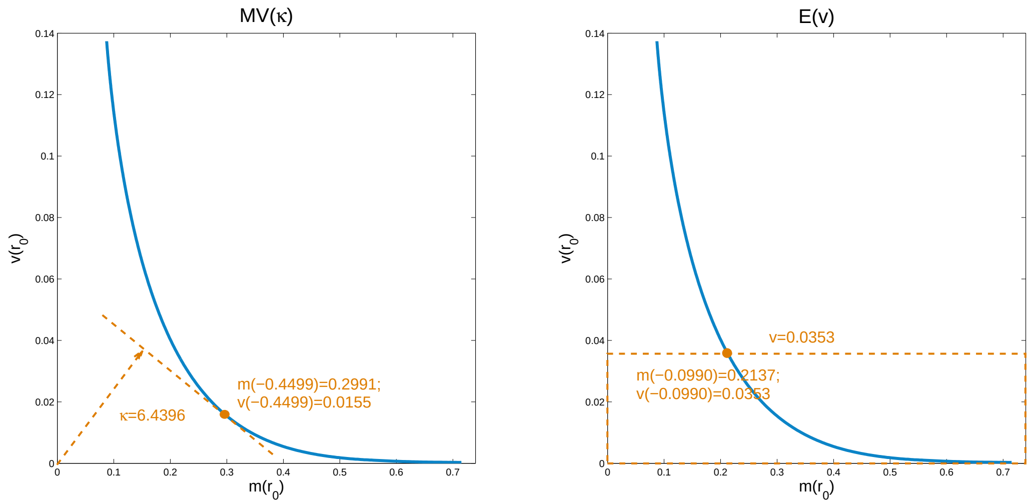

Figure 4: Determining r_0^*: $PEF[-1.4283, 1.8606]$ can be used to determine the optimal r_0^* for different problems: the left figure for $\mathbf{MV}(6.3496)$ and the right for $\mathbf{E}(0.0353)$

.

over the methods in existing literature with even less computation complexity. This chapter also studies the performance of adaptive trading strategies vs. deterministic trading strategies, and the unique Aggressive-In-the-Money behavior appeared in the mean-variance objectives.

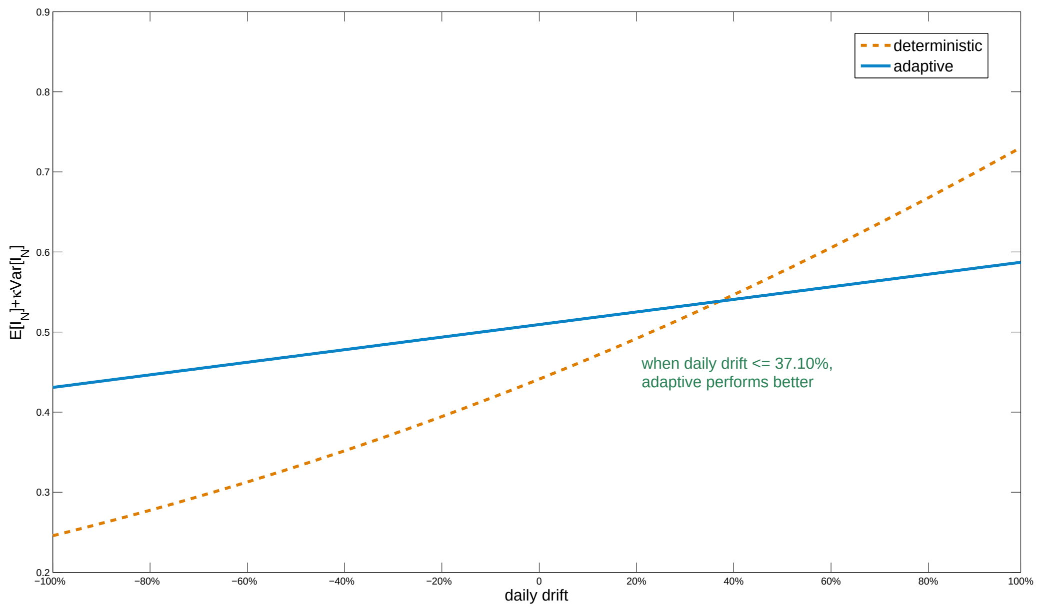

Figure 5: Comparison of mean-variance sum $\mathbf{E}[I_N|_{\pi^*(r_0^*(v))}] + 6.4396\text{Var}[I_N|_{\pi^*(r_0^*(v))}]$ of deterministic and adaptive strategy when the underlying process has non-zero drift: daily drift=100% means the mean price will increase to \$200 by the end of the day when $S_0 = \$100$. Here $v = 0.0353$. Adaptive strategy produced smaller mean-variance as long as the daily drift is below 37.10%. For a buying problem, adaptive strategy tends to performs much better for a downward market.

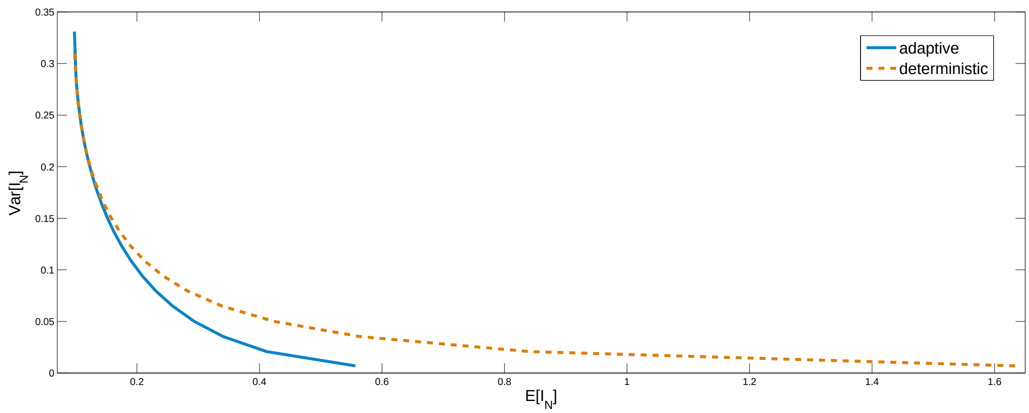

Figure 6: Efficient frontier of adaptive vs. deterministic strategy: 60 different levels of risk aversion level κ are used to compare adaptive and deterministic strategy. These risk aversion levels correspond to 60 deterministic half lives spanning from 0.1525 hours to 3.25 hours, with 0.0525hrs increase. In terms of risk aversion factor, it varies from $\kappa = 0.0336$(half life=3.25 hours) to $\kappa = 120.0750$(half life=0.1525 hours). We can see adaptive strategy works particularly when κ is large, and they coincide with deterministic strategy when κ is small.

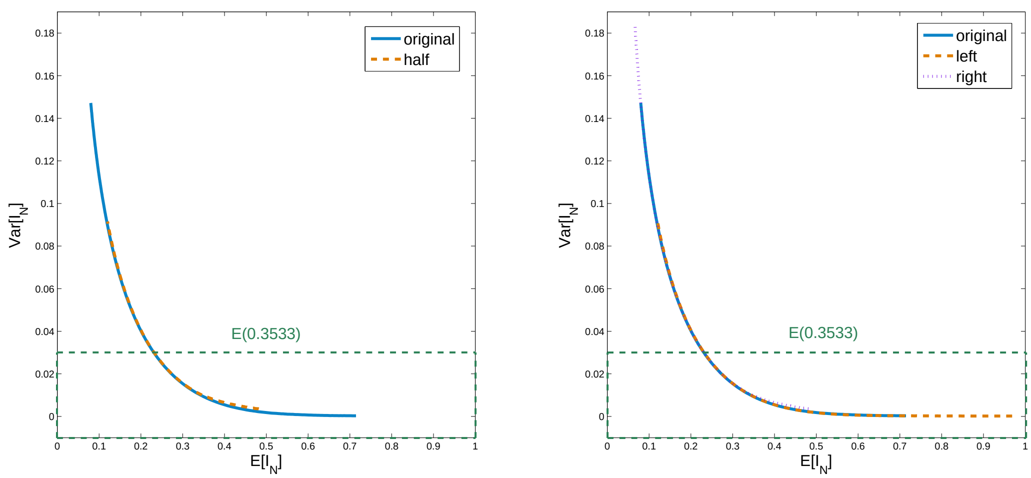

Figure 7: Robustness of $[Z_0, Z_K]$: the pseudo efficient frontier generated through state space $[Z_0, Z_K]$ defined rather than 52 are compared. The left figure shrink $[Z_0, Z_K]$ width as half, and the right figure shifts $[Z_0, Z_K]$ sideways. We can see that the transformed pseudo efficient frontier coincides with the original one, which fluctuations in the value of Z_0 and Z_K do not have a significant impact on the derived strategy for auxiliary problems.

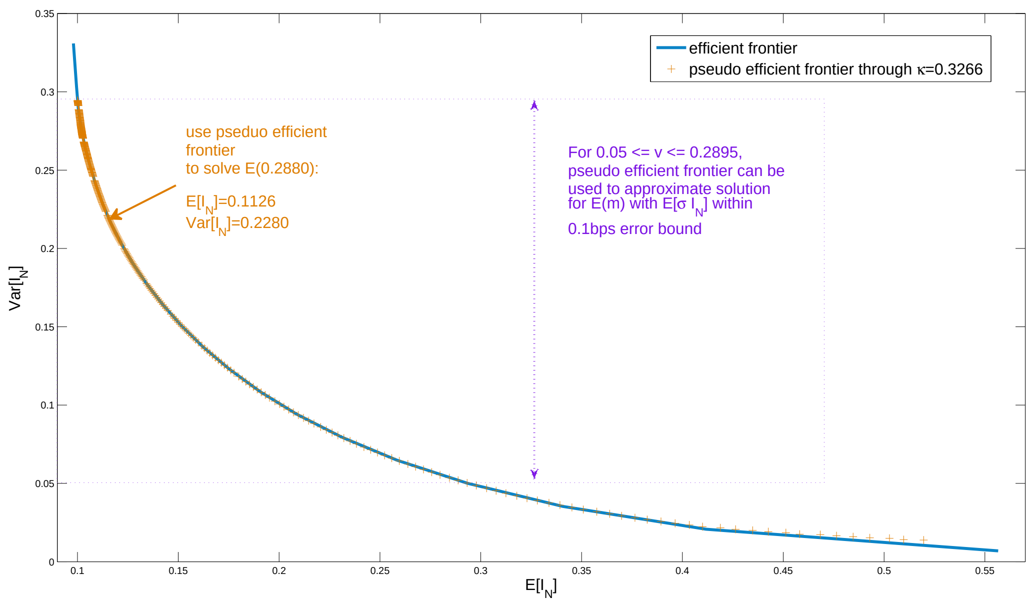

Figure 8: EF vs. $PEF[Z_0(0.3266), Z_K(0.3266)]$: EF is generated in the same way as the adaptive EF in Figure 6, which is based on 60 different risk aversion κ' within $[0.0336, 120.0750]$. A particular such pseudo efficient frontier is plotted for $\kappa = 0.3266$: $PEF[Z_0(0.3266), Z_K(0.3266)]$. We can see that it matches closely to part of EF. Therefore, we can use pseudo efficient frontier to solve $\mathbf{E}(v')$ for different levels of v' that is close to $\text{Var}[I_N|_{\pi^*(r_0^*(0.3266))}]$. Figure 9 will show that when $0.05 \leq v' \leq 0.2895$, the strategy generated from EF and $PEF[Z_0(0.3266), Z_K(0.3266)]$ has very close expected implementation shortfall.

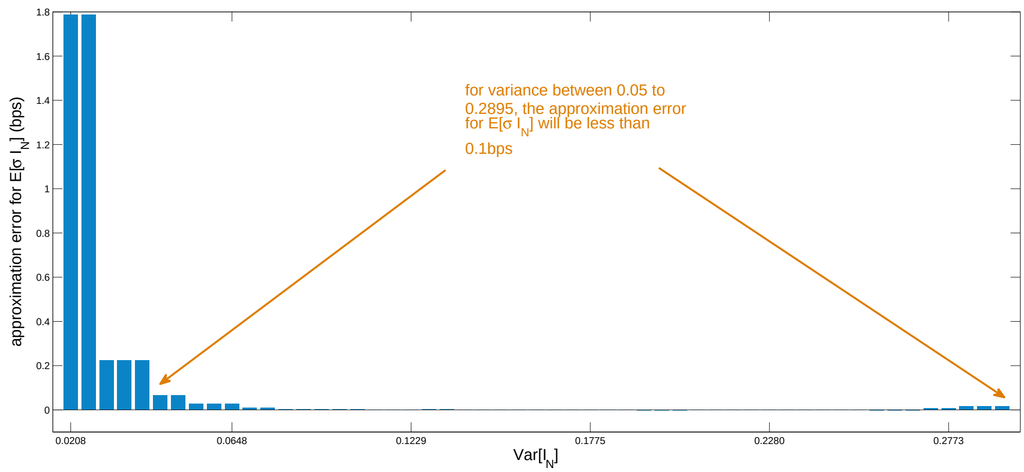

Figure 9: Approximation error of implementation shortfall based on $PEF[Z_0(0.3266), Z_K(0.3266)]$: the bar represents the difference of $\mathbf{E}[\sigma I_N]$ for the solution of $\mathbf{E}(v')$ with $v' \in [0.0208, 0.2895]$ generated by EF and $PEF[Z_0(0.3266), Z_K(0.3266)]$ We can see that when $0.05 \leq v' \leq 0.2895$, these two strategies perform very closely with the difference of $\mathbf{E}[\sigma I_N]$ within 0.1bps.

CHAPTER III

TRADE SCHEDULING WITH VWAP BENCHMARK

3.1 Introduction

Besides the arrival price benchmark used in Chapter 2, another commonly used benchmark is Volume weighted average price (VWAP). The VWAP of a stock over a period of time is the average price paid per share during the given period, typically of full day or part day horizon. VWAP trading is a strategy of ex-ante buying or selling a fixed number of shares at an average price that ex-post tracks the market VWAP benchmark as closely as possible.

According to [9], VWAP execution orders represent about 50% of all institutional investor trading. It is popular for several reasons. First, unlike IS (Implementation Shortfall) strategy where the trader is risk averse, and charges a large risk premium because of the size of the risk, investors can save on the risk premium by using VWAP trading because market directional risk remains the investor's responsibility. Second, the VWAP benchmark encourages traders to spread their trades out over time to avoid the risk of trading at prices that are at the extremes of the day. Therefore, it can reduce the market impact and prevent manipulation of market prices. Third, VWAP orders can be used to evaluate a traders' performance since the effect of market directional movement is excluded in comparisons with market VWAP. Last, VWAP's computational simplicity and transparency is a major advantage, especially in markets where detailed trade level data are difficult to obtain.

A VWAP trader wishes to get the average execution price as close to the market

VWAP as possible to avoid price movement risk. A common approach of achieving a VWAP, as in [29], is to use automated trading strategies to participate proportionately through the trading. This way the trader could disguise his trades by mimicking the actions of the other market participants: he trades more whenever others trade more. Orders can be split up for execution over the day in accordance with the historical volume profile, which tends to have a U-shape. The problem with such a strategy based on time pattern, however, is that the actual realized volume on any given day can depart significantly from the historical average.

As a result, there are a large literature in econometrics on modeling the intra-day trading volume, which is then used to update trading decision adaptively. For example, [11] concentrates on volume dynamics of a group of stocks, decomposing excess volume into common and idiosyncratic components. The authors in [13] use Component Multiplicative Error Models to estimate the intra-day volume. [25] improves on [11]'s framework and incorporates intra-day noise into the volume prediction framework. However, after obtaining the volume prediction for both the next period and the remainder of the day, these papers tend to use a deterministic approach to make trading decisions. For example, the trading size for the next period in [11] is the product of the remaining unexecuted shares and the ratio of next periods' predicted volume over predicted volume for the remainder of the day. Therefore, the "adaptiveness" of these strategies reflects only within the updates of new volume prediction.

Rather than incorporating sophisticated econometric model of intra-day volume and price processes, this chapter takes an alternative approach of deriving an adaptive VWAP trading strategy from a rather simple assumption of volume and price processes. The majority of our numerical results are based on the assumption that volume and price are two independent Markovian processes. Although this assumption may not be realistic in real environment, our numerical examples in both simulated and real environment shows that this assumption not only greatly simplifies the calibration process, it also allows us to build a dynamic programming model that is trackable for meaningful analysis and results. This is particularly important as dynamic programming often involves state space with extra high dimensions. For example, in this chapter, the state space has eight dimensions, which makes direct computation of a value function infeasible.

There are three major contributions in this chapter. First, we derive a theoretical result which shows that under certain market assumptions, the optimal adaptive strategy is independent of the realized execution VWAP and market VWAP up to the current time. As a corollary of this property, we prove that for a mean-variance problem, the optimal solution is independent of risk aversion factor. Second, as often used in Approximate Dynamic Programming, we come up with a reasonable way to approximate the value function such that the optimal adaptive strategy is equivalent to a cumulative curve matching problem, i.e. make sure the sequentially executed order percentages get as close as possible to the expected cumulative market volume percentage. This is one of the approaches often used by practitioners. Therefore, this chapter provides an important link between the theoretical framework and actual industry practice. The adaptive strategy defined this way indeed outperforms the deterministic strategy proposed in [29] in both simulation data and real market data. Lastly, to compensate for potential model misspecification and avoid the adverse impact of large volume spike/lull, we add a two sided bound constraint over the cumulative curve of the deterministic strategy. The goal is to prevent ourselves from over-reaction or under-reaction when the market volume largely deviates from the historical average profile. This simple adaptive strategy with controlled bands protection significantly outperforms the method proposed in some recent articles, such as [11], that only depend on the sophisticated intraday volume prediction. Therefore, our work shows that the benefits of using a simple and tractable dynamic programming approach may outweigh the benefits of using a sophisticated intraday volume prediction model with only a deterministic decision making framework.

The remainder of this chapter is organized as follows. Section 2 begins with a formulation the problem and presents the main theoretical result of this chapter which claims a simple dependency of optimal adaptive strategy on a two dimension variable under certain market assumption. This simple dependency structure suggests a value function approximation that results in an adaptive strategy similar to a cumulative curve matching problem. However, the detailed derivation of both the dependency structure as well as the value function approximation will be left to Section 4. At the end of Section 2, we introduce the controlled adaptive strategy. Section 3 presents the performance comparison of three types of strategies - deterministic, uncontrolled adaptive and controlled adaptive - when applied to both simulation data and real market data. Section 4 fills the analytical details missed in Section 2. It formulates the VWAP trading problem under a strict dynamic programming framework by defining the state variable, cost function, and eventually deriving the Bellman equation. We then proved the simple dependency structure of the adaptive strategy and illustrate how it can be applied to approximate the value function to achieve the curve matching problem introduced in Section 2. Section 5 concludes the chapter.

3.2 Cumulative Curve Matching

This section introduces the adaptive VWAP trading problem, and presents the major result of this chapter: a solution through matching the cumulative volume curve.

3.2.1 Problem Setup

Imagine a scenario where a trader buys a certain amount of shares for a given horizon(such as 1 day). He splits the parent order to participate proportionately in the market's volume, such that his average trading price tracks market VWAP as close as possible. Divide 1 day into n intra-daily bins by $t_0, t_1, ..., t_n$ where $t_0 = 0$ and $t_n = 1$ day. At each time $t_i(i = 1, 2, ..., n)$, we define the incremental market volume(excluding the trader's portion) during $(t_{i-1}, t_i]$ as ΔV_i and the price change as ΔP_i. The cumulative volume by time t_i excluding the trader's execution is V_i:

$$V_i = V_{i-1} + \Delta V_i \quad \text{for } i = 1, 2, ..., n. \tag{60}$$

Similarly, the price at time t_i is

$$P_i = P_{i-1} + \Delta P_i \quad \text{for } i = 1, 2, ..., n. \tag{61}$$

where P_i represents the market volume weighted average price for the period $(t_{i-1}, t_i]$ which will not be known until time t_i. For convenience, we shift and normalized the price process, such that $P_0 = 0$ and P_i represents the price change with respect to first period's VWAP. Let $\mathcal{F}_i$ be the σ-field generated by the exogenous process $\{P_1, V_1, P_2, V_2, ..., P_i, V_i\}$.

At time 0, the parent order size v_{n-1} is given exogenously, which should also be scaled by the total number of outstanding shares[1]. At $t_i(i = 0, 1, ..., n-1)$, after observing the latest market volume V_i and price P_i, we need to determine the child order size

$$\Delta v_i = v_i - v_{i-1} \geq 0 \tag{62}$$

to be placed during the upcoming period $[t_i, t_{i+1})$. Since we do not allow selling execution, v_i is a non-negative non-decreasing series. For notational ease let $v_{-1} = 0$.

[1] Although the parent order size has a index of $n-1$, it is actually known at the beginning of the order, and hence $\mathcal{F}_0$ measurable

If the trader has executed a positive number of shares v_i by time t_{i+1}, his volume weighted average price (VWAP) is defined as

$$q_{i+1} := \sum_{j=0}^{i} \frac{\Delta v_j}{v_i} P_{j+1} \quad \text{for } i = 0, 1, ..., n-1; \tag{63}$$

while the corresponding market VWAP by time t_{i+1} is

$$Q_{i+1} = \sum_{j=0}^{i} \frac{\Delta V_{j+1} + \Delta v_j}{V_{i+1} + v_i} P_{j+1} \quad \text{for } i = 0, 1, ..., n-1. \tag{64}$$

Otherwise, if $v_i = 0$, we define $q_{i+1} = 0$ and if $V_{i+1} + v_i = 0$, we define $Q_{i+1} = 0$.

Note that the definition in (63) assumes the trader has the perfect limit order placement model[2] that completes the child order of Δv_j shares exactly at market VWAP of P_{j+1}. This may not be the case in practice as the actual average transaction price may fluctuate around market VWAP or even worse due to a combination of poor timing and market impact which often moves the price adversely. However, this uncertainty between transaction price and market VWAP diminishes as we increase the number of bins and make more frequent decisions. As the number of bins goes to infinity, the trader's execution price within each period will be the same as market VWAP. For this reason, we ignore the market impact in this chapter, and focus our efforts on order splitting.

Our goal is try to align the order splitting with the market volume such that the order's total VWAP q_n gets as close as possible to the market VWAP Q_n:

$$\min \mathbb{E}\Big[(q_n - Q_n)^2\Big]. \tag{65}$$

At first glance, the symmetric mean square measure (65) ignores the directional difference of $q_n - Q_n$ and penalizes both potential price improvement and execution

[2] Limit order placement model explores market microstructure and decides which order types(market order, limit order, etc) and at which price levels to place orders.

loss. Therefore, some natural alternative objectives can be:

$$\min \mathbb{E}\Big[d_0(q_n - Q_n) + (q_n - Q_n)^2\Big] \quad \text{where } d_0 \geq 0 \tag{66}$$

or

$$\min \Big[\mathbb{E}(q_n - Q_n) + \kappa \text{Var}(q_n - Q_n)\Big] \quad \text{where } \kappa > 0 \tag{67}$$

Note that we assume this is a buy order. With similar argument as in Chapter 2 of this thesis, the solution of the mean-variance problem (67) is also a solution of (66) for the proper value of d_0. Therefore, in the following, we will focus on solving linear-quadratic objective (66).

There are two types of strategies that solves 65: deterministic strategies and adaptive strategies. The child order sizes $\Delta v_i (i = 0, 1, ..., n-1)$ are fixed ex-ante at order arrival for deterministic strategies, while adaptive strategies expand the decision space to allow inter-temporal adjustment during the trading horizon.

3.2.2 Main Theoretical Result

So far we have not made any assumption of the market processes $\{P_i, V_i\}_{i=1}^{i=n}$. Although there are numerous empirical study on the relationship between price and volume process, the main theoretical result of this chapter is based on a very simple assumption of $\{P_i, V_i\}_{i=1}^{i=n}$:

$$\begin{aligned}
&\{\Delta V_i\}_{i=1}^n \text{ are independent non-negative random variables}\\
&\ \text{with } \mathbb{E}[\Delta V_i] = \mu_i \text{ and } \text{Var}[\Delta V_i] = \sigma_i^2;\\
&\{\Delta P_i\}_{i=1}^n \text{ are i.i.d random variables with } \mathbb{E}[\Delta P_i] = 0 \text{ and } \text{Var}[\Delta P_i] = \sigma_P^2.\\
&\text{Furthermore, } \{\Delta V_i\}_{i=1}^n \text{ and } \{\Delta P_i\}_{i=1}^n \text{ are mutually independent.}
\end{aligned} \tag{68}$$

(68) does not take into consideration the potential positive correlation between price volatility and market volume, nor does it incorporate the positive autocorrelation often observed in intraday volume processes. Instead, (68) focuses mainly on the

mean of individual period volume μ_i and the variance σ_i^2. However, this simplification allows us to achieve a simple optimal decision structure, which results in a simple and robust adaptive strategy. Specifically, [29] has proved that the optimal deterministic strategy trades proportionally to a pre-defined intraday volume curve:

$$\Delta v_i = \mathbb{E}\Big[\frac{\Delta V_{i+1}}{V_n}\Big]. \tag{69}$$

This chapter instead focuses the on adaptive strategy. As will be presented in detail in Section 3.4, (68) is a special case of the assumption under Proposition 1, which claims that:

Property 1. *Under the assumption of (68), the optimal adaptive decision Δv_i^* only depends on the executed quantity v_{i-1}, and the cumulative market volume V_i:*

$$\Delta v_i^* = g_i(v_{i-1}, V_i). \tag{70}$$

This may seem to be a trivial result. But in Section 3.4 a strict modeling of the problem (66) using dynamic programming will reveal a seven or eight dimension state space. This means Δv_i^* should depend on not only v_{i-1} and V_i, but also other variables, such as d_0, P_i, q_i and Q_i. The claim in the Property above basically excludes all these other dependencies. For example, $q_i - Q_i$ represents the difference between the executed VWAP and the market VWAP, which turns out to be irrelevant in optimal decision. The Property above also claims that the optimal adaptive strategy is independent of d_0 in (66). As a result, (65)-(67) all share the same solution. This implies that under assumption (68), there is no meaningful method to outperform the market VWAP(i.e. making a directional bet). The only question we can answer is how to execute the parent order such that executed VWAP gets as close as possible to the market VWAP.

