University College London

Master Thesis

Optimal Execution Under Nonlinear Transient Market Impact Model

Author:

Weiguan Wang

Supervisor:

Dr. Johannes ruf

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Financial Mathematics
Department of Mathematics

September 2015

Declaration of Authorship

I, Weiguan Wang, declare that this thesis titled, 'Optimal Execution Under Nonlinear Transient Market Impact Model' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ________________________________

Date: ________________________________

UNIVERSITY COLLEGE LONDON

Abstract

Department of Mathematics

Master of Science

Optimal Execution Under Nonlinear Transient Market Impact Model

by Weiguan Wang

We study the optimal strategy to execute a large order in the context that the market impact has a nonlinear transient form. We first review two classic temporary-permanent models. Then we formulate the constrained optimisation problem under the transient model following the paper [11] and discuss the regularity and irregularity of a general market impact model. As we are investigating a nonlinear transient impact model, we resort to sequential quadratic programming (SQP), a numerical optimisation of cost functional. We verify the conclusion of [11] that in the strongly concave instantaneous impact case, the optimal strategy is characterised as several single-term large purchases separated by long-lived yet weak sales in a buy program. This indicates the existence of transaction-triggered price manipulation, and it deteriorates as nonlinearity increases. We innovatively illustrate the trend of optimal strategy as the degree of non-linearity changes. We also investigate the effect of discretisation from a new aspect and find that a higher frequency of trading enables a trader to better exploit the benefit of price manipulation and this benefit is more significant in strongly concave case.

Acknowledgements

I would like to thank Dr. Juan Miguel Montes, my industry supervisor. He suggested this interesting topic to me, which is also of practical importance. He is busying doing his job, however he is still willing to meet me several times and offers me more than two hours every time I met him. During the past three months, he helped me familiarize the problem, discussed with me and pointed out a direction that I can focus on. He is kind, patient and knowledgeable. I really appreciate his contribution to my dissertation.

Moreover, I would like to thank Dr. Johannes Ruf, my academic supervisor. He chose this research area for me. I didn't know much about this area before I started this dissertation. At this moment, this area fascinates me very much. Also he offered useful advice on academic writing and enlightened me on some questions.

Contents

List of Figures

Chapter 1

Introduction

The financial markets have seen the rapid evolution of algorithmic trading. Traders use computer programs to implement various strategies This allows traders and financial institutions to react to changes in market conditions more rapidly and efficiently. In Additional, in financial markets like the US, financial institutions usually play a more important role than individual investors because the institutional investors, such as pension funds and hedge funds, have larger total assets and more advanced strategies. As a result, we have witnessed a substantial increase in the growth of equity trading over the past several decades.

A large financial institution usually has significant ability to affect the market condition, because its trading is normally of higher frequency and larger volume. Thus the trader should carefully choose a proper strategy to execute the order, in terms of the time, velocity and venue, i.e. exchange or dark pool.

Trading cost, often called execution cost because it appears during the execution of trading strategy, consists of bid-ask spread, commission and price impact; these aspects affect the investment performance to a large extent. According to Perold, Andre, and William (1988) [29], in their investigation of a representative fund, they observed that the fund would outperform the market by 20% if it were executed by 'paper' transaction. 'Paper' means that your order can always be fulfilled at any quantity and free of commissions, without affecting the market condition. However, the actual performance of that fund only outperformed the market average by 2%, which is the result of execution cost. This fact implies that the execution cost is unexpectedly large, and strategies that could optimize the cost should be developed, especially for large financial institutions whose trading accounts for a significant proportion of daily volume.

From the literature, there are two major classes of market impact model. The first one separates the price impact into two components: temporary impact, which only affects the trading that triggered it; and permanent impact, which affects the future price dynamics and trading afterwards. The Bertsimas and Lo model (1998) [6] and Almgren and Chriss model (1999) [4] belong to this class. However, empirical data shows that the price impact of executing an order decays with time, as seen in Moro, Esteban and Vicente (2009) [26]. This means the market impact model should have a transient form. This class of impact model has two components: an instantaneous impact function and a decay kernel.

From empirical study, the instantaneous impact function in a transient model is strongly concave, while the decay kernel is asymptotically a power law function. The goal of liquidating a large position in a finite time horizon imposes a constraint on the optimization problem. Due to the nonlinearity of cost functional, finding the solution to this constrained optimization problem is challenging. For a practical trading strategy, a solution to the discretized version of the problem is to be found. Usually numerical optimization techniques are required to deal with this kind of problem. One numerical algorithm that is suitable for solving this non-linear constrained optimization problem is Sequential Quadratic Programming. This state-of-the-art optimisation technique, in terms of speed, accuracy and percentage of successful convergence, is based on derivative of Lagrangian function and constraint. It tackles the optimization problem by constructing a quadratic sub-problem and converging to the local minimum iteratively. By the local sequential quadratic programming, the convergence to a local minimum is guaranteed. In order to search the global minimum within the feasible region, we implement the Multistart scheme. This scheme runs the local sequential quadratic programming from each start point and then obtains multiple local minima. The one with the lowest objective value among the local minima is regarded as the global minimum.

We obtain the optimal liquidating strategy under different market conditions and discretizations by using the Multistart SQP. We examine the optimal strategies in cases where the instantaneous impact function is linear, slightly concave and strongly concave, respectively. In each case, we then characterize the behaviour of optimal strategies. One feature we should pay attention to particularly is the existence of price manipulation. The weak form of price manipulation is transaction-triggered price manipulation, which can reduce execution cost by intermediate selling (buying) during a buy (sell) order. Price manipulation is disallowed and the existence of price manipulation in an optimal strategy implies that the model needs to be further investigated and our method calls for improvement.

This dissertation is organized as follows. In Chapter 2, we review two classes of market impact model by illustrating their representative models in detail. In Chapter 3, we explain how to formulate this constrained optimization problem under the transient impact model and discuss the regularity and irregularity of the transient model. We also illustrate the functional form of each component and present the discretized expression of the problem. In Chapter 4, we explain the local sequential quadratic programming algorithm and Multistart with parallel computing scheme used to obtain the optimal trading strategy. In Chapter 5, we present the numerical results given by SQP and characterize the optimal strategy to discuss the existence of price manipulation. In Chapter 6, we summarize the findings, mention the weakness of our method, and possible future improvement.

Chapter 2

Market impact model

The relation between order flow and price changes has attracted considerable attention in recent years(see Hasbrouck 2007 [16]; Bouchaud *et al.*, 2004 [7]). This is partly due to the increasing tendency towards a full automation of exchanges and the discovery of new statistical regularities of the microstructure of financial markets.

2.1 Market price impact

Market price impact refers to the fact the execution of a large order will influence the prevailing market price. From market observation, usually the asset price will tend to increase if a large buy order is posed, and a large sell order will probably decrease the asset price. This impact is usually adverse to the market participant, creating additional cost for executing the order. Studies (e.g. see Brinson, Hood and Beebower (1986) [8]) have shown that portfolio managers are unable to match the performance of various passive benchmarks and under-perform by about one or two percent. The researches on the market price impact have shown that the price impact of block trades is not trivial and to some extent explains the poor performance of portfolio managers. However in some cases, the participant can make use of this impact. For instance, the central bank can sell the government bond to increase the corresponding interest rate.

The market price impact of a large trade is more pronounced in an asset market without high liquidity. In this kind of market, there is insufficient liquidity to permit the immediate execution of a large order without eating into the limit order book. Thus in most cases, it is not optimal to immediately execute the whole order in this kind of market.

The particular importance of the market price impact can be seen from the famous event "Flash Crash" of May 6, 2010. According to CFTC-SEC (2010), an important contribution to the occurrence of this event is the rapid execution of a large E-Mini contract in just 20 minutes. By CFTC-SEC (2010) [9] : On May 6, 2010, the prices of many U.S.-based equity products experienced an extraordinary rapid decline and recovery.[...] At 2:32 p.m., a large fundamental trader initiated a sell program to sell a total of 75,000 E-Mini contracts. On normal occasion when this trader initiated the sell order of similar size, the automated execution algorithms took into account price, time and volume, and it took more than 5 hours for the order to be executed. However, on May 6, when the market was already under stress, the sell algorithm chosen by the large trader only target trading volume, without considering price or time, executed the sell program extremely rapidly in just 20 minutes.

2.2 Optimal execution strategy

In the presence of market price impact, we need to find an optimal execution strategy in order to minimize the adverse price impact due to our trading. If we execute the whole order immediately, we are incurred with huge execution cost but we do not need to bear any future price uncertainty. In another case in which we execute the order gradually, we are likely to incur smaller execute cost but with higher future price uncertainty. Thus according to Macey and O'hara (1996) [25], there is no clear definition of best execution. The position of a typical financial institution is usually relatively large, so their trading will influence the market. As a result, they almost always choose to divide the whole order into smaller parts, trading gradually, to mitigate the market perturbation caused by their trading. For this reason, optimal execution is not a single amount to trade. It should be a strategy, i.e. a sequence of orders that execute the whole position during the horizon and are subject to the change of market conditions.

In order to find the best execution strategy, one starts with setting up a model that both describes the evolution of the asset price and how trades affect the asset price dynamics. Then we need to specify a criteria to judge what strategy is better than the others. In addition, one needs to choose a certain criteria in order to characterize the preference of an investor. In this sense, the optimal strategy is the minimizer of the criteria among all feasible trading strategies under certain constraints.

There are various kinds of risk criteria, which we summarize as follows and is based on Gatheral and Schied (2013) [14]:

- *Mean-variance* optimization. This risk criteria is similar to the mean-variance optimization in portfolio choice problem. It aims to minimize the functional of the form

  $$\mathbb{E}[\mathcal{R}_{\mathcal{T}}(\mathcal{X})] + \lambda var(\mathcal{R}_{\mathcal{T}}(\mathcal{X})),$$

  where $\mathcal{R}_{\mathcal{T}}$ donotes the revenue of an order execution strategy, the $var(Y)$ denotes the variance with respect to measure $\mathbb{P}$ of a random variable Y, and $\lambda \geq 0$ is a risk aversion parameter. This problem is investigated by Almgren and Chriss(1999)[3, 4]. And in Lorenz & Almgren (2011) [24], in contrast to standard formulation in which the mean-variance optimal strategy are static, they show that substantial improvement is possible by using dynamic trading strategies and that the improvement is larger for large initial positions. In Konishi and Makimoto (2001) [21], they define the measure of opportunity cost as the standard deviation of the asset price movement and minimize the transaction cost, the sum of the execution cost and the opportunity cost.

- *Expected cost minimization* corresponds to the minimization of the functional of the form

  $$\mathbb{E}\left[\sum_{t=1}^{T} P_t S_t\right]$$

  where P_t is the price of the asset at time t, which follows a specified distribution under trading impact, and S_t is the amount of asset to be executed during the tth trading interval. In Bertsimas and Lo (1998) [6], they show that given a fixed block $\bar{S}$ of shares to be executed within a finite number of periods T and an impact function that captures the impact of trading on asset price dynamics under different market condition, an optimal sequence of trades exists, which minimizes the expected cost of executing $\bar{S}$ within T periods.

- *Expected-utility maximization* corresponds to the maximisation of the functional form
$$\mathbb{E}\left[u(\mathcal{R}_T(X))\right],$$
where $u : \mathbb{R} \longmapsto \mathbb{R}$ is a utility function, which by definition is concave and non-decreasing. The advantage of this kind of functional is the time consistency, in contrast to the mean-variance functional. In Schied and Schöneborn [31], they want to find the optimal strategy that maximize the utility of liquidating a block of stock. They address this question in the continuous-time liquidity model introduced by Almgren [5] where the impact cost is linear and time horizon is finite.

2.3 Temporary and permanent price impact

One class of the market impact models that have been proposed so far assumes that the price impact consists of two components. The first component is the temporary impact. This kind of impact only affects the individual trade that has triggered it. By Almgren and Chriss [4], the economic intuition behind the temporary effect is that a trader plan to buy n_k units of asset during the interval t_{k-1} and t_k. The trader chooses to divide the large order into smaller ones in order to better exploit liquidity. As the trader is buying asset, the price of the asset increases gradually, because the buy orders exhaust the liquidity at current level of price thus have to trade at more adverse level. They assume that this change of price is temporary and the liquidity will restore and a new equilibrium will appear. The second component of this class of model is the permanent impact, which affects all the trades after the one triggered it.

2.3.1 The Bertsimas and Lo model

In Bertsimas and Lo model [6], they consider a problem as follows: an investor seeking to acquire a large position of $\bar{S}$ shares of stock in a finite horizon $[0, T]$. Since the demand curve for even the most liquidate asset is not perfectly elastic, the trader will divide the order into smaller packages distributed over the course of the finite time. Presumably, the way in which it is optimal to trade depends on the way in which the asset price evolves and how the trade affects the price dynamics, i.e. price impact.

With little loss of generality, time is measured in number of units. The unitary length can be arbitrarily long that is practically sensible. For instance, if the whole order has to be executed with in one day from market open at 9:30 a.m. to 4:00 p.m., setting the length of a period to be 15 minutes yields $T = 26$. In other cases, the combination of T and length of period can be adjusted accordingly.

In the following, denote by S_t the number of shares to be executed in period t, where $t = 1, ..., T$ and by P_t the corresponding execution price. In the model, Bertimas and Lo minimize the expected execution cost under the liquidation constraint, which can be expressed as follows:

$$\min_{S_t} \mathbb{E}\left[\sum_{t=1}^{T} P_t S_t\right],$$

subject to the constraint

$$\sum_{t=1}^{T} S_t = \bar{S}.$$

In Bertsimas and Lo model [6], they investigate many kinds of price dynamics for the asset P_t. In the literature, the law of motion can be decomposed into two parts: the dynamics of price without the trading impact and the impact of trading S_t units on the execution price P_t. In the very basic model, the dynamics of P_t in the absence of trading impact follows an arithmetic random walk, which can be expressed as

$$P_t = P_{t-1} + \epsilon_t, \mathbb{E}\left[\epsilon_t \mid S_t, P_{t-1}\right] = 0.$$

In the simplest setting, the impact due to trade is a linear function of trade size, with an amplifying parameter θ, so that the impact is added to the non-impact price P_{t-1}, generating the effective price for trading S_t. Then the price dynamics of P_t can be written as:

$$P_t = P_{t-1} + \theta S_t + \epsilon_t, \theta > 0, \mathbb{E}\left[\epsilon_t \mid S_t, P_{t-1}\right] = 0,$$

where ϵ_t are identically and independently distributed normal random variable.

There are a few implausible empirical implication of this kind of specification: independent increment, permanent price impact, linear impact and positive probability of negative price. However it still motivates us of a more complicated and realistic models.

