Optimal Trading Under Non-negativity Constraints Using Approximate Dynamic Programming

Shahin Abbaszadeh[1*], Tri-Dung Nguyen[1,2], Yue Wu[1]

[1]Management School, University of Southampton
[2]School of Mathematical Sciences, University of Southampton

Abstract

In this paper we develop an extended dynamic programming (DP) approach to solve the problem of minimising execution cost in block trading of securities. To make the problem more practical, we add non-negativity constraints to the model and propose a novel approach to solve the resulting DP problem to near optimal results. We also include time lags in the problem state to account for the autoregressive behaviour of most financial securities as a way of increasing problem sensitivity to variability of prices and information. The computation times achieved for the proposed algorithm are fast and provide the possibility of live implementation. We demonstrate the benefits offered by the new approach through numerical analysis and simulation runs in comparison to the classic model without the non-negativity constraints.

Keywords: Dynamic programming, Finance, Optimization, Stochastic processes.

1 Introduction

1.1 Background and Literature Review

The growth in equity trading in recent decades which have been largely due to the ever increasing amounts of funds available to institutional investors such as pension and mutual funds, has triggered an interest in more effective management of trading costs. These costs are often called transaction costs or execution costs, which include commissions, bid/ask spreads, opportunity costs of waiting and price impact from trading. Loeb (1983) was among the first to recognise the importance of execution costs and discusses different aspects of trading cost with relation to capitalisation and spread of the stock and the funds available to the trader. Perold (1988), among others, documents that hypothetical portfolios or passive benchmarks constantly outperform the market and the portfolio manager by almost 20% per year. Chan and Lakonishok (1993) argue that since trading in equity markets is increasingly dominated by institutional investors, the implementation shortfall that Loeb (1983) and Perold (1988) discuss, may be due to the costliness of executing individual trade transactions that result in a more costly overall execution of the order. This overall transaction cost prompts the traders to break their order down into smaller transaction units which is then executed over a certain time period. A trading strategy that minimises the expected execution cost of the trade is defined as best execution strategy.

Chan and Lakonishok (1993) study the institutional effects of trade on equity prices. They argue that since trading in equity markets is increasingly dominated by institutional investors, the

*Correspondence: Management School, University of Southampton, SO17 1BJ Southampton, United Kingdom; s.abbaszadeh@soton.ac.uk

importance of transaction costs are increasing. They offer three explanations for price changes triggered by large trade volumes: (i) *short-run liquidity costs* that arise when there is not an immediate buyer or seller, and the attempt to attract the buyer or seller translate to a price concession, (ii) *imperfect substitution* where there is no particular stock as a substitute and buyer offers a premium for a large transaction, and (iii) *information effects* where the amount of trade reveals information about the trade that is incorporated into the subsequent prices. This and many other studies document portfolio managers' inability to outperform various passive benchmarks, despite considerable effort to analyse and select stocks.

Bertsimas and Lo (1998) were the first to consider the use of dynamic programming in acquiring the best execution strategy. They argue that the act of trading affects price dynamics that will result in changes to the future trading costs (since the demand for financial securities is not perfectly elastic, the price impact of current trades, however small, can affect the course of future prices). Also they observe that trading takes time as large trades need to be executed over numerous periods. They propose and solve the dynamic trading problem with the use of dynamic programming framework. Bertsimas and Lo (1998) offer an analytic solution to the transaction cost problem in the case of a single stock. They solve the problem using dynamic programming methods and offer numerical analysis based on simulation runs on a scenario representing a real problem. Bertsimas et al. (1999) provide a similar method to the single stock model presented by Bertsimas and Lo (1998) to address the portfolio case. They also develop a new price impact model and an approximate model to solve the problem with non-negativity constraints for a specific case.

Chakravarty (2001) suggests that the medium sized trading done by institutions (which he argues is the optimal way if the trader intends to sell big block of shares and prevent big shocks to the market i.e. adverse price impact) has a disproportionate effect on the cumulative price change compared to trading in big chunks. Alexander and Peterson (2007) study the effects of trade clustering on various parts of the market. They argue that the size of the clusters tend toward 100, 500 and 5000 clusters which is mostly used by stealth and highly informed traders. They study the effects of clustering and block trading on the price of the underlying stock and conclude that the price impact from medium sized clusters are much higher than the small or big sized clusters. Domowitz and Yegerman (2005) review some of the early algorithmic trading services and give a brief result as to their efficiency compared to manual and other types of trading considering parameters such as trade size, type of security, etc. Butenko et al. (2005) provide models and algorithms for the problem of liquidating a certain amount of securities with or without the consideration of risk as a factor in decision making. Engle and Ferstenberg (2006) propose an extension to execution cost problem by incorporating the "risk return" trade-off into the problem. They achieve this through combination of transaction cost model with the portfolio planning problem.

Almgren and Chriss (2001) examine the relation between the risk (different levels of liquidity and the traders' idea of this) and optimal execution strategy. They obtain closed form solutions for the optimal trading strategy for any level of risk aversion from the trader. Almgren (2003) offers an update on the Almgren and Chriss (2001) modelling of the portfolio/stock trading costs by introducing a more robust price impact model. He incorporates trading-enhanced risk, an additional volatility measure that corresponds to the change in price following the demand in a more rapid execution of large blocks (liquidity premium demanded by the market is not deterministic). Kissell et al. (2004) build on the work of Bertsimas and Lo (1998) and Almgren and Chriss (2001) and give a more detailed breakdown of price change sources and analyse the causal relationship between these. They offer best execution strategy for three different scenarios: cost minimisation; balancing the risk vs cost; and price improvement. Subramanian and Sherali (2010) quantify the liquidity risk which corresponds to the difference between market price and realisation price from a traders position. They find an optimality condition for block trading.

He and Mamaysky (2005) offer an optimal execution strategy which is similar to that of Bertsimas and Lo (1998), but present a different price impact model based on Merton (1971). Huberman and Stanzl (2005) consider a linear price impact and argue that in a market without arbitrage opportunity, their proposed linear price impact model is optimal. They extend the work of Bertsimas and Lo (1998) by considering a risk averse trader. Hasbrouck and Seppi (2001) study various price dynamics along with other liquidity related issues such as focusing on the market as a whole and the effect of inter-company trades on the liquidity of a security. Kissell and Malamut (2006) introduce a framework that can accommodate different aspects of trading behaviour. A system that can act aggressively/patiently at times of favourable/unfavourable price changes would benefit investors greatly. The criteria also include the change in the dynamics of price changes as well as the different initial prices the algorithms can take.

1.2 Contributions and Paper Structure

Although Bertsimas and Lo (1998) acquire the closed form solution and the resulting trade strategy is the optimal execution strategy, it does not take into account the non-negativity constraints which would lead to a short-selling situation when in practice it is not possible to do so. If the price change is severe in an otherwise normal trade scenario, during a buy operation it might be optimal to sell and vice versa, while in practice the trader will not be allowed to operate based on that insight. This shortcoming might not affect a large portion of daily trades, but since it is a significant probability when trading large volumes of securities in a volatile market, the expected cost of ignoring the non-negativity constraints is still considerable. We develop an approximate dynamic programming approach to circumvent this costly possible scenario.

Our first contribution to the literature is the inclusion of non-negativity constraints in the formulation. To the best of our knowledge, the inclusion of non-negativity constraints has not been treated in the relevant literature. This constraint, however, increases the complexity of this problem substantially, as dynamic programming does not lend itself readily to constraints. We present here an approximate dynamic programming approach that enables us to solve a large combination of problems to near optimality through a generalised platform.

Secondly, we contribute to the literature through development of a bespoke Approximate Dynamic Programming method that combines both above contributions into a more complex problem. Our method offers a generalised platform through which a large combination of problems can be solved fast and near optimally.

The rest of this paper is structured as follows: we first present the modelling approach and results outlined by Bertsimas and Lo followed by our proposed extensions to their model in section 2. We then propose an approximate dynamic programming method in section 3 that handles the added sign constraints to the problem. We present the results and insights gained through numerical analysis in section 4 followed by conclusion and a look at the future possible directions for this research.

2 Models for Optimal Trade Execution

2.1 Basic Model

Consider a situation where a trader wants to buy a number S of shares in T consecutive periods of equal length. It is assumed that the price dynamics are known and the price p at period t follows an autoregressive process where it is related to the prices in the previous periods. It is also assumed that the effect of trade volume on the price is known to the trader. An additional information source is assumed, reflecting any complementary data the decision maker might use to infer the price behaviour. This additional source of information, denoted henceforth by x_t might be an index of the market where the share is traded or expert knowledge available to the

trader.

Consider s_t to denote the current number of stocks available at each stage and u_t to be the decision variable which is the trade volume in the current period. Based on this information the decision maker wants to optimise the number of shares traded in each period in order to minimise the overall cost of execution of the S shares within the T sequential periods.

