

The City College of New York

CSC 59867-E: Senior Project II

AI Agents for Decision Making in the Real World

Financial Agents, Stock Portfolio Orchestration, and Trading

By Saptarashmi Bandyopadhyay
Email: sbandyopadhyay@ccny.cuny.edu
Assistant Professor of Computer Science
City College of New York and Graduate Center at the City University of New York

October 5, 2026 CSC 59867 - E

Saptarashmi Bandyopadhyay

Today's Agenda

Part 01	Macro Portfolio Orchestration (DRL Framework)
Part 02	Market Microstructure & Multi-Agent Simulation
Part 03	Optimal Trade Execution (Almgren-Chriss Framework)
Part 04	In-Class Problem: Optimal Sizing of Trade Execution

opadhyay

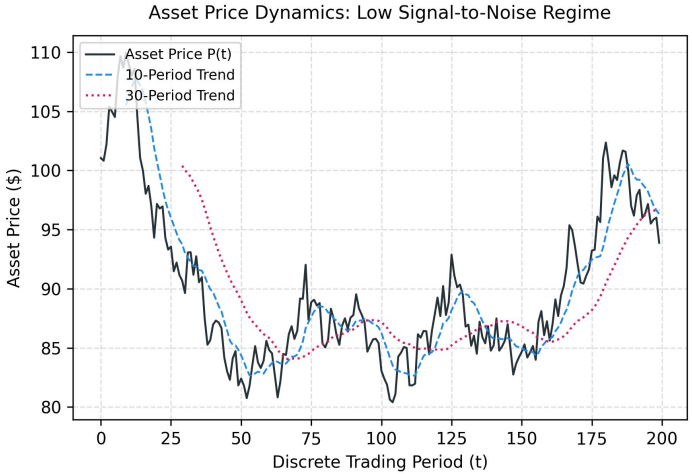
Macro Portfolio Orchestration

Saptarashmi Bandyopadhyay

Finance as an Agentic Actionable Environment

Predicting the next price move is tough: market data is noisy, and small changes can throw off a forecast.

- Instead of predicting every price, can we learn how to allocate our money?
- **Recall Non-stationarity**: Markets keep changing. Investors adjust their strategies, so an agent cannot assume the same market dynamics will hold forever.
- **Costly physical friction**: Rebalancing has a price. Commissions, slippage, and unfavorable execution prices all eat into returns.
- **Action space on the probability simplex**: Here the agent chooses portfolio weights w□ that add up to 1, not simply a buy or sell label.



Actual asset prices alongside rolling averages for different periods

Saptarashmi Bandyopadhyay

Portfolio Optimization as a Markov Decision Process

Observation state tensor (X□): We give the agent a rank-3 tensor of recent closing, high, and low prices, normalized using the latest closes over window n.

Price relative vector (y□): Quantifies period-to-period price movements across m non-cash assets plus cash (index 0). $v_{i,t}$ **is the price** (specifically, the closing price) of asset i at trading period t

Continuous action vector (w□): The agent outputs portfolio weights, with $w_{□,i} \geq 0$ and $\sum w_{□,i} = 1$.

Objective function: We want the average cumulative log return to be as large as possible over horizon t_f.

$$\mathbf{y}_t = \left(1, \frac{v_{1,t}}{v_{1,t-1}}, \frac{v_{2,t}}{v_{2,t-1}}, \dots, \frac{v_{m,t}}{v_{m,t-1}}\right)^\top, \quad R = \frac{1}{t_f}\sum_{t=1}^{t_f} \ln(\mu_t \mathbf{y}_t^\top \mathbf{w}_{t-1})$$

Saptarashmi Bandyopadhyay

Transaction Friction & Portfolio-Vector Memory (PVM)

Passive weight drift: Prices move even when we do nothing. That changes the weights from w□-1 to w'□ before the next rebalance.

Transaction remainder factor (μ□): μ□ tells us what fraction of the portfolio remains after paying buy and sell commissions c_s and c_p.

Recursive remainder solver: Because μ□ appears on both sides of the update, we solve for it iteratively using a contraction mapping.