Based on all these assumptions, the solution to this problem can be obtained by stochastic dynamic programming. Bertimas and Lo find that the optimal execution, i.e. the solution to the optimization problem is

$$S_1^* = S_2^* = ... = S_T^* = \bar{S}/T,$$

which means the best execution strategy is simply trading evenly over the course of the finite horizon.

This surprisingly simple strategy as the best execution is because the price impact does not depend either on the current price P_{t-1} or the remaining order to be executed. Thus the package executed in each period should be independent and same.

Then Bertimas and Lo make an improvement to the very basic model. They add another component X_t which is serially-correlated to reflect the correlation between serial prevailing prices P_t and affect the execution price linearly. Thus, keeping the impact function linear in S_t,

$$\begin{aligned} P_t &= P_{t-1} + \theta S_t + \gamma X_t + \epsilon_t, \theta > 0 \\ X_t &= \rho X_{t-1} + \eta_t, \eta \in (-1, 1) \end{aligned}$$

where ϵ_t and η_t are independent white noise processes with mean 0 and variance σ_ϵ^2 and σ_η^2, respectively.

Bertimas and Lo argue that the presence of X_t in the price dynamics captures one of the two kinds of information, one of which is the changing market condition and the other of which is private information about the security. For the first kind, X_t may be the return on market index or a common factor of most assets being considered. Here, ρ measures the sensitivity of the particular asset to that factor. In the other case where X_t represent the private information, this indicates that the trader can have an individual judgement on the future evolution of the asset price and thus makes use of this judgement to construct the optimal strategy. Again by dynamic programming, Bertimas and Lo obtain a solution of the following form

$$S^*_{T-i} = \sigma_{w,i} W_{T-i} + \sigma_{x,i} X_{T-i}, \tag{2.1}$$

for $i = 0, 1, ..., T-1$ where

$$\sigma_{w,i} = \frac{1}{i+1}, \sigma_{x,i} = \frac{\rho b_{i-1}}{2a_{i-1}}$$

and

$$\begin{aligned}
a_i &= \frac{\theta}{2}\left(1 + \frac{1}{i+1}\right), a_0 = \theta, \\
b_i &= \gamma + \frac{\theta \rho b_{i-1}}{2a_{i-1}}, b_0 = \gamma, \\
c_i &= \rho^2 c_{i-1} - \frac{\rho^2 b_{i-1}^2}{4a_{i-1}}, c_0 = 0, \\
d_i &= d_{i-1} + c_{i-1}\sigma_\eta^2, d0 = 0.
\end{aligned}$$

From solution (2.1), we can see that the strategy consists of two parts, the first part is dividing the remaining position evenly and the second part can be seen as an adjustment due to the existence of the serially-correlated information. Thus if the correlation coefficient $\rho_0 = 0$, this execution strategy will reduce to the naive strategy, as the information X_t is totally unpredictable.

This linear impact model with information has several defects. For example, the impact is still permanent, which contradicts some empirical evidence that shows the impact should be separated into permanent and temporary components (see Chan and Lakonishok [10]). Moreover, the percentage change decreases as the asset price increases in this model, which also contradicts empirical data (see Loeb (1983) [23]). In order to improve this model, they proposed linear-percentage temporary impact model.

Let the price at time t P_t be the sum of two parts, the no-impact price $\tilde{P}_t$ and the transaction-triggered impact Δ_t. For the no-impact price $\tilde{P}_t$, a plausible and observable proxy is the mid-point of the bid-ask price. Also, to ensure the non-negativity of asset price, they assume a geometric Brownian motion for the no-impact price $\tilde{P}_t$, which is expressed below,

$$\tilde{P}_t = \tilde{P}_{t-1} exp(Z_t)$$

where Z_t here is IID normal random variable, i.e. $Z_t \sim N(\mu_t, \sigma_t^2)$. For the price impact part, they assume that the percentage of the price impact to no-impact price $\tilde{P}_t$ is a linear function of transaction size S_t and serially-correlated information indicator X_t. It can be expressed as follows:

$$\begin{aligned} \Delta_t &= (\theta S_t + \gamma X_t)\, \tilde{P}_t \\ X_t &= \rho X_{t-1} + \eta_t \end{aligned}$$

where η_t is the white noise process that each has mean 0 and variance σ_η^2. As above, variable X_t is an indicator of private information or market condition, and it is set to be an $AR(1)$ process to incorporate predictability. The parameter ρ and θ represent the sensitivity of impact Δ_t to information X_t and trade size S_t, respectively.

The LPT specification has several advantages over the linear impact model. First, under some restrictions on impact Δ_t, the non-negativity of asset price P_t is guaranteed. Second, the percentage price impact increases linearly with the trade size. Third, by decomposing the asset price into no-impact price and price impact, the price impact becomes temporary, which mean it does not affect future prices.

In Bertimas and Lo 1998 [6], they give a close-form solution for the LPT specification, in which the optimal execution strategy is a combination of linear function of the state variable X_{T-k}, the information indicator, and W_{T-k}, the order to be executed.

2.3.2 The Almgren and Chriss model

The model proposed in Almgren and Chriss (1999) [4] considers the aim of minimization the combination of uncertainty and transaction cost arising from temporary and permanent impact. Taking the volatility as a penalty for late execution of the order, they construct an efficient frontier for a simple linear cost model in the class of time-dependent liquidation strategy, which is in a similar way to the portfolio choice problem.

In contrast to Bertimas and Lo (1998) [6] where they define the optimal execution as the strategy which minimize the expected execution cost, Almgren and Chriss works in a more general framework. They aim to maximize the utility of trading revenue. They define utility as a linear combination of expected execution cost and variance of trading. Adding the variance of execution cost into the determination of optimal strategy has an economic justification. Suppose, the trader can either choose to execute the whole order immediately or to execute gradually over the horizon. Especially in trading illiquid and volatile asset, immediate trading has very high execution but with no future uncertainty. Trading gradually will probably has smaller expected cost, but the trader has to bear the risk of future price fluctuation.

How to penalize for deferring the trading is a rather subjective issue and it should depend on how risk-averse that investor is. For example, if we are considering an investor with high risk tolerance and long trading horizon like a mutual fund, the penalty can be set to a small constant times variance of cost. For investor who is highly risk-averse, the penalty is set to be a large constant times variance of cost.

Consider a trading strategy for liquidating a single security. Suppose we have to liquidate a block of X units of a security before time T. The trading is to be executed in N equal length time intervals, i.e. $\tau = T/N$. Then discrete times $t_i = i\tau$, where $i = 0, 1, ..., N-1, N$. A trading trajectory is defined as a sequence $x_0, x_1, ..., x_N$, where x_i is the number of asset held at time t_i. Liquidation condition requires $x_0 = X$ and $x_N = 0$.

They define a new variable by the equation $n_i = x_{i-1} - x_i$. Then n_i is the units of asset sold in interval $[t_{i-1}, t_i)$, and we have the following relation between x_i and n_i:

$$x_i = X - \sum_{j=1}^{i} n_j = \sum_{j=i+1}^{N} n_j, i = 0, 1, ..., N.$$

This can be extended to more general framework where there is simultaneous buying and selling several securities. They distinguish two kinds of trading strategy, static and dynamic. Static strategy is determined prior to trading, using the information available at t_0, while dynamic strategy depends on the information up to and including time t_{k-1}.

Suppose that the initial price of the security is P_0, then the initial value of the position is XP_0. In this model, the evolution of price of the security depends on two exogenous factors and one endogenous factor. The word 'exogenous' means that it is independent of our trading and is determined by the demand and supply of the security in the market. They are assumed to be drift and volatility. The endogenous factor is the impact triggered by the trading, which is called market impact. As in Bertimas and Lo [6], it distinguishes between temporary impact, which only influence the trade that triggered it, and permanent impact, which affects all the remaining trades equally and will give rise to a new equilibrium price.

Again they assume discrete arithmetic random walk in order to remain the tractability:

$$P_i = P_{i-1} + \sigma\tau^{1/2}\xi_i - \tau g(\frac{n_i}{\tau}),$$

for $i = 0, 1, ..., N$. Here σ refers to volatility, ξ_i is a normally distributed random variable with mean 0 and unit variance. $g(\cdot)$ is a function of average trading rate during time interval from t_{i-1} to t_i, and it is actually a measure of permanent impact due to our trading. Note that there is no drift term in the expression, meaning that we have no information regarding the evolution of the price.

To model temporary price impact due to trading, they introduce a linear function of average trading rate, $g(n_i/\tau)$, describing the temporary price drop (or increase) per share due to trading at average rate n_i/τ in $[t_{i-1}, t_i)$. Hence the effective price for the order during this interval is

$$\tilde{P}_i = P_{i-1} - h(\frac{n_i}{\tau}).$$

We can see that $h(\cdot)$ does not appear in the expression for P_i, which implies temporary impact does not affect the next equilibrium price P_i.

Then they define the capture of a trajectory as the total revenue obtained from executing the whole position. This is the sum of each trade size times the corresponding execution price, and they arrive at

$$\sum_{i=0}^{N} n_i\tilde{P}_i = XP_0 + \sum_{i=1}^{N}\left(\sigma\tau^{1/2}\xi_i - \tau g(\frac{n_i}{\tau})\right) x_i - \sum_{i=1}^{N} n_i h(\frac{n_i}{\tau}). \tag{2.2}$$

The permanent impact function $g(\cdot)$ and temporary $h(\cdot)$ in equation (2.2) can be chosen to reflect suitable market microstructure. We can see from the expression that the first term is the initial book value of position and the following three terms are caused by three factors. The term $\sum \tau g(n_i/\tau)x_i$ is the loss caused by the permanent impact. The term $\sum \sigma\tau^2\xi_i x_i$ is the total impact of volatility. And the term $n_i h(n_i/\tau)$ is the temporary price decrease due to selling at average rate n_i/τ, which only affects the portion that has triggered it.Then they define implementation shortfall as the difference between $XP_0 - \sum n_i\tilde{P}_i$, which is the same as Perold (1998) [29].

If the variance of trading cost is taken into account, a rational trader will always seek to minimize the expectation of shortfall for a given level of variance of shortfall. Almgren and Chriss (2001) [4] define a trading strategy to be optimal or efficient if there is not a strategy which has lower expected shortfall for the same or lower variance. Thus expressed mathematically, we may construct the efficent frontier by solving the constrained optimization problem as follows

$$\min_{x:V(x)\leqslant V_*} E(x)$$

for a maximal variance V_*. Since this is a convex optimization problem where the constraint $\{V(x) \leqslant V_*\}$ and objective function $E(x)$ are both convex, there must be a local minimizer $x_*(V_*)$ which is also global minimizer. Thus the optimal strategies has a single parameter V_*, maximal variance. This family is called the efficient frontier of optimal trading strategy.

To solve this constrained optimization problem, they introduce a Lagrange multiplier λ to change the constrained problem to an unconstrained on by penalty function. Then the problem becomes

$$\min_x \left(E(x) + \lambda V(x)\right).$$

If $\lambda > 0$, this will ensure a unique optimal solution as the objective function $E+\lambda V$ is strictly convex.

In the paper, the temporary impact function is set to

$$h(\frac{n_i}{\tau}) = \epsilon sgn(n_i) + \frac{\eta}{\tau}n_i,$$

and the permanent impact function is set to be

$$g(v) = \gamma v$$

Then the necessary condition for the minimizer of objective function $E(x)+\lambda V(x)$ is that the gradient is zero. Thus they obtain a linear difference equation,

$$\frac{1}{\tau^2}(x_{j-1} - 2x_j + x_{j+1}) = \tilde{\kappa}^2 x_j$$
$$\tilde{\kappa}^2 = \frac{\lambda\sigma^2}{\tilde{\eta}} = \frac{\lambda\sigma^2}{\eta(1 - \frac{\lambda}{2\eta})}$$

They obtain a specific solution with constraint $x_0 = X$ and $x_N = 0$, which is

$$x_j = \frac{sinh(\kappa(T - t_j))}{sinh(\kappa T)} X, j = 0, .., N.$$

The solution above is static optimal solution for a given level of variance of cost. This static strategy ignores the arrival of news. So they continue to investigate how close these static strategy is to being globally optimal. And they investigate three types of information. The first is serial correlation. They proved that the improvement of incorporating serial correlated information into price dynamics is small. More importantly, the improvement does not depend on portfolio size. The second kind of information is scheduled news. They proved that the anticipated news can temporarily shift the parameters of the dynamics of the asset price significantly. And the global strategy turns out to be piecewise static. It means we can obtain the optimal strategy at time t_0, having the same trajectory up to the scheduled release time of the news, and according to the outcome of this anticipated news, we decide at that time which stategy we should take. The third kind of information is unanticipated news. According to Almgren and Chriss [4], if one makes the simplifying assumption that the new information is either scheduled or anticipated, the optimal strategy is always same as static trading before the arrival of news. After the news is released, trading strategy becomes the static optimal strategy which adjusts to parameter changes in price dynamics.

2.4 Transient price impact

The transient model implies that the market impact of trading order decays with time. This class of model consists of two components. The first one is the impact as a function of trading rate, and the second component is the decay kernel. There is empirical study verifying this feature of market impact, see Moro, Esteban and Vicente (2009) [26]. They study the impact of large trading order that is executed incrementally, which they call hidden order, and they find the market impact is strongly concave in London stock exchange and Spanish stock market. Moreover, they find the market impact grows according to a power law as time goes and the impact revert to $0.5 \sim 0.7$ times its peak value. This type of model has two component.

2.4.1 Linear transient model

One of the first linear transient models is proposed by Obizhaeva & Wang (2013) [28]. In their model they develop a general framwork for a limit order book to capture the dynamics of supply/demand. They show that the optimal strategy of execution does not depend on static property such as bid-ask spread, rather it depends on dynamic property like the resilience after a trade.

In the extended model of Obizhaeva & Wang (2013) [28], a market order which trades dX_t shares of asset at time t is put on the limit order book, where the limit order follows a uniform distribution with a density of limit order q. It means there are qdP limit orders available in the price interval from P to $P + dP$, and it does not depend on current price P. According to Gathetal [14], in a buy program, $dX_t > 0$, after putting the market order in the limit order book, it will drives the price up to

$$P^X_{t+} = P^X_t + qdX_t,$$

here the superscript means that the price is the one under trade impact. The decay component of transient model is modeled by a function called decay kernel, which is a mapping $G : \mathbb{R}_+ \mapsto \mathbb{R}_+$. Then we say the price impact created at time t by placing the market order dX_t is $qdX_t = G(0)dX_t$. As time goes by, at time $s > t$, the trading impact of order dX_t decays to $G(s - t)dX_t$. Hence, the cumulative price impact until time t created by placing several orders dX during interval $[0, t]$ is the sum of the impact decayed to time t over that period, which can be expressed as

$$dP_t^X = P_t^0 + \int_0^t G(t-s)dX_s$$

By Gatheral (2012) [15], one can show that the expected cost of a general order execution strategy is

$$\mathbb{E}\left[\mathcal{C}_T(X)\right] = \frac{1}{2}\mathbb{E}\left[\int_0^T \int_0^T G(\mid t-s \mid)dX_s dX_t\right];$$

From a result obtained by Gatheral (2012) [15], we have

Theorem 2.1. *Let the decay kernel G be a nonincreasing convex function. Then there exists a unique optimal strategy X^* for each combination of X_0 and T. Moreover, X_t^* is a monotone function, which means there is not transaction-triggered price manipulation.*

Chapter 3

Formulation of problem

3.1 Optimisation problem

In Gatheral (2010) [13], they propose that an absolutely continuous order execution strategy π results in a price process of the form

$$S_t^\pi = S_t^0 + \int_0^t f(\dot{x_s}) G(t-s) ds.$$

Here, S_t^0 denotes the price without trading impact at time t, S_t^π denotes the price under impact of trading strategy π . $f(\dot{x}_s)$ represents the instantaneous impact when placing an order x_t at time t, $\dot{x}_s$ is the rate of trading, i.e. derivative of position X with respect to time, and $G(\cdot\,)$ represent the decay kernel as before.

