Adaptive Algorithmic Trading with Market Impact

Shi-Jie Deng
-joint work with Jim Dai and Ralph Yuan

Georgia Institute of Technology

SWUFE Banking Symposium 2011
June 27-July 2, 2011

Content

- Motivation
- Price Dynamics
  - Market Impact
  - Implementation Shortfall
- Optimal Execution Problem
  - Objective Function
  - Deterministic vs. Adaptive
- Linear-Quadratic Approach
  - Linear-Quadratic Auxiliary Problems
  - Solving Auxiliary Problems
- Numerical Result
  - Numerical Approximation
  - Full Price Adaptivity

Brief Introduction to Algorithmic Trading

- Some numbers on algorithmic trading:(data source: J.P.Morgan)
  - At least 30% of US equity markets is traded algorithmically: 360-750 billion shares/year.
  - A typical brokerage firm trades $1-5 trillion/year algorithmically. A saving of 1 bps(1bps=0.01%) translates to billions of dollars for the firm.
  - A typical algorithmic trading strategy might submit 500-1000 child orders.
- Definition of algorithmic trading:
  - The use of computer programs to automatically trade large pre-defined orders for financial securities in electronically accessible markets.
  - The goal of algorithmic strategies is not to maximize trading P&L but to effectively manage trading costs and risk.

Motivation

- Investors need to trade large sizes, far larger than the market place can absorb immediately without adverse price impact: buying drives price up and selling drives it down.
- Large orders(parent order) need to be parceled out in smaller sizes(child orders) and traded over relatively large periods of time, such as one day.
- However, extended trading hours incur uncertainty of the order's true execution cost, such as price volatility and liquidity risk.
- Since the order is executed over a long period it makes sense to use some form of 'average price' when judging the quality of a strategy. Both the mean and the risk of the 'average price' should be considered.

Model Setup

- Assume a buying problem with parent order size X_0 shares in one day. Divide 1 day into N periods by $t_0 = 0, t_1, ..., t_N$. Each period execute a child order of size $y_0, y_1, ..., y_{N-1}$:

$$\sum_{i=0}^{N-1} y_i = X_0.$$

- Stock price is determined by the trading activity of both exogenous traders and our trader;
- Price determined by exogenous traders: arithmetic Brownian motion with zero drift:

$$S_t = S_0 + \sigma S_0 B_t.$$

where volatility σ is assumed constant and known.

Market Impact

- Incurred by the act of our trading, i.e. demanding liquidity and thus changing the supply and demand balance.
- Different models:
  - Permanent impact: representing value information to other traders;
  - Temporary impact: representing cost of demanding liquidity;
  - Resilient impact: representing elastic liquidity as in Obizhaeva and Wang(2006).
- Assume only linear temporary market impact for simplicity of presentation:

$$\tilde{S}_{t_i} = S_{t_i} + \eta_i v_i y_i$$

  where y_i is the child order size, $v_i = \frac{y_i}{1 \text{ day}/N}$ is its execution speed and η_i is the market impact factor, which represents liquidity level.
- Same market impact model as Almgren and Lorenz(2011).

Implementation Shortfall

- The implementation shortfall(or slippage) is defined as the difference between the final execution cost with the parent order's initial notional value:

$$I = \sum_{i=0}^{N-1} \tilde{S}_{t_i} y_i - X_0 S_0$$

Objective

- The optimal execution is a trade-schedule that minimizes the risk adjusted cost for the trade:

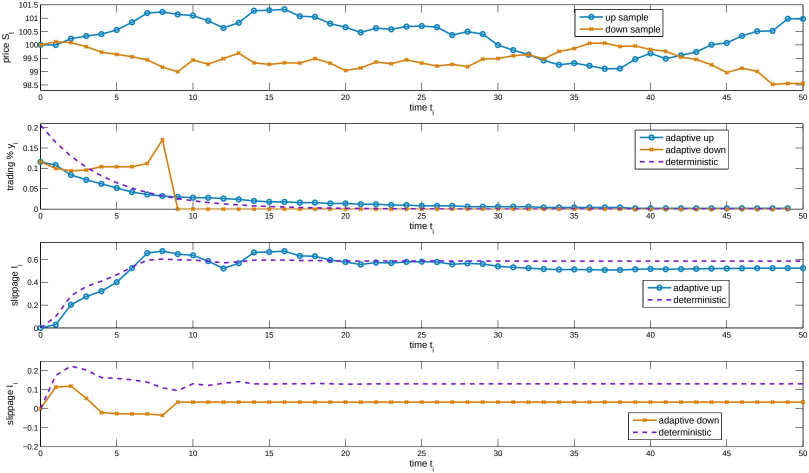
$$\mathbf{MV}(\kappa) : \min \left(\mathrm{E}[I_N] + \kappa \mathrm{Var}[I_N] \right)$$

  where $\kappa \geq 0$ is the risk aversion factor.
- By varying κ, we obtain a family of optimal execution strategies that create an "efficient frontier".
- Two extreme case:
  - when $\kappa = 0$, risk neutral, $y_i^* = \frac{X_0}{N}$, VWAP strategy.
  - when $\kappa = +\infty$, disregard market impact and trade everything at the beginning: $y_0^* = X_0$.
- A good strategy should balance the trade off between market impact cost from fast trading with the volatility risk from slow trading.

Deterministic vs. Adaptive Strategy

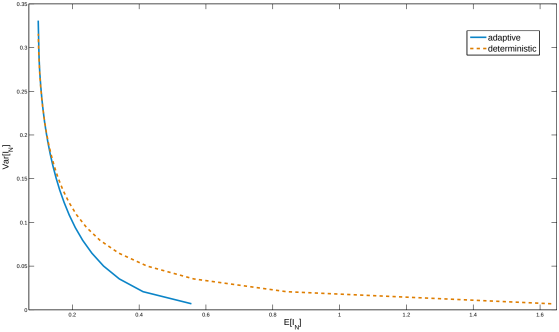
- Entire trade schedule is fixed in advance, first proposed by Almgren and Chriss(2001).
- Allow adjustment of child order size in response to price movement observed during execution.
  - The child order size y_i will not be determined until the price S_{t_i} has been observed. $y_0, y_1, ..., y_{N-1}$ are adapted to the filtration generated by price process S_{t_i}.
  - Since mean-variance objective suffers inconsistency in dynamic programming framework, we clarify the mean-variance objective, particularly the risk aversion factor κ is fixed at the order arrival time, and should not be modified during the trading process.
  - Aggressive-in-the-money(AIM): buy more when the price falls and buy less when the price rises.
  - Adaptivity pattern(AIM or passive-in-the-money(PIM)) depends on objective function(or utility function). Schied and Torsten(2009) shows that mean-variance objectives falls into the category of increasing absolute risk aversion(IARA) utility, whose adaptive strategy should be AIM.

Sample Trading Examples



Efficient Frontier Comparison

- Adaptive strategy should always work no worse than deterministic strategy.
- Adaptive strategy works particularly well when the trader is more risk averse, and two strategies coincide when risk aversion factor κ is small:



Linear-Quadratic Auxiliary Problems

- Dynamic Programming is not directly amenable for the minimization of mean-variance objective since the variance operator does not satisfy the smoothing property:

$$\forall 0 \leq s \leq t, \mathrm{Var}[\mathrm{Var}(\cdot|\mathcal{F}_t)|\mathcal{F}_s] \neq \mathrm{Var}(\cdot|\mathcal{F}_s).$$

- Instead, the expectation operator satisfies the smoothing property and the following linear quadratic problem can be solved by standard dynamic programming: for $r_0 \in \mathbf{R}$

$$\mathbf{LQ}(r_0) : \min E[r_0 I_N + I_N^2].$$

- **Proposition**: The optimal strategy of the original mean-variance problem is a subset of the union of solutions of all linear quadratic problems. Let

$$\begin{aligned} \Pi_{\mathbf{MV}}(\kappa) &= \{\pi|\pi \text{ is an adaptive strategy minimizing } \mathbf{MV}(\kappa)\}, \\ \Pi_{\mathbf{LQ}}(r_0) &= \{\pi|\pi \text{ is an adaptive strategy minimizing } \mathbf{LQ}(r_0)\} \end{aligned}$$

then $\Pi_{\mathbf{MV}}(\kappa) \subset \bigcup_{r_0 \in \mathbf{R}} \Pi_{\mathbf{LQ}}(r_0)$.