3.2.3 Curve Matching

Although the optimal decision only depends on two variables, its functional form depends on the value functions, which are functions over seven dimension state space. The result in Property suggests a natural value function approximation that results in approximate optimal decisions. Particularly, when the order size is small with respect to the market volume: $v_{n-1} \ll V_n$, using the value function approximation, we derive the following heuristic trading policy:

$$\Delta v_i^* = g_i(v_{i-1}, V_i) = \min\left(v_{n-1} - v_{i-1}, \max\left(0, \mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right] v_{n-1} - v_{i-1}\right)\right) \quad (71)$$

Ignoring the boundary constraints, (71) is equivalent to

$$\frac{v_{i-1} + \Delta v_i^*}{v_{n-1}} = \mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right] \quad (72)$$

(72) reveals that the optimal trading size Δv_i^* is the size that matches the prospective cumulative execution percentage with the expected cumulative market volume percentage for the next period. Therefore, the (approximate) optimal strategy is a cumulative curve matching strategy. It constantly updates the cumulative market curve based on realized volume V_i, and tries to match $\frac{v_i}{v_{n-1}}$ with expected $\frac{V_{i+1}}{V_n}$. Although (72) may look very natural, it is important to point out this is not the only matching scheme. Rather than matching on cumulative terms, an alternative is to match the next period's size with predicted proportion of the next period's volume over the remaining trading horizon's volume:

$$\Delta v_i = \mathbb{E}\left[\frac{\Delta \hat{V}_{i+1}}{\sum_{j=i+1}^{n} \Delta \hat{V}_j}\right](v_{n-1} - v_{i-1}). \quad (73)$$

where $\Delta \hat{V}_j (j = i+1, i+2, ..., n)$ are predicted market volume given all previous periods' volume up to time t_i. In fact, this is commonly used in the econometrics literature, such as in [11] and [25]. Under this approach, the cumulative market volume is no longer explicit in the decision function, which only directly depends on v_{i-1}. The dependency of V_i is implicit through the prediction of future volume $\Delta\hat{V}_j(j = i+1, i+2, ..., n)$. The dependency of the trading decision over v_{i-1} and V_i is hence separated. In contrast, (71) combine v_{i-1} and V_i organically, which can results in a better performance in numerical test.

The simple dependency of the optimal adaptive decision Δv_i^* over just two variables: v_{i-1} and V_i is based on the assumption (68). Although (68) may not be fully consistent with the market microstructure property, we believe it nonetheless captures the most important component of the adaptive VWAP problem. This is illustrated fully in the numerical section where our simple adaptive strategies not only outperform deterministic strategies, but also perform as well as (or even better than) strategies based on (73) when sophisticated time series models are used to predict future volumes. Therefore, we can argue that allowing a more complex market process assumption is secondary to the tractability of a simpler solution format (71) under the less sophisticated assumption in (68).

One final missing piece in (71) involves the estimation of $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right]$, which requires some assumption of the distribution of $(\Delta V_{i+1}, \Delta V_{i+2}, ..., \Delta V_n)$ conditioned on V_i. The independence assumption of period volume in (68) allows us to treat conditional expectation as marginal expectation, which greatly simplify the estimation from real market data. According to [41], the expected ratio of two random variables Y and Z follows

$$\mathbb{E}\left[\frac{Y}{Z}\right] \approx \frac{\mathbb{E}Y}{\mathbb{E}Z} - \frac{\mathrm{Cov}(Y,Z)}{(\mathbb{E}Z)^2} + \frac{\mathbb{E}Y}{(\mathbb{E}Z)^3}\mathrm{Var}Z \tag{74}$$

Utilizing (74), under the assumption of (68),

$$
\begin{aligned}
& \mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle| V_i\right] \\
= & \frac{\mathbb{E}[V_{i+1}|V_i]}{\mathbb{E}[V_n|V_i]} - \frac{\operatorname{Cov}(V_{i+1}, V_n|V_i)}{(\mathbb{E}[V_n|V_i])^2} + \frac{\mathbb{E}[V_{i+1}|V_i]}{(\mathbb{E}[V_n|V_i])^3}\operatorname{Var}[V_n|V_i] \quad (75) \\
= & \frac{V_i + \mu_{i+1}}{V_i + \sum_{j=i+1}^n \mu_j} - \frac{\sigma_{i+1}^2}{\left(V_i + \sum_{j=i+1}^n \mu_j\right)^2} + \frac{V_i + \mu_{i+1}}{\left(V_i + \sum_{j=i+1}^n \mu_j\right)^3} \sum_{j=i+1}^n \sigma_j^2 \quad (76)
\end{aligned}
$$

Some of the literature approximates the expected ratio through just the first order terms, i.e. $\mathbb{E}\left[\frac{Y}{Z}\right] \approx \frac{\mathbb{E}Y}{\mathbb{E}Z}$, which only requires first moment estimation of future volume. However, empirical testing shows that additional second and third order approximation as in (74) and (76) provides additional improvement in both deterministic and adaptive strategy with additional effort of second moment estimation of future volume process.

3.2.4 Controlled Adaptive Strategy

Note that in (72), the left hand side consists only of traders' decision, while the right hand side only involves market volume. Since $\Delta v_i \in [0, v_{n-1} - v_{i-1}]$, the left hand side of the trader's cumulative curve is constrained within $\left[\frac{v_{i-1}}{v_{n-1}}, 1\right]$. If the prospective cumulative market curve $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle| V_i\right]$ falls outside of this boundary, the decision becomes binary, either trading nothing or everything remaining. In other words, by adding the boundary constraint back, (71) is equivalent to

$$\frac{v_{i-1} + \Delta v_i^*}{v_{n-1}} = \min\left(UB_{i-1}, \max\left(LB_{i-1}, \mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle| V_i\right]\right)\right) \quad (77)$$

where the upper and lower bounds are defined as

$$
\begin{aligned}
UB_{i-1} &= 1 \quad (78) \\
LB_{i-1} &= \frac{v_{i-1}}{v_{n-1}}. \quad (79)
\end{aligned}
$$

In practice, intraday volume process often presents strong dispersion from the historical average. The explanation comes from the equities' particular events, such as an earning announcement, dividend payments, changes in the management board, etc., which have direct influence on the intraday volume process. For example, a volume spike of ΔV_i may leave V_i exceptionally large. Under the assumption (68), future volume processes are independent processes. This may lead to a higher than usual estimate of $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right]$. As a result, following (77), the trader may decide to allocate a larger quantity Δv_i to the next period to catch up with the market cumulative curve. This may not be the optimal decision ex-post if it turns out the rest of the day also observes a higher than usual trading volume. To mitigate the adverse impact of a sudden volume spike, practitioners often impose stricter bounds based on an ex-ante deterministic volume profile $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\right]$:

$$UB_{i-1}(e) = \min\left(\mathbb{E}\left[\frac{V_{i+1}}{V_n}\right] + e, 1\right) \tag{80}$$

$$LB_{i-1}(e) = \max\left(\mathbb{E}\left[\frac{V_{i+1}}{V_n}\right] - e, \frac{v_{i-1}}{v_{n-1}}\right). \tag{81}$$

where e is a trader specified value between $[0, 1]$, which allows the extent of control the trader would like to impose on the adaptive strategy. For example, $e = 0$ means the trader strictly follows the ex-ante cumulative volume profile - the deterministic strategy. Alternatively, if $e = 1$, (80)-(81) is equivalent to (78)-(79), which is an uncontrolled adaptive strategy. In the following, we will refer to the combination of (77), (80) and (81) as the controlled adaptive strategy with notation Adaptive(e). Empirical testing reveals that the controlled adaptive strategy performs significantly better than the uncontrolled one when applied to real market data.

3.3 Numerical Results

In this section, we address the issue of the execution performance of the adaptive strategy. In particular, we are interested in comparing the performance of three strategies:

the deterministic strategy (111)(i.e. Adaptive(0)), the uncontrolled adaptive strategy (i.e. Adaptive(1)), and controlled adaptive strategies (Adaptive(e) for $0 < e < 1$).

3.3.1 Test Setup

Consistent with our previous analysis, we will assume the parent order size is considerably smaller than the market volume: $v_{n-1} \ll V_n$, and the market process follows the assumption of (68). Under these assumptions, the (approximate) optimal strategy of Adaptive(e) for any $0 \leq e \leq 1$ can be easily derived through curve matching of (77), where the upper and lower bound are defined in (80)-(81) and the conditional expectation of $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right]$ follows (76). Therefore, the (approximate) optimal decision requires the estimation of the mean μ_i and variance σ_i^2 of period volume ΔV_i for $i = 1, 2, ..., n$. The estimation procedure will be described below.

The empirical study focuses on VWAP orders with a one-day horizon. The data set consists of the (volume weighted average) price and volume history with one minute frequency for 500 component stocks from the S&P 500 from 01/01/2012 - 12/31/2012, which contains about 255 trading days. We carried out tests on four different period lengths: 5, 10, 15 and 30 minutes. Since a trading day lasts 390 minutes, the order horizon is separated into 78, 39, 26, 13 periods accordingly. Given a fixed frequency, the period VWAP and aggregated volume are computed for both training and test purposes.

The test is carried out on a per (stock,date) pair with a 20-day rolling window. More specifically, starting from the 21st business day, for each stock and date t, we will use the market data of the stock for the previous 20 business days (from date $t - 20$ to date $t - 1$) to estimate the mean and variance of period volume: μ_i and σ_i^2

for $i = 1, 2, ..., n$. Then these estimations can be plugged into (76), (77), (80) and (81) for decision making on date t. Therefore, we have about (255-20)×500=117500 out-of-sample tests. After removing erroneous and incomplete intra-day data, the number of (stock,date) pairs shrink to 112185. We chose 20 as the number of days because it maintains a good balance between model relevance and statistical significance. A longer training data set may fail to capture the dynamic links that prevail in the market process.

For each (stock,date) out-of-sample pair, the market VWAP Q_n and the order's VWAP $q_n(e)$ are computed when the order follows the Adaptive(e) strategy ($0 \leq e \leq 1$). Consistent with objective (65), the strategy' performance is measured by how close $q_n(e)$ and Q_n are in absolute terms: $|q_n(e) - Q_n(e)|$. When the strategy is applied to panel data across different stock and date combinations, we use the normalized mean absolute error(MAE) to estimate the strategy Adaptive(e)'s performance:

$$\text{MAE}(e) = 10000\mathbb{E}\left[\left|\frac{q_n(e) - Q_n}{Q_n}\right|\right] \tag{82}$$

where the multiplier of 10000 makes MAE(e) in terms of bps unit and expectation is based on sample mean. Smaller values of MAE(e) imply better strategies.

The following subsections will present different strategies' MAEs under different setups. We first look at aggregated statistics over all (stock,date) tests and stock-specific results. These results will present a significant advantage of using adaptive strategies, particularly controlled adaptive strategies. We then look into some studies to understand how different intraday volume risks play roles in adaptive strategies' performances. In the end, we carry out robustness tests with respect to control variable e and alternative ways to compute expected volume percentage.

Table 5: Statistics of mean absolute error(MAE)

	period	Deterministic			Adaptive(1)			Adaptive(0.05)		
		mean	std	Q95	mean	std	Q95	mean	std	Q95
sim	15	4.702	5.936	14.906	3.993	5.703	12.803	3.966	5.448	12.617
real	5	6.520	11.437	21.692	6.314	14.325	21.126	5.579	11.009	18.886
	10	6.397	11.289	21.364	6.191	13.562	20.619	5.528	10.802	18.673
	15	6.294	11.169	21.036	6.108	13.136	20.323	5.490	10.678	18.423
	30	6.024	10.837	20.322	5.904	12.116	19.581	5.391	10.437	18.074

3.3.2 Aggregated Result

In this section, three strategies: Deterministic, Adaptive(0) and Adaptive(0.05) are tested for both simulated data and actual market data. For each (stock,date) pair, market data are resampled in four frequencies - 5, 10, 15 and 30 minutes - while simulated data are generated based on 15 minute frequency. In the simulation, both volume ΔV_i and price ΔP_i are assumed to be normal, and follow assumption (68), where $\mu_i, \sigma_i^2 (i = 1, 2, ..., n), \sigma_P^2$ are calibrated from the previous 20 days' market data. For the controlled policy, we use $e = 0.05$ as a representative for the range $e \in (0, 1)$, which means the adaptive strategy can deviate 5% on both sides from the deterministic strategy. As will be shown later, the performance of controlled adaptive strategies is robust with respect to difference choices of e value.

Table 5 summarizes three statistics of mean absolute error (82) for tests over all (stock,date) pairs: mean, standard deviation(std) and 95% quantile(Q95). The adaptive strategy is designed to minimize mean MAE; therefore, mean MAE is the most important benchmark. Both std and Q95 contain important information about how reliable and robust these strategies are, which are particularly important for practitioners.

Since both the simulated data as well as Adaptive(0) are based on assumption (68), the test on simulated data reveals the potential upper limit of the performance improvement that can be achieved over the deterministic strategy . As shown in Table 5, this limit is about $15.08\% = \frac{4.702-3.993}{4.702}$. Interestingly, controlled adaptive strategy Adaptive(0.05) achieves a similar performance, which shows that the bound of 5% is not too restrictive and most sample trading trajectories do not violate the 5% constraint.

In contrast, the controlled strategies significantly outperform the uncontrolled strategy in all four tests based on real data. Adaptive(0) saves on average 0.18bps of MAE compared with the Deterministic strategy, while Adaptive(0.05) increases this advantage to 0.81bps. In percentage terms, the Deterministic strategy lags Adaptive(0) by 2.8% and Adaptive(0.05) by 14.7%, which is close to the potential upper limit exhibited in the simulation test. Our numerical results show that by applying a two side bounds on the cumulative execution curve, the controlled adaptive strategy can "recover" a significant proportion of potential improvements that are disrupted by simplified assumption of volume and price processes. Furthermore, besides the advantage of closing the gap from market VWAP, the controlled strategy also produced more stable and robust performance as its MAE standard deviation and 95% quantile lead the other two strategies in tests of all four frequencies.

As a comparison, the adaptive strategy in [11] is based on 73 where future volumes are predicted through sophisticated time series models. Their adaptive strategy shrinks the MAE from 10.06bps in the Deterministic strategy (called "classical strategy" in their paper) to 8.98bps in the dynamic strategy. This is only a 10.74% improvement. For a better comparison, we applied the Adaptive(0) and Adaptive(0.05) on a majority (30 out of 40) of the European stocks tested in [11]. A detailed stock specific comparison is presented in Appendix B.2. The test result is consistent with what we observed in American stock tests in Table 5. This not only confirms the superiority of the controlled adaptive strategy over the deterministic strategy, but more importantly, the superiority (or at least comparability) over adaptive strategies based purely on volume prediction with sophisticated econometrics models.

3.3.3 Stock Specific Result

Table 6: Winning percentages on aggregated stock MAEs

strategy	Deterministic	Adaptive(1)	Adaptive(0.05)
winning	0.60%	8.40%	91.00%
	36.20%	63.80%	N/A
	0.60%	N/A	99.40%
	N/A	8.60%	91.40%

Different stocks posses different volume and price risks. This section looks at stock specific result where the average MAEs' over a year for each stock are collected. From now on, we will restrict our tests to data of 15 minutes frequency. When the three strategies are compared, Adaptive(0.05) is superior to the other two in trading 455 of the stocks(91%), the uncontrolled adaptive strategy takes the second place by leading in 42 stocks(8.47%), and the Deterministic strategy is only optimal for three stocks(0.60%). This is reported in the first row of Table 6. The rest of the rows in Table 6 show all possible comparisons of two strategies where Adaptive(0.05) dominates the other two strategies.

Table 7: Stock specific improvement of adaptive vs. deterministic strategy

x	y	regression	R^2	p-val(intercept)	p-val(slope)
MAE(0)	MAE(0-1)	$y = -0.0998 + 0.0457x$	0.0358	0.1666	1.32E-05
MAE(0)	MAE(0-0.05)	$y = 0.0163 + 0.1252x$	0.6198	0.595	5.04E-106

Another significant advantage of the controlled adaptive strategy is that it performs best exactly for those stocks where the Deterministic strategy fails most in

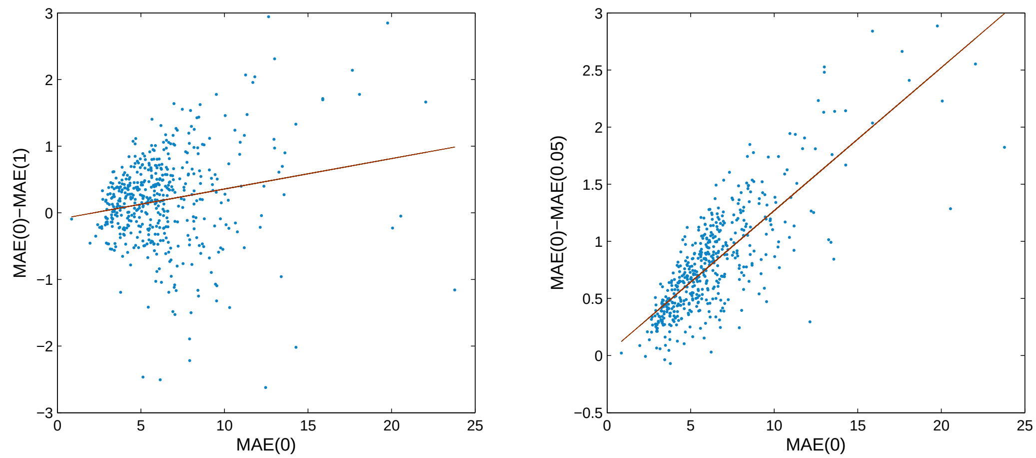

Figure 10: Dependency between deterministic strategy's MAE and the gain/loss for adaptive strategy

tracking the market VWAP. This is verified by studying whether the adaptive strategies are able or unable to correct the large tracking errors of the Deterministic strategy. Figure 10 shows the scatter plot of MAE(0) on the x-axis against the gain or loss observed by applying either uncontrolled or controlled adaptive strategies on the y-axis, with each point corresponding a stock. The statistics of two regressions are presented in Table 7. The slope coefficients in both regressions are positive, with p-values smaller than 1%, which means the larger the tracking error of the deterministic strategy, the larger the gain of the adaptive strategies. Quantitatively, Adaptive(1) decreases the tracking error by 4.57% while Adaptive(0.5) decreases it by 12.52%. The R^2 coefficient for Adaptive(0.05) is particularly large, indicating a strong linear positive link between MAE(0) and MAE(0-0.05)(short form for MAE(0)-MAE(0.05)). In fact, even for these few stocks where Adaptive(0.05) fails to outperform the Deterministic strategy, the loss is smaller than 0.1bps. However, for those stocks with MAE(0) larger than 15bps, the improvement of switching to Adaptive(0.05) is at least 1.3bps. This result confirms that our controlled adaptive strategy is a real improvement since it is not only better in average terms, but it also is most beneficial precisely when it is mostly needed.

3.3.4 Analysis Based on Volume Risks

Table 8: Test results of two specific samples

stock	date	mean	stdev	skew	kurt	MAE(0)	MAE(1)	MAE(0.05)
FAST	2012/02/13	-0.109	0.283	-0.793	2.714	27.829	7.330	16.196
AIG	2012/05/04	0.630	0.588	5.155	22.640	15.996	27.752	3.987

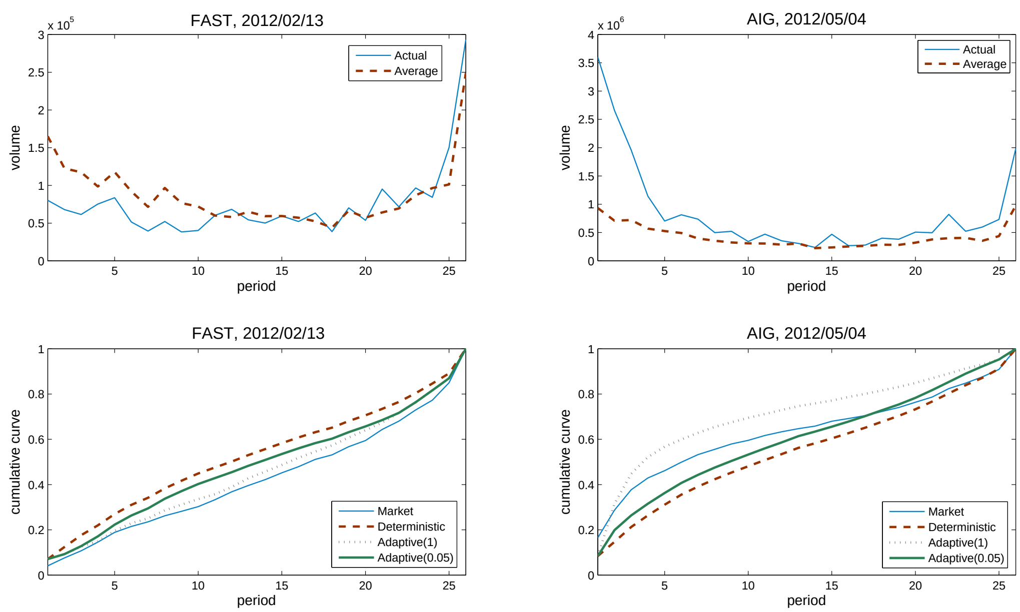

Figure 11: Market volume and cumulative curves for two specific examples

The discrepancy of the test results performed over simulation data and actual market data suggests that model misspecification may be the cause of Adaptive(1)'s underperformance. This section takes a closer look at this issue under different volume risks. First, we take a look at two actual examples: FAST on 02/13/2012 and AIG on 05/04/2012. Table 8 presents intraday volume statistics as well as three strategies' performances. The mean, stdev, skew and kurt columns refer to the mean, standard deviation, skewness and kurtosis of normalized excess period volume: $\frac{\Delta V_i - \mu_i}{\mu_i}$ where μ_i is the average of the i-th period volume of the stock during the previous 20 days.

The top row of Figure 11 plots the actual market volume ΔV_i and historical average μ_i for $i = 1, 2, ..., 26$. The bottom rows plot four cumulative curves: the cumulative market volume percentage and cumulative execution percentage for three strategies. Naturally, the closer the cumulative action curve is to the actual cumulative market volume curve, the smaller the MAE of the strategy.

In the FAST case, the period volume is approximately normally distributed without a large deviation from the historical average. In this case, the uncontrolled adaptive strategy performs best with a close match with cumulative market volume curves. In the AIG case, there is a volume spike at the beginning of the day where the first period's actual volume is 4.5 times that of the historical average. The market volume spike results in aggressive trading at the beginning of the day. By the end of the fourth period, Adaptive(1) already completed 50% of the order. On the other hand, Adaptive(0.05) is protected by the upper bound and trades more slowly. It turns out that the excess volume observed in the early morning lasts for the remainder of the day (the last period's volume is about 3 times that of the historical average), which vindicates a less aggressive trading in the morning than Adaptive(1) does. The large volume spike at both ends of the day, as well as the overall and consistent larger than average volume for the rest of the day are reflected in the four moment statistics of the period volume. Particularly, a mean value of 0.63 suggests an overall larger than average volume for the whole day. The skewness of 5.155 suggests a consistency of positive excess volume for majority of periods and the kurtosis of 22.640 suggests the existence of a few periods with a particularly large volume spike.

Table 9 categorizes all (stock,date) pairs into ten groups based on the mean values of normalized excess period volume: $\frac{\Delta V_i - \mu_i}{\mu_i}$. All groups have the same sample size with group zero containing (stock,date) pairs with the lowest realized volume and group 9 with the highest realized volume with respect to the historical average. Table 10 - Table 12 present similar test results based on the standard deviation, skewness and kurtosis of the normalized excess period volume. The columns MAE(0)-MAE(1) in all four tables show that the uncontrolled adaptive strategy works best when excess volume is not significant (i.e. its mean is near zero in Table 9), within reasonable variation (i.e. not too large, not too small as in Table 10), balanced (i.e. a mix with both positive excess and negative excess, as in Table 11) and has few large dispersions from the historical average (i.e. small kurtosis in Table 12). It is interesting to point out that the smaller standard deviation also hurts Adaptive(1)'s performance. This is because the variance of period volume are used to estimate the cumulative curve in (76). The top three categories with the worst Adaptive(1) performances are those (stock,date) pairs with the largest overall excess volume (group 9 in Table 9), most consistent deviation direction (group 9 in Table 11 where almost all periods are either larger or smaller than the historical average) and existence of a few extreme large deviations (group 9 in Table 12). All of them may result in a particularly erroneous estimation in (76).

As an alternative, Adaptive(0.05) greatly improves the execution quality. As shown in column MAE(0)-MAE(0.05), out of forty groups, Adaptive(0.05) outperforms the deterministic strategy for 37. Within the three groups for which Adaptive(0.05) fails, the maximum loss is 0.169bps (group 8 in Table 12). Furthermore, similarly to our previous stock specific results in Section 3.3.3, the controlled adaptive strategy gets the highest gain precisely in groups for which the uncontrolled version suffers the greatest loss. This is illustrated in column MAE(1)-MAE(0.05). For example, the top three groups where Adaptive(1) performs the worst coincide with the top three groups where Adaptive(0.05) demonstrates the greatest improvement over the deterministic strategy.