In the Curato, Gatheral and Lillo (2014) model [11], an stochastic term modeled by Brownian motion is added, which can be expressed as follows,

$$S(t) = S_0 + \int_0^t f(\dot{x}(s)) G(t-s) ds + \int_0^t \sigma dW(s), \tag{3.1}$$

where $\dot{x}(t)$ is the rate of trading, number of shares trading in each unit of time, which is the first derivative of x with respect to time t, $v(t) = \dot{x}(t) = dx/dt$. And as usual $f(\dot{X}_s)$ represents the instantaneous impact when placing an order X_t at time t and $G(\cdot\,)$ represent the decay kernel. σ is the volatility and $W(t)$ is the standard Brownian motion. From the above expression we can see that the drift term is actually the accumulative impact at time t of all orders traded from time 0 to t.

The strategy we aim to find is the one $\pi = x(t)_{t\in[0,T]}$ that minimizes the expected execution cost and liquidates our total position X in a finite time interval $[0,T]$. Thus the expected cost of execution by following strategy $\pi = x(t)_{t\in[0,T]}$ is

$$\begin{aligned} C(\pi) &= \mathbb{E}\left[\int_0^T \dot{x}(t)(S(t)-S(0))dt\right] \\ &= \int_0^T \dot{x}(t)\mathbb{E}\left[(S(t)-S(0))\right]dt \\ &= \int_0^T v(t)\int_0^t f(v(t))G(t-s)dsdt. \end{aligned} \tag{3.2}$$

This is because the expectation of Wiener process is 0 and the trading strategy $\pi = x(t)_{t\in[0,T]}$ is deterministic as it is a static strategy, so the expectation can be put into the integral, yielding the expression as above. Moreover the liquidation constraint can be expressed as a simple integral,

$$\int_0^T v(t)dt = X. \tag{3.3}$$

This formulation of execution cost corresponds to 'implementations shortfall' as mentioned in Perold (1988) [29]. Moreover, in the Silviu, Gennady and Steven (2011) [30], they argue that a statically optimal strategy is also a dynamically optimal strategy if the expected execution cost depends on $\int_0^T v(t)S(t)$ only and $S(t)$ is a martingale. So the static optimal solution from cost function (3.2) with constraint (3.3) is also dynamically optimal strategy in this setting.

3.2 Regularity and irregularity

The functional forms of $f(\cdot)$ and $G(\cdot)$ fully specify the model. But not all combination of these two function is feasible due to issues from both financial and mathematical aspects.

3.2.1 Regularity

A minimal requirement for regularity is that this constrained optimisation problem admits an minimizer. Another requirement is that the strategy $\pi = x(t)_{t\in[0,T]}$

should be monotone function. For instance, a sell program should not involve intermediate sell programme, as this is considered illegal. In Gatheral (2013) [14], they argue that this regularity should be independent of investor specific risk-aversion. This means some risk functional can not be used to define the regularity like expected utility. It is feasible to define regularity using expected cost. In addition, they distinguish between the effects of price impact from profitable investment strategies that can arise via trend following. Hence, it is a standard assumption in literature that the price under impact S_t^X follows a driftless dynamics. Another reason for that is we are usually considering a relatively short period of horizon, making the effect of drift neglectable.

3.2.2 Irregularity

From here onwards, we introduce three kinds of model irregularities that may appear in the impact mode. The existence of these irregularities means the model is not well-specified.

The first kind is price manipulation.

Definition 3.1. (Price manipulation). A round trip is a trading trajectory that satisfies $x_0 = 0$ and $x_T = 0$. A price manipulation strategy is a round trip satisfying additional condition that

$$\mathbb{E}\left[\mathcal{R}_T(\pi)\right] > 0.$$

From this definition, we can exploit the existence of price manipulation to decrease the execution cost. For a risk-averse investor seeking to minimize risk functional $E(\mathcal{C}(\pi)) + \lambda V(\mathcal{C}(\pi))$, they may use this round trip strategy during some time interval to decrease this objective function. This probably will give out an different optimal trading strategy. However for a risk-neutral investor who seeks to minimize objective function $E[\mathcal{C}(\pi))]$, the possibility of price manipulation may lead to arbitrarily small value of objective function. It means for a risk-neutral investor, the existence may result in the non-existence of optimal strategy.

The definition of price manipulation resembles the definition of arbitrage. This implies a relation between price manipulation and arbitrage.

Definition 3.2. A portfolio is called an arbitrage portfolio if $V_0 = 0$ and $\mathbb{P}[V_T \geqslant 0] = 1$ and $\mathbb{P}[V_T > 0] \neq 0$ under probability measure $\mathbb{P}$.

There is a link between arbitrage and price manipulation. According to Huberman and Stanzl (2004) [17], in some models, repeating price manipulation may lead to a weak form of arbitrage, called quasi-arbitrage. But there is difference between price manipulation and arbitrage. The definition of arbitrage is in the 'almost-surely' sense. It implies that when pricing a derivative, we are looking for a portfolio that exactly replicate the payoff of that derivative to be price. If such portfolio exist, the initial value of that price should be exactly the same as the market price of that derivative, otherwise, there is an arbitrage opportunity. However, price manipulation is in the 'average' sense. The existence of price manipulation does not ensure that every trade can improve the trader's situation, but in average, it will decrease the execution cost. When we are looking for an optimal execution strategy, we are actually looking for a minimizer of a certain risk functional for a particular investor. And this fact should be independent of any investor's risk preference, implying it should be defined in risk-neutral way. Thus it is reasonable to define regularity condition for any impact model in terms of expected cost.

However, in Alfonsi, Schied and Slynko (2012) [2], by analyzing model with linear instantaneous and permanent impact components, they discover another kind of price manipulation, which they call transaction-triggered price manipulation.

Definition 3.3. (Transation-triggered price manipulation) A market impact model adimits transaction-triggered price manipulation if the revenue of a sell (buy) program can be increased by intermediate buy (sell) orders. Mathematically, there exists X_0, $T > 0$, and a corresponding order execution strategy $\tilde{\pi}$ such that

$$\mathbb{E}[\mathcal{R}_T(\tilde{\pi})] > sup\{\mathbb{E}[\mathcal{R}_T(\pi)] | \pi \in \Pi \text{ is a decresing (increasing) function of time.}\}$$

where Π is the set of all feasible strategies.

Alfonsi, Schied and Slynko [2] prove that the price impact must decay as a convex non-increasing function of time to ensure the nonexsitence of this kind of transaction-triggered price manipulation along with standard price manipulation.

3.3 Functional form of components

Some empirical studies have given some clues on the functional form of two component, $f(\cdot)$ and $G(\cdot)$, which completely specifies the model.

The instantaneous impact function $f(\cdot)$ is strongly concave. The evidence comes from Moro, Esteban and Vicente (2009) [26]. They study the impact of large trading order that is executed incrementally, which they call hidden order, and they find the market impact is strongly concave in London stock exchange and Spanish stock market. Also by Lillo, Farmer and Mantegna [22], based on data from New York Stock Exchange, on a double-logarithmic scale, the slope of each curve varies from roughly 0.5 for small transactions in higher-capitalization stocks, to about 0.2 for larger transactions in lower-capitalization stocks. This means a power law is suitable for describing the behavior of instantaneous price impact.

In addition, Bouchaud, Gefen, Potters and Wyart [7], by analyzing the trade at Paris Bourse, the decay kernel is found to be asymptotically a power law function

$$G(\tau) \sim \frac{1}{\tau^{\gamma}}.$$

The nonlinearity presenting in both the decay kernel and instantaneous impact function give rise to the possible existence of price manipulation as defined in 3.1. We first present a proposition obtained by Gatheral and Schied (2013) [14],

Proposition 3.4. *Assuming the price process as equation* (3.1)*, consider a model with general nonlinear instantaneous impact function* $f(\cdot)$ *and nonincreasing decay kernel* $G(\cdot)$ *with* $G(0) := \lim_{t\downarrow 0} G(t) < \infty$*, then this model adimits price manipulation.*

For this reason, we should consider decay kernel of the form $G(t-s) = (t-s)^{-\gamma}$, which means the decay kernel is singular only at the origin. In Gatheral (2010) [13], the regularity requirement of no price manipulation imposes a restriction on the possible combination of parameter γ and σ. In detail, for the power law instantaneous function of form $f(\dot{\pi}) \propto sign(\dot{\pi})\,|\dot{\pi}|^{\sigma}$ and a power law decay function of form $G(t-s) = (t-s)^{-\delta}$, the necessary conditon for the absence of price manipulation is,

$$\gamma + \delta \geqslant 1, \gamma \geqslant \gamma^{*} = 2 - \frac{\log 3}{\log 2} \simeq 0.415$$

As it is the necessary condition, it does not preclude the possibility of price manipulation. It has been proved that there is a model which satisfies the necessary condition but still admits price manipulation. However, as in Curato, Gatheral and Lillo (2014) [11], we only consider the situation where this necessary condition is satisfied.

3.4 An approach of solving the problem

In the case when $\delta = 1$, this problem reduces to optimization with linear impact function. This problem has been well studied. A proposition is proposed by Gatheral [14], which implies,

Theorem 3.5. *Suppose $G(\cdot)$ is positive definite. Then π^* is the minimizer of cost function $C(\pi)$, if and only if there exist a constant λ such that π^* satisfies the following*

$$\int_0^T G(|t-s|)d\pi(s) = \lambda$$

The solution for the case where $G(t-s) = (t-s)^{-\gamma}$ is given by equation (2.8) and (2.9) in Curato, Gatheral and Lillo (2014) [11].

3.4.1 Stationarity condition

The optimization problem for nonlinear case is mathematically more complicated. A progress has been made by Dang (2014) [12]. In the paper of Curato, Gatheral and Lillo (2014) [11], they argues that given $f \in C^1(\mathbb{R})$ and $G \in L^1[0,T]$, for the class of functions x satisfying

- x is absolutely continuous on (0,T),
- $f \circ v \in L^1[0,T]$,

the necessary condition for the stationarity of the functional of equation (3.2) holds:

$$\int_0^T G(|t-s|)F(v(s),t)ds = \lambda, \tag{3.4}$$

where

$$F(v(s),t) = \begin{cases} f(v(s)), & s \leqslant t \\ v(s)f'(v(t)), & s > t. \end{cases} \tag{3.5}$$

We note that apart from the non-linearity from $G(|t-s|)$, $F(v(s),t)$ is also a source of non-linearity. In fact, $F(v(s),t)$ also depends on f', the first derivative of instantaneous impact function, i.e. the response of impact to trading rate, which involves the future trading rate v in the equation. And equation (3.4) implies that it can not be converted into a weakly singular nonlinear Fredholm of the first kind for the trading rate, where there should be no interaction between present and future times, by Curato (2014)[11].

3.4.2 Discretized expression

The following derivation of disvretized expression is based on Curato, Gatheral and Lillo (2014) [11]. Applying discretized homotopy analysis method to equation (3.4), we first split the time interval $[0,T]$ into N subinterval of equal length. Denote by t_i the times, where $t_i = iT/N, i \in \{0,1,...,N\}$. This will give non-linear system of N equations in the variables $v_i = v(t_i)$ where $i \in \{1,...,N\}$

$$\sum_{j=1}^{N} G_{ij}F_{ij}(v) = \lambda.$$

We note here the index i refers to the time times, for each fixed i there should be one equation. And the index j here refers to the discretization of integral interval. The nonlinear function $F(\cdot\,)$ of equation (3.5) becomes a matrix of dimension N by N

$$F_{ij} = \begin{cases} f(v_j), & j \leq i \\ v_j f'(v_i), & j > i. \end{cases}$$

The decay kernel $G(t-s)$ becomes a Toeplitz real symmetric N by N matrix given by

$$G_{ij} = \int_{t_{i-1}}^{t_i} \int_{t_{j-1}}^{t_j} G(|t-s|)dsdt$$

For our case when the decay kernel follows a power law $G(\tau) = \tau^{-\gamma}$, we have $s \in [t_{j-1}, t_j]$ and $t \in [t_{i-1}, t_i]$. As $i > j$, then $t_j \leq t_{i-1}$ and $s \leq t$. Thus by doing this simple integration we have for $i > j$

$$\begin{aligned} G_{ij} &= \int_{t_{i-1}}^{t_i} \int_{t_{j-1}}^{t_j} (t-s)^{-\gamma} ds dt \\ &= \frac{1}{(1-\gamma)(2-\gamma)} \left(\frac{T}{N}\right)^{2-\gamma} \{(i-j+1)^{2-\gamma} - 2(i-j)^{2-\gamma} + (i-j-1)^{2-\gamma}\} \end{aligned}$$

And for cases where $i = j$,

$$G_{ii} = \frac{2}{(1-\gamma)(2-\gamma)} \left(\frac{T}{N}\right)^{2-\gamma}.$$

As we are considering peicewise constant trading strategy, the liquidation constraint can be expressed as

$$\sum_{i=1}^{N} v_i = \frac{NX}{T}. \tag{3.6}$$

Finally, the execution cost (3.2) can be written in terms of discrete approximation as

$$C[v] = \sum_{i=1}^{N} \sum_{j=1}^{N} v_i f(v_j) A_{ij}, \tag{3.7}$$

where A_{ij} matrix describe the decay kernel $G(t-s)$

$$\begin{aligned} A_{ij} &= 0, j > i, \\ A_{ii} &= G_{ii}/2; \\ A_{ij} &= G_{ij}, j < i. \end{aligned}$$

In conclusion, the best execution problem becomes an optimization problem as follows,

$$\begin{aligned} C[v] &= \sum_{i=1}^{N} \sum_{j=1}^{N} v_i f(v_j) A_{ij} \\ \text{subject to } \sum_{i=1}^{N} v_i &= \frac{NX}{T}. \end{aligned} \tag{3.8}$$

Chapter 4

Numerical algorithms

4.1 The reason for applying numerical method

We restrict our trading strategy to the class of piecewise constant strategies. Thus we divide the time horizon $[0, T]$ into N subintervals of equal length. Then we seek to minimize the cost function (3.7) by determining N trading rates v_i under the liquidation constraint that all the asset held should be traded. In the linear impact case, when $f(v) \propto v$, the discrete cost function reduces to a N-dimensional quadratic form. We can see by looking at:

$$\begin{aligned} C[v] &= \sum_{i=1}^{N} \sum_{j=1}^{N} v_i f\left(v_j\right) A_{ij} \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} v_i v_j A_{ij} \end{aligned}$$

where A_{ij} are only functions of time, not depending on any trading rates v_i. This kind of problem can be classified as a quadratic programming problem.