The general form of the basic problem (Bertsimas and Lo, 1998) is:

$$\begin{aligned}
\min \quad & \mathbb{E}\left[\sum_{t=1}^{T} p_{t+1} u_t\right], \\
s.t. \quad & \sum_{t=1}^{T} u_t = S, && (1)\\
& s_1 = S, \quad s_{T+1} = 0, && (2)\\
& s_{t+1} = s_t - u_t, \qquad t = 1, 2, ..., T-1 && (3)\\
& x_{t+1} = \rho x_t + \eta_t, \qquad t = 1, 2, ..., T && (4)\\
& p_{t+1} = p_t + \beta x_t + a u_t + \epsilon_t, \ t = 1, 2, ..., T && (5)
\end{aligned}$$

where η_t and ϵ_t are independent white noise processes with mean 0 and variance σ_ϵ^2 and σ_η^2 respectively. a is assumed to be positive and ρ to be bounded between 1 and -1, and β is the effect of information on price dynamics and is a given at the start of the optimisation period. Constraints (1) and (2) explain the dynamics of the stock at hand and ensure that all the shares are executed during the T periods. Constraint (3) is to ensure that the number of stocks (to execute) in the coming period (s_{t+1}) is the current number of stocks available (s_t) minus the trade volume in the current period. And constraints (4) and (5) express the dynamics of information and price evolution accordingly. Finally, the objective function is the expected cost over the trading period.

Throughout this study the execution is assumed to be a buy operation. However, the results would be applicable to a sell operation without any loss in generality. The objective of such problem would be to maximise revenue. Another point to be noted is that while the occurrence of negative prices in the model is possible, we implicitly assume that the parameters of the model are of such values as to make this unlikely. Thus we treat the model without such limitation.

2.2 Extending the Basic Model

An important aspect of the basic model is the absence of non-negativity constraints. Based on the variance of the information variable and also the price, the optimal solution might suggest a negative trade (i.e. sale in a buy operation). In a real situation where the short-selling option is not available, a naive strategy is to change the negative trades to zero. However, such a solution would significantly reduce the intended benefits of the model e.g. minimisation of transaction costs.

In the coming section we address these limitations by offering a flexible framework that provides near optimal results while handling the added complexity of non-negativity constraints and offer a generalised platform through which a large combination of problems can be solved fast and near optimally.

We first extend the model in vector form and present the closed form solution in line with the results of Bertsimas and Lo (1998). we augment the state space as follows:

$$\underbrace{\begin{bmatrix} p_{t+1} \\ s_{t+1} \\ x_{t+1} \end{bmatrix}}_{y_{t+1}} = \underbrace{\begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}}_{A} . \underbrace{\begin{bmatrix} p_t \\ s_t \\ x_t \end{bmatrix}}_{y_t} + \underbrace{\begin{bmatrix} a \\ -1 \\ 0 \end{bmatrix}}_{b} u_t + \underbrace{\begin{bmatrix} \epsilon_t \\ 0 \\ \eta_t \end{bmatrix}}_{\omega_t} .$$

The above formulation can be written as:

$$y_{t+1} = Ay_t + bu_t + \omega_t.$$

The cost at period t is:

$$\begin{aligned} g(y_t, u_t, \omega_t) &= p_{t+1}u_t \\ &= y_{t+1}^\intercal e_1 u_t \\ &= (Ay_t + bu_t + \omega_t)^\intercal e_1 u_t \\ &= y_t^\intercal (A^\intercal e_1)u_t + b^\intercal e_1 u_t^2 + \omega_t^\intercal e_1 u_t, \end{aligned}$$

where e_1 represents the first column of the identity matrix.

At the final stage ($t = T$), the value function is equal to the expected value of cost function with respect to ω_T. We substitute the decision variable u_t with s_T which is the only available decision at $t = T$, and obtain:

$$u_T = s_T = e_2^\intercal * y_T.$$

Thus,

$$\begin{aligned} V_T(y_T) &= \mathbb{E}[g(y_T, u_T, \omega_T)] \\ &= \mathbb{E}[y_T^\intercal (A^\intercal e_1)u_T + b^\intercal e_1 u_T^2 + \omega_T^\intercal e_1 u_T] \\ &= y_T^\intercal (A^\intercal e_1)e_2^\intercal y_T + b^\intercal e_1 y_T^\intercal (e_2 e_2^\intercal) y_T \\ &= y_T^\intercal \left((A^\intercal e_1)e_2^\intercal + b^\intercal e_1 (e_2 e_2^\intercal)\right) y_T \\ &= y_T^\intercal K_T y_T, \end{aligned}$$

where $K_T = (A^\intercal e_1)e_2^\intercal + b^\intercal e_1(e_2 e_2^\intercal)$.

For the second last period, i.e. stage $(T-1)$, we have:

$$\begin{aligned} V_{T-1}(y_{T-1}) &= \min_{u_{T-1}} \mathbb{E}[g(y_{T-1}, u_{T-1}, \omega_{T-1})V_T(y_T)] \\ &= \min_{u_{T-1}} \mathbb{E}[y_{T-1}^\intercal (A^\intercal e_1)u_{T-1} + b^\intercal e_1 u_{T-1}^2 + \omega_{T-1}^\intercal e_1 u_T + y_T^\intercal K_T y_T] \\ &= \min_{u_{T-1}} \mathbb{E}[y_{T-1}^\intercal (A^\intercal e_1)u_{T-1} + b^\intercal e_1 u_{T-1}^2 + \omega_{T-1}^\intercal e_1 u_T \\ &\quad + (Ay_{T-1} + bu_{T-1} + \omega_{T-1})^\intercal K_T (Ay_{T-1} + bu_{T-1} + \omega_{T-1})] \\ &= \min_{u_{T-1}} \Big\{ (b^\intercal e_1 + b^\intercal K_T b)u_{T-1}^2 + y_{T-1}^\intercal \left(A^\intercal e_1 + A^\intercal (K_T + K_T^\intercal)b\right) u_{T-1} \\ &\quad + y_{T-1}^\intercal A^\intercal K_T A y_{T-1} + \mathbb{E}[\omega_{T-1}^\intercal K_T \omega_{T-1}] \Big\}. \end{aligned}$$

Since $V_{T-1}(y_{T-1})$ is a quadratic equation on u_{T-1} and we assume $a > 0$, we will have $(b^\intercal e_1 + b^\intercal K_T b) > 0$.

Lemma 1. *$\alpha_t = (b^\intercal e_1 + b^\intercal K_{t+1} b) > 0$ is true for all $t = 1, ..., T-1$.*

Proof. We demonstrate this result in appendix A.1. □

We acquire the optimal decision for period $(T-1)$ as:

$$u_{T-1}^* = -\frac{y_{T-1}^\intercal L_{T-1}}{2\alpha_{T-1}},$$

where $L_{T-1} = A^\intercal e_1 + A^\intercal (K_T + K_T^\intercal)b$, and $\alpha_{T-1} = b^\intercal e_1 + b^\intercal K_T b$.

We can then rewrite $V_{T-1}(y_{T-1})$ as:

$$V_{T-1}(y_{T-1}) = y_{T-1}^{\intercal} K_{T-1} y_{T-1} + C_{T-1},$$

where

$$K_{T-1} = \left(A^{\intercal} K_T A - \frac{L_{T-1} L_{T-1}^{\intercal}}{4\alpha_{T-1}} \right),$$
$$C_{T-1} = \mathbb{E}[\omega_{T-1}^{\intercal} K_T \omega_{T-1}], C_T = 0.$$

To find the closed form solution for the case of extended state, we assume that:

$$V_{t+1}(y_{t+1}) = y_{t+1}^{\intercal} K_{t+1} y_{t+1} + C_{t+1}.$$

Then, we have:

$$\begin{aligned}
V_t(y_t) &= \min_{u_t} \mathbb{E}[g(y_t, u_t, \omega_t) + V_{t+1}(y_{t+1})] \\
&= \min_{u_t} \mathbb{E}[y_t^{\intercal}(A^{\intercal} e_1) u_t + b^{\intercal} e_1 u_t^2 + y_{t+1}^{\intercal} K_{t+1} y_{t+1} + C_{t+1}] \\
&= \min_{u_t} \mathbb{E}[y_t^{\intercal}(A^{\intercal} e_1) u_t + b^{\intercal} e_1 u_t^2 + (Ay_t + bu_t + \omega_t)^{\intercal} K_{t+1}(Ay_t + bu_t + \omega_t) + C_{t+1}] \\
&= \min_{u_t} \Big\{ (b^{\intercal} e_1 + b^{\intercal} K_{t+1} b) u_t^2 + y_t^{\intercal} \left(A^{\intercal} e_1 + A^{\intercal}(K_T + K_T^{\intercal}) b \right) u_t \\
&\quad + y_t^{\intercal} A^{\intercal} K_{t+1} A y_t + \mathbb{E}[\omega_t^{\intercal} K_{t+1} \omega_t] + C_{t+1} \Big\}.
\end{aligned}$$

The above quadratic equation holds for all t, resulting in the optimal solution:

$$u_t^* = -\frac{y_t^{\intercal} L_t}{2\alpha_t},$$

where

$$\begin{aligned}
L_t &= A^{\intercal} e_1 + A^{\intercal}(K_{t+1} + K_{t+1}^{\intercal}) b, \\
\alpha_t &= b^{\intercal} e_1 + b^{\intercal} K_{t+1} b, \\
K_t &= A^{\intercal} K_{t+1} A - \frac{L_t L_t^{\intercal}}{4\alpha_t}, \\
K_T &= (A^{\intercal} e_1) e_2^{\intercal} + b^{\intercal} e_1 (e_2 e_2^{\intercal}), \\
C_t &= \mathbb{E}[\omega_t^{\intercal} K_{t+1} \omega_t] + C_{t+1}, \\
C_T &= 0.
\end{aligned}$$

Given the values of β, a and ρ, we can calculate K_t, L_t and C_t offline for all stages $t = 1, 2, ..., T$ by starting backward, after which the optimal solution is obtained in the forward stage. Algorithm 1 depicts the general steps of this procedure.