Table 9: Performance based on 10 levels of mean normalized excess volume

group	min	medium	max	MAE(0-1)	MAE(0-0.05)	MAE(1-0.05)
0	-0.959	-0.516	-0.431	-1.749	0.381	2.130
1	-0.431	-0.372	-0.323	-0.058	0.467	0.525
2	-0.323	-0.279	-0.238	0.788	0.600	-0.188
3	-0.238	-0.197	-0.157	1.663	1.072	-0.591
4	-0.157	-0.117	-0.075	2.193	1.535	-0.657
5	-0.075	-0.031	0.017	2.427	1.820	-0.607
6	0.017	0.070	0.130	2.331	1.760	-0.571
7	0.130	0.200	0.287	1.765	1.217	-0.548
8	0.287	0.400	0.570	-0.006	0.096	0.102
9	0.570	0.895	38.076	-7.492	-0.904	6.589

Table 10: Performance based on 10 levels of standard deviation of normalized excess volume

group	min	medium	max	MAE(0-1)	MAE(0-0.05)	MAE(1-0.05)
0	0.018	0.188	0.222	-2.050	0.092	2.142
1	0.222	0.250	0.274	0.006	0.542	0.537
2	0.274	0.297	0.320	0.760	0.819	0.059
3	0.320	0.343	0.367	1.325	1.076	-0.249
4	0.367	0.391	0.418	1.491	1.158	-0.332
5	0.418	0.448	0.480	1.490	1.146	-0.344
6	0.480	0.517	0.561	1.473	1.124	-0.350
7	0.561	0.613	0.679	1.078	0.944	-0.133
8	0.679	0.772	0.916	0.203	0.710	0.507
9	0.916	1.218	33.472	-3.915	0.433	4.347

3.3.5 Robustness

This section presents evidence to suggest that the adaptive strategy is robust. First, the robustness with respect to different sampling frequencies, such as 5, 10, 15, and 30 minutes, are presented in Section 3.3.2. Second, all the previous results with regard to the controlled adaptive strategy are represented by Adaptive(0.05). As an extension to Table 5, Figure 12 plots the mean, standard deviation and 95% quantile for MAE(e) of any Adaptive(e) with $0 \leq e \leq 0.15$ under all four sampling frequencies. With respect to minimizing the expected MAEs, all adaptive strategies with

Table 11: Performance based on 10 levels of skewness of normalized excess volume

group	min	medium	max	MAE(0-1)	MAE(0-0.05)	MAE(1-0.05)
0	-158.909	-6.458	-4.209	-2.121	0.200	2.320
1	-4.209	-3.004	-2.129	0.114	0.521	0.407
2	-2.129	-1.437	-0.879	1.063	0.848	-0.215
3	-0.879	-0.381	0.053	1.777	1.249	-0.528
4	0.053	0.460	0.829	2.216	1.551	-0.664
5	0.830	1.190	1.538	2.268	1.629	-0.640
6	1.538	1.871	2.211	2.163	1.527	-0.636
7	2.211	2.562	2.931	1.395	1.021	-0.374
8	2.931	3.338	3.835	-0.165	0.135	0.300
9	3.835	4.533	11.840	-6.850	-0.637	6.212

Table 12: Performance based on 10 levels of kurtosis of normalized excess volume

group	min	medium	max	MAE(0-1)	MAE(0-0.05)	MAE(1-0.05)
0	1.424	2.531	2.938	2.554	1.813	-0.741
1	2.938	3.322	3.719	2.124	1.442	-0.683
2	3.719	4.123	4.570	1.822	1.285	-0.537
3	4.570	5.051	5.593	1.548	1.132	-0.416
4	5.594	6.193	6.873	1.312	1.018	-0.294
5	6.873	7.636	8.516	0.916	0.789	-0.127
6	8.516	9.586	10.871	0.354	0.518	0.163
7	10.871	12.452	14.477	-0.647	0.120	0.767
8	14.477	17.307	21.822	-2.718	-0.169	2.549
9	21.823	32.823	16837.196	-5.405	0.098	5.503

$0 < e \leq 0.15$ beat the deterministic one: MAE(0). The optimal control variable that maximizes the improvement is reached between 0.06 and 0.07 for all four frequencies. Although not plotted in Figure 12, the superiority of the adaptive strategy over the deterministic strategy in terms of minimizing $\mathbb{E}[\text{MAE}(e)]$ is also true for any $e > 0$. However, this is not the case in terms of minimizing the standard deviation or 95% of MAE(e). The relaxation of the cumulative curve constraints as e increases from 0 to 1 first decreases the standard deviation (or 95% quintile) of MAE(e), but gradually increases after $e = 0.07$ and eventually exceeds the level reached at $e = 0$ of deterministic strategy.

Third, both the deterministic and adaptive strategies rely on estimations of the cumulative market volume curve $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right]$. All previous results are based on (76) which uses a third order approximation. A simpler approach, as used in the academic literature, such as [11], only approximates with first order:

$$\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right] \approx \frac{V_i + \mu_{i+1}}{V_i + \sum_{j=i+1}^{n} \mu_j}. \tag{83}$$

Table 13 repeat all tests in Table 5 with this simpler approximation. Not surprisingly, the results in Table 13 are slightly worse than those in Table 5. The interesting part is to compare strategy specific performance. Using a first order approximation, the Deterministic strategy results in about 0.103bps underperformance for market data tests (mean column under Deterministic), but the equivalent value is only 0.010bps for Adaptive(1) and 0.0002bps for Adpative(0.05). This proves that the adaptive strategies are much more robust to different methods of computing the expected ratio $\mathbb{E}\left[\frac{V_{i+1}}{V_n}\middle|V_i\right]$ than is the deterministic strategy. This flexibility allows us to disregard the second moment estimation of period volume σ_i^2 in (83) and only focus on estimating the mean of the historical period volume.

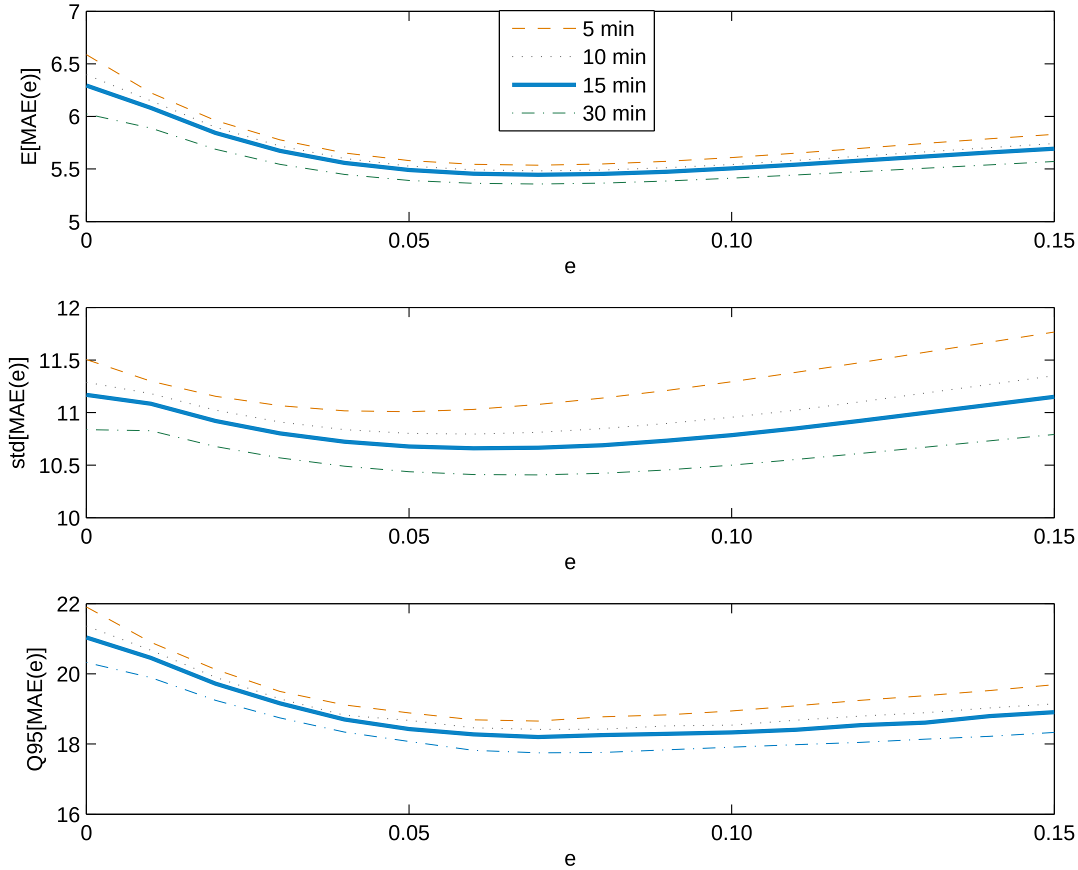

Figure 12: Statistics of Mean Absolute Error(MAE(e)) for varying e

3.4 Optimal Strategy through Approximate Dynamic Programming

This section fills the gap of analytical details that was missing in Section 2, particularly the proof of Property (70) and the derivation of the curve matching approach (71) through value function approximation.

Although most of the results and numerical tests are based on assumption (68), we will start with a more general assumption about market processes $\{P_i, V_i\}_{i=1}^{i=n}$: we assume the two dimensional process (V_i, P_i) is jointly Markovian and depends only

Table 13: Statistics of MAE when expected ratio is approximately on first order

	period	Deterministic			Adaptive(1)			Adaptive(0.05)		
		mean	std	Q95	mean	std	Q95	mean	std	Q95
sim	15	4.719	5.966	14.976	4.007	5.721	12.833	3.971	5.456	12.642
real	5	6.587	11.505	21.905	6.317	14.227	20.992	5.574	10.970	18.796
	10	6.489	11.396	21.678	6.198	13.431	20.470	5.524	10.748	18.543
	15	6.405	11.305	21.409	6.119	12.982	20.150	5.488	10.613	18.301
	30	6.166	11.031	20.823	5.925	11.912	19.401	5.403	10.354	17.992

on the cumulative market volume and price up to time $t_{i-1} : (V_{i-1}, P_{i-1})$. Naturally (68) is a special case under this assumption. As we proceed with the analysis, we may restrict the joint Markovian assumption when necessary. But starting with a more general assumption allows part of our results to be used for further research, particularly in the econometrics and the Approximate Dynamic Programming communities.

The joint Markovian assumption incorporates two empirical properties with respect to volume and price processes. First, the intra-day volume spike, often driven by economic or corporate news and announcements, tends to be followed by a larger than usual trading volume after the announcement. [11] observed that there exists a positive autocorrelation between intraday excess volumes. In our model, the lasting effect of a volume spike can be represented through ΔV_i's dependence upon realized cumulative volume V_{i-1}: a large value of V_{i-1} may imply a large ΔV_i probabilistically. Second, note that the Markovian assumption allows the possibility of a positive correlation between ΔV_i and ΔP_i^2, which is observed in empirical study. This is consistent with the "Mixture of Distribution Hypothesis" by [14] and empirical analysis by [26], which assumes all traders simultaneously receive the new price signals. As a result, there exists a contemporaneous correlation between price volatility and market trading volume.

3.4.1 Dynamic Programming Formulation

3.4.1.1 Objective Decomposition

We start by building the Bellman Equation that the optimal strategy of (66) should satisfy. To achieve that, we need to first decompose (66) as the sum of adaptive cost functions. We have

$$q_n - Q_n = \sum_{i=1}^{n} c_i. \tag{84}$$

where

$$\begin{aligned}
r_i &= \frac{\Delta v_i}{v_i} \quad \text{if } v_i \neq 0; \quad r_i = 0 \text{ if } v_i = 0 \quad \text{for } i = 0, 1, ..., n-1; \quad r_n = 0; \\
R_{i+1} &= \frac{\Delta V_{i+1} + \Delta v_i}{V_{i+1} + v_i} \quad \text{if } V_{i+1} + v_i \neq 0; \\
R_{i+1} &= 0 \text{ if } V_{i+1} + v_i = 0 \quad \text{for } i = 0, 1, ..., n-1; \\
c_i &= (P_i - Q_{i-1})(r_{i-1} - R_i) - r_i(q_i - Q_i) \quad \text{for } i = 1, 2, ..., n.
\end{aligned} \tag{85}$$

Following Chapter 2 of this thesis, (84) can be used to decompose the objective (66) as the sum of cost components:

$$d_0(q_n - Q_n) + (q_n - Q_n)^2 = \sum_{i=1}^{n} \left(d_{i-1}c_i + c_i^2 \right) =: \sum_{i=1}^{n} C_i \tag{86}$$

with the introduction of a variable d_i representing realized portion of final VWAP slippage $q_n - Q_n$ as in (84): $d_i = d_{i-1} + 2c_i = 2\sum_{j=1}^{i} c_j$. The cost component contributing to the linear quadratic objective is

$$\begin{aligned}
C_i &:= d_{i-1}c_i + c_i^2 \\
&= d_{i-1}[(P_i - Q_{i-1})(r_{i-1} - R_i) - r_i(q_i - Q_i)] \\
&\quad +[(P_i - Q_{i-1})(r_{i-1} - R_i) - r_i(q_i - Q_i)]^2.
\end{aligned} \tag{87}$$

Note that since $c_1 = 0$ from (85), $C_1 = 0$.

3.4.1.2 State Variables and Value Function

To carry on dynamic programming, we need to define state variable S_i, which summaries the past decision and exogenous processes:

$$h_i = (S_0, \Delta v_0, P_1, V_1, S_1, \Delta v_1, ..., \Delta v_{i-1}, P_i, V_i)$$

There are multiple ways to represent the state variable; we will pick the following one:

$$S_i = (d_{i-1}, v_{i-1}, V_i, r_{i-1}, R_i, q_{i-1}, Q_{i-1}, P_i). \tag{88}$$

The state variable records all information necessary to make the next optimal decision Δv_i:

- d_{i-1}: previous execution's cumulative contribution to the VWAP error;
- v_{i-1}: cumulative executed shares by the trader;
- V_i: cumulative market volume excluding the trader's;
- $r_{i-1}v_{i-1}$: child order size executed in $[t_{i-1}, t_i)$;
- $R_i(V_i + v_{i-1})$: total market volume during $[t_{i-1}, t_i)$;
- q_{i-1}: trader's realized VWAP;
- Q_{i-1}: market's realized VWAP;
- P_i: current price.

Since the next decision Δv_i follows immediately once S_i is determined, S_i is also referred to as a pre-decision state variable in Approximate Dynamic Programming community [35].

The cost function defined on the state variable is $C_i(S_i, \Delta v_i) = d_{i-1}c_i + c_i^2$ with

$$
\begin{aligned}
c_i(S_i, \Delta v_i) &= (P_i - Q_{i-1})(r_{i-1} - R_i) \\
&\quad - \frac{\Delta v_i}{v_{i-1} + \Delta v_i}\Big[(1 - r_{i-1})q_{i-1} + r_{i-1}P_i - (1 - R_i)Q_{i-1} - R_iP_i\Big] \quad (89)
\end{aligned}
$$

The transition function $S_{i+1} = S_{i+1}^M(S_i, \Delta v_i, \Delta V_{i+1}, \Delta P_{i+1})$ is

$$q_i = (1 - r_{i-1})q_{i-1} + r_{i-1}P_i; \quad (90)$$

$$Q_i = (1 - R_i)Q_{i-1} + R_iP_i; \quad (91)$$

$$d_i = d_{i-1} + 2\Big[(P_i - Q_{i-1})(r_{i-1} - R_i) - \frac{\Delta v_i}{v_{i-1} + \Delta v_i}(q_i - Q_i)\Big]; \quad (92)$$

$$v_i = v_{i-1} + \Delta v_i; \quad (93)$$

$$r_i = \frac{\Delta v_i}{v_i}; \quad (94)$$

$$P_{i+1} = P_i + \Delta P_{i+1}; \quad (95)$$

$$V_{i+1} = V_i + \Delta V_{i+1}; \quad (96)$$

$$R_{i+1} = \frac{\Delta V_{i+1} + v_i r_i}{V_{i+1} + v_i}. \quad (97)$$

Note from (90) and (91) that the update of q_i and Q_i depend solely on S_i, and do not incorporate any new decision Δv_i or new information $(\Delta V_{i+1}, \Delta P_{i+1})$. The initial state is $S_0 = (d_{-1}, v_{-1}, V_0, r_{-1}, R_0, q_{-1}, Q_{-1}, P_0) := (d_0, 0, 0, 1, 1, 0, 0, 0)$.

The transition function $S_{i+1} = S_{i+1}^M(S_i, \Delta v_i, \Delta V_{i+1}, \Delta P_{i+1})$(i.e. (90)-(97)) is decomposed into two parts:

$$S_i^v = S_i^{M,v}(S_i, \Delta v_i); \quad (98)$$

$$S_{i+1} = S_{i+1}^{M,W}(S_i^v, P_{i+1}, V_{i+1}). \quad (99)$$

We propose that the post-decision state variable has the form

$$S_i^v = (d_i, v_i, q_i, Q_i, P_i, V_i, r_i)$$

which is one dimension less than the pre-decision state S_i. In terms of transition functions, (98) is based on (90)-(94) while keeping (P_i, V_i) unchanged. Once P_{i+1} and

V_{i+1} are known, the transition (99) is based on (95)-(97).

The backward induction based on post-decision state variables is as follows:

for $i = n$**:**

$$J_{n-1}(S^v_{n-1}) = \mathbb{E}\Big[C_n(S_n)\Big|S^v_{n-1}\Big]. \tag{100}$$

for $i = 1, 2, ..., n-1$**:**

$$J_{i-1}(S^v_{i-1}) = \mathbb{E}\Big[\min_{\Delta v_i}\big(C_i(S_i, \Delta v_i) + J_i(S^v_i)\big)\Big|S^v_{i-1}\Big] \tag{101}$$

where the optimal action for $i = n-1$ is fixed as $\Delta v^*_{n-1} = v_{n-1} - v_{n-2}$, and

$$\Delta v^*_i(S_i) = \arg\min_{\Delta v_i}\big(C_i(S_i, \Delta v_i) + J_i(S^v_i)\big); \tag{102}$$

for $i = 0$**:**

$$\Delta v^*_0(S_0) = \arg\min_{\Delta v_0} J^v_0(S^v_0); \tag{103}$$

Note that in (101), given the post-state variable S^v_{i-1}, the decision Δv_i is made after observing the new information P_i and V_i. This allows the flexibility of using observed market data, which arrives after trading decision Δv_i, to replace the expectation in (101). Using a post-decision state variable not only decreases the dimension of the value function, but most importantly, allows us the potential flexibility of directly estimating the value function with real data without ever knowing their distribution. However, in later section of the chapter, we do assume a simple distribution for the stochastic processes (P_i, V_i) for a quick test using market data. But the structure we built here through post-decision state variables lays a flexible ground for further research through ADP techniques([35]).

3.4.2 Optimal Adaptive Strategy

The optimal trading size Δv_i^* allocated within $[t_i, t_{i+1})$ depends on the value function J_i as in (102). Traditional dynamic programming uses backward induction (101) to estimate value functions sequentially. However, this method suffers the curse of dimensionality as the state variable has at least seven dimensions, which makes it computationally infeasible to work with a tabular representation with reasonable resolution. Approximate Dynamic Programming solves the curse of dimensionality by approximating the value function with lower dimensionality representation, either through state aggregation or features regression. In our problem, we also need to resort to function approximation in Section 3.2 to achieve a structurally simple and computationally feasible solution. But as a first step, we will derive an interesting analytical result for the optimal trading size Δv_i^*.

3.4.2.1 Optimal Trading Size

One important property of the value function is that it is the expected sum of the cost functions following the optimal strategy. For fixed i:

$$J_i(S_i^v) = \mathbb{E}\Big[\sum_{j=i+1}^{n} C_j(S_j, \Delta v_j^*(S_j))\Big| S_i^v\Big] \tag{104}$$

where $\Delta v_{i+1}^*(S_{i+1}), ..., \Delta v_{n-1}^*(S_{n-1})$ all follow the unknown optimal adaptive strategy. For notational ease, we will drop the star sign and functional expression below and assume the trading action Δv_j is a function of S_j for $j > i$, given $S_i^v = \{d_i, v_i, q_i, Q_i, P_i, V_i, r_i\}$ is fixed and known[3].

A closer look into (104) reveals a decomposition of $J_i(S_i^v)$ into two parts: one that can be expressed analytically through S_i^v, and the other that is a conditional

[3]Note that given S_i^v, the decision at time i: $\Delta v_i = (1 - r_i)v_i$ is a known scalar. Therefore, the notional simplification is not applied on Δv_i.

expectation of a future stochastic process. More importantly, this decomposition can be used to simplify the optimal strategy $\Delta v_i^*(S_i)$:

Propersition 2 1. *For $i = 0, 1, ..., n-2$:*

$$\begin{aligned} J_i(S_i^v) &= d_i K(S_i^v) + K^2(S_i^v) - (d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v] \quad (105) \\ \Delta v_i^*(S_i) &= \arg\min_{\Delta v_i} \Big[-(d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v] \Big] \quad (106) \end{aligned}$$

where $K(S_i^v)$ is $\mathcal{F}_i$-measurable, while U_I depends on future stochastic process $\{(P_j, V_j)\}_{j=i+1}^n$ and future optimal decision $\Delta v_{i+1}, ..., \Delta v_{n-1}$:

$$\begin{aligned} K(S_i^v) &= -(1-r_i)(q_i - Q_i) + \frac{v_i}{v_{n-1}}(1-r_i)(q_i - P_i); \quad (107) \\ U_I &= \frac{[V_i + (1-r_i)v_i](Q_i - P_i)}{V_n + v_{n-1}} + \sum_{j=i}^{n-1} \left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}} \right) \Delta P_{j+1}. \quad (108) \end{aligned}$$

What is unknown in (106) is the first and second conditional moment of U_I. A particular complication lies in the fact that the total market volume V_n appearing in the denominator of (108) may not be independent of $\{\Delta P_j\}_{j=i+1}^n$. However, under the following assumptions, the decision function can be simplified:

Proposition 3 1. *For $i = 0, 1, ..., n-2$, assume*

1. $\mathbb{E}[\Delta P_{i+1}|\mathcal{F}_i] = 0$;
2. *given $\mathcal{F}_i$, $\sum_{j=i+1}^n \Delta V_j$ and ΔP_{i+1} are conditionally independent;*

Then:

$$\Delta v_i^*(S_i) = \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}} \right)^2 \Delta P_{j+1}^2 \middle| S_i^v \right] \quad (109)$$

Note that under the Markovian assumption, ΔV_{i+1} and ΔP_{i+1} are allowed to be conditionally correlated. Furthermore, future volume $\Delta V_{j+1} (j \geq i+1)$ is allowed to be correlated with previous change ΔP_{i+1}. The second assumption in Proposition 1 essentially restricts the causality flow between volume and price processes to only one direction: future price can depend on the previous volume process, but not the reverse. This assumption is not ungrounded. In [16], the authors studied the values of 30 stocks in DJIA at 5-minute intervals. They found that there exists significant causality flowing from trading volume to return volatility in at least 12 stocks, which is consistent with our assumption since the conditional volatility of ΔP_i depends upon the previous $\{\Delta P_j, \Delta V_j\}_{j=i-1-l}^{j=i-1}$, where l corresponds to the number of lags that shows causality. They also found that the reverse causality from volatility to volume was generally much smaller, and achieved statistical significance in only two out of 30 stocks. A special case that satisfies the second assumption of Proposition 1 is when $\{V_i\}_{i=1}^n$ is a Markovian process by itself, but $\{\Delta P_{i+1}\}$ is allowed to depend on (V_i, P_i) jointly, which has a simple one lag causality flow from volume to price volatility.

(109) greatly simplifies (106). Particularly, only two volume components of S_i^v, v_i and V_i, are kept in (106) while all other components, including P_i, q_i, Q_i, d_i, are no longer "visible". (109) seems to suggest that the expectation conditioned on S_i^v can be simplified as an expectation conditioned only on (v_i, V_i) and consequently the optimal decision Δv_i^* only depends on executed volume v_{i-1} and cumulative market volume V_i. However, there is a catch. Since the future decisions $\Delta v_{i+1}, \Delta v_{i+2}, ..., \Delta v_{n-1}$ are functions of futures $S_{i+1}, S_{i+2}, ..., S_{n-1}$, which all evolve from current state S_i, all these future decisions also depend on P_i, q_i, Q_i, d_i. Luckily, if we assume both volume and price are independent of the current state variable, we can prove through backward induction that the optimal decision only depends on executed volume and cumulative market volume.

Propersition 4 1. *For $i = 0, 1, ..., n-2$, under the condition of Proposition 1 and the assumption that future market information $\sigma(\{\Delta V_j\}_{j=i+1}^n, \{\Delta P_j\}_{j=i+2}^n)$ is independent with post-decision state S_i^v, then the optimal decision only depends on executed* *quantity v_{i-1}, and cumulative market volume V_i:*

$$\Delta v_i^*(S_i) = \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2 \Delta P_{j+1}^2 \middle| (v_i, V_i)\right] =: g_i(v_{i-1}, V_i). \tag{110}$$

Proposition 1 shows that when future market processes are independent of past realization, the optimal decision only depends on how much the trader has executed, and how much the market has traded. Although our eventual goal is to minimize the difference between the trader's total VWAP q_n and the market's total VWAP Q_n, neither of the partially observed VWAPs, q_i and Q_i, plays a meaningful role guiding future trading action. In addition, the optimal decision $\Delta v_i(S_i)$ does not rely on d_{i-1}, which is a function of the linear quadratic coefficient d_0 in the objective function (66):

Corollary 1. *Under the assumption in Proposition 1, the optimal trading decision is independent of d_0 in (66).*

Therefore, the optimal solutions for (65) and (66) are the same. Furthermore, using an approach similar to the one used in Chapter 2 of this thesis, the solution of the mean-variance problem (67) becomes a subset of the linear-quadratic problem (66). Under the assumption that there exists a unique solution to the linear-quadratic problem, both (66) and (67) share the same solution for any risk aversion factor λ. This implies that under our market assumption, there is no meaningful method to outperform the market VWAP, i.e. making a directional bet. The only question we can answer is how to execute the parent order such that our VWAP gets as close to the market VWAP as possible.

A special case under the assumption of Proposition 1 is (68), which makes $(V_i, P_i)_{i=1}^n$ a particularly simple example of a two dimensional Markovian process. Its simplicity ignores the potential causality flow from volume to price as assumed in Proposition

1. It also assumes the future market volume ΔV_i is independent of the previous cumulative volume V_i, which is inconsistent with elastic effects of a news driven volume spike. However, the advantage of a simple decision structure as in (110) outweighs the disadvantage of model misspecification, which will be demonstrated in the numerical test.

3.4.2.2 Function Approximation and Curve Matching

In Proposition 1, if we assume $\mathrm{Var}[\Delta P_{j+1}^2|(v_i, V_i)] = \sigma_P^2$(for $j \geq i$) to be a constant, the optimal decision $\Delta v_i^* = g_i(v_{i-1}, V_i)$ does not depend on price processes anymore:

$$\Delta v_i^* = g(v_{i-1}, V_i) = \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] \tag{111}$$

The optimal decision depends on both future volume processes as well as future optimal decisions $\Delta v_{i+1}, \Delta v_{i+2}, ..., \Delta v_{n-1}$, which are unknown. Using methods similar to those used in Approximate Dynamic Programming, we can approximate

$$\mathbb{E}\left[\sum_{j=i+1}^{n-1} \left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2 \middle| (v_i, V_i)\right]$$

as an explicit function of (v_i, V_i) without dependency over future decisions.