The general form of a quadratic problem (QP) is as follows,

$$\begin{aligned} \min_{x} \; q(x) &= \frac{1}{2} x^T G x + x^T c \\ \text{subject to } a_i^T x &= b_i, i \in \mathcal{E} \\ d_i^T x &= e_i, i \in \mathcal{I}, \end{aligned}$$

where x, c and a_i, d_i are all n-dimensional vectors, G is symmetric $n \times n$ matrix, $\mathcal{E}$ and $\mathcal{I}$ is the set of indices representing the number of equality and inequality constraints. Quadratic programs can always be solved in a finite amount of calculation steps, but for problems with different features for objective function and different numbers of inequality constraints, the effort required to solve the problem varies largely. In the case where the Hessian G is positive semi-definite, we call it convex quadratic program, and this problem is similar to solving a linear program in terms of difficulty. For our case, A_{ij} is clearly positive definite as $G(\tau) = |\tau|^{-\gamma}$ is positive semi-definite. So the effort required to solve this optimization problem is similar to a linear program.

In the nonlinear case, the objective function involves terms that are neither quadratic or linear, so the numerical minimization involves finding local minima of a complicated nonlinear function of N trading rates v_i. These local minima are not necessarily to global minimum.

In the general case, we need to perform a non-convex optimization for the cost functional expressed in terms of N trading rates v_i, subject to the liquidation constraint. This is because, as shown by the empirical data, the instantaneous impact function is strongly concave and the decay kernel is asymptotically as a power law function. Thus for nonlinear case, we need to use numerical methods.

4.2 Local sequential quadratic programming

The sequential quadratic programming is one of the state-of-the-art approaches to solving non-linear optimization problem, it is especially powerful when dealing with nonlinear optimization with significant non-linearity in the constraints. The sequential quadratic programming is an iterative method. In each step, the algorithm solves a quadratic sub-problem and generates an iteration step, and then use the solution to construct the next iterate. Convergence to local minimum is then guaranteed.

Consider an equality constrained problem of the following form

$$\begin{gathered} \min_x f(x) \\ \text{subject to } c(x) = 0, \end{gathered} \tag{4.1}$$

where $f : \mathbb{R}^N \mapsto \mathbb{R}$ and $c : \mathbb{R}^N \mapsto \mathbb{R}$ are smooth functions. The idea of SQP is to model (4.1) by a quadratic sub-problem at iterate x_k and use this minimizer to construct next iterate. The difficulty of this algorithm is to properly design the quadratic sub-problem. In Jorge and Stephen (2006) [27], they describe a simplest way of derivation of SQP, where they apply Newton's method to the Karush-Kuhn-Tucker optimality condition.

4.2.1 Karush-Kuhn-Tucker optimality condition

First we need to introduce some concepts on constrained optimization theory before we move on to the derivation of sequential quadratic programming.

Consider a general constrained optimization problem in the form of

$$\min_{x \in R^n} f(x) \text{ subject to } \begin{cases} c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geqslant 0, i \in \mathcal{I}, \end{cases} \tag{4.2}$$

where objective function $f(\cdot)$ and constraint function c_i are all smooth on hyperplane $\mathbb{R}^n$, and there should be finite number of equality and inequality constraints, which means $\mathcal{E}$ and $\mathcal{I}$ are finite sets.

Definition 4.1. The active set $\mathcal{A}(x)$ at any feasible x consists of indices of equality constraints from $\mathcal{E}$ and the indices of the inequality constraints for which $c_i = 0$; i.e.

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i = 0\} \tag{4.3}$$

At a feasible point x, the inequality constraint is said to be inactive if $c_i(x) > 0$, and the inequality constraint is active if $c_i(x) = 0$.

Definition 4.2. (LICQ.) Given a point x and the active set $\mathcal{A}(x)$ defined as definition 4.1, we say linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\Delta c_i(x), i \in \mathcal{A}(x)\}$ is linearly independent.

We define the Lagrangian function of problem (4.2) as

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x). \tag{4.4}$$

The following condition is a necessary condition for optimality, called first-order necessary conditions or Karush-Kuhn-Tucker condition.

Theorem 4.3 (First-Order necessary condition)**.** *Suppose that x^* is a local minimum, that function $f(\cdot)$ and $c_i(\cdot)$ are all smooth, meaning the first derivative exists, and LICQ holds at x^*. Then there is a Lagrange multiplier vector λ^* with $i \in \mathcal{E} \cup \mathcal{I}$ such that the following conditions are satisfied at (x^*, λ^*)*

$$\begin{aligned} \Delta_x \mathcal{L}(x^*, \lambda^*) &= 0 \\ c_I(x^*) &= 0, \;\; for\;\; all\;\; i \in \mathcal{E} \\ c(x^*) &\geqslant 0, \;\; for\;\; all\;\; i \in \mathcal{I} \\ \lambda_i^* &\geqslant 0, \;\; for\;\; all\;\; i \in \mathcal{I} \\ \lambda_i^* c_i(x^*) &= 0, \;\; for\;\; all\;\; i \in \mathcal{E} \cup \mathcal{I}. \end{aligned} \tag{4.5}$$

From equation (4.4), the Lagrangian function in our case where there is only one constraint is

$$\mathcal{L}(v, \lambda) = C[v] - \lambda \left(\sum_{i=1}^{N} v_i - \frac{NX}{T} \right).$$

Or we can change the constraint to another equivalent form,

$$\frac{T}{NX} \sum_{i=1}^{N} v_i = 1,$$

and this gives rise to another equivalent form of Lagrangian function which is

$$\mathcal{L}(v, \lambda) = C[v] - \lambda(\frac{T}{NX} \sum_{i=1}^{N} v_i - 1)$$

In general case, the Jacobian matrix of the constraint is defined as

$$J(x)^T = \left[\Delta c_1(x), \Delta c_2(x), ..., \Delta c_m(x) \right],$$

where Δc_i here denotes the gradient of constraint $c_i(x) = 0$. That is,

$$\begin{aligned} J(x)^T &= \begin{bmatrix} \nabla c_1(x) & \nabla c_2(x) & ... & \nabla c_m(x) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial c_1}{\partial x_1} & \frac{\partial c_2}{\partial x_1} & \cdots & \frac{\partial c_m}{\partial x_1} \\ \frac{\partial c_1}{\partial x_2} & \frac{\partial c_2}{\partial x_2} & \cdots & \frac{\partial c_m}{\partial x_2} \\ \\ \frac{\partial c_1}{\partial x_n} & \frac{\partial c_2}{\partial x_n} & \cdots & \frac{\partial c_m}{\partial x_n} \end{bmatrix} \end{aligned}$$

In our one constraint case, the Jacobian becomes

$$J(v)^T = \begin{bmatrix} \frac{T}{NX} \\ \frac{T}{NX} \\ \cdots \\ \frac{T}{NX} \end{bmatrix}$$

The first-order condition (KKT condition) (4.3) of the one equality constrained optimization problem can be written as a system of $N+1$ equations in the $N+1$ unknowns $v_i, i = 1, 2, ..., N$ and λ:

$$F(v, \lambda) = \begin{bmatrix} \nabla f(v) - J(v)^T \lambda \\ c(v) \end{bmatrix} = 0 \tag{4.6}$$

where

$$\nabla f(v) = \begin{bmatrix} \frac{\partial c(v)}{\partial v_1} \\ \frac{\partial c(v)}{\partial v_2} \\ \cdots \\ \frac{\partial c(v)}{\partial v_N} \end{bmatrix}.$$

In our case,

$$\begin{aligned} \frac{\partial c[v]}{\partial v_1} &= \sum_{j=1}^{N} f(v_j) A_{1j} + v_3 f'(v_1) A_{31} + ... + v_N f'(v_N) A_{N1} \\ &= \sum_{j=1}^{N} f(v_j) A_{1j} + f'(v_1) \sum_{j=1}^{N} v_j A_{j1} \end{aligned}$$

Similarly, we have derivatives with respect to other components,

$$\frac{\partial C[v]}{\partial v_i} = \sum_{j=1}^{N} f(v_j)A_{ij} + f'(v_i)\sum_{j=1}^{N} v_j A_{ji}.$$

Any solution (x^*, λ^*) of the equality-constrained problem (3.8) for which $J(x^*)$ has full rank satisfies the first order condition (4.6).

One approach to solve the non-linear equations (4.6) is by using the Newton's method.

4.2.2 Newton's method

In linear case, Newton's method constructs a quadratic approximation by taking the second order Taylor series expansion of objective function f around iterate v_k. The solution to this quadratic model is the Newton step at current iterate. In the nonlinear case, Newton's method is constructed in a similar way, but using linear Taylor approximation which consits of only the first two terms, i.e. function value and its gradient at current iterate x_k.

Theorem 4.4 (Multidimensional version of Taylor's theorem)**.** *Assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable in convex set $\mathcal{A}$ and x and $x+t$ are vectors in $\mathcal{A}$. We then have*

$$f(x+t) = f(x) + \int_0^1 J(x+st)tds.$$

We can approximate the second term in the above expression by Jacobian at the kth iterate, that is $J(x)t$. Thus we have a linear model $M_k(p)$ for approximating $f(x+t)$ at iterate k, writing

$$M_k(t) := f(x_k) + J(x_k)t$$

Newton's method takes the step t_k to be the solution to equation $M_k(t) = 0$, i.e. $t_k = -J(x_k)^{-1}f(x_k)$.

Thus the procedure of Newton's method can be described formally as follows,

First Pick a starting point x_0.

```
for loop k=1,2,...
    Calculate the Newton step $t_k$ by
        $J(x_k)t_k = -f(x_k);$    (4.7)
    and then generate the next iterate $x_{k+1}$ by $x_{k+1} \leftarrow x_k + t_k$;
end for
```

The reason why we use a linear model for deriving Newton step t_k is that a linear model usually has a solution and converge to the solution rapidly. However, the Newton's method has some drawbacks that we should take into account when examining the results form Newton method.

- The algorithm may not result in a solution if the starting point is far away from it. When $J(x_k) = 0$, the Newton may not even be defined.
- The gradient $J(\cdot\,)$ may not be easy to obtain.
- When the number of variable n tends to be large, it is time-consuming to calculate the Newton step.

The Jacobian of (4.6) with respect to v and λ is given by $F'(v, \lambda)$, where the first column is the derivative with respect to v and the second column is the derivative with respect to λ,

$$F'(v,\lambda) = \begin{bmatrix} \nabla^2_{vv}\mathcal{L}(v,\lambda) & -J(v)^T \\ J(v) & 0 \end{bmatrix}$$

The Newton steps from the kth iterate (v_k, λ_k) can be obtained to generate the next iterate by

$$\begin{bmatrix} v_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} v_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} t_k \\ t_\lambda \end{bmatrix}.$$

Here the Newton step is the solution to the Newton-KKT system, that is, it satisfies the following system of equation,

$$\begin{bmatrix} \nabla^2_{vv}\mathcal{L}_k & -J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} t_k \\ t_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_k + J_k^T\lambda_k \\ -c_k \end{bmatrix}. \tag{4.8}$$

The sufficent condition for which the Newton iteration is well defined is that the KKT matrix is non-singular at (v_k, λ_k). The matrix is non-singular if, by Nocedal (2006) [27],

- The Jacobian of constraints $J(v)$ has full row rank;
- The Hessian $\nabla^2_{vv}\mathcal{L}(v,\lambda)$ is positive definite when (x,λ) satisfies $d^T\nabla^2_{vv}d > 0$ for all $d \neq 0$ such that $J(v)d = 0$.

Under these assumptions, the Newton's method is quadratically convergent and is a good algorithm for solving equality-constrained problem, provided the initial point is too far away to local minimum.

4.2.3 SQP framework

The SQP framework views Newton step and Newton-KKT system in another way. Suppose our choice of modelling optimization problem (4.1) changes to use quadratic program at iterate (v_k, λ_k) as following

$$\begin{aligned} \min_{t} \;\; & f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{vv} \mathcal{L}_k p \\ \text{subject to } & A_k p + c_k = 0. \end{aligned} \tag{4.9}$$

If the above assumption that the KKT matrix is non-singular is satisfied, this optimization problem has a unique solution (p_k, l_k) to system of equations

$$\nabla^2_{vv}\mathcal{L}_k p_k + \nabla f_k - J_k^T l_k = 0 \tag{4.10}$$

$$J_k p_k + c_k = 0. \tag{4.11}$$

If A_k^T is subtracted from both sides of the first equation in (4.8), we obtain

$$\begin{bmatrix} \nabla^2_{vv}\mathcal{L}_k & -J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_k + \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}.$$

Then we set $\lambda_{k+1} := p_k + \lambda_k$, and by comparing with equation (4.10), we have $\lambda_{k+1} = l_k$ and p_k that solves (4.9).

Both the Newton viewpoint and the SQP framework can generate the iterate (v_{k+1}, λ_{k+1}) required to solve the problem. However, the Newton approach can facilitate our analysis while the SQP framework can be used to derive a practical algorithm.
Algorithm description:

Start Pick an initial point (v_0, λ_0); set $k = 0$.

Do until a specified tolerance is satisfied. Calculate$\nabla^2_{vv}\mathcal{L}_k, f_k, \nabla f_k, c_k$ and $J(v_k)$; Solve system of equation (4.10) to get p_k and l_k; Update the iterate $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow l_k$;

Repeat(end)

We find that in the objective function (4.9), the second term $\nabla f_k^T p$ can be replaced by the gradient of Lagrangian function $\nabla\mathcal{L}(v_k, \lambda_k)$. This is because of constraints in (4.9) and (4.1),

$$\begin{aligned} \nabla f_k^T p &= \left[\nabla_v \mathcal{L}_k + J(v_k)^T \lambda_k\right] p \\ J(v_k)^T p &= -c(v_k) = 0 \end{aligned}.$$

This fact motivates us a way of choosing a proper quadratic model (4.9): first the general nonlinear optimization problem of form (4.1) can be transformed to the corresponding Lagrangian function, and then derive a quadratic approximation of the Lagrangian function and a linear approximation of the constraint to form the quadratic SQP subproblem (4.9).