The closed form solution arrived at here is compatible with the results of Bertsimas and Lo (1998).

2.3 Including Non-negativity Constraints

The basic model in Bertsimas and Lo (1998) does not include non-negativity constraints to avoid negative trade volumes when the variations in the prices or the market warrant negative trading. An optimal policy that contains a negative trade in a block trade buying operation would be void in real trading situations since it is counter-intuitive and is not allowed in many systems (Bertsimas and Lo, 1998). The inclusion of non-negativity constraints would improve the performance of the trade regime substantially. However, adding non-negativity constraints to a dynamic programming problem increases the complexity of the problem considerably. Bertsimas and Lo (1998) provide a closed form solution for the problem with non-negativity constraints

Algorithm 1: Closed form solution in vector form

input : β, a and ρ
output: u^*

$K_T = (A^\intercal e_1)e_2^\intercal + b^\intercal e_1(e_2 e_2^\intercal)$
for $t = (T-1) \rightarrow 1$ **do**
  $L_t = A^\intercal e_1 + A^\intercal(K_{t+1} + K_{t+1}^\intercal)b$
  $\alpha_t = b^\intercal e_1 + b^\intercal K_{t+1} b$
  $K_t = A^\intercal K_{t+1} A - \frac{L_t L_t^\intercal}{4\alpha_t}$
$y_1 = y_0$
for $t = 1 \rightarrow T$ **do**
  $u_t^* = -y_t^\intercal L_t / 2\alpha_t$

when price dynamics follow a special pattern:

$$p_t = \theta p_{t-1} + a x_t u_t + \epsilon_t,$$

$$log x_t = log x_{t-1} + \eta_t.$$

However, as Huberman and Stanzl (2005) conclude, a linear price impact model is the most representative formulation for price impact in most real-world situations. The above special form of price evolution formula concerns only a limited case and is not an appropriate substitute for price impact in practical situations. In this paper we assume a linear price evolution model such as (**??**).

If we assume non-negativity must hold, the optimal trade size is decided by:

$$\begin{aligned} u_t^* &= \max(0, \min(s_t, -\frac{y_t^\intercal L_t}{2\alpha_t})) \\ &= \max(0^\intercal y_t, \min(e_2^\intercal y_t, -\frac{y_t^\intercal L_t}{2\alpha_t})), \end{aligned}$$

resulting in three different possible outcomes for each period. In the context of dynamic programming, after each stage in the backward progress through algorithm, the state space of the problem grows threefold (up to 3^T in the final stage), which results in considerably larger problems. We discuss the solution approach for handling this in the next section.

3 Approximate Dynamic Programming

Approximate dynamic programming is a range of approximation tools that are devised to address the inherent problems of dynamic programming framework in dealing with constraints. Classic approaches to dynamic programming are unable to deal with exponential growth of the computational requirements as the number of states increase. Unless the problems are defined to very restrictive assumptions, a simulation of the system is more easily constructed than a model. In this paper we employ a value function approximation method where we replace the piecewise value function which is resulted from the addition of non-negativity constraints with a single value function that best captures the characteristics of the three functions.

3.1 An Approximated Value Function

In order to overcome the added complexity of the non-negativity constraints, we approximate the cost to go at period $(t+1)$ with a quadratic function:

$$V_{t+1}(y_{t+1}) = y_{t+1}^\intercal Q_{t+1} y_{t+1} + B_{t+1}^\intercal y_{t+1} + C_{t+1},$$

where Q_t and B_t are quadratic and linear coefficients of the value function during period t respectively and C_t is the constant term.

Following the Bellman procedure, we have:

$$\begin{aligned} V_t(y_t) &= \min_{0\le u_t\le S_t} \mathbb{E}\left[g(y_t,u_t)+V_{t+1}(y_{t+1})\right] \\ &= \min_{0\le u_t\le S_t} \mathbb{E}[y_t^\intercal(A^\intercal e_1)u_t + b^\intercal e_1 u_t^2 + w_t^\intercal e1 u_t \\ &\quad +(Ay_t+bu_t+w_t)^\intercal Q_{t+1}(Ay_t+bu_t+w_t)+B_{t+1}^\intercal(Ay_t+bu_t+w_t)+C_{t+1}] \\ &= \min_{0\le u_t\le S_t} [b^\intercal e_1 + b^\intercal Q_{t+1}b]u_t^2 + [y_t^\intercal(A^\intercal e_1 + A^\intercal(Q_{t+1}+Q_{t+1}^\intercal)b)+B_{t+1}^\intercal b]u_t \\ &\quad +[(Ay_t)^\intercal Q_{t+1}(Ay_t)+B_{t+1}^\intercal Ay_t+\mathbb{E}(\omega_t^\intercal Q_{t+1}\omega_t)+C_{t+1}]. \end{aligned}$$

If there were no constraints on u_t, then the optimal u_t would be

$$u_t^* = -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t},$$

where $L_t = (A^\intercal e_1)+A^\intercal(Q_{t+1}+Q_{t+1}^\intercal)b$, $\beta_t = B_{t+1}^\intercal b$, and $\alpha_t = b^\intercal e_1 + b^\intercal Q_{t+1}b$.

Considering the three possible outcomes based on where u_t^* is situated in the state space, we will have three possible optimal u_t, each of which will yield different value functions:

$$u_t = \begin{cases} 0 & \text{if } -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} < 0, \\ S_t & \text{if } -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} > S_t, \\ -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} & \text{if } 0 \le -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} \le S_t \end{cases}$$

Let us define

$$U_1 = \{y| -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} < 0\},$$

$$U_2 = \{y| -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} > s_t\},$$

$$U_3 = \{y|0 \le -\frac{y_t^\intercal L_t+\beta_t}{2\alpha_t} \le s_t\},$$

to be mutually exclusive domains for y. If we replace the $u_{t,j}$ $(j=1,2,3)$ in $V_t(y_t)$ with above results, we will have the following optimal cost-to-go at each stage t $(t=1,2,...,T)$:

Case j=1 : $y \in U_1$

$$V_t(y_t) = y_t^\intercal(A^\intercal Q_{t+1}A)y_t + B_{t+1}^\intercal Ay_t + \mathbb{E}(\omega_t^\intercal Q_{t+1}\omega_t) + C_{t+1}.$$

Case j=2 : $y \in U_2$

$$V_t(y_t) = y_t^\intercal(A^\intercal Q_{t+1}A + L_t e_2^\intercal + e_2\alpha_t e_2^\intercal)y_t + [B_{t+1}^\intercal A + \beta_t e_2^\intercal]y_t + \mathbb{E}(\omega_t^\intercal Q_{t+1}\omega_t) + C_{t+1}.$$

Case j=3 : $y \in U_3$

$$V_t(y_t) = y_t^\intercal[A^\intercal Q_{t+1}A - \frac{3L_tL_t^\intercal}{4\alpha_t}]y_t + [B_{t+1}^\intercal A - \frac{3L_t^\intercal\beta_t}{2\alpha_t}]y_t + [\mathbb{E}(\omega_t^\intercal Q_{t+1}\omega_t) + C_{t+1} - \frac{3\beta_t^2}{4\alpha_t}].$$

In all three cases, this results in the quadratic form

$$V_t(y_t) = y_t^\intercal \Phi_{t,j} y_t + \Pi_{t,j} y_t + \Omega_{t,j},$$

with $\Phi_{t,j}$, $\Pi_{t,j}$ and $\Omega_{t,j}$ being defined appropriately according to the three aforementioned cases.

We now approximate the value function based on the three distinct functions resulting from each possible u_t by finding a quadratic function over the whole state space where the square distance from each of the above functions to it are minimum. For simplicity and clarity, we

assume a linear price impact with AR(1). These results can be extended to AR(m) without the loss in generality. If we assume linear price formulation and no time lag in the price evolution, A and b have the following formats:

$$A = \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}, \quad b = \begin{bmatrix} a \\ -1 \\ 0 \end{bmatrix}.$$

Lemma 2. *$\Phi_{t,j}$ and consequently Q_t have the following format:*

$$\Phi_{t,j} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix}, Q_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & x_1 & x_2 \\ 0 & x_3 & x_4 \end{bmatrix}.$$

Proof. see appendix A.2. □

Under linear price impact, the approximate value function has the following form :

$$\hat{V}_t(y_t) = y_t^\intercal Q_t y_t + B_t^\intercal y_t + C_t,$$

where

$$Q_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & q_{t,1} & q_{t,2} \\ 0 & q_{t,3} & q_{t,4} \end{bmatrix}, \text{ and } B_t = \begin{bmatrix} 0 \\ r_{t,1} \\ r_{t,2} \end{bmatrix}.$$

We need to find (Q_t, B_t, C_t) that minimises the least squares equations.

$$(Q_t, B_t, C_t) = \underset{(\{q_t\},\{r_t\},C_t)}{\operatorname{argmin}} \int_{y_t} (V_t(y_t) - \hat{V}_t(y_t))^2 dy_t,$$

where

$$\begin{aligned} & \int_{y_t} (V_t(y_t) - \hat{V}_t(y_t))^2 dy_t = \\ & \int_{U_1} [y_t^\intercal (Q_t - \Phi_{t,1}) y_t + (B_t^\intercal - \Pi_{t,1}) y_t + C_t - \Omega_{t,1}]^2 dy_t \\ + & \int_{U_2} [y_t^\intercal (Q_t - \Phi_{t,2}) y_t + (B_t^\intercal - \Pi_{t,2}) y_t + C_t - \Omega_{t,2}]^2 dy_t \\ + & \int_{U_3} [y_t^\intercal (Q_t - \Phi_{t,3}) y_t + (B_t^\intercal - \Pi_{t,3}) y_t + C_t - \Omega_{t,3}]^2 dy_t. \end{aligned}$$

Figure 1 depicts a simplified graph that shows the approximated value function based on the three value functions that are optimal in each region.