For $i + 1 \leq j \leq n$, define the functions:

$$h(j) = \frac{v_{j-1}}{v_{n-1}}; \tag{112}$$

$$H(j) = \frac{V_j + v_{j-1}}{V_n + v_{n-1}}. \tag{113}$$

$h(j)$ represents the cumulative percentage of shares that has been executed by the trader after the jth trade, while $H(j)$ is its market volume counterpart. Both functions are non-decreasing with respect to j and both of them have the same right boundary value: $h(n) = H(n) = 1$. $h(j)$ and $H(j)$ do not appear in the value function separately. Instead, they always appear as a pair: $h(j) - H(j)$. In an ex-post fashion where we know the market volume in each period, $h(j)$ should equal $H(j)$. However, both the optimal decision and the volume process are stochastic, hence the difference between $h(j)$ and $H(j)$ fluctuates, and eventually decreases(or increases) to 0 when $j = n$. Whatever shape $\{H(j)\}_{j=i+1}^{j=n}$ may have, the future optimal strategy should track the curve of $H(j)$ closer as j increases. Hence, the absolute value of $h(j) - H(j)$ should also decrease. Therefore, we assume $h(j) - H(j)$ is a linear function starting at $h(i+1) - H(i+1) = \frac{v_i}{v_{n-1}} - \frac{V_{i+1}+v_i}{V_n+v_{n-1}}$ and ending at $h(n) - H(n) = 0$:

$$h(j) - H(j) \approx \frac{n-j}{n-i-1}\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1}+v_i}{V_n+v_{n-1}}\right) \tag{114}$$

Taken (112) and (113) into (111):

$$\begin{aligned}
g(v_{i-1}, V_i) &= \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] \\
&\approx \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \left(\frac{n-j}{n-i-1}\right)^2 \mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1}+v_i}{V_n+v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] && (115) \\
&= \arg\min_{\Delta v_i} \frac{(n-i)(2n-2i-1)}{6(n-i-1)} \mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1}+v_i}{V_n+v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] && (116) \\
&= \min\left(v_{n-1} - v_{i-1}, \max\left(0, \frac{\mathbb{E}\left[\frac{V_{i+1}V_n}{(V_n+v_{n-1})^2} \middle| V_i\right]}{\mathbb{E}\left[\frac{V_n^2}{(V_n+v_{n-1})^2} \middle| V_i\right]} v_{n-1} - v_{i-1}\right)\right) && (117)
\end{aligned}$$

The expectation part of (116) is a convex quadratic function of Δv_i, with constraint $0 \leq \Delta v_i \leq v_{n-1} - v_{i-1}$. The detailed derivation from (116) to 117 is is listed in the Appendix B.1. The solution (117), which exists and is unique, reduces the optimal decision's dependence on each remaining periods' market volume and the future market decision. The remaining dependence is on just two unknown sources: the next period's market volume ΔV_{i+1} and the sum of the remaining periods' total volume $V_n - V_i$.

Note that the approximation (115) does not require satisfaction of the assumption of Proposition 1. The major assumption underlying the linear approximation (114) is that $\Delta v_{i+1}, ..., \Delta v_{n-1}$ follow the optimal trajectory that shrinks the gap between $h(j)$ and $H(j)$. This can be applied naturally to (109) in Proposition 1 as well. The seven dimension state space of S_i^v can be aggregated to only two dimensions: (v_i, V_i). This technique is commonly used in the Approximate Dynamic Programming literature to solve the curse of dimensionality. The conclusion in Proposition 1 by focusing only upon (v_i, V_i) provides the inspiration and theoretical support for using the dimension reduction approach in (115).

When the order size is small with respect to the market volume: $v_{n-1} \ll V_n$, $\frac{V_{i+1}+v_i}{V_n+v_{n-1}} \approx \frac{V_{i+1}}{V_n}$. Therefore, (116) can be approximately by

$$\arg\min_{\Delta v_i} \frac{(n-i)(2n-2i-1)}{6(n-i-1)} \mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1}}{V_n}\right)^2 \middle| (v_i, V_i)\right]$$

Following similar procedure as Appendix B.1, we can prove (71).

3.5 *Conclusion*

This chapter deviates from a common approach in the literature of using a sophisticated volume prediction model to improve VWAP trading over the deterministic strategy. We built a dynamic programming approach based on fairly simple assumptions about market volume and price processes. The simple assumption not only made the dynamic programming problem trackable for theoretical analysis, it also allowed us to come up with an intuitive approximation framework that simplified VWAP trading into a curve matching problem, which is exactly the approach the broker-dealer industry often use in practice. To compensate for potential model misspecification, we introduced a controlled adaptive strategy which not only beat the deterministic strategy, but also outperformed the prediction-driven adaptive strategies in the existing literature.

Further research can be conducted on how to incorporating sophisticated intraday volume and price models into a tractable dynamic programming problem, particularly the inclusion of heavy tail and autocorrelation of the volume process. This will likely result in a dynamic programming problem with larger state space and a trackable solution may resort to the recent advances in Approximate Dynamic Programming or Reinforcement Learning.

CHAPTER IV

OPTIMAL TRADING WITH MARKET AND LIMIT ORDERS

4.1 Introduction

Technology innovation has completely transformed how trading is done today in financial markets. Today, the majority of execution orders in the U.S. stock markets are traded electronically in a systematic fashion through computer algorithms. Although there are different execution preferences specified by investors, most algorithmically traded orders go through three steps: trade scheduling, optimal order placement and smart order routing. In the trade scheduling step, a large parent order is split into gradually executed smaller child orders. Once a child order size is determined, it is fed into the optimal order placement model (OOPM) to be executed within a specified duration. OOPM decides how many shares of the child order will be submitted as a market order, and how many shares as limit orders and at which limit prices. If there are multiple trading venues available[1], smart order routing (SOR) is needed next to route the market or limit order to the appropriate trading destination, based on the order type, size and limit price. Simply put, an execution order will be split repeatedly in three ways: split in time by trade scheduling, then split in order type by OOPM and lastly split in space by SOR. An execution algorithm is essentially an algorithm that decides how to split the order in these three dimensions.

What further complicates the splitting process is that investors prefer algorithms that are adaptive to market condition changes. This is not surprising, as an adaptive

[1] In US equity markets, there are more than ten active exchanges(or electronic crossing networks) and more than 40 dark pools.

strategy that accurately incorporates market dynamics should always outperform a deterministic strategy that fixates all splitting at the beginning of the parent order.

An optimal trading algorithm should be designed with the consideration of different market dynamic risks[2]. Unfortunately, these risks are often in conflict with each other, so different steps of algorithmic trading are used to address these conflicts. For example, if the trading cost is measured against the arrival price, a trade scheduling problem tries to find the right balance between fast trading with a large liquidity risk (risk associated with the the adverse price reaction due to one's own trading action, namely, market impact) and slow trading with a timing risk (risk associated with price fluctuations). On the other hand, a optimal order placement model(OOPM) tries to find the optimal balance between paying higher prices by using market orders (i.e. liquidity risk) and facing the uncertainty of incomplete order fills by using limit orders namely, execution risk. Specifically, market orders guarantee immediate execution but come with the costs of spread and market impact. Limit orders can save on the spread, but the orders may not be completely filled. Thirdly, a smart order routing model tries to balance order flow, queue size and fee structures[3].

The three stages approach, widely used in financial industry, is a perfect example of the "divide and conquer" tactic when dealing with different market dynamic risks, which include aforementioned liquidity risk, timing risk and execution risk[4]. For example, the trade scheduling problem addresses the conflicts between liquidity

---

[2]The term 'risk' here is a general term which includes various uncertainties in the evolution of market related variables, such as price, volume, etc.

[3]Different exchanges charge differently for orders executed on their venues, sometimes even with a rebate to attract liquidity

[4]As a summary, the three types of market dynamic risks we are interested in this chapter are:

1. timing risk: risk associated with price volatility;
2. liquidity risk: risk associated with short term supply and demand imbalance, which includes market impact, bid-ask spread, etc. It is often associated with market order submission;
3. execution risk: risk associated with incomplete order transaction, which is often associated with limit order submission;

risk and timing risk while OOPM addresses the conflicts between liquidity risk and execution risk. However, by addressing these risks separately, we may potentially restrict ourselves to suboptimal strategies. One such restriction is the assumption that child orders will always be completely executed in the given period. This assumption simplifies the modeling in the trade scheduling problem. For example, in order to guarantee child orders' complete execution, the authors in two trade scheduling papers: [33] and [1], assume child orders are submitted as market orders. The side effect of this restriction shows up in the downstream OOPM where any unfilled limit orders will be resubmitted as a market order at the end of the period. For example, [18], [21], [6] and [22] analyze order placement strategies that only use market orders as a fall-back should there be any shares left unexecuted. This may not be optimal, such as in the case when spread is large. What if we do not force market order submission at the end of a period and return the unexecuted shares back to the upstream scheduling problem for the next period? This is indeed the option traders sometimes choose as a badly timed market order often raises the eyebrows of investors. However, this behavior makes the original scheduling suboptimal as it never incorporates the possibility of OOPM returning unexecuted shares.

A natural question arises: why not extend the trade scheduling problem from submitting purely market orders to a combined use of market and limit orders? Will this outperform strategies that only submit market orders, assumed for example in [33] and [1]? In other words, rather than addressing the market dynamic risks separately, why not combine the trade scheduling and optimal order placement model together and find an optimal trade-off among liquidity risk, timing risk and execution risk? Will this unified approach improve the execution performance, compared with the approach that treats scheduling and OOPM separately? These are the two main questions we would like to answer in this chapter.

In this chapter, we provide a discrete-time adaptive strategy that minimizes the mean-variance cost of the execution slippage. However, since variance is not a time consistent measure, we cannot directly use dynamic programming to solve it. Instead, we use the same risk diversification framework as the one in Chapter 2 of this thesis to circumvent this issue by solving a family of auxiliary linear-quadratic problems.

To highlight the conflicts among different market dynamic risks, we used simplified assumptions to capture the key risks. Specifically, the liquidity risk associated with market orders is modeled as the sum of the spread and the linear temporary market impact. Timing risk is modeled through price volatility. With regard to execution risk associated with limit orders, we assume that limit orders are submitted only at the prevalent best quote prices (ask prices for sell orders and bid prices for buy orders). When we submit a buy limit order, execution risk is modeled as aggregated sell volume that is eligible to cross with our buy limit order. Naturally, only the sell volume that is submitted at or below our buy limit price can be included. Similarly, for a sell limit order, execution risk is represented by aggregated buy volume submitted at or above the limit sell price that is eligible to cross with our sell limit order. Furthermore, to address different rebates/fees for market and limit orders, we also include them into the modeling.

With these simplifications, the problem is modeled as a finite time horizon dynamic programming problem with two dimensional decision variable and a four dimensional state variable. The decision variable includes market and limit order size, while the state variable includes slippage from the arrival price up to the current time, the latest price change, the current spread and the number of remaining shares and time periods.

Like the optimal strategy from the scheduling problem, the unified strategy trades faster at the start of the order. This is a natural result of the arrival price benchmark. However, under the same market condition, the unified strategy will trade slower than the scheduling strategy at the beginning of the order. This is due to the fact that the unified strategy can wait to submit limit orders at better price levels in the future to take advantage of the price and spread fluctuations and save on market impact. Orders with different sizes are treated differently under the unified strategy. Small orders predominantly use limit orders for execution. Waiting for limit orders to be filled, the unified strategy may not submit market orders until later periods to complete (parent order) trading. On the other hand, large orders require submission of both market and limit orders. If a large number of the limit orders are filled relatively quickly, then submission of market orders may be scaled back accordingly. Besides order size, other factors that affects the unified approach's trading behavior include the trader's risk aversion level and the stock's liquidity condition.

To answer the two questions we proposed, we compare this unified strategy with two different benchmark strategies. One benchmark is a strategy similar to the one discussed in Chapter 2 of this thesis and [3], where only market orders are submitted. The other benchmark is a strategy that separates scheduling and OOPM into two steps. For a fair comparison, the OOPM component in this benchmark is simulated by submitting all child order shares at best quote at the beginning of the period and executing whatever is left as a market order at the end of the period. This is like combining Chapter 2 with a simplified version of [22]. We test these three strategies in out-of-sample test. Even with a conservative estimate of aggregated liquidity that undermines the potential of the unified strategy, the unified approach still outperforms the other two benchmarks consistently under different market settings.

The major contribution of this chapter is to provide a general framework for solving dynamic discrete time execution problem where different market dynamic risks can be integrated in a systematic way. The framework allows us to optimally diversify various risks over time. Therefore we will refer this as a risk diversification framework. Although the model here uses relatively simple assumptions (such as that the market impact is temporary and linear, liquidity flow is an independent time series, etc), it can handle more sophisticated assumptions (such as that market impact decays over time, liquidity flow is auto-correlated due to the consistent short term order flow imbalance, etc). The framework also allows the extension of submitting limit orders at multiple price levels, or use other price benchmarks rather than arrival price.

To our knowledge, papers directly addressing the discrete time trading problem in a mean-variance framework is not very common in both academia and industry. Most trading models are built and solved in continuous time and use it to approximate the discretized optimal solutions (such as [24]). However, modeling in discretized time allows much more flexible assumptions. For example, lead-lag relationship, autocorrelation in liquidity, is not easy to model in continuous time. On the other hand, we choose variance as a risk objective because the simple mean-variance approach has the practical advantage that risk and reward are expressed as two real variables that are easily understood and displayed in a two-dimensional picture. This is an important preference for practitioners in contrast to more mathematically sophisticated utility function formulations. More importantly, the framework we provide here allows the practitioner to efficiently solve the optimal strategies for different risk aversion levels all at once, which makes the two dimensional risk-reward plot easily available.

Once a trading model is built using our framework. There are in general two ways to solve them. One is through traditional backward induction, while the other through forward induction with approximated value functions. The latter approach can be used to significantly decrease the state space dimension, which is the key idea behind approximate dynamic programming(ADP) and reinforcement learning(RL). More importantly, ADP/RL approach does not require direct statistical inference of market variables. The end product of data training is not a parametric/non-parametric description of how the market looks, but directly how to trade. A well designed ADP/RL model can pick up market evolution and reflect it into its decision making process without presenting itself to the model user. We believe this is an important direction for future innovation in algorithmic/systematic trading. Although we present our numerical results based on backward induction solution, the theoretical framework we built here provides an analytical context which can be leveraged by techniques in ADP/RL. We will leave this part of work for future research.

Trade scheduling problems began to receive academic interest when [8] studied it for a risk-neutral trader to minimize market impact cost. By introducing the variance of the slippage as a risk measure, [2] improved upon [8]s result , and derived a deterministic strategy that minimize the mean-variance of the execution cost. Since then, numerous academic studies have appeared that present different variations of the scheduling problem. Some of them use more sophisticated market microstructure assumptions, such as a market impact model based on the shape of the limit order book([33], [1], [36]). Others improve on the deterministic trading strategy by developing adaptive strategies that adjust to market condition evolution (Chapter 2 of this thesis ,[31],[19]).

Although studies of strategies that submit both limit and market orders have appeared, such as [5], it is not until very recently that an optimal order placement model has received academic attention, such as [22] and [15]. [15] consider a one period unified problem that combines order placement and smart order routing. [22] considers a multi-period problem where a risk-neutral trader decides on quantities of limit orders at multiple price levels; whatever is not executed at the end of the given horizon will be executed as market order. We will use a simplified version of [22] to simulate the order placement model in our numerical test.

To our knowledge, besides [24], there is almost no other chapter that combines the scheduling and optimal order placement problems. Both [24] and this chapter combine three risks together: liquidity risk, timing risk and execution risk. However, this chapter differs significantly from [24] in two ways. First, [24] considers a continuous trading problem where orders adjust continuously while our model directly solves the discretized problem. As mentioned before, discretized settings not only fit reality better, but more importantly, can capture a lead-lag relationship in market dynamics which cannot be modeled through continuous models. Our model also fits industry practice as we are using a mean-variance framework, which can be easily interpreted by practitioners while [24] aims to minimize a cost utility function. Second, and more importantly, while the model in [24] is based on a specific market microstructure model from the result of econophysics study, our model provides a general framework that can be easily extended with various market assumptions. This framework allows the practitioner the flexibility to choose between modeling complexity and computation complexity.

This chapter is organized as follows. In Section 4.2, we describe market dynamics, order types, and model different market dynamic risks, such as market impact and eligible liquidity. All these components are integrated together to develop a definition of the execution cost. We then introduce the mean-variance objective as well as the auxiliary linear-quadratic objectives. Section 4.3 derives the Bellman equations that can be used to solve the auxiliary linear-quadratic problems, and we introduce two approaches to solve them: backward induction and forward induction through ADP/RL. Section 4.4 provides detailed description on how to numerically solve the Bellman equations through backward induction. It also discuss how to estimate eligible liquidity from real quote tick-by-tick data. We then test three strategies (a unified strategy that submits both market and limit orders, a strategy that only submits market orders, and a strategy that separates scheduling OOPM into two sequential steps) in out-of- sample tests and compare their performance.

4.2 Problem Formulation

We take the perspective of a risk-adverse trader facing the problem of executing a large order within a given period of time. The goal is to minimize the risk-adjusted execution cost, which is measured against the arrival price. The problem set up and formulation is similar to Chapter 2 of this thesis with one major difference: Chapter 2 allows only market orders while this chapter includes limit orders as well. In the following, we will start by introducing the market dynamics, and incorporate them into the definition of slippage. In the end, we will introduce the risk-adjusted execution cost by discussing mean-variance objective functions as well as the auxiliary linear-quadratic objective functions.

4.2.1 Market Dynamics

Imagine a scenario where a trader is given a buy order of X shares of a certain stock for a given horizon T (normally lasting from a few minutes to a few hours). In this section, we will focus on a buy order; the modeling of a sell order is completely analogous.

To avoid a potential massive market impact by buying all X shares at once, the trader often splits the large order (parent order) into gradually executed smaller ones (child orders). In this chapter, we assume the child orders are equally spaced in time. More specifically, total trading duration T can be divided into n equal periods by $t_0, t_1, ..., t_n$ where $t_0 = 0$ and $t_n = T$. The number of periods n is often determined based on trader's discretion, fee structure as well as infrastructure setup. In general, a period lasts from few seconds to few minutes. Child orders decisions are made at the first n discrete time points: $t_0, ..., t_{n-1}$.

4.2.1.1 Price Dynamics

The security price is determined by both our execution and trading activities of other exogenous traders who submit their orders independently. We assume the latter part is the fundamental price of the security and we model it as a random walk:

$$P_i = P_{i-1} + \epsilon_i \quad for \quad i = 1, 2, ..., n \tag{118}$$

where P_i is the fundamental price at time t_i. P_0 is referred as the arrival price as this is the price when the execution order arrives at the trading desk. An observable proxy for P_i is the market mid quote. As a special case, we can assume the price increments ϵ_i follow independent and identically distributed (i.i.d) normal distributions with zero mean and a standard deviation of σ. Traditionally it is the price return $\frac{P_i - P_{i-1}}{P_{i-1}}$, rather than price difference $P_i - P_{i-1}$ modeled as i.i.d. random variables. However, the trading horizon in our problem is relatively short. Therefore, the assumption that the price difference is i.i.d is not a major divergence from short-term asset price dynamics.

4.2.2 Limit Order and Market Order

As mentioned previously, for $i = 0, 1, ..., n-1$, t_i is the time that the trader makes the $(i+1)$-th child order decision. At each t_i, the trader has the option to submit market and/or limit orders to get access to market liquidity in the upcoming period $[t_i, t_{i+1})$.

A limit order is an order to trade a certain amount of a security at a given price. Limit orders are posted to an electronic trading system and the outstanding aggregated limit orders from both buy and sell orders form the limit order book (LOB). The LOB is represented by multiple price levels and their associated posted quantity. Two price levels are of particular importance in the LOB: the ask price is the lowest price for which there is an outstanding limit sell order, and the bid price is the highest price for the outstanding buy limit order. If we assume the spread between the bid and ask prices at time t_i is s_i and the mid quote is the fundamental price P_i, then the ask price is $P_i + \frac{1}{2}s_i$ and bid price is $P_i - \frac{1}{2}s_i$.

In practice, for a buy order, patient traders can post limit orders at multiple price levels at and/or below the bid price. To simplify modeling in this chapter, we will only post the limit order at the bid price for the buy order. We use y_i to denote the quantity the trader wants to post at bid price $P_i - \frac{1}{2}s_i$ at time t_i. We do not allow any intermediate selling. Therefore, y_i should be non-negative, $y_i \geq 0$. Furthermore, we assume any remaining shares from the previous limit order submitted at t_i will be canceled at time t_{i+1}. Therefore, only one limit order can be active at a time, i.e. only by canceling his current limit order can the trader create a new one. This simplification is important in reducing modeling complexity. Otherwise, the model needs to record all previous outstanding limit orders' fill status, which will result in a state space that grows exponentially over time.

A market order is an order to buy/sell a certain amount of a security at the best available price in the LOB. For example, a buy market order with size x_i will be matched with sell limit orders at ask price $P_i + \frac{1}{2}s_i$. If there is not enough liquidity at ask price, the remaining shares will be matched with the sell limit orders available at the next price level until all x_i shares are executed. Therefore, a market buy order will "eat off" the sell limit orders on LOB with the lowest price and potentially pushes the ask price at a new higher level. In this chapter, we assume the aggregated liquidity at or above ask price level is unlimited. In other words, the completion of market order is guaranteed.

Unlike buy limit orders, where we can guarantee the upper bound for execution price, market buy orders often incur extra liquidity risk beyond ask price, i.e. market impact. The existence of market impact is the reason why traders do not execute the parent order all at once. There are extensive literature studying market impact in the framework of optimal execution, such as Chapter 2 of this thesis,[1] and [33]. The key in our problem is to model the causality relationship that large market order leads to large adverse price deviation. Therefore, we use the simplistic assumption that market impact is temporary and proportional to trading speed. Specifically, we assume the market order with size x_i executed during $[t_i, t_{i+1})$ has an average price of

$$P_i + \frac{1}{2}s_i + \eta\frac{x_i}{T/n} \tag{119}$$

where $\frac{x_i}{T/n} = n\frac{x_i}{T}$ is the average trading speed during period $[t_i, t_{i+1})$. For notational simplicity, we can adjust the time unit such that $T = 1$. Then (119) becomes $P_i + \frac{1}{2}s_i + n\eta x_i$. The market impact factor η represents the liquidity premium, with smaller value of η implying higher liquidity. According to [33], the linear market impact structure is a result of the assumption that LOB has a block shape with same limit order queue lengths at different price levels. Furthermore, we assume more sell limit orders will be attracted to the order book after our trader exhausts the liquidity during period $[t_i, t_{i+1})$, and the price will recover completely to the fundamental price level P_{i+1}.

4.2.2.1 Eligible Liquidity

A market order pays a higher price, but guarantees execution. On the other hand, a limit order saves on spread and market impact[5], but it does not ensure fill completion. Any limit orders that are not completed now have to be executed in the future, which are exposed to future market impacts and/or price fluctuations. Therefore, using limit orders introduces another type of risk: execution risk. We capture execution risk by eligible liquidity. For the buy limit order submitted at price $P_i - \frac{1}{2}s_i$ during $[t_i, t_{i+1})$, its eligible liquidity is defined as the aggregated shares of sell orders submitted at/below $P_i - \frac{1}{2}s_i$ during $[t_i, t_{i+1})$ that is eligible to cross the buy limit order. In other words, it is the number of shares that will be crossed if we submit an infinitely large limit buy order: $y_i = \infty$. Larger eligible liquidity means limit orders will be easily filled and hence incurs less execution risk. We use l_i for the notation of eligible liquidity during $[t_i, t_{i+1})$ that can potentially cross with our limit order submitted at $P_i - \frac{1}{2}s_i$. Note

[5]Recently research(see [23]) points out that limit orders also have market impact. For the simplicity of the analysis, we do not assume that a limit order has a market impact. However, an addition of a simple linear limit order market impact does not add extra modeling complexity.

that l_i will not be known until t_{i+1}. By t_{i+1}, the filled quantity for the limit order will be known as well:

$$y_i \wedge l_i := \min(y_i, l_i).$$

Naturally, the larger the spread s_i, the less likely the limit order will be filled, as it costs more for market sell orders to cross the spread. Therefore, s_i and l_i should be negatively correlated, which can be captured in our model. Furthermore, we do not require $\{l_i\}_{i=0}^{n-1}$ to be identically distributed. This allows us the flexibility to model intraday seasonality of the eligible liquidity, which often has a U-shape similar to intraday volume.

Not all sell orders submitted during $[t_i, t_{i+1})$ can be counted as eligible liquidity l_i. It depends on the market structure where the stock is traded. Take the U.S. equity market for example. The LOBs in the U.S. equity market follow a price/time priority rule for crossing, i.e. buy limit orders with higher price will be crossed first. Furthermore, the U.S. equity market is highly fragmented and investors can trade stocks in multiple displayed electronic systems. As a result of the Order Protection Rule[6], a large sell market order submitted at a trading venue/exchange (for example Venue A) will first cross with buy limit orders at the highest price (i.e. prevailing bid price, such as \$10) in Venue A. If the sell market order is not completely filled, it will be routed to other venues to cross with buy limit orders at the same bid price \$10. It is not until the buy liquidity at \$10 from all venues are depleted will the sell market order return to Venue A to cross with buy limit orders at next bid price level (such as \$9.99) and lowers the bid price from \$10.00 to \$9.99. Following this rule, if we submit a buy limit order at price $P_i - \frac{1}{2}s_i = \$10$ at t_i, and if during $[t_i, t_{i+1})$ we observe sell market orders crossing at \$9.99, or sell limit orders posted at \$9.99, we

[6]Order Protection Rule is one of the provisions of Reg NMS(National Market System) that aims to ensure that both institutional and retail investors get the best possible price for a given trade by comparing quotes on multiple exchanges. If a better price is quoted elsewhere, the trade must be routed there for execution, and not “traded through” at its current exchange.

can guarantee that our buy limit order will have been filled. Otherwise, if the market has less selling pressure during $[t_i, t_{i+1})$, whether or not our buy limit order at \$10 submitted at t_i will be crossed depends not only on the venue at which we submitted our order, but also the number of shares waiting in front our buy limit order at \$10 at this venue.

Therefore, it is not easy to estimate the accurate distribution of eligible liquidity l_i. To circumvent this difficult problem, we will instead give a fairly conservative estimate of l_i, which is easier to observe. In fact, for a buy limit order submitted at \$10 at t_i, we will only count a subset of the limit sell orders posted at \$9.99 during $[t_i, t_{i+1})$, which is a subset of the true l_i. The detailed procedure will be provided in Section 4.4. We can show that even under this conservative liquidity assumption which may reduce its performance potential, the unified trading strategy with both limit and market orders still outperforms the strategy that relies solely on market orders.

However, in actual practice, once we collect enough data of submitted limit orders and their associated filled sizes, we can use the censored maximum likelihood method to directly estimate a parametric distribution of eligible liquidity based on our own historical execution data. Hence the previous described procedure based on market data will no longer be used. We will revisit this issue in Section 4.4.

4.2.2.2 Fee Structure

One last component of market dynamics is the order fee structure. To attract market participants to provide liquidity, some electronic trading venues reward them with a rebate when their submitted limit orders are filled. On the other hand, these systems charge a fee when market orders are submitted to take liquidity. We will use f^l as the rebate for providing liquidity, and f^m as the fee charged for the market order. For example, according to Nasdaqs website, adding liquidity rewards $f^l = \$0.00295$ per share while taking liquidity risks $f^m = \$0.0029$ per share. Most electronic trading venues give rebates for limit orders and charge fees for market orders. However, there are also venues with exactly the opposite fee structure. For these venues, both f^l and f^m will be negative.