4.2.4 Multistart strategy

One way of finding a global minimum in a constrained nonlinear optimization problem is by multistart strategy. This algorithm starts the local solver, which in our case is SQP local solver, from multiple start points to sample multiple basins of attraction.

Definition 4.5. Provided an objective function is smooth, the opposite of gradient i.e. $-\nabla f(x)$ will give the direction of quickest descent. The basin of attraction is the set of initial points for which the direction of descent leads to same local minimum.

The most basic multistart strategy is to generate uniformly distributed starting points and then run local solver from these points. After obtaining these multiple local minima, we take the solution with the smallest objective function value as the global minimum. In theory, this algorithm can reach a global minimum with probability one as the number of start points tend to infinity. In Matlab, multistart algorithm has several advantages when compared to GlobalSearch,

- Parallel computing can be used in 'MultiStart'.
- Use other local solver other than 'fmincon'
- Customized starting point can be included in the starting set as well as random start point.

Generate start points

Matlab global optimization toolbox provides several ways to set start points for the local solver.

- Passing a positive integer k to function $run(ms, problem, k)$. In this way, Matlab generate $k-1$ uniformly distributed random start points from which the local solver runs. Together with an initial start point in problem structure, k start points are used in total.
- Pass a 'RandomStartPointSet' object. By passing a 'RandomStartPointSet' object to $run(ms, problem, \cdot\,)$, Matlab generates a specified amount of uniformly distributed random variable as start points.
- Passing a 'CustomStartPointSet' object. By the same way, it allows the local solver to start from the points supplied by the user.

In our case, we have a liquidation constraint (3.6), which also applies to all the start points. According to Gatheral (2014) [11], where they test various kinds of distribution for the start points including uniform and Dirichlet distribution, they find qualitatively similar optimal execution strategies.

Thus we choose a procedure as follows to generate the multiple start points: first we generate $N-1$ identically independently uniformly distributed random variable, then use the constraint $v_N = N{*}X/T - \sum_{i=1}^{N} v_i$ to calculate the remaining variable. Here, we choose $v_i \sim U(-4.9, 4.9), i = 1, 2, ..., 99$.

Each row of the generated matrix represents a start point. This matrix is then used to create a 'CustomStartPointSet' object, which is then passed to function $run(\cdot\,)$.

Create problem structure

One of the ingredients for multistart method is the problem structure. The problem structure contains information in terms of what kind of algorithm the local solver use, the objective function, bounds, constraints and so on. In Matlab, the problem structure can be created in two ways, exporting from the optimization toolbox or use ‘createOptimProblem’ function. We choose to use createOptimProblem function as it is more convenient to change its options.

The procedure is first to define and create all the variable required, such as objective function handle, equality and inequality constraint vector and upper and lower bounds and so on.

By running the local solver set with default options, we find the local solver exit prematurely before it converge to the local minimum. This issue can be solved simply by change some options in the problem structure. From our experiment, when setting the maximum number of iteration to 600 and the maximum number of function valuation to 100000, more than 90% of iteration can exit with a positive exit flag, meaning the local solver actually converges to the corresponding local minimum.

This can be easily done by creating an option structure using ‘optimoption’ function as follows,

```
opts=optimoptions(@fmincon,‘MaxIter’,600,‘MaxFunEvals’,100000);
```

Set up solver object and parallel pool

A solver object contains the preference in terms of global option for the optimization. For instance, one can set the property ‘Display’ to ‘iter’ to show information after each local solver finishes running. The information displayed include number of function valuation, first-order optimality, function value and exit flag. A positive exit flag implies the local solver has find a local minimum, otherwise it means the local solver stops prematurely.

Matlab enables the user to run local solver from multiple start points in parallel. In contrast to serial computing where the computation is executed one after another, parallel computing is a way of execute computation simultaneously. It is based on the principle that a large computation task can usually be divided into smaller and similar subtasks, which then can be sent to a number of connected workers and then return the results to the server. Thus parallel computing can be run in a multicore processor or when user has access to a network of processors.

The results of running multistart is usually random due to the random set of start points, thus the time it costs is also random. Normally, parallel computing will reduce the time required to run the optimization, and this benefit will be more pronounced as the number of start points becomes larger. From the data provided in Matlab user's guide [20], when there are 1000 start points to run, parallel computing usually reduce the time cost by 1/3. In our numerical experiment, we will use a two-core processor, and by starting function 'matlabpool' using the 'local' profile, we are able to connect to 4 workers, which means we can run 4 local solvers for 4 different start point simultaneously.

Output sturcture

Matlab can record most of the information we need to analyze the result. By running the following code,

```
[xmin,fmin,flag,outpt,allmins]=run(ms,optimproblem,tpoints);
```

one can keep five kinds of information which help us examine the reliability of the numerical results.

- *xmin*: 'xmin' is of the same dimension as the start points. This variable stores the global solution found by running the local solver from all the feasible start points.
- *fmin*: This is a scalar and it stores the funcation value of the global minimum point, i.e. the smallest objective function value among all the local minima.
- *flag*: It is the integer that indicate why the algorithm terminates. In our multistart algorithm, each time of running the local solver with a different start point will result in an exitflag. In the case of using SQP algorithm, an exit flag of '1' implies that the first order optimality is less than 'option.TolFun' and maximum violation of constraint is less than 'option.TolCon'. This

means the solver has converged to a local minimum. An integer of '0' implies that the solver stops prematurely. This is because the number of function valuation exceeds the user-specified 'options.MaxFunEvals' or the number of iteration exceed user-specified 'options.MaxIter'. This problem can be settled by changing the options.

- *outpt*: 'outpt' is a struct containing information after running the global optimization. Usually, it keeps the number of total function valuation, total, successful, incomplete and no solution local solver. It also offer a field briefly summarizing the result of optimization.
- *allmins*: Multistart generates a vector of 'GlobalOptimSolution' object, the dimension of which is the number of local solver exiting with a positive flag. Each element of the vector stores the information of an individual local minimum, ordered by the objective function value from the lowest to the highest. This information on each local minimum includes the location, objective function value, exit flag, outpt struct and the start point used to reach the minimum.

No guarantee for convergence to global minimum

Once again we note that there is no guarantee for the convergence to the global minimum by using any of the algorithm provided in the Global Optimization toolbox in Matlab. The most straightforward way to check is to increase the number of start points, that is, running the local solver from additional start points. If that does not generate a smaller function value, it means we have reached a global minimum. Another way of improvement is by tightening the region. When dealing with practical problems, there usually exist a reasonable region in which we believe the global minimum lies. Thus by bounding the region, we have, to some extent, the assurance that Multistart finds the global minimum.

Chapter 5

Numerical results

In this chapter, we will investigate how the optimal strategy behaves in different parametric sets. We will first investigate the behaviour of optimal strategy in the linear impact case, and we will take the this case as a benchmark for comparing with the nonlinear cases.

In general, the instantaneous function is of the form $f(v) \propto sign(v) \mid v \mid^{\delta}$. Study on market data shows it behaves as a strongly concave function. By Lillo, Farmer and Mantegna [22], the exponent varies from roughly 0.5 for small transactions in higher-capitalization stocks, to about 0.2 for larger transactions in lower-capitalization stocks. The general form of the decay kernel is $G(\tau) = \tau^{-\gamma}$ as shown by market data that the decay kernel is asymptotically a power law function. The requirement of no transaction-triggered price manipulation restricts the possible combination of parameters. By Gatheral (2010) [13], the necessary conditon should be

$$\gamma + \delta \geqslant 1, \gamma \geqslant \gamma^* \simeq 0.415.$$

Note that the above is only necessary condition, it does not preclude the existence of transaction-triggered price manipulation. In the following sections, the numerical results also indicate the existence of price manipulation even when this necessary condition is satisfied. From here onwards, we will always examine the cases where this necessary condition is satisfied.

5.1 Linear impact case

When the exponent is equal to 1, the impact function reduces to a linear function, causing the objective function to be a quandratic form, which substantially simplifies optimization problem of finding the optimal strategy. In the following, we are always considering X as a proportion of market volume and we will set $X = 10\%$ unless otherwise specified. Thus we set $\delta = 1$ and $\gamma = 0.5$. We present the optimal trading strategy in each period in figure 5.1.

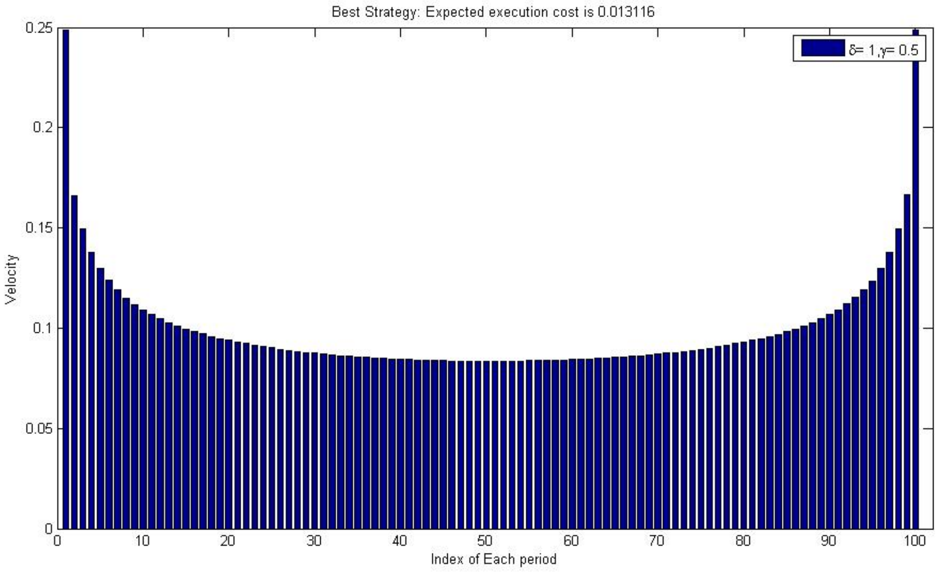


Figure 5.1: The optimal trading strategy given by SQP when $X = 0.1$. The total finite horizon T is divided into 100 trading periods. The vertical axis is the constant average trading rate in each subinterval. The lowest expected execution cost is 0.013116.

From our numerical result, the optimal strategy in the case of linear impact function has a good regularity. We find from the above graph, the trading strategy is consecutive, that is, the trader need to trade in all of these subintervals. Also, except in the beginning and end of the horizon, this strategy is quite close to the VWAP strategy, where the order execution spreads evenly during the horizon T. More importantly, a buy program does not include selling in any of these subinterval, resulting in a monotone and single-sign trading strategy. In terms of regularity, this means this transient market impact model with linear instantaneous impact function does not adimit transaction-triggered price manipulation.

5.2 Nonlinear impact case

From here onwards, we will investigate the nonlinear instantaneous impact function and the corresponding optimal trading strategy. We will vary the nonlinearity of instantaneous impact function, i.e. δ, from the slightly nonlinear case to the strongly nonlinear case to investigate how the optimal strategy will change in each situation.

5.2.1 Slightly nonlinear case

Fixing the exponent of the decay kernel γ to 0.5, we change the exponent of instantaneous impact function $f(\cdot)$ to slightly less than 1, taking 0.95 as an example. Remaining all other parameter unchanged, that is: the total position X is 0.1, the number of total start points for the SQP algorithm is 1000, time horizon $T = 1$. We present the numerical result in figure 5.2.

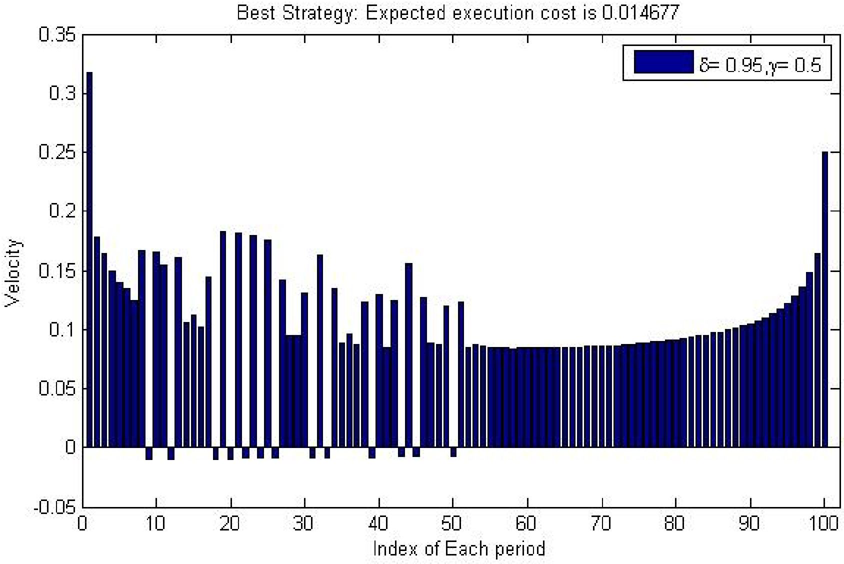


Figure 5.2: The optimal trading strategy given by SQP when $X = 0.1, T = 1, N = 100$. The vertical axis is the constant average trading rate in each subinterval. The lowest expected execution cost is 0.014677.

From the graph, we can see the strategy in the nonlinear instantaneous impact function case is in general similar to the one in linear case. However, there is one significant difference: the trading strategy is no longer monotone or single-signed. During the $10th$ to $50th$ subintervals, there exists selling for a buy program.

The amount of sell is relatively small compared with those buy orders in other subintervals, and these small sell orders are of similar size. Also, immediately after executing a sell order in each subinterval, there will be a higher than usual amount of buy order. This means the trader can exploit the benefit of selling a small amount of asset before placing a relatively large buy order. This is what we defined as transaction-triggered price manipulation.

We should mention here placing intermediate sell orders in a buy programme is considered illegal. The reason why we can not regularize this issue is because the numerical algorithm we are using, i.e. SQP, is based on derivative of Lagrangian function. If we add non-negativity constraint to the problem structure, its derivative at the points for which at least one subinterval has no trading is not well-defined. This indicates SQP algorithm is not suitable for this situation. There are other numerical algorithms that does not depend on derivative and thus can apply to the situation where non-negativity is taken into account. This will be future improvement beyond this dissertation.

Another characteristic we should tell from the graph is that the sell orders lie in a certain range of time rather than spreading sparsely within the whole horizon. When the impact function is more concave, the number of periods when sell order is placed increases and they are more broadly spread across the whole horizon, which we can see in figure 5.3. Also we can see the magnitude of each sell order also increases. All these changes may imply that the benefit from transaction-triggered price manipulation becomes more evident as the degree of non-linearity increases.