The quadratic coefficient $(Q_t - \Phi_{t,j})$ is

$$Q_t - \Phi_{t,j} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & q_{t,1} - k_{t,j,1} & q_{t,2} - k_{t,j,2} \\ 0 & q_{t,3} - k_{t,j,3} & q_{t,4} - k_{t,j,4} \end{bmatrix},$$

and the linear coefficient is

$$B_t^\intercal - \Pi_{t,j} = \begin{bmatrix} 0 \\ r_{t,1} - f_{t,j,1} \\ r_{t,2} - f_{t,j,2} \end{bmatrix},$$

where $k_{t,j,i}$ is the i_{th} element of $\Phi_{t,j}$ corresponding to that of Q and $f_{t,j,i}$ is the i_{th} element of $\Pi_{t,j}$ corresponding to elements of B_t. The above results also render p, the first element of state space, irrelevant for the rest of the calculations. Finally we have $C_t - n_{t,j}$ as the constant term.

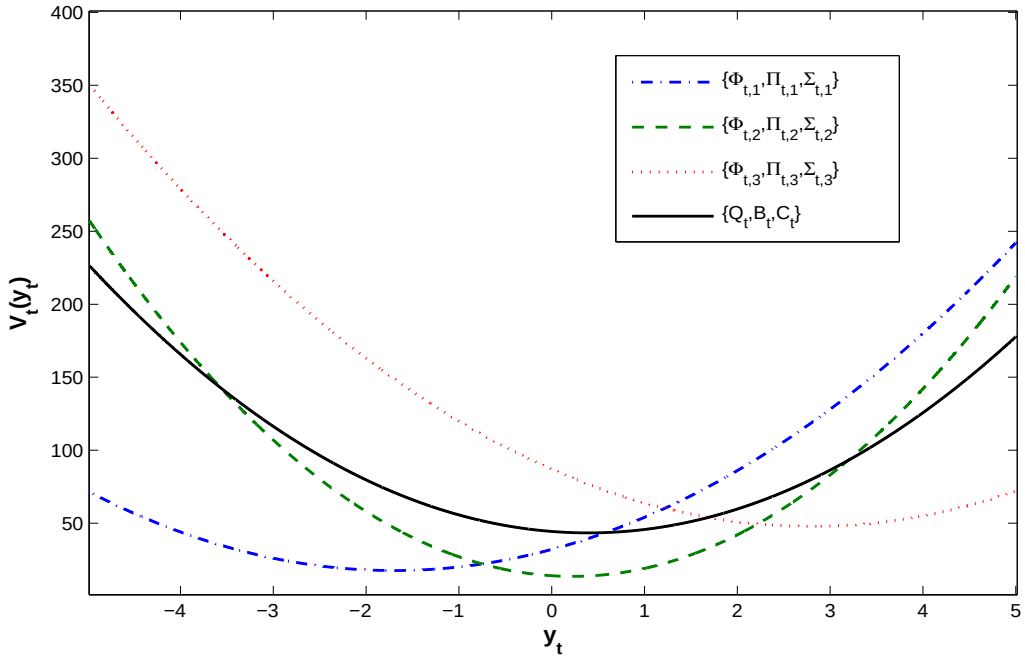

Figure 1: The three regions (representing U in cases $j = 1, 2, 3$) and approximation of the three value functions that are optimal in each region.

At each stage t, we need to solve the following minimisation problem:

$$(\{q\},\{r\},C) = argmin_{q,r,n} \int_y (V(y) - \hat{V}(y))^2 dy$$

where

$$\begin{aligned}
& \int_y (V(y) - \hat{V}(y))^2 dy \qquad (6)\\
= & \int_{U_1} ([s\ x] \begin{bmatrix} q_1 - k_{1,1} & q_2 - k_{1,2} \\ q_3 - k_{1,3} & q_4 - k_{1,4} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} + \begin{bmatrix} r_1 - f_{1,1} & r_2 - f_{1,2} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} \\
& + C - n_1)^2 dx ds \\
+ & \int_{U_2} ([s\ x] \begin{bmatrix} q_1 - k_{2,1} & q_2 - k_{2,2} \\ q_3 - k_{2,3} & q_4 - k_{2,4} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} + \begin{bmatrix} r_1 - f_{2,1} & r_2 - f_{2,2} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} \\
& + C - n_2)^2 dx ds \\
+ & \int_{U_3} ([s\ x] \begin{bmatrix} q_1 - k_{3,1} & q_2 - k_{3,2} \\ q_3 - k_{3,3} & q_4 - k_{3,4} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} + \begin{bmatrix} r_1 - f_{3,1} & r_2 - f_{3,2} \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} \\
& + C - n_3)^2 dx ds.
\end{aligned}$$

The three regions are obtained as follows:

- Case $j = 1$ (U_1):

$$\frac{-y_t^\intercal L_t - \beta_t}{2\alpha_t} < 0 \Leftrightarrow y_t^\intercal L_t > -\beta_t \Leftrightarrow \begin{bmatrix} s & x \end{bmatrix} \begin{bmatrix} l_2 \\ l_3 \end{bmatrix} > -\beta_t \Leftrightarrow$$

$$l_2 s + l_3 x > -\beta_t \Leftrightarrow x > \frac{-\beta_t - l_2 s}{l_3}$$

- Case $j = 2$ (U_2):

$$-(\frac{y_t^\intercal L_t + \beta_t}{2\alpha_t}) > s_t \Leftrightarrow y_t^\intercal L_t < -\beta_t - 2\alpha_t s_t \Leftrightarrow \begin{bmatrix} s & x \end{bmatrix} \begin{bmatrix} l_2 \\ l_3 \end{bmatrix} < -\beta_t - 2\alpha_t s_t \Leftrightarrow$$

$$l_2 s + l_3 x < -\beta_t - 2\alpha_t s_t \Leftrightarrow x < \frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3}$$

- Case $j = 3$ (U_3):

$$0 \leq -(\frac{y_t^\intercal L_t + \beta_t}{2\alpha_t}) \leq S_t \Leftrightarrow -2\alpha_t S_t - \beta_t \leq y_t^\intercal L_t \leq -\beta_t \Leftrightarrow$$

$$-2\alpha_t S_t - \beta_t \leq \begin{bmatrix} s & x \end{bmatrix} \begin{bmatrix} l_2 \\ l_3 \end{bmatrix} \leq -\beta_t - \beta_t - 2\alpha_t s_t < l_2 s + l_3 x < -\beta_t \Leftrightarrow$$

$$\frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3} < x < \frac{-\beta_t - l_2 s}{l_3}$$

where l_2 and l_3 are elements of L_t: $\begin{bmatrix} 0 & l_2 & l_3 \end{bmatrix}^\intercal$.

Considering both x and s have an additional range that they must comply with: ($x \in [-P_0, P_0]$ and $s \in [0, s_t]$), we have the following ranges to apply in the formulation. Let elements of x and s be parameters a_1, a_2, b_1 and b_2 such that $x \in [a_1, a_2]$ and $s \in [b_1, b_2]$.

$$U_1 = \{(x,s) | x \in [a_1 = \frac{-\beta_t - l_2 s}{l_3}, a_2 = P_0], s \in [b_1 = 0, b_2 = s_t]\},$$

$$U_2 = \{(x,s) | x \in [a_1 = -P_0, a_2 = \frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3}], s \in [b_1 = 0, b_2 = s_t]\},$$

$$U_3 = \{(x,s) | x \in [a_1 = \frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3}, a_2 = \frac{-\beta_t - l_2 s}{l_3}], s \in [b_1 = 0, b_2 = s_t]\}$$

Applying these ranges on (6) and expanding we will have the following which can be used to obtain the optimal values for Q, R and C.

$$\int_y (V(y) - \hat{V}(y))^2 dy = \int_0^{s_t} \int_{\frac{-\beta_t - l_2 s}{l_3}}^{P_0} \boldsymbol{F} \;\; dxds$$

$$+ \int_0^{s_t} \int_{-P_0}^{\frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3}} \boldsymbol{F} \;\; dxds + \int_0^{s_t} \int_{\frac{-\beta_t - 2\alpha_t s_t - l_2 s}{l_3}}^{\frac{-\beta_t - l_2 s}{l_3}} \boldsymbol{F} \;\; dxds$$

where

$$\begin{aligned} \boldsymbol{F} =& ((q_1 - k_{j,1})s^2 + (q_4 - k_{j,4})x^2 + ((q_2 - k_{j,2}) + (q_3 - k_{j,3}))sx \\ &+ (r_1 - f_{j,1})s + (r_2 - f_{j,2})x + C - n_j)^2 \end{aligned}$$

The above equations are quadratic functions on $(\{q\}, \{r\}, n)$ and can be rewritten as $R^\intercal M R + O^\intercal R + N$, where R is the vector $[q_1, q_2, q_3, q_4, r_1, r_2, C]$. M is a symmetric matrix derived from the coefficients of the above formulation, and O is the linear coefficient.