Solving the Original MV problem

- Assume we can solve $\mathbf{LQ}(r_0)$ for any $r_0 \in \mathbf{R}$, then an optimal strategy $\pi^*(\kappa) \in \Pi_{\mathbf{MV}}(\kappa)$ that solves $\mathbf{MV}(\kappa)$ should satisfy

$$\pi^*(\kappa) = \arg \min_{\pi \in \bigcup_{r_0 \in \mathbf{R}} \Pi_{\mathbf{LQ}}(r_0)} \left(\mathrm{E}[I_N|_\pi] + \kappa \mathrm{Var}[I_N|_\pi] \right)$$

- However, solving $\mathbf{LQ}(r_0)$ through dynamic programming for each $r_0 \in \mathbf{R}$ separately is computationally infeasible. Can we solve all auxiliary problems all at once?

Solving Auxiliary Problems

- Yes, we can! Take r_0 as a state variable given at the initial period, and

$$r_i = r_0 + 2 \times \text{realized partial slippage before time } t_i$$

Therefore, r_i is observable by time t_i, and it is a function of past child orders and price movement.

- More importantly, the objective linear-quadratic function can be decomposed as a sum of cost functions:

$$r_0 I_N + I_N^2 = \sum_{i=0}^{N-1} C_i(X_i, r_i, y_i, S_{t_{i+1}} - S_{t_i}).$$

where X_i is the remaining shares for execution.

- The cost function $C_i(X_i, r_i, y_i, S_{t_{i+1}} - S_{t_i})$ is determined by three parts: state variable (X_i, r_i), decision variable y_i such that $0 \le y_i \le X_i$ and exogenous stochastic process $S_{t_{i+1}} - S_{t_i}$.

Solving Auxiliary Problems, cont'd

- Bellman backward induction: Let $V_i(X_i, r_i)$ be the ith value function at state (X_i, r_i):
  - At last period, buy all shares left unexecuted $y^*_{N-1} = X_{N-1}$:

$$V_{N-1}(X_{N-1}, r_{N-1}) = C_{N-1}(X_{N-1}, r_{N-1}, X_{N-1}, 0).$$

  - For other periods: $i = N-2, N-3, ..., 0$:

$$V_i(X_i, r_i) = \min_{0 \le y_i \le X_i} \mathrm{E}[C_i(X_i, r_i, y_i, S_{t_{i+1}} - S_{t_i}) + V_{i+1}(X_{i+1}, r_{i+1}) | (X_i, r_i)].$$

    and the optimal child order size y^*_i is the minimizer of the RHS of the above equation.

Numerical Approximation

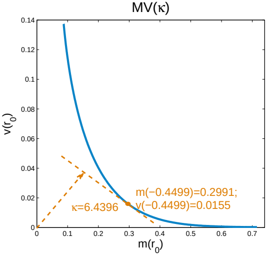
- Shrink the range of r_i from $\mathbf{R}$ to an interval $[Z_0, Z_K]$. The particular choices of Z_0 and Z_K are based upon deterministic strategy's performance during Monte Carlo simulation. Discretize $[Z_0, Z_K]$ into K intervals where r_i's take values at grid points $Z_0, Z_1, ..., Z_K$.
- Expectation in Bellman equation is computed through Gaussian-Legendre quadrature.
- When solving the minimization problem in the Bellman equation, we resort to deterministic strategy to provide a smaller search range within $[0, X_i]$.
- Once the Linear Quadratic problems are solved, $(X_0, r_0 = Z_k)$ for $(k = 0, 1, ..., K)$ is assigned as the initial state respectively to test the performance of $\mathbf{LQ}(r_0)$ through simulation. Sample mean $m(Z_k)$ and variance $v(Z_k)$ are computed for each Z_k.