4.2.2.3 Execution cost per period

In summary, at time $t_i (i = 0, 1, ..., n-1)$, the trader will submit a market order with size x_i and a limit order of size y_i at the prevailing bid price $P_i - \frac{1}{2}s_i$. Both order sizes are assumed nonnegative: $x_i \geq 0$ and $y_i \geq 0$. The associated execution cost is:

$$\left(P_i + \frac{1}{2}s_i + f^m + n\eta x_i\right)x_i + \left(P_i - \frac{1}{2}s_i - f^l\right)(y_i \wedge l_i). \tag{120}$$

Since the trader needs to complete the parent order with size X by the end of time T, only the market order is submitted at time t_{n-1}. In other words, we require:

$$\begin{aligned} y_{n-1} &= 0 && (121) \\ \sum_{i=0}^{n-1}(x_i + y_i \wedge l_i) &= X. && (122) \end{aligned}$$

4.2.3 Trading Objectives

Execution cost is often measured as the difference between the final average trade price and a pre-defined benchmark price[7]. The sign is taken such that positive cost represents of loss of value: buying for a higher price or selling for a lower price. Some common benchmarks are the arrival price, close price, volume weighted average price and time weighted average price. In this chapter we focus on the problem using arrival price as the benchmark.

For a buy order with fixed parent order size, the execution cost with arrival price benchmark can be equivalently defined as the difference between the final dollar values

[7] In this chapter, we exclude commissions, taxes and fees from the calculation of the final average trade price. This is due to the fact that these costs are direct and predictable, hence play less important roles in the quantitative analysis of execution quality

paid for the purchase and the notional value based on arrival price, which is called implementation shortfall (or slippage):

$$I := \sum_{i=0}^{n-1}\left[\Big(P_i + \frac{1}{2}s_i + f^m + n\eta x_i\Big)x_i + \Big(P_i - \frac{1}{2}s_i - f^l\Big)(y_i \wedge l_i)\right] - XP_0 \qquad (123)$$

Similar to Chapter 2, by introducing a variable to record the remaining unexecuted shares, we can rewrite (123). More specifically, define

$$z_0 = X, \quad , z_i = z_0 - \sum_{j=0}^{i-1}(x_j + y_j \wedge l_j)$$

then (123) is equivalent as

$$I = \sum_{i=0}^{n-1} c_i \qquad (124)$$

where

$$c_i = z_i\epsilon_i + \Big(\frac{1}{2}s_i + f^m + n\eta x_i\Big)x_i + \Big(-\frac{1}{2}s_i - f^l\Big)(y_i \wedge l_i) \quad \text{for} \quad i = 0, 1, ..., n-1 \quad (125)$$

and

$$\epsilon_0 := 0. \qquad (126)$$

where ϵ_0 is defined in (118). The detailed derivation from (123) to (124)-(126) is presented in Appendix C.1. A special case for (125) is the last period cost when $i = n-1$. At time t_{n-1}, we will submit all remaining shares as a market order:

$$x_{n-1} = -z_{n-1} \quad \text{and} \quad y_{n-1} = 0.$$

Therefore, the eligible liquidity l_{n-1} will never be referred in the last period's cost c_{n-1}:

$$c_{n-1} = z_{n-1}\epsilon_{n-1} - \Big(\frac{1}{2}s_{n-1} + f^m - n\eta z_{n-1}\Big)z_{n-1}$$

(123) captures the liquidity risk(sum of half spread and market impact) as well as execution risk. If the objective is just to minimize the average execution cost, the optimal approach is to trade as slowly as possible to minimize the liquidity risk. However, this exposes the order to a possible large adverse deviation from the arrival

benchmark price, particularly if the stock is volatile. As in [2] and Chapter 2, we introduce variance in the objective function to penalize timing risk induced by slow trading. Our goal is to minimize both the mean and variance of the slippage:

$$\mathbf{MV}(\kappa) : \min_{\Pi} \mathbb{E}I + \kappa \mathrm{Var}[I] \tag{127}$$

where $\kappa \geq 0$ is the risk aversion factor and Π stands for all feasible strategies:

$$\begin{aligned} \Pi = \quad \{ \quad & \pi = (x_i, y_i)_{i=0}^{n-1} | \forall i, x_i, y_i \text{ are random variables adapted to } \mathcal{F}_i; \\ & \pi \text{ satisfies (121) and (122)} \}. \end{aligned} \tag{128}$$

In (128) we introduced the definition of feasible strategies where $\mathcal{F}_i$ is the σ-field generated by all the information available up to time t_i. More specifically,

$$\mathcal{F}_i := \sigma(s_0, l_0, \epsilon_1, s_1, ..., l_{i-1}, \epsilon_i, s_i).$$

By definition, a feasible strategy is an adaptive strategy that adjusts trading decisions according to quickly-changing market conditions as well as the previous trading decisions.

For a finite time horizon problem such as $\mathbf{MV}(\kappa)$, dynamic programming is often the first choice of tools to be considered. However, a direct application of dynamic programming is not possible since the variance operator does not satisfy the smoothing property, i.e. $\forall 0 \leq s \leq t$, $\mathrm{Var}[\mathrm{Var}(\cdot|\mathcal{F}_t)|\mathcal{F}_s] \neq \mathrm{Var}(\cdot|\mathcal{F}_s)$. To address this problem, [31] achieves the Bellman backward induction through the decomposition of variance objective function by Law of Total Variance. Chapter 2 and Chapter 3 of this thesis take an alternative approach by solving a family of auxiliary linear-quadratic problems. According to Chapter 2, the latter approach is structurally simpler and numerically more efficient. Therefore, we will resort to an approach similar to Chapter 2 and Chapter 3 of this thesis.

Consider a family of linear quadratic problems for $r_0 \in \mathbf{R}$:

$$\mathbf{LQ}(r_0)\text{:} \ \min_{\Pi} \mathbb{E}[r_0 I + I^2]. \tag{129}$$

where Π is the same set of feasible strategies defined in (128). Similar to κ in (127), r_0 also plays the role of determining risk aversion level. However, they have reverse effects. Larger value of κ in $\mathbf{MV}(\kappa)$ implies higher risk aversion while larger value of r_0 in $\mathbf{LQ}(r_0)$ implies lower risk aversion tolerance.

Proposition 1 in Chapter 2 establishes the relationship between the solutions of mean-variance problems and linear quadratic problems:

Proposition 1. *Let*

$$\begin{aligned}
\Pi_{\boldsymbol{MV}}(\kappa) &= \{\pi|\pi \in \Pi \textit{ and } \pi \textit{ is a minimizer of } \boldsymbol{MV}(\kappa)\},\\
\Pi_{\boldsymbol{LQ}}(r_0) &= \{\pi|\pi \in \Pi \textit{ and } \pi \textit{ is a minimizer of } \boldsymbol{LQ}(r_0)\},\\
\Pi_{\boldsymbol{LQ}} &= \bigcup_{r_0 \in \mathbf{R}} \Pi_{\boldsymbol{LQ}}(r_0),
\end{aligned}$$

then $\Pi_{\boldsymbol{MV}}(\kappa) \subset \Pi_{\boldsymbol{LQ}}$. *More specifically, if* $\pi^*(\kappa) \in \Pi_{\boldsymbol{MV}}(\kappa)$*, then* $\pi^*(\kappa) \in \Pi_{\boldsymbol{LQ}}(r_0^*)$ *where*

$$r_0^* = \frac{1}{\kappa} - 2\mathbb{E}[I|_{\pi^*(\kappa)}]. \tag{130}$$

The proof of Proposition 1 can be found in Chapter 2. The equation (130) shows that by choosing the appropriate r_0^*, we can derive the optimal strategy of the $\mathbf{MV}(\kappa)$ by solving auxiliary problem $\mathbf{LQ}(r_0^*)$. However, according to (130), r_0^* depends on $\pi^*(\kappa) \in \Pi_{\mathbf{LQ}}(r_0^*)$, which would not be available until we know the value of r_0^*. Thus we have a "chicken and egg" dilemma.

Alternatively, Chapter 2 structured a framework that can solve $\mathbf{LQ}(r_0)$ for all $r_0 \in \mathbf{R}$ simultaneously by introducing r_0 as a state variable. Assume $\pi(r_0)$ solves $\mathbf{LQ}(r_0)$, the solution for $\mathbf{MV}(\kappa)$ is achieved by finding the appropriate r_0 that minimizes the mean-variance benchmark:

$$r_0^*(\kappa) = \arg\min_{r_0 \in \mathbf{R}} \{\mathbb{E}[I|_{\pi(r_0)}] + \kappa \mathrm{Var}[I|_{\pi(r_0)}]\}.$$

According to Proposition 1, $\pi(r_0^*(\kappa)) \in \Pi_{\mathbf{MV}}(\kappa)$ solves the original mean-variance problem. A detailed description of this two step process can be found in (3.8)-(3.10) in Chapter 2. Therefore, we will focus on solving the linear-quadratic problem $\mathbf{LQ}(r_0)$ from now on.

4.3 Optimal Adaptive Strategies

4.3.1 Bellman Equations

To solve $\mathbf{LQ}(r_0)$ through dynamic programming, we first decompose the objective function into a sum of cost functions. As in Chapter 2, we introduce the linear-quadratic weight r_0 as a state variable, which evolves following:

$$r_{i+1} = r_0 + 2\sum_{j=0}^{i} c_j \quad \text{for} \quad i = 0, 1, ..., n-2. \tag{131}$$

Note that $\frac{1}{2}(r_{i+1} - r_0)$ can be interpreted as a partial realization of the slippage I that is $\mathcal{F}_i$-measurable. In other words, it records information about how the trader performs up to time t_i.

Once r_i is introduced, the objective can be decomposed as:

$$\begin{aligned} & \mathbb{E}[r_0 I + I^2] \\ = \; & \mathbb{E}\Big[r_0 \sum_{i=0}^{n-1} c_i + \Big(\sum_{i=0}^{n-1} c_i\Big)^2 \Big| \mathcal{F}_0\Big] \\ = \; & \mathbb{E}\Big[\sum_{i=0}^{n-1} (r_i c_i + c_i^2) \Big| \mathcal{F}_0\Big] \\ = \; & \mathbb{E}\Big[\sum_{i=0}^{n-1} \mathbb{E}\Big[r_i c_i + c_i^2 \Big| \mathcal{F}_i\Big] \Big| \mathcal{F}_0\Big] \end{aligned}$$

and the cost function C_i is defined as:

$$C_i = \mathbb{E}\Big[r_i c_i + c_i^2 \Big| \mathcal{F}_i\Big] \tag{132}$$

where the conditional expectation is taken over future liquidity l_i.

Define the state variable $S_i := (r_i, z_i, \epsilon_i, s_i)$, which includes information about aggregated trading performance up to time t_i: r_i, remaining number of shares z_i, latest price change ϵ_i and current spread s_i. The cost (132) can be written as a function of state S_i and action (x_i, y_i):

$$\begin{aligned} C_i(S_i, x_i, y_i) &:= \mathbb{E}\Big[r_i c_i + c_i^2 \Big| \mathcal{F}_i\Big] \\ &= \mathbb{E}\Bigg[r_i\Big[z_i\epsilon_i + \Big(\frac{1}{2}s_i + f^m + n\eta x_i\Big)x_i + \Big(-\frac{1}{2}s_i - f^l\Big)(y_i \wedge l_i)\Big] \\ &\quad + \Big[z_i\epsilon_i + \Big(\frac{1}{2}s_i + f^m + n\eta x_i\Big)x_i + \Big(-\frac{1}{2}s_i - f^l\Big)(y_i \wedge l_i)\Big]^2 \Bigg| \mathcal{F}_i\Bigg] \end{aligned} \tag{133}$$

When $i = n-1$, the conditional expectation operator is not necessary for the previous period's cost since only the market order is submitted and future liquidity l_{n-1} does not come into effect:

$$\begin{aligned} C_{n-1}(S_{n-1}) &= r_{n-1}\Big[z_{n-1}\epsilon_{n-1} - \Big(\frac{1}{2}s_{n-1} + f^m - n\eta z_{n-1}\Big)z_{n-1}\Big] \\ &\quad + \Big[z_{n-1}\epsilon_{n-1} - \Big(\frac{1}{2}s_{n-1} + f^m - n\eta z_{n-1}\Big)z_{n-1}\Big]^2 \end{aligned}$$

Once the cost function is defined, the Bellman equations as well as the optimal trading decisions follow naturally:

For $i = n-1$:

$$J_{n-1} = C_{n-1}(S_{n-1}) \tag{134}$$

$$x^*_{n-1}(S_{n-1}) = -z_{n-1} \tag{135}$$

$$y^*_{n-1}(S_{n-1}) = 0. \tag{136}$$

For $i = n-2, ..., 0$:

$$J_i(S_i) = \min_{x_i, y_i}\Big(C_i(S_i, x_i, y_i) + \mathbb{E}\Big[J_{i+1}(S_{i+1})\Big|S_i\Big]\Big) \tag{137}$$

$$\Big(x_i^*(S_i), y_i^*(S_i)\Big) = \arg\min_{x_i, y_i}\Big(C_i(S_i, x_i, y_i) + \mathbb{E}\Big[J_{i+1}(S_{i+1})\Big|S_i\Big]\Big) \tag{138}$$

where the minimization is over two dimensional space $\{(x_i, y_i)|x_i \geq 0, y_i \geq 0, x_i + y_i \leq z_i\}$ to make sure we always submit buy orders and the sum of submitted order size does not exceed the remaining shares z_i.

4.4 *Numerical Tests*

In this section, we will first introduce a formulation based on scaled variables. Next we will describe how market variables can be calibrated from real market data and be used to solve Bellman equations through backward induction. Finally, we will compare the unified strategy with the other two benchmarks. In the following, we will use **ML** to refer to the "market-limit" strategy that submits both market and limit orders, **MO** to refer to the 'market only' strategy that submits only market order, and **TS** to refer to the "two steps" strategy that separates scheduling and optimal order placement and treats them sequentially. We will compare their relative performances under different order sizes, market impacts and risk-aversion settings. We will show that **ML** has consistent advantages over the other two strategies, particularly for illiquid stocks, less risk averse traders and large orders. We will also study the trading behavior of **ML** and try to understand how the market dynamic risks we built in the previous analysis get reflected in its trading decisions.

For illustration purpose, we will focus on a trading problem of executing certain Apple Inc (AAPL) shares within one hour. We allow a trading frequency of once per minute. In other words, $T = 1$ hour, $n = 60$ and each period is one minute. The tests based on other S&P 500 stocks convey similar results.

4.4.1 Scaling

For better analysis and interpretation of the test results, we scale the slippage by notional value, similar to the procedure in Chapter 2:

$$\tilde{I} := \frac{I}{XP_0}.$$

Accordingly, price related variables are scaled by arrival price P_0. This includes price P_i, price change ϵ_i, spread s_i, rebate for posting liquidity f^l and fee for taking liquidity

f^m for $i = 0, 1, ..., n$:

$$\tilde{P}_i = \frac{P_i}{P_0}, \quad , \tilde{\epsilon}_i = \frac{\epsilon_i}{P_0}, \quad , \tilde{s}_i = \frac{s_i}{P_0}, \quad , \tilde{f}^l = \frac{f^l}{P_0}, \quad , \tilde{f}^m = \frac{f^m}{P_0}. \tag{139}$$

Size related variables are scaled by the total order size X, including parent order size itself, market order size x_i, limit order size y_i, remaining shares z_i and eligible liquidity l_i:

$$\tilde{X} = 1, \quad , \tilde{x}_i = \frac{x_i}{X}, \quad , \tilde{y}_i = \frac{y_i}{X}, \quad , \tilde{z}_i = \frac{z_i}{X}, \quad , \tilde{l}_i = \frac{l_i}{X}. \tag{140}$$

Cost related variable r_i is scaled by total notional value:

$$\tilde{r}_i = \frac{r_i}{XP_0}$$

Market impact is scaled according to:

$$\tilde{\eta} = \frac{\eta X}{P_0}.$$

and risk aversion factor λ is scaled according to:

$$\tilde{\lambda} = \frac{\lambda}{(XP_0)^2}$$

Then the slippage (123) can be rewritten as

$$\tilde{I} = \sum_{i=0}^{n-1} \left[\left(\tilde{P}_i + \frac{1}{2}\tilde{s}_i + \tilde{f}^m + n\tilde{\eta}\tilde{x}_i \right) \tilde{x}_i + \left(\tilde{P}_i - \frac{1}{2}\tilde{s}_i - \tilde{f}^l \right) (\tilde{y}_i \wedge \tilde{l}_i) \right] - \tilde{X}\tilde{P}_0 \tag{141}$$

Note that (141) has exactly the same form as (123). Similarly, the mean-variance objective $\mathbf{MV}(\kappa)$ is equivalent to the same form:

$$\min_{\Pi} \mathbb{E}\tilde{I} + \tilde{\kappa}\text{Var}[\tilde{I}]$$

and linear-quadratic objective $\mathbf{LQ}(r_0)$ is equivalent to the same form:

$$\min_{\Pi} \mathbb{E}[\tilde{r}_0 \tilde{I} + \tilde{I}^2].$$

Therefore, all the analysis in Section 4.2 and Section 4.3 can be carried out in the exact same way and we can treat the results in these two sections as if they were based on scaled variables. To simplify notation, we will ignore the tilde symbol from now on. When we need to refer to unscaled variables, we will point it out specifically.

4.4.2 Calibration for Market Exogenous Variables

As mentioned in Section 4.4.7.2, we will use traditional backward induction to compute value functions. This requires the computation of expectations in (137). Hence we need to know the distribution of three market exogenous variables: price change $\epsilon_i (i = 1, 2, ..., n-1)$, spread $s_i (i = 0, 1, ..., n-1)$ and eligible liquidity $l_i (i = 0, 1, ..., n-2)$.

We are going to make some simplified assumptions about these market exogenous variables. The positive result from the out-of-sample test will show that these assumptions are accurate enough to reflect actual market dynamics. Specifically, we assume $\{\epsilon_i\}_{i=1}^{n-1}, \{s_i\}_{i=0}^{n-1}$ and $\{l_i\}_{i=0}^{n-2}$ are mutually independent series with no autocorrelation. ϵ_i follows a normal distribution with zero mean and standard deviation σ[8]. For most actively traded US stocks with small price levels, the spread normally just takes a few different values, such as zero cents, one cent, etc. If there are fewer than 10 different spread levels in the training data set, we will just use a discrete distribution to model the spread. Otherwise, we will use a zero-inflated Weibull distribution to model the spread.

The calibration for eligible liquidity requires special attention as it is not directly observable. Rather than following the exact definition described in Section 4.2.2.1, we will instead estimate only a portion of the true eligible liquidity that can be directly observed. The idea is that if we can show that **ML** can beat **MO** even with a smaller liquidity estimation, then **ML** will definitely be able to outperform **MO** under true market liquidity, which is larger than the conservative estimate.

Here is the conservative approach we used for estimating the eligible liquidity submitted at the bid price $P_i - \frac{1}{2}s_i$ at time t_i. Recall the two groups of liquidity

[8] Recall that our model allows s_i and l_i to be correlated, which fits market reality better. However, for training simplicity, we ignore their correlation and simply assume independence between s_i and l_i in this section.

in Section 4.2.2.1. We will only focus on the first group as the liquidity from the second group depends on the venue in which we execute our order as well as our queue position. More specifically, we collect tick-by-tick ask quote data[9] within time period $[t_i, t_{i+1})$. We aggregate the quote size at the best bid prices if the line satisfies all the following conditions: 1) ask price is below $P_i - \frac{1}{2}s_i$ (i.e. we only accept ask liquidity which is submitted at a lower price than our buy limit order); 2) ask price is no higher than previous ask prices (i.e. we don't count liquidity when ask price is updated upwards); 3) updated time is at least one second after t_i (due to latency, sell liquidity immediately appearing after t_i may not be eligible for our buy limit order). If the ask price has not changed from the previous line, we only include additional quote size from the previous line. Although a similar procedure can be carried out for tick-by-tick trade data, we intentionally exclude them because historical trades happen when our limit buy order does not exist. Since our limit buy order may prevent potential sell limit orders crossing the spread (i.e. market impact from limit order), it may be optimistic to include historical trade data that happened below $P_i - \frac{1}{2}s_i$ when there is no limit buy order sitting there. This concern is much less significant for an estimation from quote data.

We can carry out this conservative approach for both the training and testing data sets for both the ask quote and bid quote data (the bid quote data is used for selling execution problems). It appears that eligible liquidity frequently takes zero values (i.e. there is no liquidity on one direction over a period of time), so we will use a zero-inflated Weibull distribution to fit the training data set. The reason we use a Weibull distribution is because it can model non-zero data with heavy tails.

In actual practice, the conservative estimation may make **ML** underachieve its true potential at the beginning. Based on the feedback from actual limit order fills,

[9]Tick-by-tick ask quote data includes all the updates for ask price and/or its associated quote size. The database has three columns: updated time, ask price, and quote size at ask price. If either of these two values changes, a new line will be generated in the database

the conservative estimation can be adjusted higher from time to time to reflect the true liquidity distribution. As we collect more and more actual limit order fills, these data can be used directly for estimating liquidity through censored maximum likelihood method (see [27] for censored MLE). At this stage, the conservative estimation will no longer be used.

In this section, we will use tick-by-tick quote data for Apple Inc (AAPL) from 02/01/2012- 12/31/2012. We exclude the first and last 30 minutes' data as they often include too much noise. Data from 02/01/2012-06/30/2012 will be used for calibration and the rest will be used for testing. Note from (140) that scaled liquidity $\tilde{l}_i$ depends on parent order size. To illustrate for now, we will assume the parent order size is 10% of the average one hour trade volume. The average one hour trading volume for AAPL during 02/01/2012 -06/30/2012 is 2594420, hence the assumed parent order size is 259442. The calibration for other parent order sizes can be easily derived by adjusting the scale parameter below. Table 14 presents the parametric distribution for aggregated one minute frequency data based on the training set:

Table 14: Parametric distribution of market exogenous variables

r.v.	distribution
e_i	normal with mean 0 and stdev 7.198bps;
s_i	with prob 0.000552 to take zero value, else Weibull with scale 1.864bps and shape 0.122;
l_i	with prob 0.357 to take zero value, else Weibull with scale 0.0423 and shape 0.760.

There are three other market variables that need to be pre-specified: the rebate for posting liquidity, f^l; the fee for taking liquidity, f^m; and the market impact, η. In this test, we assume that unscaled f^l = \$0.003/share and unscaled f^m = \$0.003/share, which are close to AAPL's primary listed exchange(NASDAQ)'s fee structure. Rather than scaling f^m and f^l by arrival price P_0, as in (139), which may change from trade to trade, we will scale them by average price \$563.11. Therefore, the scaled $f^l = f^m = 0.0533$bps, which is a small number compared with the spread or market impact. In other words, the rebate/fee has little impact on trading AAPL.

For the market impact, we use a similar assumption as in [2] and Chapter 2 where the (scaled) market impact is proportional to the order's percentage of the market volume (POV)

$$\begin{aligned} \eta &= \text{market impact scale} \times \frac{\text{unscaled order size}}{\text{average trading volume within } T} \\ &= \text{market impact scale} \times \text{POV} \end{aligned} \tag{142}$$

where the market impact scale(mis) is pre-specified. POV represents the relative order size with respect to market volume. Larger orders have a higher value of market impact η, and hence are harder to trade. For example, if mis = 70bps, and POV=10%, $\eta = 7$ bps, which means if you immediately execute an order worth of 10% of market average volume, you will incur an immediate market impact of 7 bps. In the following tests, we will consider various scenarios with different combinations of market impact scale and POV.

4.4.3 Backward Induction

We use a similar numerical procedures that was used in Chapter 2 to solve Bellman equations (134)-(138). It includes the following few steps. More details are available in Chapter 2.

First, for each state variable, we need to specify an interval within which it is likely to lie. For z_i, this will be $[0, 1]$. For e_i, we will use $[-3\sigma, 3\sigma]$. For s_i, it should be $[0, 99\%$ quantile$]$. Since $r_n = r_0 + 2I$, r_n is proportional to slippage value when $r_0 = 0$. Therefore, the range of r_i values should cover two times of most likely observed slippage values. To achieve this, we test an equal split strategy with only market order (i.e. $x_i = \frac{1}{n}, y_i = 0$ for $i = 0, 1, ..., n-1$) over the training data. It turns out the largest absolute slippage is 56.252bps. We multiply it by four to set the range for the values of r_i: [-225.008,225.008].

Second, we discretize each state variable within its respective interval. For example, for $z_i \in [0, 1]$, we discretize it with 10 equally spaced values: $(0, 0.1, 0.2, ..., 1)$. We carry out the same procedures for r_i, s_i, e_i, all with resolution 10. Hence for $i = 0, 1, ..., 59$, state space S_i is represented by $11 \times 11 \times 11 \times 11$ grid points and value function $V_i(S_i)$ is represented by a table of 11^4 values. For any state S_i that does not fall onto the grid points, its value $V_i(S_i)$ is estimated through spline interpolation(see [17]).

Third, for $i = n - 1, n - 2, ..., 0$, value functions $V_i(\cdot)$ for 11^4 grid points are computed following (134) and (137). The expectation is computed using Gaussian Legendre quadrature, following the same approach used in Chapter 2. In the training process, the discretized actions (x_i, y_i) share the same grid points with the ones used for z_i, i.e. (0,0.1,0.2,...,1). The optimal decision (x_i^*, y_i^*) is found by enumerating all possible grid points within the action space $\{(x_i, y_i)|x_i \geq 0, y_i \geq 0, x_i + y_i \leq z_i\}$.

Fourth, once all value functions V_i are known, (135),(136) and (138) can be used to derive the optimal trading actions. However, in order to increase action flexility, we increase the resolution of (x_i, y_i) and make it adjustable to z_i. Specifically, given remaining shares z_i, both x_i and y_i will use grid points with resolution 20: $(0, \frac{z_i}{20}, 2\frac{z_i}{20}, ..., z_i)$ instead of (0,0.1,0.2,...,1).

4.4.4 Testing

In this section, we will present the result of a trade simulation based on AAPL's data from 07/01/2012-12/31/2012. We extract 10 trading tests from each date. These are buy and sell execution orders with 5 trading horizons: 10:00am-11:00am, 11:00am-12:00pm,...,14:00pm- 15:00pm. We have complete and valid data for 121 days between 07/01/2012 and 12/31/2012, which means we have 1210 test samples.

Three pre-specified parameters will impact a strategy's performance: market impact scale (mis), order size (relative to market volume) and risk aversion level. We will carry out our test analysis based on different combinations of these three values. Specifically, we choose two mis for experimentation: 50 bps and 100 bps. The majority of U.S. stocks should have mis falling into this bucket. For order size, we consider four different possibilities: 5%,10 according to (142), larger orders have bigger market impact factor η. Second, larger orders will have a smaller percentage of limit orders that can be filled by fixed amount of eligible liquidity. In other words, it is harder to fill a larger order, regardless of if it is executed through market orders or limit orders. Third, the risk aversion level can be adjusted by assigning different r_0 values. Large r_0 values imply less risk aversion. We test trading strategies for $r_0 \in (-75, -50, -25, ..., 150, 175)$. Note that unlike mis and order size, changes in the value of r_0 normally does not require new training. This is because, as mentioned previously, (134)-(138) is computed for $r_0 \in [-225.008, 225.008]$. In summary, we will train 8 models, which include all combination of mis$\in$ (50bps, 100bps) and POV$\in$ (5%, 10%, 20%, 30%). Each model will be tested given 11 different values of $r_0 \in (-75, -50, -25, ..., 150, 175)$.