The resulting equation is of quadratic form and yields an optimal solution.

4 Numerical Analysis

Our numerical analysis consists of two studies. In section 4.1 we simulate the example used in Bertsimas and Lo (1998) to illustrate the comparative advantages gained through our approximate dynamic programming method with non-negativity constraints. We compare our method (ADP) with their closed form solution (B&L) and a naive strategy where the trade is divided into equal

Algorithm 2: Approximate Dynamic Programming

input: α_t, β_t, L_t, $\{Q_t^0, B_t^0, C_t^0\}$ for all t
output: $\{Q_t^*, B_t^*, C_t^*\}$

for $i = 1 \to M$ **do**
  for $t = (T-1) \to 1$ **do**
    for $j = 1 \to 3$ **do**
      $\{\Phi_{t,j}^i, \Pi_{t,j}^i, \Omega_{t,j}^i\} = \texttt{F1}(Q_{t+1}^{i-1}, B_{t+1}^{i-1}, C_{t+1}^{i-1}, j)$
  for $t = (T-1) \to 1$ **do**
    $\{Q_t^i, B_t^i, C_t^i\} = \texttt{F2}(\ \Phi_{t,j}^i, \Pi_{t,j}^i, \Omega_{t,j}^i)$
  if $VF\{Q_t^i, B_t^i, C_t^i\} < VF\{Q_t^{i-1}, B_t^{i-1}, C_t^{i-1}\}$ **then**
    for $t = 1 \to T$ **do**
      $\{Q_t^*, B_t^*, C_t^*\} = \{Q_t^i, B_t^i, C_t^i\}$

$\{\Phi,\Pi,\Omega\}$ = **Function** `F1` (Q, B, C, j)
if $j = 1$ **then**
  $\Phi = A^\intercal Q A$
  $\Pi = B^\intercal A$
  $\Omega = \mathbb{E}(\omega_t^\intercal Q \omega_t) + C$
if $j = 2$ **then**
  $\Phi = A^\intercal Q A + L_t e_2^\intercal + e_2 \alpha_t e_2^\intercal$
  $\Pi = B^\intercal A + \beta_t e_2^\intercal$
  $\Omega = \mathbb{E}(\omega_t^\intercal Q \omega_t) + C$
if $j = 3$ **then**
  $\Phi = A^\intercal Q A - \frac{3 L_t L_t^\intercal}{4\alpha_t}$
  $\Pi = B^\intercal A - \frac{3 L_t^\intercal \beta_t}{2\alpha_t}$
  $\Omega = \mathbb{E}(\omega_t^\intercal Q \omega_t) + C - \frac{3\beta_t^2}{4\alpha_t}$

$\{Q, B, C\}$ = **Function** `F2` $(\Phi_t, \Pi_t, \Omega_t)$

$$\begin{aligned}\underset{(\{Q\},\{B\},C)}{\operatorname{argmin}} \int_{U_1} &[y_t^\intercal (Q_t - \Phi_{t,1}) y_t + (B_t^\intercal - \Pi_{t,1}) y_t + C_t - \Omega_{t,1}]^2 dy_t \\ &+ \int_{U_2} [y_t^\intercal (Q_t - \Phi_{t,2}) y_t + (B_t^\intercal - \Pi_{t,2}) y_t + C_t - \Omega_{t,2}]^2 dy_t \\ &+ \int_{U_3} [y_t^\intercal (Q_t - \Phi_{t,3}) y_t + (B_t^\intercal - \Pi_{t,3}) y_t + C_t - \Omega_{t,3}]^2 dy_t\end{aligned}$$

Period	σ_ϵ^2=0.01		σ_ϵ^2=0.1		σ_ϵ^2=1	
	B&L	ADP	B&L	ADP	B&L	ADP
2	0.26	12.78	0.31	13.05	0.38	13.31
4	0.45	14.57	0.51	15.77	0.58	16.89
6	0.67	19.43	0.78	21.93	0.87	24.43
8	1.00	28.85	1.11	33.32	1.22	38.19
10	1.37	45.95	1.53	52.84	1.68	59.84
12	1.92	70.15	2.13	80.01	2.35	89.70
14	2.60	103.12	2.82	116.53	3.04	129.81
16	3.32	147.59	3.56	164.93	3.80	182.29
18	4.19	204.63	4.49	226.87	4.75	249.02
20	5.05	276.41	5.35	303.77	5.63	330.96

Table 1: Execution time (in milliseconds) of B&L and ADP algorithms for variable number of periods (T) and market volatility rates (σ_ϵ^2 values)

sizes to be executed over the trading horizon. In this case we test the efficiency of each optimal strategy with regard to different realisations of information variable x. Since the variable x is the main driving factor in the volatility of prices in the scenario presented by Bertsimas and Lo (1998), we test the performance of the algorithms against different rates of variance in x.

In section 4.2 we simulate trading of a security from London Stock Exchange on a specific date based on intra-day trade information of that day. This is to illustrate the effectiveness of our method in a practical setting and its advantage over the classic and naive methods that do not include the non-negativity constraints.

We code the algorithms in MATLAB and run all the experiments on a 64-bit Windows 7 workstation having 4GB of RAM and quad-core Intel CPU at 2.6GHz. All the experiments were very fast and as such we forego a detailed examination of the time performance. As a brief guideline, table 1 shows the time taken for a single run of each of the algorithms for various trading periods and market volatility rates.

4.1 A Simulated Example

The example involves execution (buy) of 100000 shares over $T = 20$ periods. The current price is \$50. The parameters are set as follows: $a = 5 \times 10^{-5}$, $\beta = 5$, $\rho = 0.5$ and $\sigma_\epsilon^2 = (0.125)^2$. For a full description of the values and the reasoning behind the choice of the values, interested readers are referred to Bertsimas and Lo (1998). In summary, a is chosen to yield a price impact of \$500000 if the trader executes the 100000 shares in one transaction. The standard deviation of ϵ_t is calibrated to be one tick (12.5 cents) per period.

We assign different values to σ_η^2 (variance of the information evolution, a white noise process, representing the overall market volatility) in order to test the efficiency of the algorithms with regard to the market behaviour where it might drive the prices significantly up or down and render the non-negativity constraints relevant. For example if the optimal strategy where we do not apply the non-negativity constraints is to sell during a buy operation, the original method would likely offer negative trade as part of its optimal solution. However, if the assumption is that short-selling is not possible, then the original algorithm would not be able to accommodate the above cases. The naive strategy on the other hand is unable to utilise the swings in the price to minimise the incurred overall cost. The approximate dynamic programming method, as expected, provides superior results compared to the original method of Bertsimas and Lo (1998) and the naive strategy under a volatile market condition when the assumption is that a buy operation cannot include sales.

Table 2 depicts the values of the Expected execution cost for a single random run. Table 2 shows that the expected value function improves substantially for the ADP method compared

σ_η^2	ADP	B&L	% Diff.
0.001	5211140	5218652	0.14
0.002	5183163	5196262	0.25
0.005	5127863	5151103	0.45
0.01	5072956	5105838	0.65
0.02	5003845	5048489	0.89
0.05	4885584	4947445	1.27
0.1	4785283	4863176	1.63
0.2	4727752	4820594	1.96
0.5	4951978	5047339	1.93
1	5030476	5141406	2.21
2	4848128	5004119	3.22
5	4494581	4739153	5.44
10	4090485	4435450	8.43
20	3511292	3999008	13.89
50	2362097	3133048	32.64
100	1066983	2157133	102.17

Table 2: Expected execution cost in example execution of 100000 shares

to B&L when the variance of the market volatility increases. The naive strategy is equivalent in almost all instances of σ_η^2 to that of B&L and hence not shown here. As we are dealing with the volatility of the market, the more important aspect of the improvement of the ADP method over the other methods is apparent in the actual execution cost over the whole trade period as can be seen in Figure 2, which depicts the realized execution cost for various σ_η^2 rates. To achieve this, we run each algorithm with random realisations of x and calculate the execution cost at each stage. This graph shows the average of total execution cost for 50 simulation runs for each method.

The performance of these three methods are very similar when the σ_η^2 rates are very small and the probability of negative trade being optimal is relatively low. Since B&L method provides optimality with regard to the uncertainty in the market and includes short-selling in its policies, when the short-selling is prohibited, it loses that advantage. This is reflected in the average execution cost values in σ_η^2 rates of 1 and above. The ADP, on the other hand, takes into account the non-negativity constraints and provides optimal policies in which the majority of the trade is performed when the price has swung down and little trade when the prices are high. In other words, ADP is better equipped to take advantage of price variations without the need for short-selling. Note the sharp drop in ADP execution costs when the volatility increases beyond 1, where it is able to execute with higher probability a larger amount of the security in lower prices leading to lower overall execution cost.

4.2 An Empirical Example

The example shown in section 4.1 proves the suitability of the ADP approach in situations where short-selling is not allowed. However to gauge the improvement that this method will yield in a real-world situation we test the algorithm on real securities traded in the market and compare it to that of B&L as a benchmark on what would be the optimal policy when we ignore the non-negativity constraints.