5.2.2 Strongly concave impact case

When γ is 0.5, the smallest feasible δ is 0.5, which still satisfies the no-dynamic-arbitrage necessary condition. As before, we fix the total position X to be 0.1, the number of total start points for the SQP algorithm to be 1000, time horizon T to be 1, and we present the numerical result in figure 5.4.

From this graph we can see, the optimal strategy is not monotone either, and it becomes more irregular than the slightly nonlinear impact case. The number of subinterval where we should buy substantially decreases, while the number of

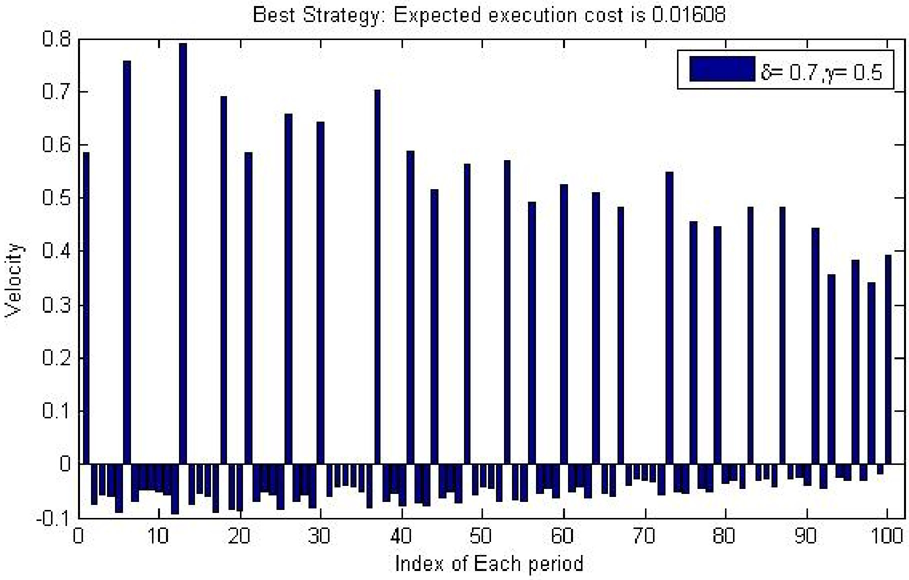


FIGURE 5.3: The optimal trading strategy given by SQP when $X = 0.1, T = 1, N = 100$. The vertical axis is the constant average trading rate in each subinterval. The lowest expected execution cost is 0.01608.

subintervals where we should sell increases. Specifically, the optimal trading strategy consists of several large, single-period buy orders, separated by long-term but small sell orders. Also we can see there is at least one period of selling before every burst of buying. And by observing the optimal strategy under other parametric sets, the trading for the last period of a buy program is always buying. This characteristic again implies the existence of transaction-triggered price manipulation as defined in definition 3.3, which means the trader can benefit from executing a sell order before executing a buy program.

Another characteristic we should notice is the negative expected execution cost. In this case where $\delta = 0.5$ and $\gamma = 0.5$, the lowest expected cost is -0.00059449, which indicates there exist a possibility of price manipulation defined as 3.1. This means we can devise a round trip strategy, by which we have positive expected revenue. Note here, this round trip strategy does not necessarily lead to positive revenue every time we execute. But in expectation, this will result in positive revenue. Because the repeated price manipulation can lead to a weak-form arbitrage, called quasi-arbitrage, see Huberman and Stanzl (2004) [17]. This implies the non-linear transient impact model (3.1) with $f(v) = v^{\delta}, \delta < 1$ is not well-defined as it admits arbitrage opportunity.

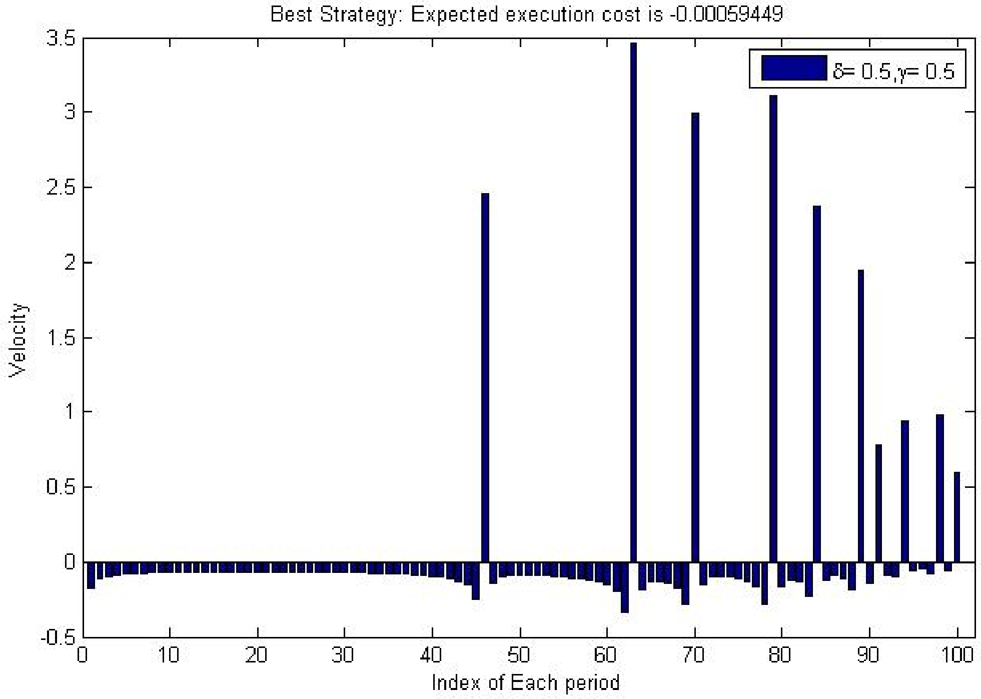


FIGURE 5.4: The optimal trading strategy given by SQP when $X = 0.1, T = 1, N = 100$. The vertical axis is the constant average trading rate in each subinterval. The lowest expected execution cost is -0.00059449.

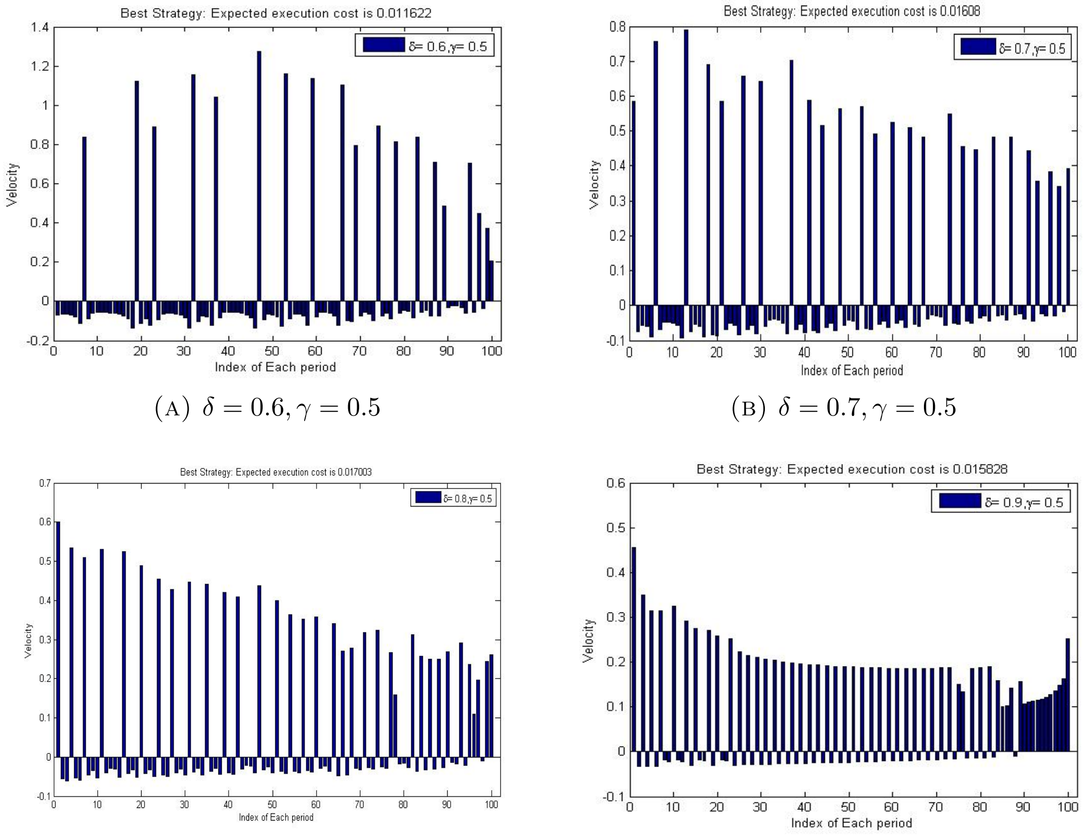
We still need to mention that we have not incorporated the non-negativity constraint as the SQP algorithm can not deal with this situation. Other numerical algorithm should be used in order to take the non-negativity constraint into consideration.

5.3 Impact of discretisation

In our numerical scheme, we divide the finite time horizon into 100 equally length subintervals, then we calculate the optimal trading strategy in different parametric sets. We list all our results in figure 5.5.

Then we need to investigate if the way we discretize the finite horizon affects the behaviour of optimal trading strategy and the lowest expected execution cost. That is, whether N affects the optimal strategy.

We will first investigate the impact of N in one set of parameter, where $\delta = 0.55, \gamma = 0.5$. As before, we fix X to be 0.1, and use 1000 start points in Multistart



(A) $\delta = 0.6, \gamma = 0.5$

(B) $\delta = 0.7, \gamma = 0.5$

(C) $\delta = 0.8, \gamma = 0.5$

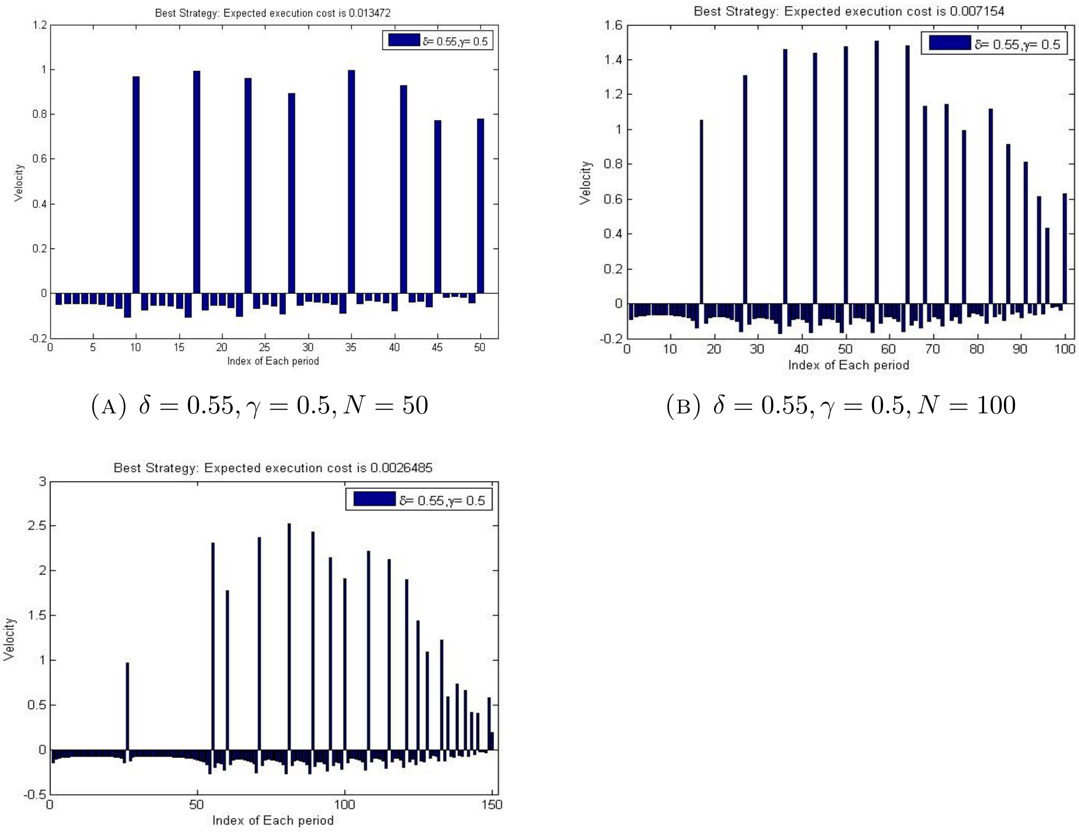
(D) $\delta = 0.9, \gamma = 0.5$

FIGURE 5.5: Four different parametric sets($\delta = 0.6, \gamma = 0.5$),($\delta = 0.7, \gamma = 0.5$),($\delta = 0.8, \gamma = 0.5$),($\delta = 0.9, \gamma = 0.5$) respectively. The number of subintervals is fixed to be 100, with $X = 0.1$.

method. We choose N to be 50,100,150 respectively. The SQP method will give the optimal strategy as in figure 5.6.

We can see that these three figures are qualitatively similar. In each case, the optimal strategy is composed of several bursts of buying separated by small but long-term selling. This shape of optimal trading strategy implies the existence of transaction-triggered price manipulation.

However, there are several differences. The first thing we should notice is that, in the case of finer discretization, the absolute value of both buying and selling orders are larger than that of the case where the horizon is less finely discretized. Another difference we can notice from these three plots is that the minimal expected cost decreases as the number of subintervals increases. This may imply that a higher frequency of trading enables the trader to better exploit the benefit of transaction-triggered price manipulation.



(A) $\delta = 0.55, \gamma = 0.5, N = 50$

(B) $\delta = 0.55, \gamma = 0.5, N = 100$

(C) $\delta = 0.55, \gamma = 0.5, N = 150$

FIGURE 5.6: Optimal execution strategy with same set of parameters, under three different discretizations, N=50, N=100 and N=150 respectively. X is fixed to be 10% of unitary market volumn.

In order to verify the hypothesis that higher frequency of trading enables the better exploiting price manipulation, we draw the lowest expected execution cost as a function of degree of non-linearity δ, while fixing all other parameters. That is, $X = 0.1, T = 1, \gamma = 0.5$. We run the test under two kinds of discretization, dividing the finite horizon into 50 and 100 sunintervals respectively. The result is presented in figure 5.7.