Before we compare the performances of the three strategies, we will give a brief description of **MO** and **TS**. The **MO** strategy is based on Chapter 2, with the same market impact assumptions. We can arrive at that strategy in our model by setting $l_i \equiv 0$, $s_i \equiv 0$ for $i = 0, 1, ..., n-1$ and $f^m = f^l = 0$. Under these assumptions, since limit orders will never be filled, the optimal **ML** strategy will always submit market orders: $x_i^{MO} \geq 0, y_i^{MO} = 0$ for $i = 0, 1, ..., n-1$. By ignoring spread and rebates/fees, we essentially focus only on the trade-offs between market impact and timing risk. Once the solution of scheduling problem **MO** is computed, its associated **TS** can be easily derived. At time t_i, **MO** computes total shares x_i^{MO} it targets to execute within $[t_i, t_{i+1})$. **TS** will submit all x_i^{MO} shares as limit order: $y_i^{TS} = x_i^{MO}$. Whatever is left unexecuted at $t_{i+1}-$ will be submitted as a market order of size $(y_{i-1}TS - l_i) \wedge 0$. Since actual price is a continuous process: $P_{t_{i+1}-} = P_{t_{i+!}}$, submitting the market order at $t_{i+1}-$ is equivalent to submitting it at t_{i+1}. In summary, **TS** follows the rules:

1. For $i = 0$: $x_i^{TS} = 0$, $y_i^{TS} = x_i^{MO}$;

2. For $i = 1, 2, ..., n-2$: $x_i^{TS} = (y_{i-1}TS - l_i) \wedge 0$, $y_i^{TS} = x_i^{MO}$;

3. For $i = n-1$: $x_i^{TS} = x_i^{MO}$.

In the following, we will start with a specific execution order and understand trading behaviors for **ML**, **MO** and **TS**. Then we will present the aggregated comparison result for all 1210 test samples. In the end, we will focus on **ML** and analyse its trading behavior under different market dynamics.

In terms of computation time, it takes around three days to run the test with aforementioned resolution of 11×11×11×11 using a computer cluster with 30 CPUs. The majority of computation time will be spent in the training process. Once value functions are computed, decisions can be computed much more quickly.

4.4.4.1 One Test Sample

Assume two execution orders of trading 259442 shares (POV=5%) of AAPL from 10:00am to 11:00am on 09/11/2012. One execution order is for buying, the other for selling. The market impact scale is set as 50 bps and the risk averse level is determined by setting $r_0 = 125$. Figure 13 plots the trading paths for the three strategies.

The five plots on the left focus on **ML**, while the one on the right focus on **MO** and **TS**. The first left plot depicts mid quote P_i and spread s_i for $i = 0, 1, ..., 59$. The stock price movement for this particular sample has a V shape with the lowest price achieved around the 21st minute. The second plot depicts the eligible liquidity for both a buy limit order and a sell limit order . Note that the eligible liquidity often takes zero values and eligible liquidity for buy and sell orders are negatively correlated. For example, the liquidity for the sell order between $i = 2$ and $i = 7$ is almost zero while the liquidity for the buy order is much larger. This suggests that

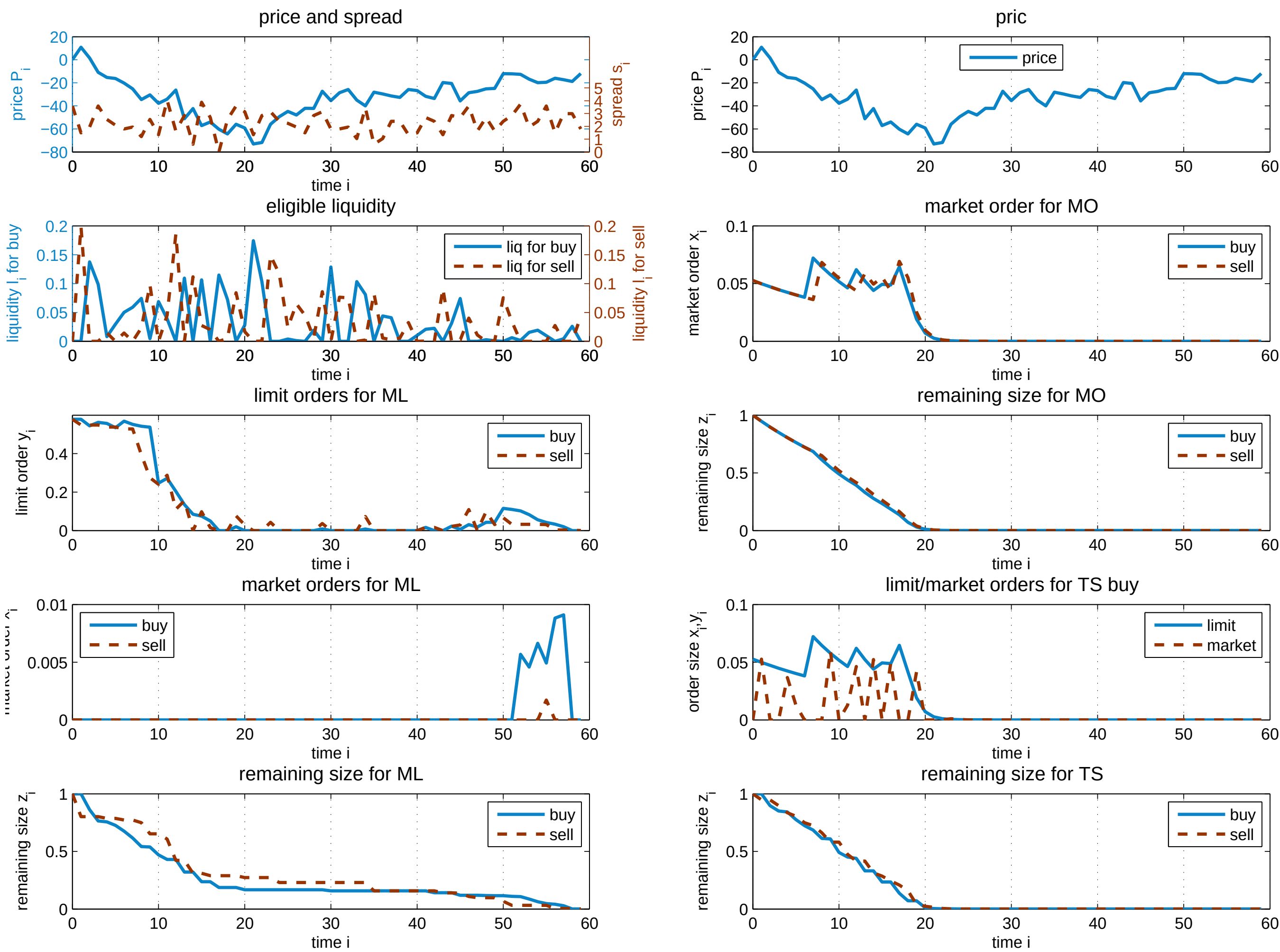

Figure 13: Trading paths for sample orders with mis=50 bps, POV=5% and $r_0 = 125$

there is an order flow imbalance with selling activities outweighs buying activities, which is consistent with the price drop observed in the first left plot during $[t_2, t_7]$. The third and fourth plots depict the limit orders y_i^{ML} and market orders x_i^{ML} submitted during this hour. The fifth left plot records the number of remaining shares z_i^{ML}. The first plot on the right is a copy of the price movement that appears in the first left plot. We do not plot the spread as it is not used in the decision making of **MO** and **TS**. The second and third plots on the right depict market orders x_i^{MO} and remaining shares z_i^{MO} for **MO**. The fourth plot on the right depicts both market and limit orders x_i^{TS}, y_i^{TS} for only the buy order using **TS** while the fifth right plot records the remaining shares z_i^{TS} for both buy and sell orders. We omit depicting the market/limit orders for the sell order using **TS** in the fourth right plot as they are similar to buy order case.

Figure 13 reveals a few key properties of **ML** and its difference from the other two strategies.

First, all three strategies are front loaded, such that there is more trading volume in the beginning. This is a natural result of the arrival price benchmark. However, **ML** trades more slowly than the other two strategies; while most of the orders done with **MO** and **TS** are complete by t_{20}, **ML** has about a quarter of the total shares left at time t_{20}. By trading slower, **ML** is able to take advantage of potential future limit order fills at a better price level. This is clearly illustrated by looking at limit orders submission y_i^{ML} during $[t_{18}, t_{50}]$, where limit orders are submitted whenever spread s_i is large, particularly for the sell order. However, **ML** needs to balance this benefit with the timing risk of slow trading and the execution risk associated with leaving a large unexecuted quantity till the end. This balance is reflected in the increased limit order submission y_i^{ML} after t_{45}.

Second, unlike **MO** and **TS**, for the relatively small order (POV=5%), market orders are playing only complementary roles. Market orders x_i^{ML} are not submitted until the last ten minutes, and only in small quantities. Limit orders are used predominantly to execute majority of the order. However, this is not the case for larger orders as they are constrained by the amount of eligible liquidity l_i available to fill the limit orders. Instead larger orders need to rely on market orders.

Third, all three strategies are "aggressive-in-the-money"(AIM), which means that whenever the trader outperforms (for a buy problem, it means small slippage), the adaptive strategy tends to trade more aggressively to reduce future risk[10]. This can be illustrated by comparing buy and sell orders. Note that the stock price is on a downward trend until t_{21}. When the benchmark is the arrival price, a buy order will outperform a sell order in a downward price trend. This results in faster trading for buy orders than sell orders. For example, in the first 20 minutes, larger limit orders y_i^{ML} are submitted in **ML** and larger market orders are submitted in x_i^{MO} for buy orders than sell orders.

Table 15: Trading slippages for sample orders with mis=50 bps, POV=5% and $r_0 = 125$

strategy	buy	sell
ML	-20.766	27.723
MO	-14.618	49.070
TS	-24.232	43.784

The actual slippage performance for this sample is listed in Table 15. Comparing the average slippage of buy and sell orders, **ML** outperforms **MO**, and **MO** outperforms **TS**. Actually, this is also the case for the risk-adjusted cost for aggregated tests, as shown in the following section.

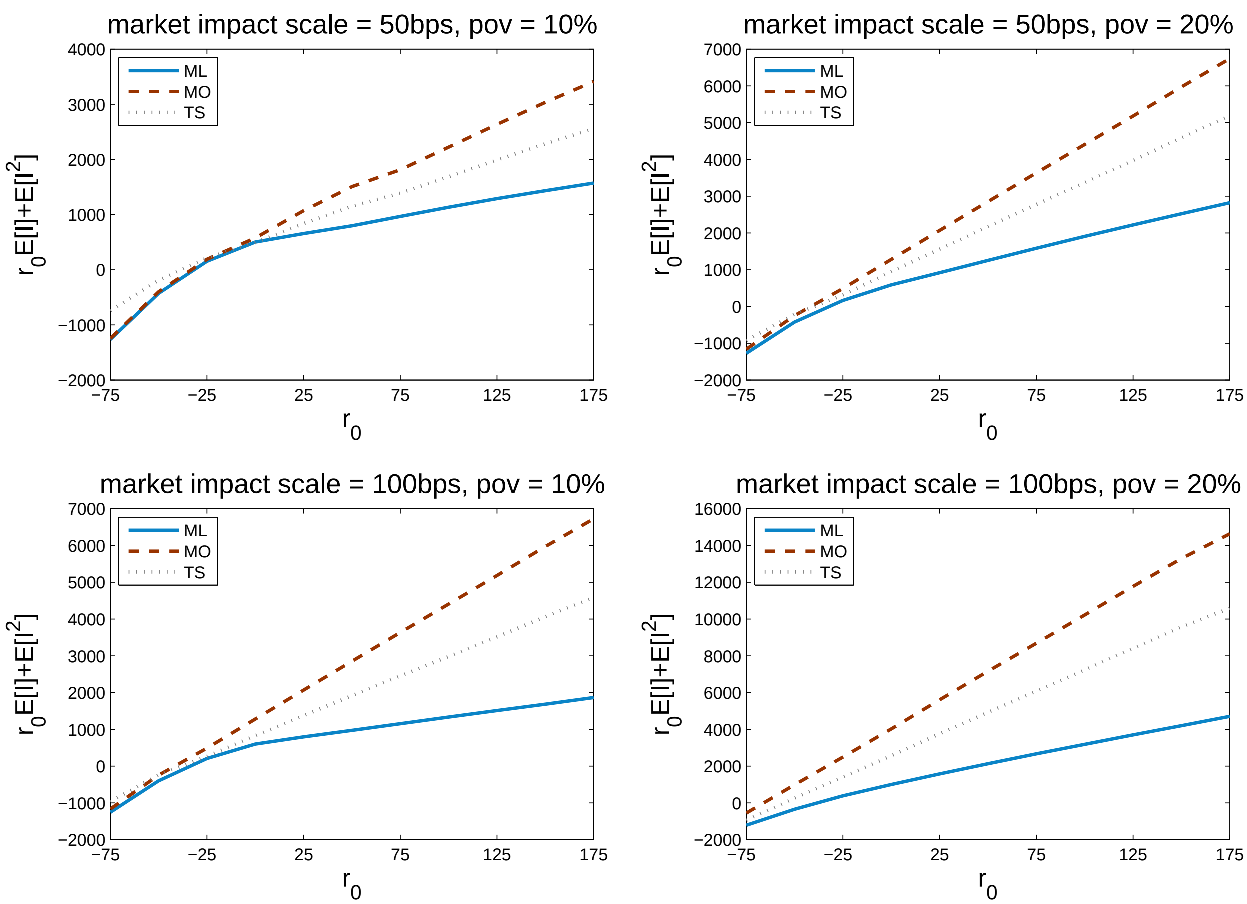

Figure 14: Sample objective values $r_0\mathbb{E}[I] + \mathbb{E}[I^2]$

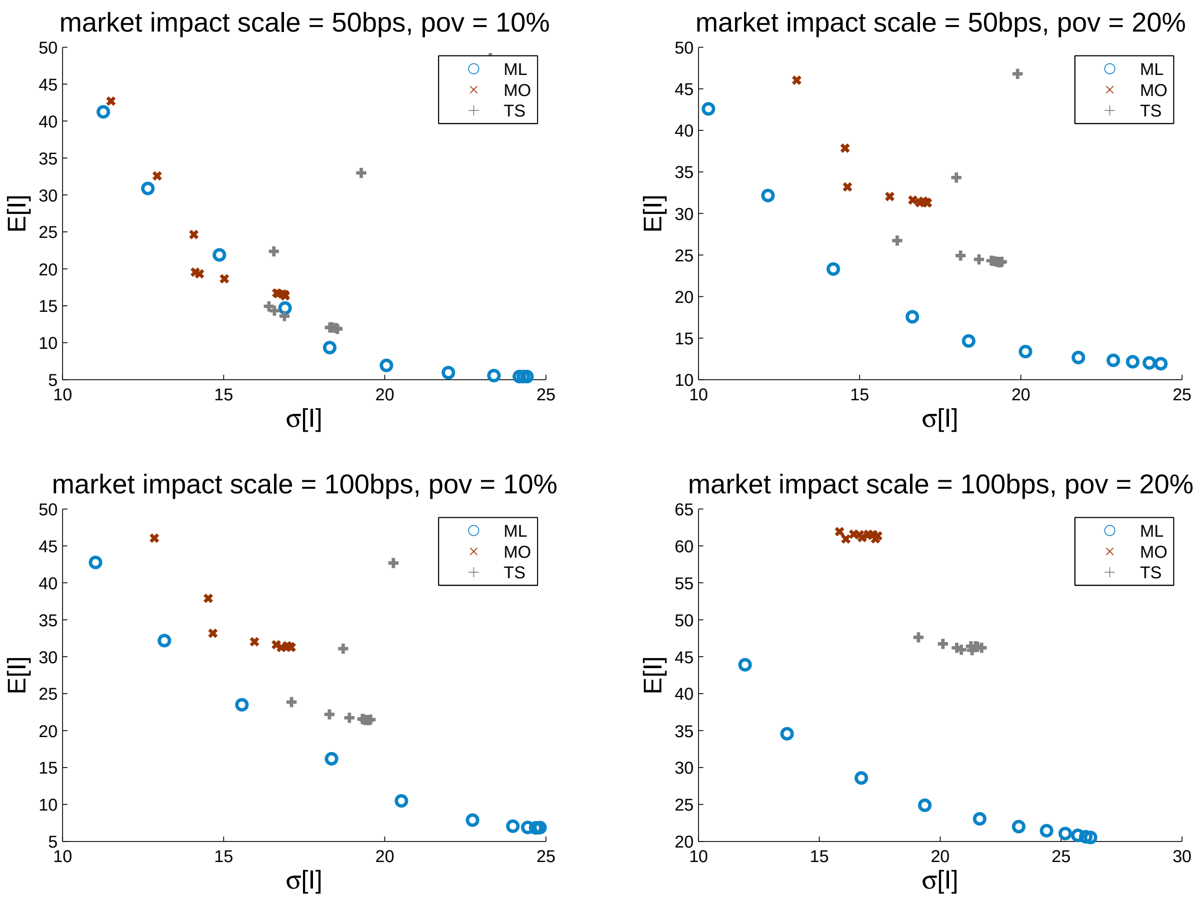

Figure 15: Pseudo efficient frontier

4.4.5 Aggregated Performance

Figure 14 and Figure 15 present the aggregated performance of **ML**, **MO** and **TS** based on 1210 samples from 07/01/2013-12/31/2013. We choose two market impact scales, i.e. 50 bps and 100 bps and two POV rates, i.e. 10% and 20% for presentation purposes. Figure 14 presents the sample linear-quadratic cost $r_0\mathbb{E}[I] + \mathbb{E}[I^2]$ for different values of $r_0 \in (-75, -50, ..., 175)$. Figure 15 presents the same result through a scatter plot to highlight the mean-risk (i.e. $(\text{Stdev}[I(r_0)], \mathbb{E}[I(r_0)])$) trade-offs.

Both figures suggest that **ML** consistently outperforms the other two strategies. Theoretically, if our model can accurately describes market dynamics(such as the distribution of eligible liquidity), **ML** should always outperform **MO** and **TS** as the action spaces for the latter two strategies are subsets of the action space of **ML**. However, in practice, certain assumptions have to be made to simplify modeling. The potential of **ML** may be greatly undermined when our assumption significantly deviates from market reality, which often leads to poor performance in out-of-sample test based on real market data. This is particularly problematic if market dynamics is not stationary and goes through regime-switching process between training period and testing period. The fact that **ML** outperforms **MO** consistently even under conservative eligible liquidity assumptions suggests the potential advantage of **ML** is significant enough to overcome any shortcomings in accurate representation of market dynamics.

The advantage of **ML** over **MO** and **TS** is more significant for large orders (POV), and under the assumption of a large market impact scale. When mis is large, **ML** can save execution costs by submitting more limit orders to save on market impact. From the perspective of order size, large POV reflects increased trading difficulty in both

[10] This can be interpreted through the value of r_i: a small slippage by time t_i implies a small value of r_i. Since r_i plays the reverse role of risk aversion factor, a small r_i implies more risk taking for future trading. Details of this analysis can be found in Chapter 2.

market and limit orders. Specifically, when POV is large, market impact factor *eta* will be increased, which increases the difficulty of trading through market orders. At the same time, the proportion of shares filled through limit orders will be decreased, which increases the difficulty of trading through limit orders. The former factor will prompt **ML** to use fewer market orders while the latter factor will result in fewer limit orders submission as well. The fact that **ML** has a larger advantage with larger POV orders suggests that the increment in market impact factor plays a more dominant role than the decrease in the percentage of limit order fills.

4.4.6 Trading Patterns of ML

In this section, we will focus on **ML** and study its submission of market and limit orders under different market settings, including order size, risk aversion level and market impact scale. We achieve this by plotting the average trading curves of $\{y_i^{ML}\}_{i=0}^{59}$, $\{x_i^{ML}\}_{i=0}^{59}$ and $\{z_i^{ML}\}_{i=0}^{59}$ over 1210 sample paths. The result is presented in Figure 16.

We use the following parameters as basic settings: $r_0 = 50$ bps, market impact scale=50 bps and POV=30%. Next we adjust one variable (either r_0, mis or POV) at a time while fixing the other two, and compare the resulting average trading paths with the average trading paths in the basic setting. Specifically, the three left plots in Figure 16 illustrate a comparison between different risk aversion levels: $r_0 = 50$ vs $r_0 = 150$. The three middle plots compare different mis assumptions: mis=50 bps vs. mis=100 bps. The three right plots compare a large order(POV=30%) vs a small order(POV=5%). The average paths for the basic setting are represented by blue lines in all three groups of plots.

Let's start with the three left plots. When r_0 increases from $r_0 = 50$ to $r_0 = 150$, **ML**(150) becomes less risk averse and trades slower (see the third left plot for z_i).

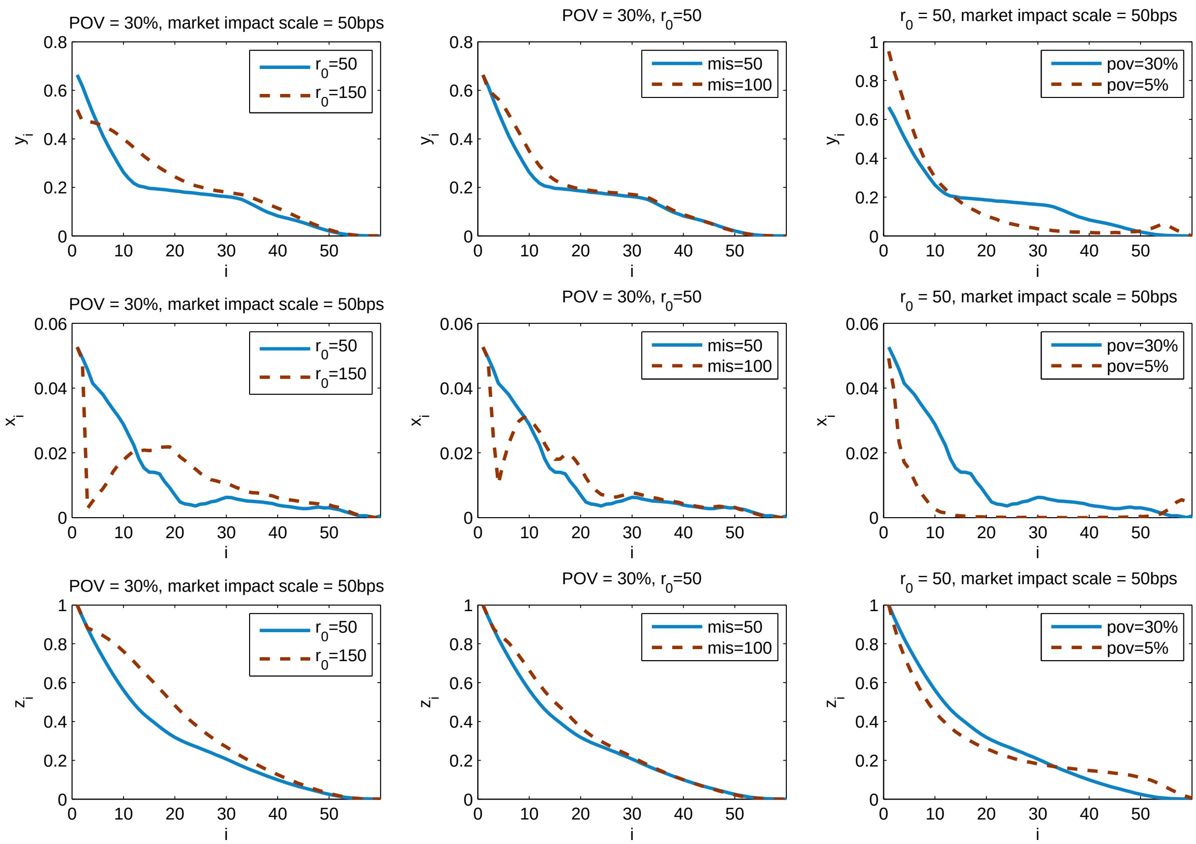

Figure 16: Average trading paths

This is because in the beginning, it submits smaller limit orders, while maintaining the same size of market orders as in the $r_0 = 50$ case. Once **ML**(150) realize submitting limit order can do a good enough job (i.e. $z_1(150) = z_1(50)$), **ML**(150) immediately switches to a more gradual approach by significantly cutting down on market order submission and focusing more on submitting limit orders. Until t_{11}, **ML**(150) submits more limit orders than **ML**(50), but fewer market orders. The coordination allows **ML**(150) to trade slower than **ML**(50), which allows it more flexibility to take advantage of future price and spread fluctuations.

Now, lets look at the middle three plots, where mis increases from 50 bps to 100 bps. We can interpret mis=50bps as the setting for a liquid stock while mis=100bps is the setting for an illiqid stock. Increasing market impact has similar effects as increasing r_0. **ML** trades illiquid stocks slower with more focus on limit orders to save market impact, particularly in the beginning. The submission of market order speeds up only around $t_1 0$.

For the three right plots, the order size is decreased from POV=30% to POV=5%. Due to increased fill probability for limit orders, **ML** posts around 95% of the small order as limit order at t_0. The majority of the small order is executed through limit orders. Similarly to the trading example we analyzed in Section 4.4.4.1, market orders play only a complimentary role in this case, which are not used between t_{15} and t_{52}. Interestingly, the trading speed for the small order slows down significantly between t_{30} and t_{50} compared with the speed for the large order (see the third right plot). As mentioned in Section 4.4.4.1, this is because **ML** would like to take advantage of potential price and spread functions for future trading. Therefore, it slows down its trading in the middle of the hour and only picks up activity at the end. This flexibility of leaving a proportion of shares till the end is a luxury only small orders can enjoy. This is because small orders can fall back on the fact that there is a plentiful supply of liquidity that can be used to fill limit orders at the end of the hour without resorting to market orders to complete the parent order.

These analyses of the adaptive trading patterns under different market dynamics reveal the sophistication of the optimal strategies that combine market and limit orders together. This complexity is achieved by recognizing the conflicts between different market dynamic risks, and modeling them in a dynamic programming framework that optimize the conflict balance in a systematic way.

4.4.7 Extension

4.4.7.1 Framework Extension

One of the major advantages of our model is its flexibility to be extended to fit various assumptions or preferences.