We considered three securities from London Stock Exchange that represent various levels of liquidity and recorded the tick price data for each of the selected stock for the duration of two weeks. FTSE100 index was considered as the information vector x during the same period. We also recorded the publicly available trade and price information of each share for a particular day during the two weeks. Out of the FTSE100 companies, we chose Lloyds Banking Group shares

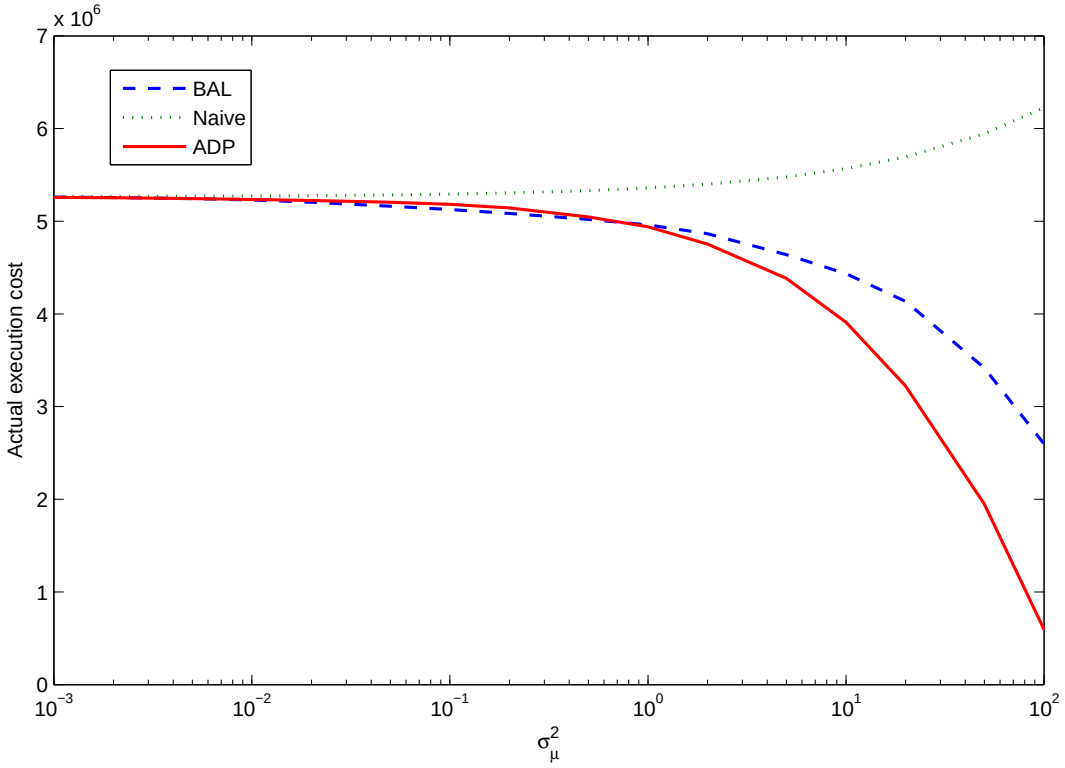

Figure 2: Actual execution cost for the three methods with variable σ_η^2 rates

as a high liquidity share, Rolls-Royce as a medium liquidity and Next Plc as representative of low liquidity securities. The choice of the three securities was not based on strict criteria but a loose interpretation of liquidity in order to evaluate the effects when applying the methods in this paper on stocks of varying liquidity. We also acquired intra-day price data for these securities where the information is available for the opening and closing price and the volume of trade during each minute. We considered the price between the opening and closing price to be the price of the share in that time period.

The first step is to estimate the parameters of $x_{t+1} = \rho x_t + \eta_t$ in order to be able to forecast the value of x from previous period. We performed a linear regression fit in R statistical software based on the historical time series data above. The parameters of information evolution ($x_{t+1} = \rho x_t + \eta_t$) which represent FTSE100 index were calculated as follows:

$$x_{t+1} = 70.91 + 0.98 x_t + \eta_t,$$

where $\mu_\eta = 0 \quad \text{and} \quad \sigma_\eta = 14.15$.

Table 3 details the coefficients of the price evolution model $p_{t+1} = p_t + \beta x_t + a u_t + \epsilon_t$ in relation to each chosen security.

Based on the parameters found in Table 3 pertaining to the behaviour of price based on the independent variables, as well as the parameters of the information evolution, we run our algorithm against that of Bertsimas and Lo (1998) to gauge the performance of these algorithms in a real-world application. We compare the results based on different number of trade periods as well as different rates of market volatility.

Figure 3 depicts the expected value function for the two methods based on 10 trade periods and varying degrees of market volatility. As can be seen, the B&L method only outperforms our method in the case of very small market volatility. As the volatility increases, the expected cost of trading the predefined number of shares increases for the B&L method. The ADP method

	Lloyds	Rolls-Royce	Next Plc
Price (1 lag)	$9.805e{-}01^{***}$	$8.776e{-}01^{***}$	$9.159e{-}01^{***}$
Std. Error	$(1.615e-02)$	$(1.886e-02)$	$(2.370e-02)$
Trade Volume	$-1.322e-10$	$1.227e-06^{+}$	$6.378e-05^{*}$
Std. Error	$(2.861e-09)$	$(1.359e-06)$	$(3.140e-05)$
UK100 Index	$7.790e-05^{+}$	$2.802e{-}02^{***}$	$4.523e{-}02^{***}$
Std. Error	$(1.573e-04)$	$(4.223e-03)$	$(1.335e-02)$
Residual Std. Err.	0.227	3.735	11.06
R^2	0.9474	0.9713	0.9612
Adj. R^2	0.9467	0.971	0.9607

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$, $^{+}p < 0.1$

Table 3: Regression analysis on the three chosen stocks

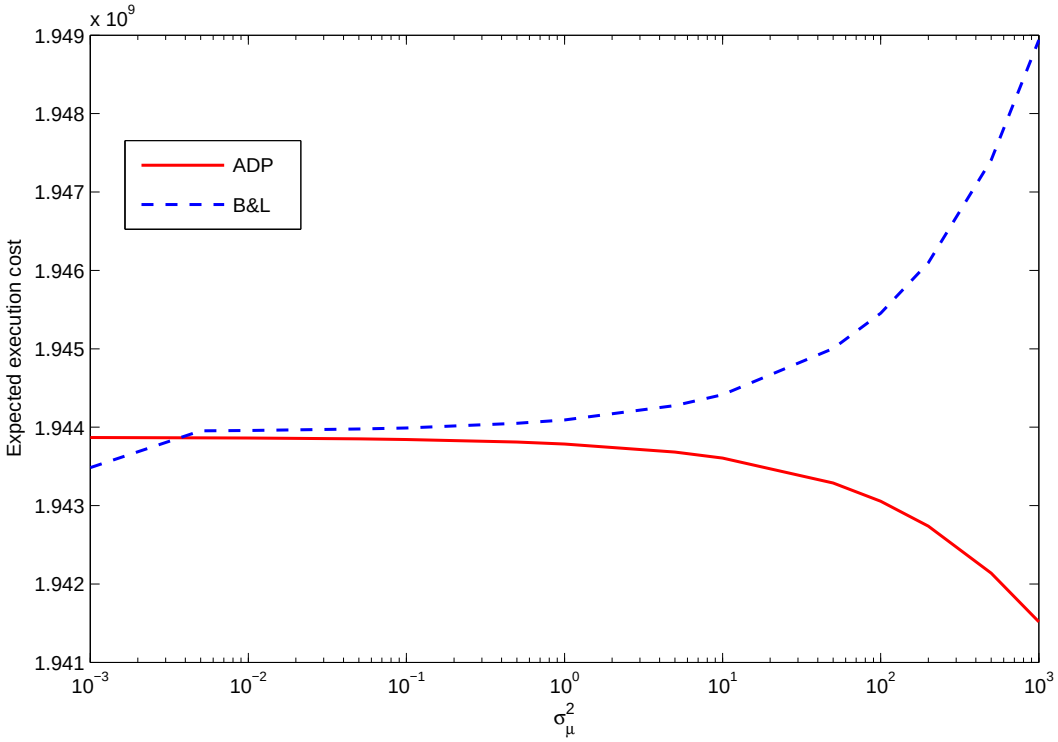

Figure 3: Expected execution cost for Lloyds share price with variable rates of σ_η^2

# of periods	ADP	B&L	% Diff.
10	8868042	8867250	-0.008
20	10595009	10586747	-0.077
30	11381698	11398890	0.151
40	11552346	11563438	0.096
50	11522654	11543933	0.184
60	11484477	11509757	0.220
70	11449253	11475887	0.232
80	11408889	11427038	0.159
90	11362621	11364245	0.014
100	11290233	11304324	0.124

Table 4: Actual execution cost realisation from each algorithm in execution of 100000 Lloyds shares and the percentage difference between the two algorithms based on the number of trading periods.

increases in performance as the volatility increases while the performance of B&L is declining. This is indicative of the ability of the ADP method in taking advantage of price swings when the B&L falls short because of neglecting the sign constraints. If short-selling is prohibited, our method adapts the policies online by taking the non-negativity into its framework from the start.