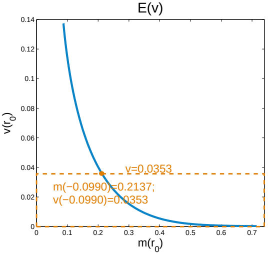
Numerical Approximation, cont'd

- Once the curve $\{m(Z_k), v(Z_k)\}_{k=0}^{k=K}$ is known, the point which produces the smallest mean-variance $m(Z_k) + \kappa v(Z_k)$ corresponds to the optimal strategy for $\mathbf{MV}(\kappa)$.
- Other equivalent objective, such as constrained minimization problem

$$\mathbf{E}(v) \quad : \quad \begin{aligned} & \min \mathrm{E}[I_N] \\ & \text{s.t. } \operatorname{Var}[I_N] \leq v \end{aligned}$$

can be solved in a similar way:





Full Price Adaptivity: Current Method

- Almgren and Lorenz(2011) solves an equivalent constraint variance minimization problem, which has a state variable of the same two dimension.
- Their decision variable contains both the child order size and an additional integrable function over the sample space of next period's price change.
- Dynamic programming suffers from the curse of dimensionality. The computation effort grows exponentially with the dimension of decision variables.
- Approximate the integrable decision function through a two dimension step function(i.e. three dimension decision variables), which corresponds to whether the price goes up or down for the next time period. This essentially restricts the child order size reacting to a binomial tree price process.

Full Price Adaptivity: Our Method

- As a comparison, our dynamic programming approach also has a state variable of two dimensions but a decision variable of only one dimension.
- More importantly, our algorithm's price adaptivity does not rely on the decision variable but the state variable r_i, which allows the decision variable to react differently to all possible future price levels, while maintaining simpler algorithm structure.
- With the same market coefficients and the same level of state variable resolution, our strategy performs consistently better than the current published result, and in some cases, saves more than 40% in terms of expected slippage under and same variance performance.

Full Price Adaptivity: Numerical Result

The implementation shortfall I_N reported below has been scaled by $\sigma X S_0$ where σ is the stock's daily volatility, such as $\sigma = 125$bps.

case	1		2	
κ	1.62		5.29	
	$\mathbb{E}[I_N]$	$\mathrm{Var}[I_N]$	$\mathbb{E}[I_N]$	$\mathrm{Var}[I_N]$
VWAP	0.1531	0.3309	0.1531	0.3309
Deterministic	0.2567	0.1439	0.4537	0.0771
Adaptive	0.2331	0.1435	0.3271	0.0771
	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{VWAP}]}$	$\frac{\mathrm{Var}[I_N]}{\mathrm{Var}[I_{VWAP}]}$	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{VWAP}]}$	$\frac{\mathrm{Var}[I_N]}{\mathrm{Var}[I_{VWAP}]}$
Deterministic	1.68	0.43	2.96	0.23
Adaptive	1.52	0.43	2.14	0.23
A&L(2011) Adaptive	1.52		2.27	
improvement %	0		5.73%	
improvement of $\mathbb{E}[\sigma I_N]$(bps)	0		2.49	

Full Price Adaptivity: Numerical Result

case	3		4	
κ	30.00		108.75	
	$\mathbb{E}[I_N]$	$\mathrm{Var}[I_N]$	$\mathbb{E}[I_N]$	$\mathrm{Var}[I_N]$
VWAP	0.1531	0.3309	0.1531	0.3309
Deterministic	1.0716	0.0274	1.9898	0.0109
Adaptive	0.4893	0.0274	0.6373	0.0109
	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{VWAP}]}$	$\frac{\mathrm{Var}[I_N]}{\mathrm{Var}[I_{VWAP}]}$	$\frac{\mathbb{E}[I_N]}{\mathbb{E}[I_{VWAP}]}$	$\frac{\mathrm{Var}[I_N]}{\mathrm{Var}[I_{VWAP}]}$
Deterministic	7.00	0.08	13.00	0.03
Adaptive	3.20	0.08	4.16	0.03
A&L(2011) Adaptive	3.92		7.09	
improvement %	18.37%		41.33%	
improvement of $\mathbb{E}[\sigma I_N]$(bps)	13.78		56.06	