Answer to Question 2: Mathematical Framework for Optimal Order Allocation

Introduction

The Optimal Order Allocation problem aims to execute a total volume of shares (S) over N trading periods, minimizing total execution cost, which includes the basic purchase price and temporary market impact. The goal is to find an allocation vector x = [x_1, x_2, ..., x_N], where x_i is the volume traded in period i, such that Σx_i = S.

Mathematical Problem Formulation

We define the following variables:

- **S:** Total shares to trade.
- **N:** Number of trading periods.
- **x_i:** Shares traded in period i.
- **P_i:** Share price at the start of period i.
- **g_i(x_i):** Temporary impact function (slippage) for trading x_i in period i.
- **h_i(x_i):** Permanent impact function (price change) for trading x_i in period i.

Objective Function:

Minimize the total execution cost, which is the sum of basic transaction costs and temporary impact costs across all periods:

Minimize: Σ [P_i * x_i + g_i(x_i)] for i = 1 to N

Where P_i is dynamically updated.

Constraints:

1. **Total Volume:** $\Sigma x_i = S$ (for $i = 1$ to N)
2. **Non-negativity:** $x_i \geqslant 0$ (for all $i = 1$ to N)
3. **Price Dynamics:** $P_{i+1} = P_i + \Delta P_i + h_i(x_i)$, where ΔP_i is random price change and $h_i(x_i)$ is permanent impact.

Impact Function Modeling:

As discussed in Question 1, $g_i(x_i)$ can be modeled as:

$g_i(x_i) = \alpha_i * x_i^k * (1 + \beta * \text{Volatility}_i) * \text{Liquidity\_Factor}_i$

And $h_i(x_i)$ can be modeled as:

$h_i(x_i) = \gamma_i * x_i$

Where γ_i is the permanent impact coefficient.

Algorithmic Framework for Solution

Given the dynamic nature of the problem, dynamic optimization or dynamic programming techniques are suitable.

Proposed Algorithm (Iterative Optimization Approach):

1. **Initialization:** Define S, N, initial P_1, and estimate initial parameters for impact functions (α_i, k, β, γ_i).
2. **Iterative Loop (for each period i from 1 to N):**
    - **Update Market State:** Update P_i based on previous period's impact and any random changes.
    - **Define Impact Function:** Update $g_i(x_i)$ and $h_i(x_i)$ parameters based on current market conditions (volatility, liquidity).
    - **Formulate Sub-Optimization Problem:** Determine x_i to minimize cost for the current period, considering remaining volume ($S_{remaining}$) and time.

- **Solve Optimization Problem:** Use numerical optimization algorithms (e.g., Quadratic Programming if functions are approximated quadratically, or Non-linear Programming for general non-linear functions). Dynamic Programming can also be used, defining a value function V(s, t) as the minimum cost to trade volume s from time t to N, solved via backward induction: **$V(s, t) = \min_{0 \leqslant x_t \leqslant s} [P_t * x_t + g_t(x_t) + V(s - x_t, t+1)]$** with $V(0, t) = 0$ and $V(s, N+1) = \infty$.

3. **Allocate x_i:** Assign the determined optimal x_i.
4. **Iterate:** Proceed to the next period until S is fully traded.

Tools and Techniques Used

- **Python:** For implementation.
- **NumPy:** For numerical operations.
- **SciPy.optimize:** For numerical optimization algorithms.
- **Pandas:** For data manipulation.
- **Matplotlib/Seaborn:** For visualization.
- **Time Series Models:** For forecasting market variables.
- **Machine Learning:** For more accurate impact function estimation.

Conclusion

This mathematical framework provides a comprehensive solution for optimal order allocation, accounting for temporary and permanent trading impacts and market dynamics. Accurate impact function modeling and appropriate optimization algorithms are crucial for minimizing execution costs and enhancing trading efficiency.