Table 4 outlines the actual execution cost of the two algorithms for the Lloyds shares. The ADP algorithm outperforms the original B&L method on most cases. When the trading periods are relatively few, there will naturally be less probability of favourable prices occurring and thus the ADP algorithm is slightly under-performing.

Figure 4 on the other hand depicts the results of the actual execution cost for the case of Lloyds share prices with both methods over varying market volatility. The benefits of the approximate dynamic programming method over the B&L method is apparent from this figure when the volatility increases.

Both other stocks show similar behaviour with varying trade periods and market volatility. Due to space limitation, the interested reader can find similar graphs to above for the Next plc and Rolls-Royce shares in appendix A.3.

5 Conclusion

Our extensions on the work of Bertsimas and Lo (1998) is aimed at bringing the problem closer to the real-world applications. We extend the AR(1) of the classic model to AR(m) and include non-negativity constraints. These extensions add significantly to the complexity of the problem. It is a well known shortcoming of the dynamic programming method that it cannot accommodate constraints very easily. We provide an approximate dynamic programming method that circumvents this. Our simulated and empirical results support the inclusion of non-negativity constraints, when the volatility of the market, and consequently of prices, are so that the probability of negative trades is significant. It improves the expected (and actual) execution cost in real circumstances and in comparison to the model by Bertsimas and Lo (1998). The difference between the performance of the models is not substantial when applied to low volatility scenarios. However, as we raise the market and price fluctuations in the model, the significance of our method over the other two methods become apparent. The benefits of the new method is also apparent in the empirical study where we test the methods on a set of data that warrants large swings in price. In both cases we observe improvement in the results which encourage the use of the new method over the previous methods under the conditions discussed.

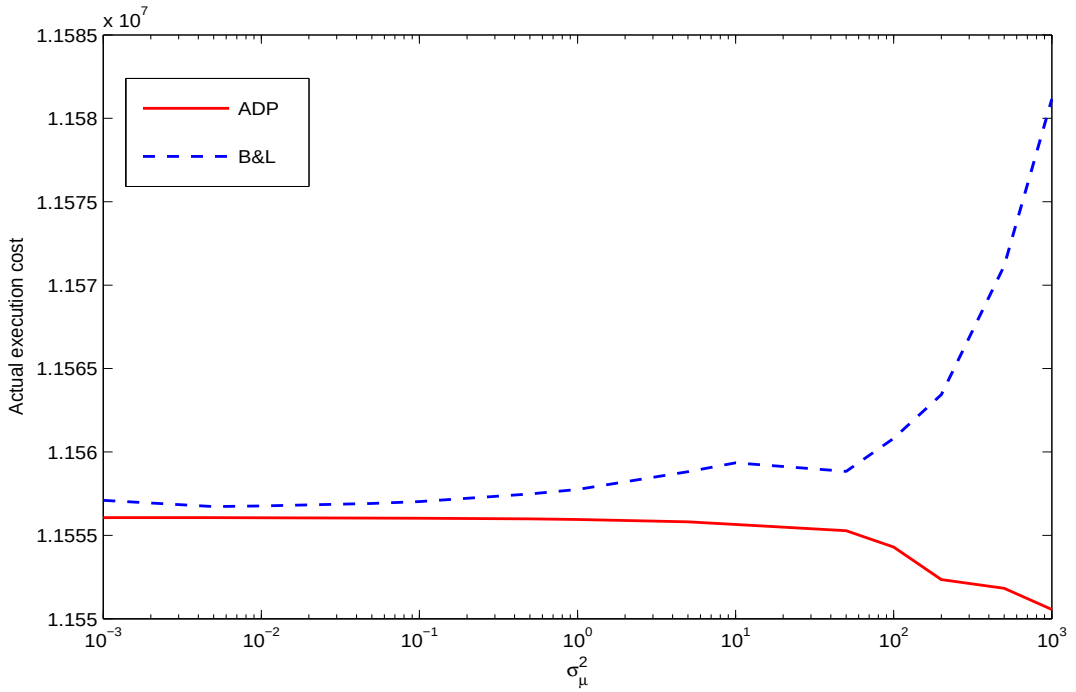

Figure 4: Actual execution cost for Lloyds share price with variable market volatility rates

References

Alexander, G., Peterson, M., 2007. An analysis of trade-size clustering and its relation to stealth trading. Journal of Financial Economics 84 (2), 435–471.

Almgren, R., Chriss, N., 2001. Optimal execution of portfolio transactions. Journal of Risk 3, 5–40.

Almgren, R. F., 2003. Optimal execution with nonlinear impact functions and trading-enhanced risk. Applied Mathematical Finance 10 (1), 1–18.

Bertsimas, D., Lo, A. W., 1998. Optimal control of execution costs. Journal of Financial Markets 1 (1), 1–50.

Bertsimas, D., Lo, A. W., Hummel, P., 1999. Optimal control of execution costs for portfolios. Computing in Science & Engineering 1 (6), 40–53.

Butenko, S., Golodnikov, A., Uryasev, S., 2005. Optimal security liquidation algorithms. Computational Optimization and Applications 32 (1), 9–27.

Chakravarty, S., 2001. Stealth-trading: Which traders trades move stock prices? Journal of Financial Economics 61 (2), 289–307.

Chan, L. K. C., Lakonishok, J., 1993. Institutional trades and intraday stock price behavior. Journal of Financial Economics 33 (2), 173–199.

Domowitz, I., Yegerman, H., 2005. The cost of algorithmic trading. Trading 2005 (1), 30–40.

Engle, R., Ferstenberg, R., 2006. Execution risk. Report, National Bureau of Economic Research.

Hasbrouck, J., Seppi, D., 2001. Common factors in prices, order flows, and liquidity. Journal of Financial Economics 59 (3), 383–411.

He, H., Mamaysky, H., 2005. Dynamic trading policies with price impact. Journal of Economic Dynamics and Control 29 (5), 891–930.

Huberman, G., Stanzl, W., 2005. Optimal liquidity trading. Review of Finance 9 (2), 165–200.

Kissell, R., Glantz, M., Malamut, R., 2004. A practical framework for estimating transaction costs and developing optimal trading strategies to achieve best execution. Finance Research Letters 1 (1), 35–46.

Kissell, R., Malamut, R., 2006. Algorithmic decision-making framework. The Journal of Trading 1 (1), 12–21.

Loeb, T. F., 1983. Trading cost: The critical link between investment information and results. Financial Analysts Journal 39 (3), 39–44.

Merton, R., 1971. Optimum consumption and portfolio rules in a continuous-time model. Journal of Economic Theory 3 (4), 373–413.

Perold, A. F., 1988. The implementation shortfall: Paper versus reality. The Journal of Portfolio Management 14 (3), 4–9.

Subramanian, S., Sherali, H. D., 2010. A fractional programming approach for retail category price optimization. Journal of Global Optimization 48 (2), 263–277.

A Appendix

A.1 Proof of Lemma 1

Lemma $\alpha_t = (b^\intercal e_1 + b^\intercal K_{t+1} b) > 0$ is true for all $t = 1, ..., T-1$.

Proof. Since $K_T = (A^\intercal e_1) e_2^\intercal + b^\intercal e_1 (e_2 e_2^\intercal)$, we have

$$\begin{aligned} K_T &= \begin{bmatrix} 1 \\ 0 \\ \beta \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} + a \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & a & 0 \\ 0 & \beta & 0 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} (b^\intercal e_1 + b^\intercal K_T b) =& a + \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & a & 0 \\ 0 & \beta & 0 \end{bmatrix} \begin{bmatrix} a \\ -1 \\ 0 \end{bmatrix} \\ =& a + \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ -a \\ -\beta \end{bmatrix} = a. \end{aligned}$$

Since a is assumed to be positive, $(b^\intercal e_1 + b^\intercal K_T b) > 0$ is true.

For K_{T-1} we have:

$$\begin{aligned} K_{T-1} =& A^\intercal K_T A - \frac{L_{T-1} L_{T-1}^\intercal}{4\alpha_{T-1}} \\ =& \begin{pmatrix} 0 & 1 & 0 \\ 0 & a - \frac{a^2}{4\,\alpha_1} & -\frac{\beta\,\rho\,a}{4\,\alpha_1} \\ 0 & \beta + \beta\,\rho - \frac{\beta\,\rho\,a}{4\,\alpha_1} & -\frac{\beta^2\,\rho^2}{4\,\alpha_1} \end{pmatrix} \end{aligned}$$

resulting in $\alpha_{T-2} = (b^\intercal e_1 + b^\intercal K_{T-1} b) = a - \frac{\alpha_{T-1}}{4}$.

Continuing in this fashion we get:

$$\alpha_t = \alpha_{t+1} - \frac{(\frac{2\alpha_{t+1}}{t})^2}{4\alpha_{t+1}}.$$

Since $\alpha_{T-1} = a$ which is assumed to be positive and since at each backward stage a value smaller than itself is deduced from it, it is concluded that α_t is a non-negative value for all $t = 1, ..., T-1$.

□

A.2 Proof of Lemma 2

Lemma $\Phi_{t,j}$ and consequently Q_t have the following format:

$$\Phi_{t,j} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix}, Q_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & x_1 & x_2 \\ 0 & x_3 & x_4 \end{bmatrix}.$$

Proof. We prove this by way of induction:

First of all, K_T has the format $\begin{bmatrix} 0 & 1 & 0 \\ 0 & a & 0 \\ 0 & \beta & 0 \end{bmatrix}$ from the formula $K_T = (A^\intercal e_1) e_2^\intercal + b^\intercal e_1 (e_2 e_2^\intercal)$ which corresponds to the general format.