First, rather than using the arrival price to measure execution cost, we can use other price benchmarks, such as Volume Weighted Average Price (VWAP), Time Weighted Average Price (TWAP) and Close Price, etc. For example, Chapter 3 of this thesis uses a similar framework to solve a scheduling problem with VWAP benchmarks. It can be combined with this chapter to provide a VWAP trading algorithm that submits both market and limit orders.

Second, in this chapter, we only submit limit orders at one price level, which is the top of the limit order book on the bid side. In practice, optimal order placement models (OOPM) often seek liquidity among multiple levels of the order book. Orders submitted on top of the book often have better fill probability while orders submitted deeper in the book (i.e. at price levels lower than the bid price) have better price improvement and less adverse selection effects from trading against informed market participants from the other side. OOPM's main job is to find the optimal balance between the execution risk and the price improvement among different limit prices. Our framework can be easily extended to problems where limit buy orders can be submitted at multiple levels at or below the bid price, and it achieves the same trade-offs OOPM tries to balance. In this case, the eligible liquidity vector $\vec{l_i} = (l_i^0, l_i^1, ..., l_i^m)$

can be defined similarly for $m+1$ different price levels. For example, l_i^0 is the eligible sell liquidity submitted at or below $P_i - \frac{1}{2}s_i$, which is the same as fore-mentioned l_i. l_i^1 is the sell liquidity submitted at or below the next price level right below $P_i - \frac{1}{2}s_i$,etc. By construction, $l_i^0 \geq l_i^1 \geq \cdots \geq l_i^m$, which captures the execution risk increment as we move deeper into the order book.

Third, a lot of market dynamic risks in this chapter are modeled using simple assumptions. However, the model we derived can be easily extended to incorporate more sophisticated assumptions that better fit market reality. For example, the linear temporary market impact can be replaced by nonlinear market impact models. One such nonlinear model can be found in appendix of Chapter 2 where market impact consists of permanent market impact and instantaneous impact with exponential decay. Impact model (119) can be considered its special case where the permanent impact is zero and instantaneous impact decays immediately. As illustrated in Chapter 2, the addition of permanent impact does not add any extra computation complexity to the problem. However, modeling exponential decay of instantaneous impact requires adding an extra dimension to the state variable which captures the cumulative decayed instantaneous impact from all previous trading activities.

Another example of the ways in which our model could be made more sophisticated is to relax the assumption that eligible liquidities $l_i (i = 0, 1, ..., n-1)$ are independent. In fact, they are often positively autocorrelated due to short term elasticity of order flow imbalance. To include the autocorrelation of l_i, we just need to add previous eligible liquidity levels $l_{i-1}, l_{i-2}, ...$ into the state space.

4.4.7.2 Training for Value Functions

Once the Bellman equations (134)-(138) are derived, the optimal trading decisions $(x_i^*(S_i), y_i^*(S_i))$ are dictated by value functions $J_i(S_i)$ for $i = 0, 1, ..., n-1$. There are two ways to solve value functions. One way, which is the one we used in the numerical section, is to use backward induction. Under the this approach, J_{n-1} of the last period will be computed first according to (134), and then use (137) recursively to compute J_{n-2},J_{n-3},...,J_0. This is a conceptually easy and robust approach, and it is guaranteed to be optimal. However, it suffers the curse of dimensionality for high dimensional state space or action space. In our problem, the state space is four dimensions while the action space is two dimensions. Both of them can be tackled through backward induction a under decent discretization resolution.

Another approach, which has gained more popularity over the past two decades, is through forward prorogation. This is often used in the approximate dynamic programming (ADP)/reinforcment learning framework(RL)(see [42],[35],[7]). In ADP/RL, value functions $J_i(S_i)$ for all $i = 0, 1, ..., n-1$ are first given an initial approximation. Once the sample exogenous data (such as price e_0, spread s_0 and eligible liquidity l_{-1} in our problem) is fed into the system for the initial period $t = t_0$, suboptimal decisions (x_0, y_0) are computed through (138), based on the approximated value functions, which leads to new exogenous information flow (e_1, s_1, l_0) and new decision making (x_1,y_1) for the next period. The process of observing new information flow and then making decisions will be repeated until time t_{n-1} when the last trade is decided. The sample observed costs (133) will be collected at the end of the trading horizon T to update value function approximations through temporal difference method(TD(λ)). Under certain assumptions (such as each state is visited infinitely often), the approximated value functions will converge to the true value functions as the number of training samples goes to infinity. The curse of dimensionality is solved as the values functions are often approximated through some parametric forms. An update to a state's value often leads to an update of the values of many more other states. This benefit is of particular importance if we want to model a more sophisticated trading problem, such as submitting limit orders at multiple price levels, which will result in higher dimensions of state and action spaces.

Besides significant computation benefits, ADP/RL does not require an explicit description of the statistical properties of the exogenous information. The actual market data is used to train the model. Unlike statistical learning, the output of the training in ADP/RL is not a descriptive inference of the market variable distribution, but direct trading decisions. This allows the ADP/RL framework to adjust naturally when the distribution of exogenous variables changes over time. The fact that we do not need to estimate the distribution of exogenous variables is particularly important for our problem as the eligible liquidity is not easily observable. Under the ADP/RL framework, the fill result of submitted limit orders will be used implicitly to update the fill liquidity levels.

Given so many advantages of ADP/RL, one of the biggest difficulties in applying this technique is finding an appropriate parametric form to approximate value functions. This is often decided on a case by case basis as there are no universal parametric forms that apply to all problems. We will leave this part for future research. Unlike in the computer science community where state variables and cost functions are constructed heuristically based on expected behavior(see [43]), the operations research community believes that rigorously defined state variable and cost functions are important for robust performance in ADP problems(see [35]). The risk diversification framework we built so far, including the state variable, cost function and Bellman equations provides an analytical context where future researchers can begin.

4.5 Conclusion

This chapter challenges the common practice in the algorithmic trading industry that separates trade scheduling and optimal order placement model(OOPM). Trade scheduling is about finding an optimal balance between liquidity risk and timing risk, while the optimal order placement model is about the balance between liquidity risk and execution risk. Instead of treating these conflicts separately, we designed a unified approach by making sequential trading decisions through both market and limit orders. The unified approach is capable of sophisticated trading patterns adaptable to different market dynamics. Out-of-sample numerical tests show that the unified approach can consistently outperform the strategy that only submits market orders or the strategy that treats scheduling and order placement separately.

One of the major contributions of this chapter is to present an adaptive discrete time framework that can be used to facilitate optimal trading decisions in a mean-variance framework. We start by recognizing three major market dynamics risks, then use simplified assumptions to model these characteristics, and eventually integrate them into a dynamic programming problem. The real market dynamics is much more sophisticated than what we assume here. However, the power of this framework is its convenience to extend to various more complicated market assumptions.

Using this framework, the model can be easily extended by including multiple levels of limit order submission, more sophisticated market impact models (for both market and limit orders) and autocorrelation of eligible liquidity flow etc. The framework can be used to solve optimal trading problems with other price benchmark, such as volume weighted average price(VWAP).

This chapter, or the framework itself, can also be used as a starting point to build adaptive dynamic programming/reinforcement learning algorithms that can solve trading problems with higher dimensions. In a world where trading is increasingly implemented in a systematic way, the ADP/RL approach shall be an important direction for future research.

APPENDIX A

APPENDIX FOR CHAPTER II

A.1 Deterministic strategy

Let the deterministic strategy $\hat{\pi} = (\hat{y}_0, \hat{y}_1, ..., \hat{y}_{N-1})$ which solves $\mathbf{D}(\kappa)$. $\mathbf{D}(\kappa)$ can be simplified as a convex problem with convex constraint:

$$\min_{\tilde{\Pi}} \sum_{i=0}^{N-1} y_i^2 + \alpha \sum_{i=1}^{N-1} x_i^2$$

where $\alpha > 0$ and $\tilde{\Pi}$ is defined as in (20).

The optimal solution $(\hat{y}_0, \hat{y}_1, ..., \hat{y}_{N-1})$ satisfies

$$\begin{aligned}
&\hat{y}_0 = \beta, \\
&\hat{y}_1 + \alpha \hat{x}_1 = \beta, \\
&\hat{y}_2 + \alpha(\hat{x}_1 + \hat{X}_2) = \beta, \\
&\vdots \\
&\hat{y}_{N-1} + \alpha(\hat{x}_1 + \cdots + \hat{x}_{N-1}) = \beta,
\end{aligned}$$

or equivalently,

$$\hat{y}_i + \alpha \sum_{j=1}^{i} \hat{x}_j = \beta, \ i = 0, 1, ..., N-1, \tag{143}$$

where 2β is the Lagrange multiplier. From (143), one has

$$\hat{y}_i = \hat{y}_{i+1} + \alpha \sum_{j=i+1}^{N-1} \hat{y}_j, \ i = 0, 1, ..., N-2,$$

which can be solved recursively. For example, by setting $y_{N-1} = 1$, one has

$$y_{N-2} = \alpha + 1, \qquad y_{N-3} = \alpha^2 + 3\alpha + 1,$$
$$y_{N-4} = \alpha^3 + 5\alpha^2 + 6\alpha + 1, \qquad y_{N-5} = \alpha^4 + 7\alpha^3 + 15\alpha^2 + 10\alpha + 1$$

Then all resulting $y_i, i = 0, 1, ..., N-1$ are scaled through $\hat{y}_i = \frac{1}{\sum_{i=0}^{N-1} y_i} y_i$.

A.2 Extension to resilient market impact

The main model presented in this chapter is based on the simple assumption of temporary market impact, where the increased price due to our trading is assumed to recover immediately before the next time period. However, in reality, it takes time for the new orders to come in to fill up the order book gap again. Furthermore, as pointed in Obizhaeva and Wang[2006], the above model has the conceptual difficulty that the price impacts of two discrete trades is independent of the time interval between the trades. For example, two buy orders close to each other in time versus far apart should generate different price dynamics, while in our assumption they lead to the same price impact. Hence the temporary impact alone will be insufficient to model the complicated market microstructure. However, our approach of using linear quadratic formulations to solve mean variance problem, as well as introducing the implied ratio as a state variable provides a general framework that works readily with other more complicated market impact models. In this appendix, we give an example by following the assumption in Obizhaeva and Wang[2006], where they introduced an extra lag variable(i.e. resilient market impact) between time interval and solved the misspecification mentioned above. However, they only provided deterministic strategies for their risk aversion problem. In the following, we will give a dynamic trading strategy for a market with permanent and resilient impacts, which is a more general assumption than our temporary market impact.

We assume three factors determining security price. The fundamental price S_t is governed by both outside supply/demand and our trading decisions. We model the latter one as the permanent impact we impose to the price and modify (1) as

$$S_{t_i} = S_a + \sigma S_a B_{t_i} + \gamma y_i \text{ for } i = 0, 1, ..., N-1 \tag{144}$$

where γ is the permanent impact factor with units (\$/share)/share and S_a is the arrival price(in our former model, S_0 is the arrival price). With this form, each y units we buy will lift up the price per share by γy, regardless of the time we take to buy them.

A third factor is the resilient market impact. At time $t_0 = 0$, if we decide to buy y_0 shares, our trading action will instantaneously lift up the price S_a to $\tilde{S}_0 = S_a + \gamma y_0 + N\eta_0 y_0$ with $\eta_0 y_0$ representing the instantaneous temporary impact. Here $\eta_0 \geq 0$ with unit (\$/share)/(share/time). Because of our buying activity, more sell orders will be attracted to the limit order book and gradually lower down the ask price. If we do not continue trading further for the remaining trading horizon, we assume the price will decrease over time with a limiting average price only reflecting the permanent impact: $\mathbf{E}\tilde{S}_\infty = S_a + \gamma y_0$. The price difference between $t = 0$ and $t = \infty$ is $N\eta_0 y_0$. We assume the security price converges to its limit exponentially:

$$\begin{aligned} \tilde{S}_t &= S_a + \gamma y_0 + N\eta_0 y_0 e^{-\rho t} \\ &= S_{t_0} + N\eta_0 y_0 e^{-\rho t} \end{aligned}$$

where the resilient factor $\rho \geq 0$ is a measure of the convergence speed. In this appendix, we assume the resilient factor is a constant. However, it can be relaxed to be a time varying parameter. If $\rho = \infty$, this is the framework that contains both permanent and temporary impact. On the other side of the spectrum, if $\rho = 0$, then only the permanent impact $(\gamma + N\eta_0)y_0$ are considered.

We can easily expand the above model from a single trade to multiple trades:

$$\tilde{S}_{t_i} = S_{t_i} + \sum_{j=0}^{i} N\eta_j y_j e^{-\rho(t_i - t_j)}.$$

Therefore, the execution price results from acombination of fundamental price process and resilient market impacts from all past trades. We can define the differences between execution price $\tilde{S}_{t_i}$ and fundamental price S_{t_i} explicitly:

$$\begin{aligned} D_{i+1} &= \tilde{S}_{t_i} - S_{t_i} \\ &= \sum_{j=0}^{i} N\eta_j y_j e^{-\rho(t_i - t_j)} \text{ for } i = 0, 1, ..., N-1. \end{aligned}$$

D_i if $\mathcal{F}_{t_{i-1}}$-measurable, and records the resilient impact from all past trades before t_i. If we define $D_0 = 0$, D_i has the following recursive form:

$$D_{i+1} = D_i e^{-\rho(t_i - t_{i-1})} + N\eta_i y_i \text{ for } i = 0, 1, ..., N-1.$$

Following similar computation, the implementation shortfall of a given feasible strategy $\pi = \{y_0, y_1, ..., y_{N-1}\}$ is

$$\begin{aligned} I_N &= \sum_{i=0}^{N-1} \Big[(N\eta_i + \frac{\gamma}{2}) y_i^2 + D_i e^{-\rho(t_i - t_{i-1})} y_i \Big] + \sum_{i=1}^{N-1} \sigma S_a (B_{t_i} - B_{t_{i-1}}) x_i + \frac{\gamma}{2} X^2 \\ &= \sum_{i=0}^{N-1} \Big[(N\eta_i + \frac{\gamma}{2}) y_i^2 + D_i e^{-\rho(t_i - t_{i-1})} y_i + \sigma S_a (B_{t_{i+1}} - B_{t_i}) x_{i+1} \Big] + \frac{\gamma}{2} X^2 \end{aligned} \quad (145)$$

by assuming $B_{t_N} = B_{t_{N-1}}$. The last term $\frac{\gamma}{2} X^2$ in (145) is independent of strategy π and its removal does not change the derivation of optimal trading strategy. Scale the remaining shortfall by the product of order's initial market value and price change's volatility:

$$\tilde{I}_N = \frac{I_N - \frac{\gamma}{2} X^2}{X S_a \sigma}. \quad (146)$$

For $i = 0, 1, ..., N-1$, let

$$\begin{aligned} \tilde{y}_i &= y_i / X, \qquad \tilde{x}_i = x_i / X, \\ \tilde{\gamma} &= \gamma X / (S_a \sigma), \qquad \tilde{\eta}_i = \eta_i X / (S_a \sigma), \\ \tilde{D}_i &= D_i / (S_a \sigma) \end{aligned}$$

Notice that under these notifications, x_i and D_i still preserve the same recursive formulation. For $i = 0, 1, ..., N-1$:

$$\tilde{x}_{i+1} = \tilde{x}_i - \tilde{y}_i, \qquad \tilde{x}_0 = 1, \quad (147)$$

$$\tilde{D}_{i+1} = \tilde{D}_i e^{-\rho(t_i - t_{i-1})} + \tilde{\eta}_i \tilde{y}_i, \qquad \tilde{D}_0 = 0. \quad (148)$$

Then the scaled implementation shortfall in (146) is

$$\tilde{I}_N = \sum_{i=0}^{N-1} \left[(N\tilde{\eta}_i + \frac{\tilde{\gamma}}{2})\tilde{y}_i^2 + \tilde{D}_i e^{-\rho(t_i - t_{i-1})}\tilde{y}_i + (B_{t_{i+1}} - B_{t_i})\tilde{x}_{i+1} \right].$$

From now on, we will omit the tilde notation.

Similar as in Section 3, the mean variance problem (14) can be solved through a family of (25), and the appropriate r_0^* can be determined using the similar methods in Section 3.3. Therefore, the main task is to derive the Bellman backward induction similar as (48).

First of all, the objective function of the (25) can be decomposed as

$$\begin{aligned} r_0\bar{I}_N + \bar{I}_N^2 &= \sum_{i=0}^{N-1} \{ r_i \left[(N\eta_i + \frac{\gamma}{2})y_i^2 + D_i e^{-\rho(t_i - t_{i-1})} y_i + (B_{t_{i+1}} - B_{t_i})(x_i - y_i) \right] \\ &\quad + \left[(N\eta_i + \frac{\gamma}{2})y_i^2 + D_i e^{-\rho(t_i - t_{i-1})} y_i + (B_{t_{i+1}} - B_{t_i})(x_i - y_i) \right]^2 \} \end{aligned}$$

where r_i is defined in a similar way as in (43): for $i = 0, 1, ..., N-2$

$$r_{i+1} = r_i + 2 \left[(N\eta_i + \frac{\gamma}{2})y_i^2 + D_i e^{-\rho(t_i - t_{i-1})} y_i + (B_{t_{i+1}} - B_{t_i})(x_i - y_i) \right]. \qquad (149)$$

Since the current execution price is influenced by all past trade's resilient impact, we should introduce D_i as our state variable for dynamic programming. Noticed that (x_i, D_i, r_i) is $\mathcal{F}_{t_i}$-measurable(more specifically, X_i and D_i are $\mathcal{F}_{t_{i-1}}$-measurable while r_i is $\mathcal{F}_{t_i}$-measurable), we use it as our state vector, which follows the transition rule of (147)-(149).

The last time period's decision $y_{N-1}(x, D, r)$ and value function $V_{N-1}(x, D, r)$ is the similar as in (2.3.2) and (47):

$$\begin{aligned} y_{N-1}^*(x, D, r) &= x; \\ V_{N-1}(x, D, r) &= r \left[(N\eta_{N-1} + \frac{\gamma}{2})x^2 + D e^{-\rho(t_{N-1} - t_{N-2})} x \right] \\ &\quad + \left[(N\eta_{N-1} + \frac{\gamma}{2})x^2 + D e^{-\rho(t_{N-1} - t_{N-2})} x \right]^2 \end{aligned}$$

While for $i = N-2, N-3, ..., 0$:

$$\begin{aligned} V_i(x,D,r) &= \min_{0\le y\le x}\{c_i((x,D,r),y) + \mathbb{E}[V_{i+1}(x-y, De^{-\rho(t_i-t_{i-1})} + N\eta_i y, \\ &\quad r + 2(N\eta_i + \frac{\gamma}{2})y^2 + 2De^{-\rho(t_i-t_{i-1})} + 2(B_{t_{i+1}} - B_{t_i})(x-y))|(x,D,r)]\} \end{aligned}$$

and $y_i(X,D,r)$ is the corresponding optimizer for the above minimization problem.

APPENDIX B

APPENDIX FOR CHAPTER III

B.1 Derivation on (117)

The following steps derive (117) from (116):

$$
\begin{aligned}
&\arg\min_{\Delta v_i} \frac{(n-i)(2n-2i-1)}{6(n-i-1)}\mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1}+v_i}{V_n+v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] \\
= \;&\arg\min_{\Delta v_i} \mathbb{E}\left[\left(\left(\frac{v_i}{v_{n-1}} - \frac{v_i}{V_n+v_{n-1}}\right) - \frac{V_{i+1}}{V_n+v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] \\
= \;&\arg\min_{\Delta v_i} \mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} \cdot \frac{V_n}{V_n+v_{n-1}} - \frac{V_{i+1}}{V_n+v_{n-1}}\right)^2 \middle| (v_i, V_i)\right] \\
= \;&\arg\min_{\Delta v_i} \left[\mathbb{E}\left[\frac{V_n^2}{(V_n+v_{n-1})^2}\middle| V_i\right]\left(\frac{v_i}{v_{n-1}}\right)^2 - 2\mathbb{E}\left[\frac{V_{i+1}V_n}{(V_n+v_{n-1})^2}\middle| V_i\right]\left(\frac{v_i}{v_{n-1}}\right)\right]
\end{aligned}
$$

Without boundary constraint, $\frac{v_i}{v_{n-1}}$ will take the minimum value at

$$
\frac{\mathbb{E}\left[\frac{V_{i+1}V_n}{(V_n+v_{n-1})^2}\middle| V_i\right]}{\mathbb{E}\left[\frac{V_n^2}{(V_n+v_{n-1})^2}\middle| V_i\right]}
$$

However, the trading decision Δv_i is bounded by $0 \leq \Delta v_i \leq v_{n-1} - v_{i-1}$, furthermore, note that $\frac{v_i}{v_{n-1}} = \frac{v_{i-1}+\Delta v_i}{v_{n-1}}$, the optimal Δv_i^* should hence satisfy (117).

B.2 Stock-specific comparison with [11]

This section tests the same stocks used in [11], and compares the relative performance of our adaptive strategy with theirs. The stock specific results are listed in Table 16, where the second and third columns are copied from Table 6 of [11] with the same measurement as (82).

The classic strategy in [11] (second column) is exactly the same as the deterministic strategy we used in (111), while [11] used (73) for their adaptive strategy (third column), where future volumes $\Delta\hat{V}_j (j = i+1, i+2, ..., n)$ are derived using econometric models(ARMA and SETAR). [11] used one year of data for 39 CAC40 stocks from 09/02/2003 to 08/30/2004. Unfortunately, we are not able to find intraday data for these stocks for the same time period. Instead, we collected 30 of these stocks' intraday data during 09/01/2012-08/31/2013. The market microstructure has changed dramatically during the past ten years, and we are not able to match the deterministic strategies' performances (second column and fifth column) for most stocks (except those in bold characters in Table 16). Nonetheless, Table 16 presents the relative performance of adaptive strategies over deterministic strategies for these two different data sets. This is illustrated in the fourth column (diff= $\frac{\text{classic-dynamic}}{\text{classic}}$) and seventh column (diff= $\frac{\text{MAE(0)-MAE(0.05)}}{\text{MAE(0)}}$). The larger the value in these two columns, the more adaptive strategies outperform deterministic strategy. To develop a fair comparison, we based our test results (fifth column to seventh column) on a frequency of 20 minutes, and a rolling training window of 20 days, which is consistent with the setup in [11]. As shown in Table 16, using the measure of relative advantage, our Adaptive(0.05) outperforms [11]'s adaptive strategy for 24 out of 30 stocks. The aggregated relative advantage of 13.41% from Adaptive(0.05) over Adaptive(1) is consistent with 14.7% observed in US stocks test(see Table 5), which is much higher than 8.27% in [11]. Taking into account the differences in the data samples, it is fair to claim that our adaptive strategies' performance is comparable to the one used in [11].

Table 16: Comparison of adaptive strategies' relative advantage between [11]'s dynamic method and our Adaptive(e)

sample data dates range	results from [11] 09/02/2003-08/30/2004			results based on Adaptive(e) 09/01/2012-08/31/2013		
companies	classic	dynamic	diff	MAE(0)	MAE(0.05)	diff
ACCOR	10.47	11.21	-7.07%	5.72	4.84	15.40%
AIR LIQUIDE	8.01	8.18	-2.12%	4.31	3.50	18.92%
ALCATEL	13.36	10.79	19.24%	16.76	15.57	7.09%
ARCELOR	11.71	10.62	9.31%	8.67	7.22	16.72%
AXA	9.30	8.89	4.41%	6.69	5.20	22.33%
BNP PARIBAS	**7.82**	7.42	5.12%	**8.03**	6.76	15.80%
BOUYGUES	17.15	17.73	-3.38%	8.42	6.18	26.55%
CAP GEMINI	23.23	14.91	35.82%	5.33	4.54	14.91%
CARREFOUR	6.28	6.38	-1.59%	8.59	6.91	19.49%
CREDIT AGRICOLE	**13.89**	11.02	20.66%	**13.67**	10.52	23.06%
DANONE	5.48	5.31	3.10%	3.92	3.43	12.34%
EADS	19.47	14.04	27.89%	9.51	8.06	15.28%
L'OREAL	8.66	8.32	3.93%	5.17	4.38	15.26%
LAFARGE	10.76	10.75	0.09%	6.57	5.63	14.21%
LVMH	11.31	9.59	15.21%	5.39	4.37	18.81%
MICHELIN	15.41	15.13	1.82%	5.94	5.49	7.57%
PERNOD-RICARD	7.75	7.45	3.87%	3.99	3.64	8.99%
SAINT GOBAIN GOBAIN	**9.79**	9.52	2.76%	**9.42**	9.25	1.85%
SANOFI-AVENTIS	9.99	8.97	10.21%	4.78	4.48	6.45%
SCHNEIDER ELECTRIC	8.65	10.27	-18.73%	6.24	5.89	5.54%
SOCIETE GENERALE	6.99	6.17	11.73%	11.48	9.40	18.09%
SODEXHO ALLIANCE	12.33	11.82	4.14%	3.97	3.52	11.29%
STMICROELECTRONICS	9.06	7.68	15.23%	13.21	11.01	16.69%
SUEZ	9.68	9.08	6.20%	17.25	16.92	1.92%
THALES	9.59	10.27	-7.09%	5.87	4.86	17.31%
TOTAL	5.28	5.08	3.79%	4.06	3.61	11.19%
VEOLIA ENVIRON.	13.00	12.86	1.08%	9.17	8.21	10.50%
VINCI (EX.SGE)	7.74	7.55	2.45%	4.54	4.09	9.79%
VIVENDI UNIVERSAL	10.95	10.20	6.85%	5.56	4.94	11.19%
average	10.80	9.90	8.27%	7.66	6.64	13.41%

B.3 Derivation of (84)

Proof. We show through induction that the following recursive equations in terms of $q_i - Q_i$ exist for $i = 1, 2, ..., n$:

$$q_i - Q_i = \sum_{j=0}^{i-1} \Big[- r_j(q_j - Q_j) + (P_{j+1} - Q_j)(r_j - R_{j+1}) \Big]. \tag{150}$$

When $i = 1$, notice that $r_0 = R_1 = 1$ and $q_0 = Q_0 = 0$:

$$q_1 - Q_1 = r_0 P_1 - R_1 P_1 = 0 = -r_0(q_0 - Q_0) + (P_1 - Q_0)(r_0 - R_1).$$

Next assume (150) holds for i, then we prove it will also holds for $i + 1$:

$$\begin{aligned}
& q_{i+1} - Q_{i+1} \\
= \;& [(1 - r_i)q_i + r_i P_{i+1}] - [(1 - R_{i+1})Q_i + R_{i+1}P_{i+1}] \\
= \;& (P_{i+1} - Q_i)(r_i - R_{i+1}) + (1 - r_i)(q_i - Q_i) \\
= \;& [-r_i(q_i - Q_i) + (P_{i+1} - Q_i)(r_i - R_{i+1})] + (q_i - Q_i) \\
= \;& \sum_{j=0}^{i} \Big[- r_j(q_j - Q_j) + (P_{j+1} - Q_j)(r_j - R_{j+1}) \Big].
\end{aligned}$$