From the graph we first notice that the optimal execution cost is never larger in the finer discretization case than that in less finely discretized case, for any degree of nonlinearity δ. This verifies our hypothesis that higher frequency of trading enables the trader better exploiting the benefit of price manipulation. When $\delta = 1$, which means the instantaneous impact funciton is linear, the optimal execution cost under the two discretization is same. It implies that under the linear transient impact model, the expected execution cost does not depend on how finely we

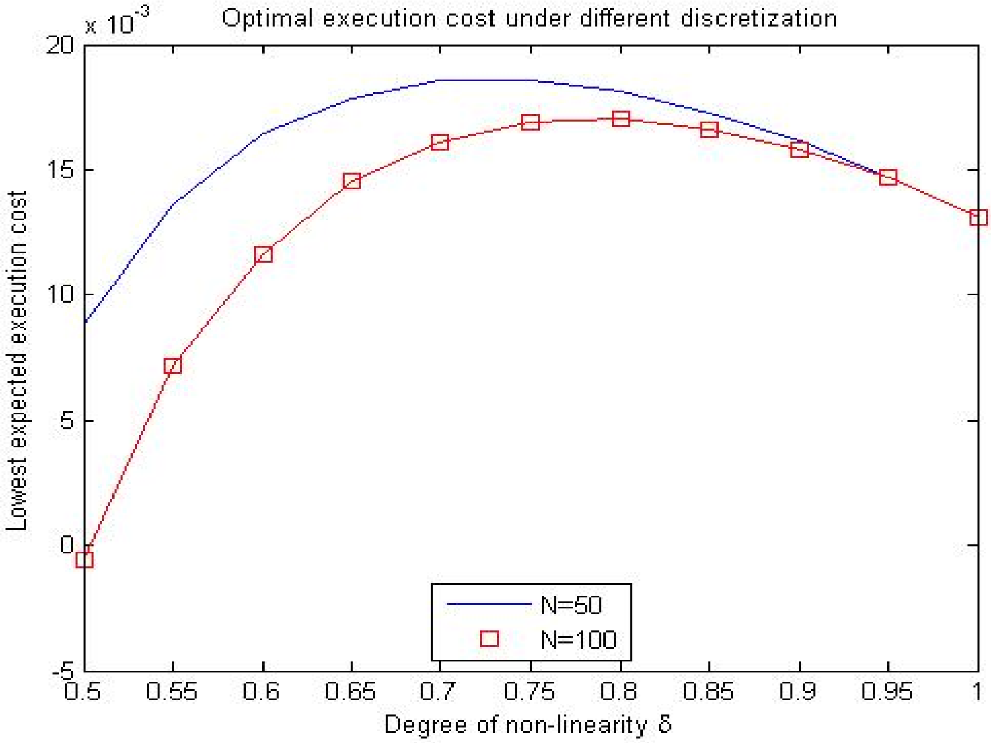


Figure 5.7: The optimal expected execution cost as a function of degree of non-linearity δ under two kinds of discretization. Other parameters are all fixed: $X = 0.1$, 1000 start points, $\gamma = 0.5$.

discretize the trading horizon. But we are not sure if this phenomenon is specific to SQP algorithm and it originates from the specification of transient model.

Further, the optimal expected execution cost is not a monotone function of degree of nonlinearity in neither of these two cases. Also we should note that the discrepancy of optimal execution cost between these types of discretization for the same parameters becomes larger as the degree of nonlinearity increases. This may imply that the benefit from finer discretization becomes more evident in the case of strongly concave instantaneous impact function.

Moreover, when $N = 100$, there is a range of degree of nonlinearity δ for which the optimal execution cost is negative. According to the tendency we observed, we may infer that this region of negative execution cost is likely to enlarge as the discretization becomes finer.

Chapter 6

Conclusion

6.1 Summary of results

In this dissertation, we review the existing two classes of market impact model. We then focus our attention on the transient impact model, especially when the instantaneous impact function is non-linear. We formulate the problem of liquidating a large amount of asset in a finite time horizon as a nonlinear constrained optimization problem. We compare the optimal strategies and the best expected execution cost.

To solve this problem, we resort to the numerical algorithm of sequential quadratic programming (SQP). This algorithm is derivative-based and is implemented by solving a properly constructed quadratic sub-problem iteratively. We investigate cases where the instantaneous impact function is slightly concave, strongly concave and linear. We also focus our attention on the possible existence of price manipulation. We find that in the linear case, the optimal execution strategy behaves well. There is larger buying at the beginning and end of the time horizon for a buy program, and there is no intermediate selling, which means no price manipulation exists. As the instantaneous impact function becomes concave, intermediate selling arises. Moreover, as the degree of nonlinearity increases, there is more intermediate selling and the optimal expected execution cost decreases incrementally. When the degree of non-linearity becomes very strong, the optimal trading strategy is comprised of short-term bursts of buying separated by long-term but small selling for a buy program. Moreover, we find that for some sets of parameters, the optimal expected execution cost becomes negative. This means that price manipulation exists, which can lead to a weak form of arbitrage: quasi-arbitrage. This suggests that this transient market impact model is not well-defined and needs to be regularized.

Moreover, we examine the impact of discretization. We find that we have qualitatively similar optimal trading strategies by different discretisation schemes, and they all admit the existence of transaction-triggered price manipulation in nonlinear cases. However, the method of discretization affects the optimal strategy. Our results show that finer discretization will lead to lower expected execution cost; in some regions of parameter, this even leads to negative expected cost. This may imply that a higher frequency of trading for the same set of parameters enables the trader to better exploit the benefit of transaction-triggered price manipulation. Furthermore, the discrepancy between two different discretization schemes tend to increase as the degree of nonlinearity increases. The discretization scheme does not affect the optimal strategy when the instantaneous function is linear.

6.2 Weakness

In this project, we focus on finding an optimal trading strategy by using numerical algorithm, sequential quadratic programming. This algorithm is based on derivative, so we are not able to impose the non-negativity constraint to the optimization problem. By using this scheme, we find the transient model admits transaction-triggered price manipulation and further in some parametric sets it admits price manipulation. As a result, we conclude that the transient model (3.1) is not well-defined. However we are still not sure if this phenomenon is a result of numerical instability, which need to be further investigated.

6.3 Future work

As we mentioned, when we add a non-selling constraint to a buy program, the derivative of Lagrangian function at points for which some components are zero is not defined, so the SQP algorithm is no longer suitable and we need to resort to other algorithms. One of the candidates would be the direct search method, especially generating set search. By using the direct search method, we may be able to find an optimal trading strategy that precludes the existence of transaction-triggered price manipulation.

Moreover, we are expecting that the functional form of $f(x)$ can be preserved no matter how fast we are trading. This kind of simplification definitely reduces the effort needed for computation and analysis. However, in practice, this is an unrealistic assumption. When we are trading at an arbitrarily high rate, the liquidity in the market order book is not enough. Thus we trade deeply into the limit order book, where liquidity is less ample. Empirical study shows that when trading is executed at a high rate, the instantaneous market impact becomes convex. This motivates us to replace the current form of $f(\dot{x})$ by a concave-convex impact function, which incorporate the penalty to excessively high trading rate. This improvement could possibly remove the existence of price manipulation.

The third improvement is adding bid-ask spread into the transient model. Price dynamics (3.1) actually models the evolution of mid-price. When an order is executed, an extra cost of half of the bid-ask spread applies. The bid-ask spread can be seen as a penalty to wrong-way trading. Wrong-way trading is intermediate selling in a buy program, or buying in a sell program. The addition of bid-ask spread provides a way to eliminate or decrease the price manipulation.

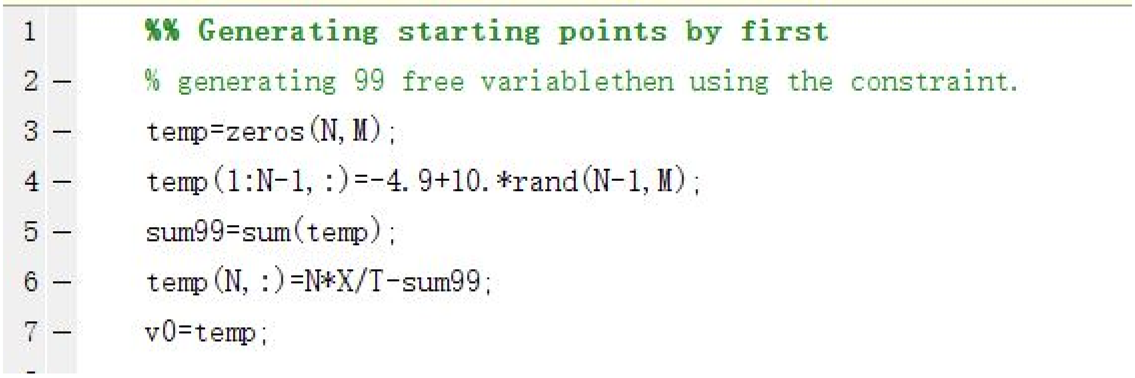
Appendix A

Matlab code

The following is the codes implementing Multistart method with parallel computing to obtain the optimal strategy by Matlab. The code consists of several sections.

A.1 Start points sampler

The first step of implementing Multistart method is to generate start points. In our case, the start points is sbuject to a constraint, i.e. $\sum_{i=1}^{N} v_i = NX/T$. This can be simply fulfilled by first generating $N - 1$ random variable and then subtracting from the sum.

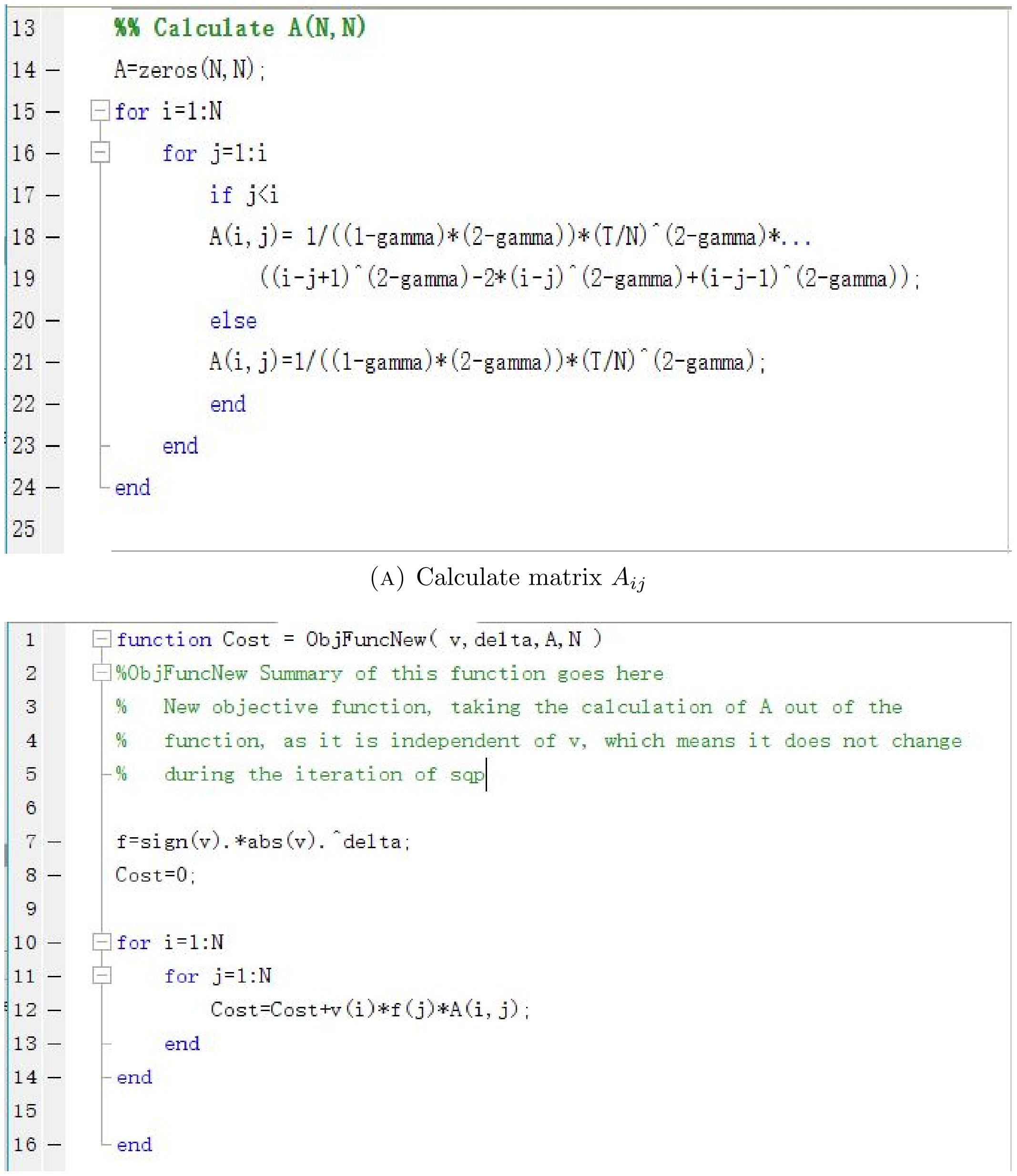


```
%% Generating starting points by first
% generating 99 free variablethen using the constraint.
temp=zeros(N,M);
temp(1:N-1,:)=-4.9+10.*rand(N-1,M);
sum99=sum(temp);
temp(N,:)=N*X/T-sum99;
v0=temp;
```

Figure A.1: Matlab code for generating multiple start points

A.2 Cost function

The liquidation cost function is expressed as 3.7, which can be coded by a nested loop.



```
%% Calculate A(N,N)
A=zeros(N,N);
for i=1:N
    for j=1:i
        if j<i
        A(i,j)= 1/((1-gamma)*(2-gamma))*(T/N)^(2-gamma)*...
            ((i-j+1)^(2-gamma)-2*(i-j)^(2-gamma)+(i-j-1)^(2-gamma));
        else
        A(i,j)=1/((1-gamma)*(2-gamma))*(T/N)^(2-gamma);
        end
    end
end