We assume K_{t+1} to be of the format $\begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix}$.

Since we have $K_t = A^\intercal K_{t+1} A - \frac{L_t L_t^\intercal}{4\alpha_t}$, in which L_t and α_t are:

$$
\begin{aligned}
L_t =& A^\intercal e_1 + A^\intercal K_{t+1} b + b^\intercal K_{t+1} A \\
=& \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & \rho \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & \rho \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} a \\ -1 \\ 0 \end{bmatrix} \\
&+ \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix} \\
=& \begin{bmatrix} 1 \\ 0 \\ \beta \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & \rho \end{bmatrix} \begin{bmatrix} -1 \\ -d_1 \\ -d_3 \end{bmatrix} + \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2\rho \\ 0 & d_3 & d_4\rho \end{bmatrix} \\
=& \begin{bmatrix} 1 \\ 0 \\ \beta \end{bmatrix} + \begin{bmatrix} -1 \\ -d_1 \\ -\beta - \rho d_3 \end{bmatrix} + \begin{bmatrix} 0 \\ a - d_1 \\ -d_2\rho \end{bmatrix} \\
=& \begin{bmatrix} 0 \\ a - 2d_1 \\ -\rho(d_2 + d_3) \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
\alpha_t =& b^\intercal e_1 + b^\intercal K_{t+1} b \\
=& a + \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} a \\ -1 \\ 0 \end{bmatrix} \\
=& a + \begin{bmatrix} a & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ -d_1 \\ -d_3 \end{bmatrix} = d_1.
\end{aligned}
$$

Thus we have:

$$
\begin{aligned}
K_t =& A^\intercal K_{t+1} A - \frac{L_t L_t^\intercal}{4\alpha_t} \\
=& \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & \rho \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & d_2 \\ 0 & d_3 & d_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & \beta \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix} \\
&-(\begin{bmatrix} 0 \\ a - 2d_1 \\ -\rho(d_2 + d_3) \end{bmatrix} \begin{bmatrix} 0 & a - 2d_1 & -\rho(d_2 + d_3) \end{bmatrix})/4d_1 \\
=& \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 & \rho d_2 \\ 0 & \beta + \rho d_3 & \rho^2 d_4 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & \varpi & \varphi \\ 0 & \varphi & \varsigma \end{bmatrix} \\
=& \begin{bmatrix} 0 & 1 & 0 \\ 0 & d_1 - \varpi & \rho d_2 - \varphi \\ 0 & \beta + \rho d_3 - \varphi & \rho^2 d_4 - \varsigma \end{bmatrix}
\end{aligned}
$$

where $\varpi = \frac{(a-2d_1)^2}{4d_1}$, $\varphi = \frac{(a-2d_1)(-\rho(d_2+d_3))}{4d_1}$ and $\varsigma = \frac{\rho^2(d_2+d_3)^2}{4d_1}$.
The end result clearly has the same form as indicated.

□

# of periods	ADP	B&L	% Diff.
10	113196572	113105567	-0.080
20	113150598	113072115	-0.069
30	113128975	113178923	0.044
40	113092688	113065843	-0.023
50	113045862	113028378	-0.015
60	113002658	113158182	0.137
70	112949417	113053358	0.092
80	112852015	113009933	0.139
90	112723957	113060720	0.298
100	112684469	112971536	0.254

Table 5: Actual execution cost realisations from running each algorithm on 100000 Rolls-Royce shares

# of periods	ADP	B&L	% Diff.
10	721966680	721800780	-0.022
20	721755698	721629741	-0.017
30	721653652	721952501	0.041
40	721365874	721687618	0.044
50	721052563	721556396	0.069
60	720865896	721908650	0.144
70	720765149	721684057	0.127
80	719697785	721385540	0.234
90	719122901	721530130	0.334
100	718250352	721282194	0.422

Table 6: Actual execution cost realisations from running each algorithm on 100000 Next plc. shares

A.3 Additional performance indicators for section 4.2

The following tables and graphs demonstrate the results shown in relation to the ADP and B&L algorithms for Rolls-Royce and Next plc. share prices.

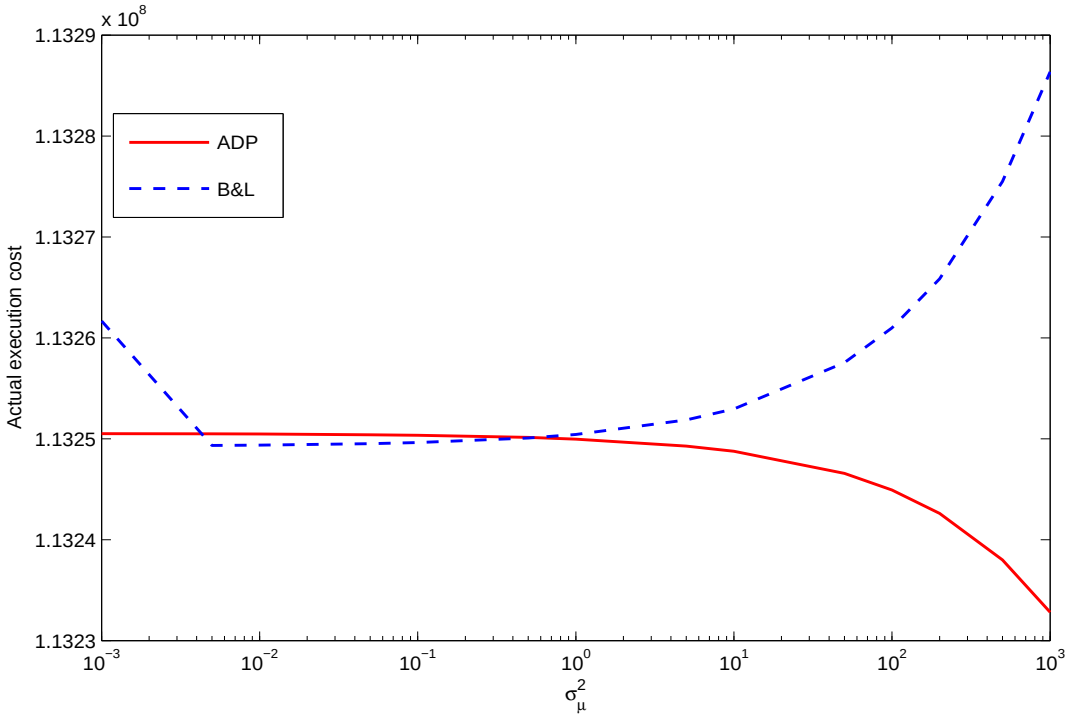

Figure 5: Actual execution cost for Rolls-Royce share price with variable market volatility

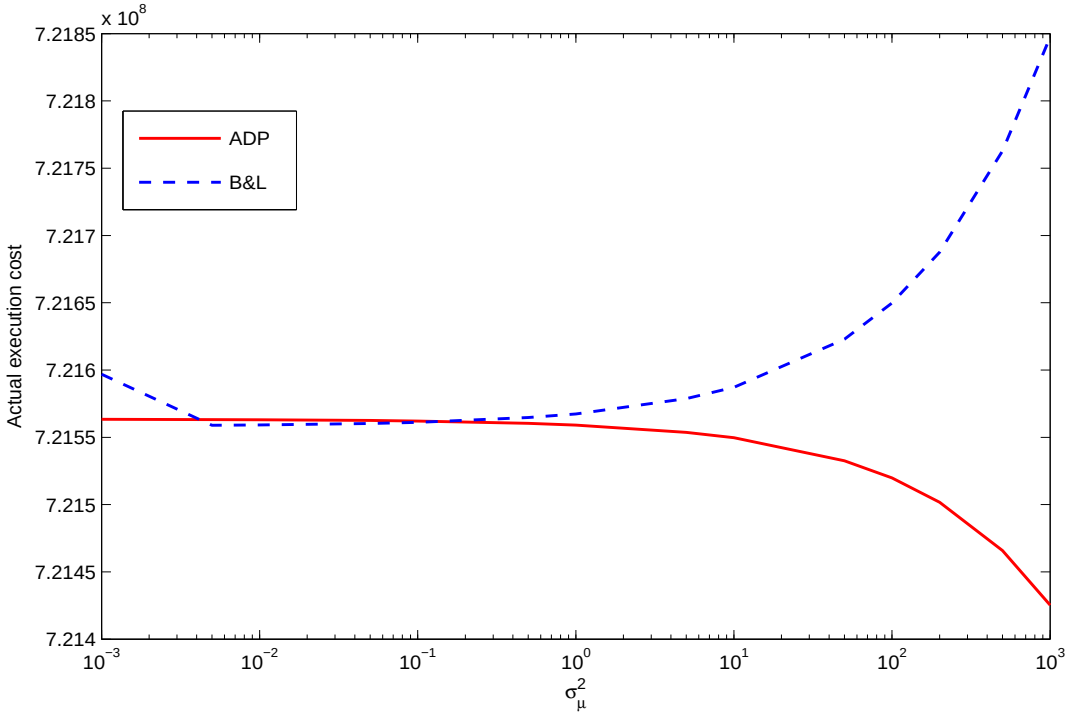

Figure 6: Actual execution cost for Next plc share price with variable market volatility