Therefore, by induction, (150) holds for $i = 1, 2, ..., n$. Particularly, for $i = n$,

$$\begin{aligned}
q_n - Q_n &= \sum_{i=0}^{n-1} \Big[- r_i(q_i - Q_i) + (P_{i+1} - Q_i)(r_i - R_{i+1}) \Big] \qquad (151) \\
(\text{notice } r_n = 0) &= -r_0(q_0 - Q_0) + \sum_{i=1}^{n} \Big[(P_i - Q_{i-1})(r_{i-1} - R_i) - r_i(q_i - Q_i) \Big] \\
&= \sum_{i=1}^{n} c_i.
\end{aligned}$$

□

B.4 Proof of Proposition 1

Proof. First, we derive (105):

The term $\sum_{j=i+1}^{n} c_j$ in (104) can be decomposed into:

$$
\begin{aligned}
& \sum_{j=i+1}^{n} c_j \\
= & \sum_{j=i+1}^{n} \Big[(P_j - Q_{j-1})(r_{j-1} - R_j) - r_j(q_j - Q_j)\Big] \\
= & \; r_i(q_i - Q_i) + \sum_{j=i}^{n-1} \Big[- r_j(q_j - Q_j) + (P_{j+1} - Q_j)(r_j - R_{j+1})\Big] \\
= & \; r_i(q_i - Q_i) + (q_n - Q_n) - (q_i - Q_i) \\
= & \; -(1 - r_i)(q_i - Q_i) + (q_n - Q_n) \\
= & \; -(1 - r_i)(q_i - Q_i) + \sum_{j=0}^{i-1} \frac{\Delta v_j}{v_{n-1}} P_{j+1} - \sum_{j=0}^{i-1} \frac{\Delta V_{j+1} + \Delta v_j}{V_n + v_{n-1}} P_{j+1} \\
& + \sum_{j=i}^{n-1} \frac{\Delta v_j}{v_{n-1}} P_{j+1} - \sum_{j=i}^{n-1} \frac{\Delta V_{j+1} + \Delta v_j}{V_n + v_{n-1}} P_{j+1} \\
= & \; -(1 - r_i)(q_i - Q_i) + \frac{(1 - r_i)v_i}{v_{n-1}} q_i - \frac{V_i + (1 - r_i)v_i}{V_n + v_{n-1}} Q_i \\
& + \sum_{j=i}^{n-1} \frac{\Delta v_j}{v_{n-1}} \Big(P_i + \sum_{k=i+1}^{j+1} \Delta P_k\Big) - \sum_{j=i}^{n-1} \frac{\Delta V_{j+1} + \Delta v_j}{V_n + v_{n-1}} \Big(P_i + \sum_{k=i+1}^{j+1} \Delta P_k\Big) \\
= & \; -(1 - r_i)(q_i - Q_i) + \frac{(1 - r_i)v_i}{v_{n-1}} q_i - \frac{V_i + (1 - r_i)v_i}{V_n + v_{n-1}} Q_i \\
& + \Big(1 - \frac{(1 - r_i)v_i}{v_{n-1}}\Big) P_i - \Big(1 - \frac{V_i + (1 - r_i)v_i}{V_n + v_{n-1}}\Big) P_i \\
& + \sum_{k=i+1}^{n} \sum_{j=k-1}^{n-1} \Big(\frac{\Delta v_j}{v_{n-1}} - \frac{\Delta V_{j+1} + \Delta v_j}{V_n + v_{n-1}}\Big) \Delta P_k \\
= & \; -(1 - r_i)(q_i - Q_i) + \frac{(1 - r_i)v_i}{v_{n-1}} (q_i - P_i) - \frac{V_i + (1 - r_i)v_i}{V_n + v_{n-1}} (Q_i - P_i) \\
& - \sum_{j=i}^{n-1} \Big(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\Big) \Delta P_{j+1} \\
= & \; K_i(S_i^v) - U_I
\end{aligned}
$$

Similar as (86), the partial sum of C_j can be expressed as a function of the partial sum of c_j as well:

$$\begin{aligned} J_i(S_i^v) &= \mathbb{E}\Big[\sum_{j=i+1}^{n} C_j \Big| S_i^v\Big] \\ &= \mathbb{E}\Big[d_i \sum_{j=i+1}^{n} c_j + \Big(\sum_{j=i+1}^{n} c_j\Big)^2 \Big| S_i^v\Big] \\ &= \mathbb{E}\Big[d_i(K_i(S_i^v) - U_I) + \Big(K_i(S_i^v) - U_I\Big)^2 \Big| S_i^v\Big] \\ &= d_i K(S_i^v) + K^2(S_i^v) - (d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v] \end{aligned}$$

□

Secondly, we prove the simplification for optimal action (106) using the above result. From (102),

$$\begin{aligned} \Delta v_i^* &= \arg\min_{\Delta v_i} \Big[C_i(S_i, \Delta v_i) + J_i(S_i^v)\Big] \\ &= \arg\min_{\Delta v_i} \Big[d_{i-1}c_i + c_i^2 + d_i K(S_i^v) + K^2(S_i^v) - (d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v]\Big] \end{aligned}$$

In the following, we show that $d_{i-1}c_i + c_i^2 + d_i K(S_i^v) + K^2(S_i^v)$ is independent of Δv_i.

Note that given S_i^v, q_i and Q_i are also known. Furthermore, $(1 - r_i)v_i = v_{i-1}$ is independent of Δv_i. Let's take the first order derivative of $d_{i-1}c_i + c_i^2 + d_i K(S_i^v) + K^2(S_i^v)$ over Δv_i:

$$\begin{aligned} \frac{\partial c_i}{\partial \Delta v_i} &= -\frac{\Delta v_i}{(v_{i-1} + \Delta v_i)^2}(q_i - Q_i) \\ \frac{\partial K(S_i^v)}{\partial \Delta v_i} &= \frac{\Delta v_i}{(v_{i-1} + \Delta v_i)^2}(q_i - Q_i) \\ \frac{\partial d_i}{\partial \Delta v_i} &= -2\frac{\Delta v_i}{(v_{i-1} + \Delta v_i)^2}(q_i - Q_i) \end{aligned}$$

Therefore

$$\begin{aligned}
& \frac{\partial}{\partial \Delta v_i}[d_{i-1}c_i + c_i^2 + d_i K(S_i^v) + K^2(S_i^v)] \\
= \quad & d_{i-1}\frac{\partial c_i}{\partial \Delta v_i} + 2c_i\frac{\partial c_i}{\partial \Delta v_i} + \frac{\partial d_i}{\partial \Delta v_i}K(S_i^v) + d_i\frac{\partial K(S_i^v)}{\partial \Delta v_i} + 2K(S_i^v)\frac{\partial K(S_i^v)}{\partial \Delta v_i} \quad (152) \\
= \quad & d_{i-1}\frac{\partial c_i}{\partial \Delta v_i} + 2c_i\frac{\partial c_i}{\partial \Delta v_i} + 2K(S_i^v)\frac{\partial c_i}{\partial \Delta v_i} - d_i\frac{\partial c_i}{\partial \Delta v_i} - 2K(S_i^v)\frac{\partial c_i}{\partial \Delta v_i} \quad (153) \\
= \quad & (d_{i-1} + 2c_i - d_i)\frac{\partial c_i}{\partial \Delta v_i} \quad (154) \\
= \quad & 0 \quad (155)
\end{aligned}$$

Since the first order derivative of $d_{i-1}c_i + c_i^2 + d_i K(S_i^v) + K^2(S_i^v)$ equals to zero for any values of Δv_i, it is independent of Δv_i. Therefore,

$$\Delta v_i^* = \arg\min_{\Delta v_i}\Big[-(d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v]\Big]$$

B.5 Proof of Proposition 1

Proof. First, the two assumptions in Proposition 1 make the following three conditional expectations have value zero: For $j \geq i$, note that $\sigma(S_i^v)$ is a sub-σ-field of $\mathcal{F}_j$, using the tower property:

$$\begin{aligned}
& \mathbb{E}\Big[\Big(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\Big)\Delta P_{j+1}\Big|S_i^v\Big] \\
= \quad & \mathbb{E}\Big[\mathbb{E}\Big[\Big(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\Big)\Big|\mathcal{F}_j\Big]\mathbb{E}\Big[\Delta P_{j+1}\Big|\mathcal{F}_j\Big]\Big|S_i^v\Big] \\
= \quad & 0 \quad (156)
\end{aligned}$$

The first equation comes from the second assumption, while the second equation comes from the first assumption. Similarly, the cross term U_I also has zero conditional expectation:

$$\mathbb{E}\Big[\frac{[V_i + (1-r_i)v_i](Q_i - P_i)}{V_n + v_{n-1}}\Big(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\Big)\Delta P_{j+1}\Big|S_i^v\Big] = 0; \quad (157)$$

For $k > j \geq i$, denote $A_{jk} = \left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j+v_{j-1}}{V_n+v_{n-1}}\right)\left(\frac{v_{k-1}}{v_{n-1}} - \frac{V_k+v_{k-1}}{V_n+v_{n-1}}\right)$, then

$$\begin{aligned} & \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)\left(\frac{v_{k-1}}{v_{n-1}} - \frac{V_k + v_{k-1}}{V_n + v_{n-1}}\right)\Delta P_{j+1}\Delta P_{k+1}\middle|S_i^v\right] \\ = & \mathbb{E}\left[\mathbb{E}\left[A_{jk}\Delta P_{j+1}\Delta P_{k+1}\middle|\mathcal{F}_k\right]\middle|S_i^v\right] \\ = & \mathbb{E}\left[\mathbb{E}\left[A_{jk}\Delta P_{j+1}\middle|\mathcal{F}_k\right]\mathbb{E}\left[P_{k+1}\middle|\mathcal{F}_k\right]\middle|S_i^v\right] \\ = & 0 \end{aligned} \tag{158}$$

Secondly, (156)-(158) can be used to simplify $-(d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v]$:

$$\begin{aligned} & -(d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v] \\ = & -(d_i + 2K(S_i^v))[V_i + (1-r_i)v_i](Q_i - P_i)\mathbb{E}\left[\frac{1}{V_n + v_{n-1}}\middle|S_i^v\right] \\ & +[V_i + (1-r_i)v_i]^2(Q_i - P_i)^2\mathbb{E}\left[\frac{1}{(V_n + v_{n-1})^2}\middle|S_i^v\right] \\ & +\mathbb{E}\left[\left(\frac{v_{i-1}}{v_{n-1}} - \frac{V_i + v_{i-1}}{V_n + v_{n-1}}\right)^2\Delta P_{i+1}^2\middle|S_i^v\right] \\ & +\sum_{j=i+1}^{n-1}\mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2\Delta P_{j+1}^2\middle|S_i^v\right] \end{aligned} \tag{159}$$

According to (106), $-(d_i + 2K(S_i^v))\mathbb{E}[U_I|S_i^v] + \mathbb{E}[U_I^2|S_i^v]$ is a function of both S_i and Δv_i. However, it can be shown only the last item in (159) depends on Δv_i. Here is the reason:

1. (152) and (153) show that $(d_i + 2K(S_i^v))$ is independent of Δv_i;

2. $(1 - r_i)v_i = v_{i-1}$ only depends on S_i;

3. (90) and (91) show that $q_i - Q_i$ depends only on S_i;

4. since the volume process is an exogenous stochastic process independent with our trading decision, $\sum_{j=i+1}^{n}\Delta V_j$ depends on S_i^v only through V_i, therefore $\mathbb{E}\left[\frac{1}{V_n+v_{n-1}}\middle|S_i^v\right] = \mathbb{E}\left[\frac{1}{V_n+v_{n-1}}\middle|V_i\right]$ is independent with Δv_i. Similar argument can

be used to get

$$\mathbb{E}\Big[\Big(\frac{v_{i-1}}{v_{n-1}} - \frac{V_i + v_{i-1}}{V_n + v_{n-1}}\Big)^2 \Delta P_{i+1}^2 \Big| S_i^v\Big] = \mathbb{E}\Big[\Big(\frac{v_{i-1}}{v_{n-1}} - \frac{V_i + v_{i-1}}{V_n + v_{n-1}}\Big)^2 \Delta P_{i+1}^2 \Big| (V_i, v_{i-1})\Big]$$

As a result,

$$\Delta v_i^* = \arg\min_{\Delta v_i} \sum_{j=i+1}^{n-1} \mathbb{E}\left[\left(\frac{v_{j-1}}{v_{n-1}} - \frac{V_j + v_{j-1}}{V_n + v_{n-1}}\right)^2 \Delta P_{j+1}^2 \middle| S_i^v\right].$$

□

B.6 Proof of Proposition 1

Proof. First, we prove the following property: for random variables X, Y, Z, if Z is independent of both X and Y, then for a continuous function f:

$$\mathbb{E}[f(X,Z)|X,Y] = \mathbb{E}[f(X,Z)|X] \tag{160}$$

Define $h(X,Y) := \mathbb{E}[f(X,Z)|X,Y]$. For $x, y \in \mathbf{R}$:

$$h(x,y) = \mathbb{E}[f(X,Z)|X = x, Y = y] = \mathbb{E}[f(x,Z)]$$

where the second equation is because Z is independent of $\sigma(X,Y)$. This implies $h(x,y)$ does not depend on y. We hence will write $h(x,y)$ as $h(x)$. Then (160) is proved based on tower property:

$$\mathbb{E}[f(X,Z)|X] = \mathbb{E}\Big[\mathbb{E}[f(X,Z)|X,Y]\Big|X\Big] = \mathbb{E}[h(X)|X] = h(X) = \mathbb{E}[f(X,Z)|X,Y].$$

Next we prove Proposition 1 through induction. For $i = n-2$, according to (109):

$$\begin{aligned}
& \Delta v_{n-2}^*(S_{n-2}) \\
= \; & \arg\min_{\Delta v_{n-2}} \mathbb{E}\left[\left(\frac{v_{n-2}}{v_{n-1}} - \frac{V_{n-1} + v_{n-2}}{V_n + v_{n-1}}\right)^2 \Delta P_n^2 \middle| S_{n-2}^v\right] \\
= \; & \arg\min_{\Delta v_{n-2}} \mathbb{E}\Bigg[\left(\frac{v_{n-2}}{v_{n-1}} - \frac{V_{n-2} + \Delta V_{n-1} + v_{n-2}}{V_{n-2} + \Delta V_{n-1} + \Delta V_n + v_{n-1}}\right)^2 \Delta P_n^2 \Bigg| \\
& v_{n-2}, V_{n-2}, d_{n-2}, q_{n-2}, Q_{n-2}, P_{n-2}, r_{n-2}\Bigg]
\end{aligned}$$

Define $X = (v_{n-2}, V_{n-2})$, $Y = (d_{n-2}, q_{n-2}, Q_{n-2}, P_{n-2}, r_{n-2})$, $Z = (\Delta V_{n-1}, \Delta V_n, \Delta P_n)$ and function $f(X, Z) = \left(\frac{v_{n-2}}{v_{n-1}} - \frac{V_{n-2}+\Delta V_{n-1}+v_{n-2}}{V_{n-2}+\Delta V_{n-1}+\Delta V_n+v_{n-1}}\right)^2 \Delta P_n^2$. Since $v_{n-1} > 0$ and $V_{n-2} + \Delta V_{n-1} + \Delta V_n + v_{n-1} > 0$, $f(X, Z)$ is a continuous function. Applying the independent assumption of $(\Delta V_{n-1}, \Delta V_n, \Delta P_n)$ and $S^v_{n-2} = (X, Y)$ into (160), we have

$$\Delta v^*_{n-2}(S_{n-2}) = \arg \min_{\Delta v_{n-2}} \mathbb{E}\left[\left(\frac{v_{n-2}}{v_{n-1}} - \frac{V_{n-1} + v_{n-2}}{V_n + v_{n-1}}\right)^2 \Delta P_n^2 \middle| v_{n-2}, V_{n-2}\right]$$

which is equivalent as $\Delta v^*_{n-2}(S_{n-2}) = \Delta v^*_{n-2}(v_{n-3}, V_{n-2})$. Therefore we proved (110) for $i = n - 2$.

Next assume (110) is true for $j = i + 1, ..., n - 2$, i.e. $\Delta v^*_j(S_j) = \Delta v^*_j(v_{j-1}, V_j)$, write $v^*_j(v_{j-1}, V_j) = v_{j-1} + \Delta v^*_j(v_{j-1}, V_j)$, then

$$\begin{aligned} \Delta v^*_i(S_i) \quad = \quad & \arg\min_{\Delta v_i} \mathbb{E}\left[\left(\frac{v_i}{v_{n-1}} - \frac{V_{i+1} + v_i}{V_n + v_{n-1}}\right)^2 \Delta P^2_{i+2}\right. \\ & \left. + \sum_{j=i+1}^{n-1} \left(\frac{v^*_j(v_{j-1}, V_j)}{v_{n-1}} - \frac{V_{j+1} + v^*_j(v_{j-1}, V_j)}{V_n + v_{n-1}}\right)^2 \Delta P^2_{j+2} \middle| S^v_i\right] \end{aligned} \quad (161)$$

Note that $v^*_{i+1}(v_i, V_{i+1})$ is a function of $v_i, V_i, \Delta V_{i+1}$. Since

$$v^*_{i+2}(v_{i+1}, V_{i+2}) = v^*_{i+2}(v^*_{i+1}(v_i, V_{i+1}), V_{i+2}),$$

v_{i+2} depends on $v_i, V_i, \Delta V_{i+1}, \Delta V_{i+2}$. Similarly, (161) is a function of $v_i, V_i, \{\Delta V_j\}^n_{j=i+1}$ and $\{\Delta V_j\}^n_{j=i+2}$. Define $X = (v_i, V_i)$, $Y = (d_i, q_i, Q_i, P_i, r_i)$, and $Z = \{\Delta V_j\}^n_{j=i+1}, \{\Delta V_j\}^n_{j=i+2}$. (110) is true according to (160) and the independence assumption of Z and S^v_i. □

APPENDIX C

APPENDIX FOR CHAPTER III

C.1 Slippage Decomposition

This section shows how (123) can be equivalently decomposed to (124)-(126).

$$
\begin{aligned}
& I \\
:= \;& \sum_{i=0}^{n-1} \left[\Big(P_i + \frac{1}{2}s_i + f^m + n\eta x_i \Big) x_i + \Big(P_i - \frac{1}{2}s_i - f^l \Big)(y_i \wedge l_i) \right] - XP_0 \\
= \;& (\text{note } P_i = P_0 + \sum_{j=0}^{i} \epsilon_j \text{ and (122)}) \\
= \;& \sum_{i=0}^{n-1} \left[\Big(\sum_{j=0}^{i} \epsilon_j + \frac{1}{2}s_i + f^m + n\eta x_i \Big) x_i + \Big(\sum_{j=0}^{i} \epsilon_j - \frac{1}{2}s_i - f^l \Big)(y_i \wedge l_i) \right] \\
= \;& \sum_{i=0}^{n-1} \left[\epsilon_i \sum_{j=i}^{n-1} (x_j + y_j \wedge l_j) + \Big(\frac{1}{2}s_i + f^m + n\eta x_i \Big) x_i + \Big(-\frac{1}{2}s_i - f^l \Big)(y_i \wedge l_i) \right] \\
= \;& \sum_{i=0}^{n-1} \left[z_i \epsilon_i + \Big(\frac{1}{2}s_i + f^m + n\eta x_i \Big) x_i + \Big(-\frac{1}{2}s_i - f^l \Big)(y_i \wedge l_i) \right] \\
= \;& \sum_{i=0}^{n-1} c_i
\end{aligned}
$$

REFERENCES

[1] Alfonsi, Aurélien, Antje Fruth and Alexander Schied, 2010, Optimal execution strategies in limit order books with general shape functions, *Quantitiatve Finance* 10, No.2, 143-157.

[2] Almgren, Robert, and Neil Chriss, 2001, Optimal execution of portfolio transactions, *Journal of Risk* 3, 5-39.

[3] Almgren, Robert and Julian Lorenz, 2007, Adaptive arrival price, *Algorithmic Trading III: Precision, Control, Execution, Brian R. Bruce, editor, Institutional Investor Journals.*

[4] Andersen, T.G., 2010, Return volatility and trading volume: An information flow interpretation of stochastic volatility, *The Journal of Finance* 51, 169-204.

[5] Avellaneda, Marco and Sasha Stoikov, 2008, High-frequency trading in a limit order book, *Quantitative Finance*, Vol.8, No.3, 217-224.

[6] Bayraktar, Erhan and Michael Ludkovski, 2012, Liquidiation in limit order books with controlled intensity, *Mathematical Finance*

[7] Bertsekas, Dimitri P, 2012, Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming, *Athena Scientific*

[8] Bertsimas, Dimitris, and Andrew Lo, 1998, Optimal control of execution costs, *Journal of Financial Markets* 1, 1-50.

[9] Bank of America, 2007, Marching up the learning curve: The second buy-side algorithmic trading survey, Feb 12 2007

[10] Bellman, Richard, 1961, Adaptive Control Processes: A Guided Tour, *Princeton University Press.*

[11] Bialkowski, J., S. Darolles, and G. Le Fol, 2008, Improving vwap strategies: A dynamic volume approach, *Journal of Banking and Finance*, 32, 1709-1722.

[12] Bouchaud, Jean-Philippe, J. Doyne Farmer and Fabrizio Lillo, 2008, How markets slowly digest changes in suppply and demand, *In Handbook of Financial Markets: Dynamics and Evolution*(Elsevier: Academic Press), 57-156.

[13] Brownlees, C, F. Cipollini, and G.M. Gallo, 2010, Intra-dailyl volume modeling and prediction for algorithmic trading, *http://ssrn.com/abstract=1393993.*

[14] Clark, P.K., 1973, A subordinated stochastic process model with finite variance for speculative price, *Econometrica* 41, 135-155.

[15] Cont, Rama and Arseniy Kukanov, 2012, Optimal Order Placement in Limit Oder Markets, *Available at SSRN 2155218*

[16] Darrat, Ali F, Shafiqur Rahman and Maosen Zhong, 2003, Intraday trading volume and return volatility of the DJIA stocks: A note, *Journal of Banking & Finance* 27, 2035-2043.

[17] Dierckx, Paul, Curve and Surface Fitting with Splines, 1993, *Monographs on Numerical Analysis, Oxford University Press*

[18] Esser, Angelika and Burkart Monch, 2007, THe navigation of an iceberg: The optimal use of hidden orders, *Finance Research Letters*, Vol. 4, Issue 2.

[19] Forsyt, Peter, 2009, Optimal trade execution: A Hamilton Jacobi Bellman Approach to Optimal Trade Execution, to appear *Applied Numerical Mathematics.*

[20] Fuh, Cheng-Der, H.W. Teng and R.H.Wang, 2010, On-line VWAP trading strategies, *Sequential Analysis* 29, 292-310.

[21] Gueant, Olivier, Charles-Albert Lehalle, and Joaquin Fernandez-Tapia, 2012, Optimal portfolio liquidiation with limit orders, *SIAM Journal on Financial Mathematics*, Vol 3, Issue 1.

[22] Guo, Xin, Adrien de Larrard and Zhao Ruan, 2013, Optimal placement in a limit order book, *Preprint*.

[23] Hautsch, Nikolaus and Ruihong Huang, 2012, The market impact of a limit order, *Journal of Economic Dynamics and Control*, Vol. 36, Issue 4.

[24] Huitema, Robert, 2013, Optimal Portfolio Exectuion using Market and Limit Orders, *Available at SSRN 1977553*.

[25] Humphery-Jenner, M.L., 2011, Optimal VWAP trading under noisy conditions, *Journal of Banking & Finance* 35, 2319-2329.

[26] Jain, P.c., G. Joh, 1988, The dependence between hourly prices and trading volume, *Jounal of Fianncial Quantitative Analysis* 23, 269-283.

[27] Ganchev, Kuzman, Michael Kearns, Yuriy Nevmyvaka and Jennifer Wortman Vaughan, 2009, Censored Exploration and the Dark Pool Problem, *Communications of the ACM*, Vol 53, No. 5.

[28] Kissell, Robert, and Morton Glantz, 2003, *Optimal Trading Strategies: Quantitative Approaches for Managing Market Impact and Trading Risk* (Mcgraw-Hill Professional).

[29] Konishi, Hizuru, 2002, Optimal slice of a VWAP trade, *Journal of Financial Markets* 5, 197-221.

[30] Li, Duan, and Wan-Lung Wu, 2000, Optimal dynamic portfolio selection: Multiperiod mean-variance formulation, *Mathematical Finance*, Vol. 10, No. 3, 387-406.

[31] Lorenz, Julian and Robert Almgren, 2009, Continuously Adaptive Arrival Price, submitted.

[32] Markowitz, Harry, 1952, Portfolio Selection, *The Journal of Finance* 7(1), 7791.

[33] Obizhaeva, Anna, and Jiang Wang, 2013, Optimal trading strategy and supply/demand dynamics, *Journal of Financial Markets* Vol. 16, No. 1.

[34] Perold, Andre, 1988, The implementation shortfall: paper versus reality, *Journal of Portfolio Management*, Spring.

[35] Powell, W., 2011, Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition *Wiley Series in Probability and Statistics*

[36] Predoiu, Silviu, Gennady Shaikhet, Steven Shreve, 2011, Optimal Execution in a General One-Sided Limit-Order Book, *SIAM Journal on Financial Mathematics* 2(1) 183-212.

[37] Press, William, Saul Teukolsky, William Vetterling and Brian Flannery, 2002, Numerical Recipes in C++: The Art of Scientific Computing, second edition, *Cambridge Unversity Press.*

[38] Reid, R.W. and S.J.Citron, 1971, On Noninferior Performance Index Vectors, *Journal of Optimization Theory and Applications* Vol. 7, No. 1.

[39] Schied, Alexander, and Torsten Schoneborn, 2009, Risk aversion and the dynamics of optimal liquidation strategies in illiquid markets, *Finance and Stochastics*, Vol. 13, No. 2, 181-204.

[40] Smirlock, M and L. Starks, 1985, A further examination of stock price changes and transaction volume, *Journal of Financial Research* 8, 217-225.

[41] Stace, A.W., 2007, A moment matching approach to the valuation of a volume weighted average price option, *International Journal of Theory and Applied Finance* 10, 95-110.

[42] Sutton, Richard S, and Andrew G. Barto, 1998, Reinformcent Learning: An Introduction, *MIT Press*

[43] Yuriy, Nevmyvaka, Michael Kearns, Amy Papandreou and Katia Sycara, 2005, Electronic Trading in Order-Driven Markets: Efficient Execution, *IEEE Conference on Electronic Commerce (CEC).*