```

(A) Calculate matrix A_{ij}

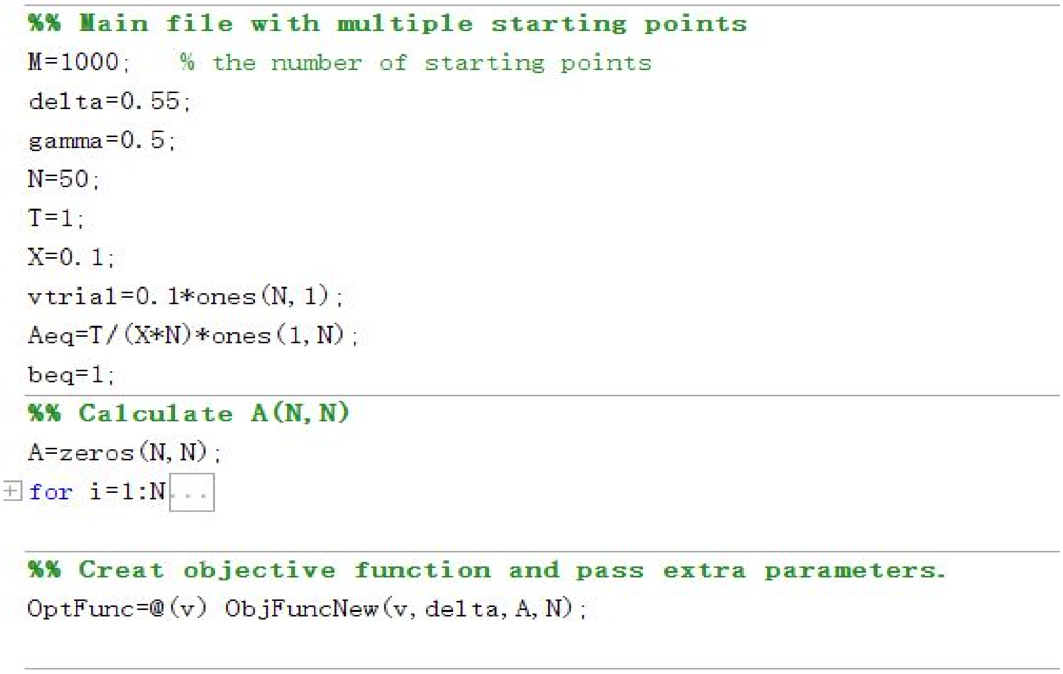
```
function Cost = ObjFuncNew( v,delta,A,N )
%ObjFuncNew Summary of this function goes here
%   New objective function, taking the calculation of A out of the
%   function, as it is independent of v, which means it does not change
%   during the iteration of sqp

f=sign(v).*abs(v).^delta;
Cost=0;

for i=1:N
    for j=1:N
        Cost=Cost+v(i)*f(j)*A(i,j);
    end
end

end
```

(B) Cost function

Figure A.2: Matlab code: discretized cost function

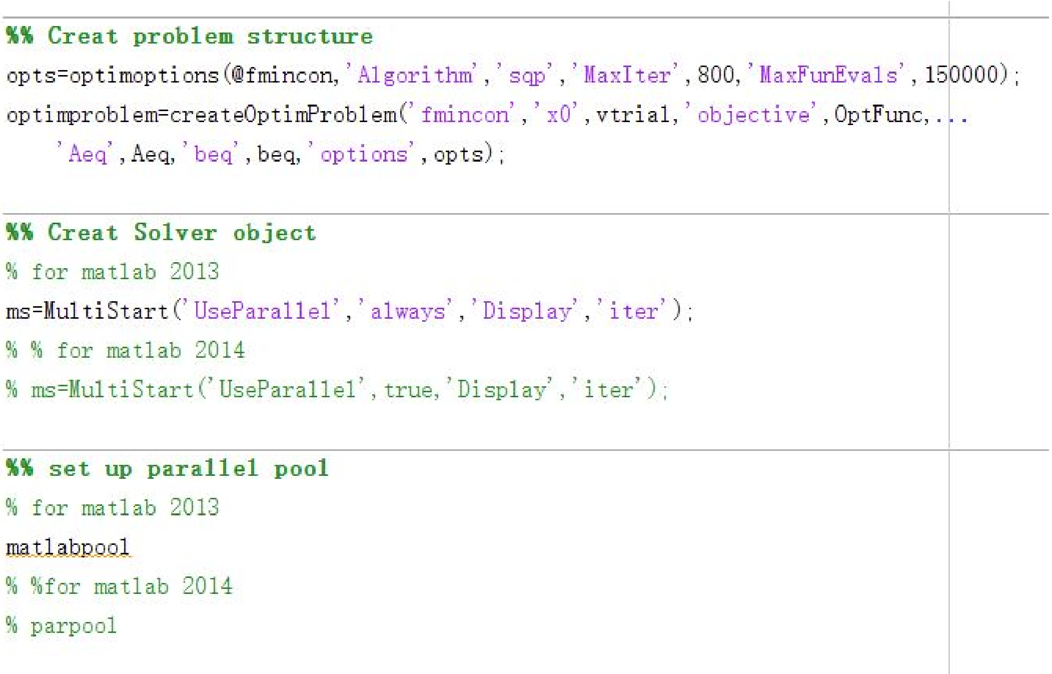
A.3 Multistart method

The steps required to use multistart method for sequential quadratic optimisation includes writing constraints, creating problem structure, create solver object and running local solver. The code is shown as follows.



```
%% Main file with multiple starting points
M=1000;   % the number of starting points
delta=0.55;
gamma=0.5;
N=50;
T=1;
X=0.1;
vtrial=0.1*ones(N,1);
Aeq=T/(X*N)*ones(1,N);
beq=1;
%% Calculate A(N,N)
A=zeros(N,N);
for i=1:N...

%% Creat objective function and pass extra parameters.
OptFunc=@(v) ObjFuncNew(v,delta,A,N);
```

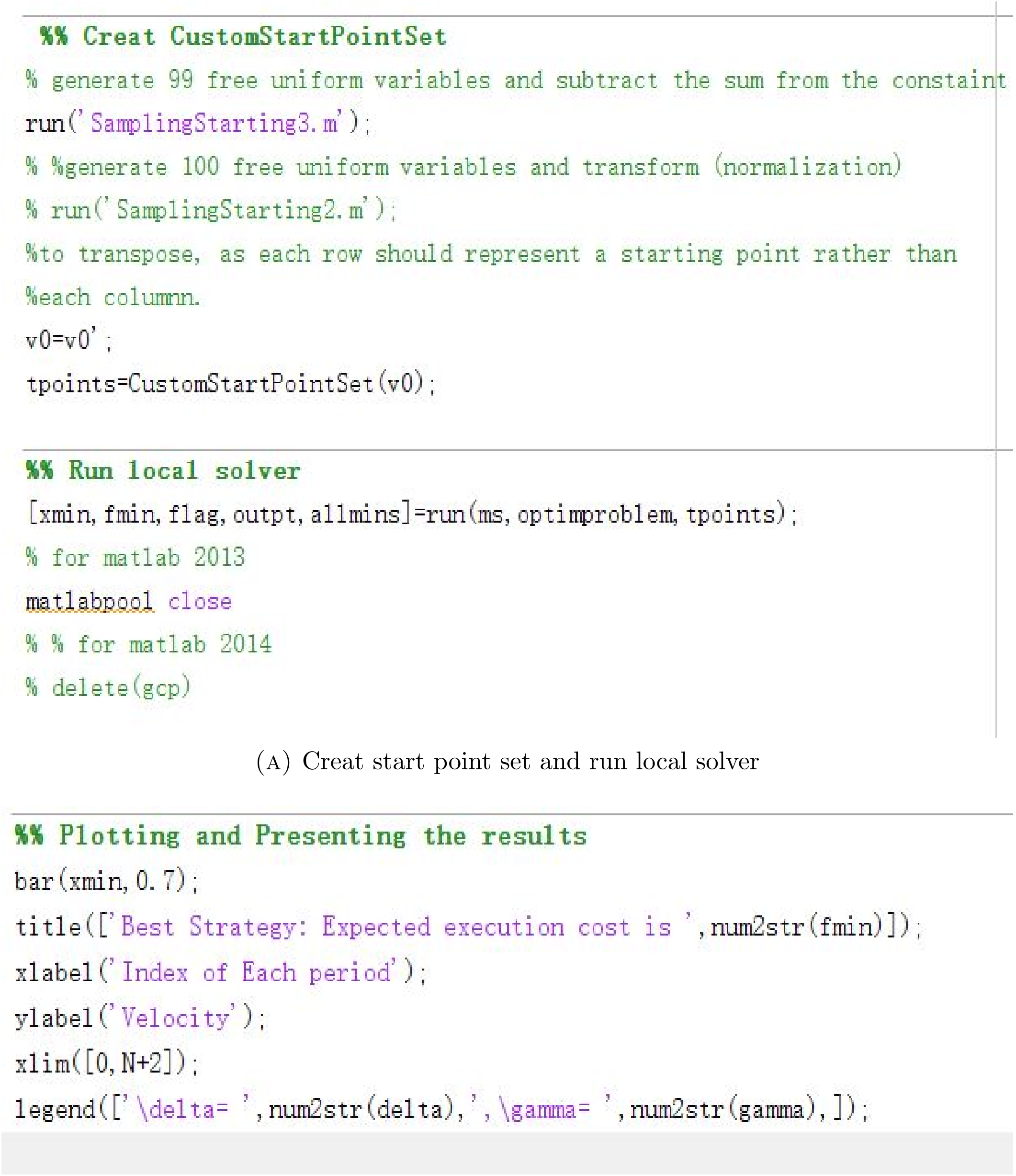
(A) Initialise prarmeter and create objective function and constraint



```
%% Creat problem structure
opts=optimoptions(@fmincon,'Algorithm','sqp','MaxIter',800,'MaxFunEvals',150000);
optimproblem=createOptimProblem('fmincon','x0',vtrial,'objective',OptFunc,...
    'Aeq',Aeq,'beq',beq,'options',opts);

%% Creat Solver object
% for matlab 2013
ms=MultiStart('UseParallel','always','Display','iter');
% % for matlab 2014
% ms=MultiStart('UseParallel',true,'Display','iter');

%% set up parallel pool
% for matlab 2013
matlabpool
% %for matlab 2014
% parpool
```

(B) Create problem structure and solver object,and set up parallel computing

Figure A.3: Initialisation, creating problem structure and solver object, and setting up parallel computing



```
%% Creat CustomStartPointSet
% generate 99 free uniform variables and subtract the sum from the constaint
run('SamplingStarting3.m');
% %generate 100 free uniform variables and transform (normalization)
% run('SamplingStarting2.m');
%to transpose, as each row should represent a starting point rather than
%each columnn.
v0=v0';
tpoints=CustomStartPointSet(v0);

%% Run local solver
[xmin,fmin,flag,outpt,allmins]=run(ms,optimproblem,tpoints);
% for matlab 2013
matlabpool close
% % for matlab 2014
% delete(gcp)
```

(A) Creat start point set and run local solver

```
%% Plotting and Presenting the results
bar(xmin,0.7);
title(['Best Strategy: Expected execution cost is ',num2str(fmin)]);
xlabel('Index of Each period');
ylabel('Velocity');
xlim([0,N+2]);
legend(['\delta= ',num2str(delta),', \gamma= ',num2str(gamma),]);
```

(B) Set plotting property

FIGURE A.4: Running local solver and plotting

Bibliography

[1] Frédéric Abergel, Jean-Philippe Bouchaud, Thierry Foucault, Charles-Albert Lehalle, and Mathieu Rosenbaum. *Market Microstructure: Confronting Many Viewpoints*. John Wiley & Sons, 2012.

[2] Aurélien Alfonsi, Alexander Schied, and Alla Slynko. Order book resilience, price manipulation, and the positive portfolio problem. *SIAM Journal on Financial Mathematics*, 3(1):511–533, 2012.

[3] Robert Almgren and Neil Chriss. Value under liquidation. *Risk*, 12(12):61–63, 1999.

[4] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Risk*, 3:5–40, 2001.

[5] Robert F Almgren. Optimal execution with nonlinear impact functions and trading-enhanced risk. *Applied Mathematical Finance*, 10(1):1–18, 2003.

[6] Dimitris Bertsimas and Andrew W Lo. Optimal control of execution costs. *Journal of Financial Markets*, 1(1):1–50, 1998.

[7] Jean-Philippe Bouchaud, Yuval Gefen, Marc Potters, and Matthieu Wyart. Fluctuations and response in financial markets: the subtle nature of 'random' price changes. *Quantitative Finance*, 4(2):176–190, 2004.

[8] Gary P Brinson, L Randolph Hood, and Gilbert L Beebower. Determinants of portfolio performance. *Financial Analysts Journal*, 51(1):133–138, 1995.

[9] SEC CFTC and US SEC. Findings regarding the market events of may 6, 2010. *Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues*, 2010.

[10] Louis KC Chan and Josef Lakonishok. The behavior of stock prices around institutional trades. *Journal of Finance*, pages 1147–1174, 1995.

[11] Gianbiagio Curato, Jim Gatheral, and Fabrizio Lillo. Optimal execution with nonlinear transient market impact. *Available at SSRN 2539240*, 2014.

[12] Ngoc-Minh Dang. Optimal execution with transient impact. *Available at SSRN 2183685*, 2014.

[13] Jim Gatheral. No-dynamic-arbitrage and market impact. *Quantitative Finance*, 10(7):749–759, 2010.

[14] Jim Gatheral and Alexander Schied. Dynamical models of market impact and algorithms for order execution. *HANDBOOK ON SYSTEMIC RISK, Jean-Pierre Fouque, Joseph A. Langsam, eds*, pages 579–599, 2013.

[15] Jim Gatheral, Alexander Schied, and Alla Slynko. Transient linear price impact and fredholm integral equations. *Mathematical Finance*, 22(3):445–474, 2012.

[16] Joel Hasbrouck. Empirical market microstructure: The institutions, economics, and econometrics of securities trading, 2007.

[17] Gur Huberman and Werner Stanzl. Price manipulation and quasi-arbitrage. *Econometrica*, 72(4):1247–1275, 2004.

[18] The MathWorks Inc. *Global Optimization Toolbox User's Guide*, 2004-2015.

[19] The MathWorks Inc. *Optimization Toolbox User's Guide*, 2004-2015.

[20] The MathWorks Inc. *Parallel Computing Toolbox User's Guide*, 2004-2015.

[21] Hizuru Konishiy and Naoki Makimoto. Optimal slice of a block trade. *Risk*, 3(4), 2001.

[22] Fabrizio Lillo, J Doyne Farmer, and Rosario N Mantegna. Econophysics: Master curve for price-impact function. *Nature*, 421(6919):129–130, 2003.

[23] Thomas F Loeb. Trading cost: the critical link between investment information and results. *Financial Analysts Journal*, 39(3):39–44, 1983.

[24] Julian Lorenz and Robert Almgren. Mean–variance optimal adaptive execution. *Applied Mathematical Finance*, 18(5):395–422, 2011.

[25] Jonathan R Macey and Maureen O'hara. The law and economics of best execution. *Journal of Financial Intermediation*, 6(3):188–223, 1997.

[26] Esteban Moro, Javier Vicente, Luis G Moyano, Austin Gerig, J Doyne Farmer, Gabriella Vaglica, Fabrizio Lillo, and Rosario N Mantegna. Market impact and trading profile of hidden orders in stock markets. *Physical Review E*, 80(6):066102, 2009.

[27] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

[28] Anna A Obizhaeva and Jiang Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32, 2013.

[29] Andre F Perold. The implementation shortfall: Paper versus reality. *The Journal of Portfolio Management*, 14(3):4–9, 1988.

[30] Silviu Predoiu, Gennady Shaikhet, and Steven Shreve. Optimal execution in a general one-sided limit-order book. *SIAM Journal on Financial Mathematics*, 2(1):183–212, 2011.

[31] Alexander Schied and Torsten Schöneborn. Risk aversion and the dynamics of optimal liquidation strategies in illiquid markets. *Finance and Stochastics*, 13(2):181–204, 2009.