Portfolio-Vector Memory (PVM): PVM keeps the agent's earlier portfolio weights w□-1 available, rather than relying on the network to remember them through gradients.

$$\mathbf{w}'_t = \frac{\mathbf{y}_t \odot \mathbf{w}_{t-1}}{\mathbf{y}_t^\top \mathbf{w}_{t-1}}, \quad \mu_t = \frac{1}{1 - c_p w_{t,0}} \left[1 - c_p w'_{t,0} - (c_s + c_p - c_s c_p) \sum_{i=1}^{m} (w'_{t,i} - \mu_t w_{t,i})^+ \right]$$

Sootarashmi Bandyopadhyay

Ensemble of Identical Independent Evaluators (EIIE)

Mini-machine factorized topology: Think of running the same small evaluator on every risky asset. The subnetworks share weights, and a final softmax sets the portfolio allocation.

Identity-agnostic valuation: We want the model to respond to price patterns, not just memorize which ticker it is looking at.

Linear computational scaling: Shared parameters let us process m assets with work that grows roughly linearly with m, rather than exploring every asset combination.

Saptarshmi Bandyopadhyay

Market Microstructure & Multi-Agent Simulation

Saptarashmi Bandyopadhyay

Market Microstructure & The Limit Order Book (LOB)

The double auction mechanism: Buyers submit bids and sellers submit asks. The exchange matches compatible orders in a continuous double auction.

Order book mechanics: Better prices come first; among orders at the same price, earlier orders get priority.

L1 vs. L2 market depth: L1 shows the best bid, best ask, and spread. L2 lets us see how many shares are available at several price levels.

Mechanical slippage: If a market order is bigger than the shares available at the best quote, it fills at worse prices farther down the book.

Sobtarashmi Bandyopadhyay

Agent-Based Interactive Discrete Event Simulation (ABIDES)

A backtest can make trades look easier than they are: our own orders may move prices. ABIDES models interacting trading agents and NASDAQ-style ITCH/OUCH messages with nanosecond timestamps.

Network & compute latency modeling: The simulator includes network delays and computation time. Those milliseconds—or less—can change which price an order gets.

Heterogeneous agent ecosystem: Market makers, momentum traders, and noise traders all act together. Their interactions determine the liquidity we observe.

Saptarshi Bandyopadhyay

Temporary vs. Permanent Drift

Selling a large block too quickly can push the price down. That means we receive less money for the shares.

Permanent market impact (g(v)): Some impact sticks around after our trades, as the market incorporates information conveyed by our orders.

Temporary market impact (h(v)): Other impact comes from temporarily consuming liquidity at the bid. It can fade when the selling pressure stops.

Linear impact formulation: Let v = n□/τ be the trading rate, where n□ shares are sold in interval k. The model uses permanent impact γv and temporary impact ε sign(v) + ηv.

$$S_k = S_{k-1} + \sigma\tau^{1/2}\xi_k - \tau g\left(\frac{n_k}{\tau}\right), \quad \tilde{S}_k = S_{k-1} - h\left(\frac{n_k}{\tau}\right)$$

Sooteroshni Bandyopadhyay

Optimal Trade Execution

Saptarashmi Bandyopadhyay

Why we can't always execute fast

Sell everything now, and we move the market. Sell too slowly, and we are exposed to price swings while we wait.

Implementation shortfall: This is what we lose relative to the initial market value when we actually execute the trades.

Trade trajectory representation: We sell X shares over N intervals of length $\tau = T/N$. At time t_k, x_k shares remain; we sell $n_k = x_{k-1} - x_k$ in interval k.

Boundary conditions: Start with $x_0 = X$ shares. By the end, the position must be fully sold: $x_N = 0$.

Sooraj Lakshmi Bandyopadhyay

Expected Shortfall and Variance

Total expected cost (E[x]): Add up the permanent price impact and the temporary trading costs across the schedule.

Total shortfall variance (V[x]): Keeping shares unsold exposes us to price changes. V[x] tracks that accumulated inventory risk.

Path-independent permanent cost: In the linear model, permanent impact adds up to (1/2)γX² no matter how we split the trades.

The Mean-Variance objective (U[x]): We minimize expected shortfall E[x] plus λ times the risk V[x]. A larger λ means we care more about risk.

$$E[x] = \frac{1}{2}\gamma X^2 + \epsilon X + \frac{\tilde{\eta}}{\tau}\sum_{k=1}^{N} n_k^2, \quad V[x] = \sigma^2 \tau \sum_{k=1}^{N} x_k^2, \quad U[x] = E[x] + \lambda V[x]$$

Sooptaroshmi Bandyopadhyay

The Execution-Efficient Frontier

Every risk preference λ gives us a different execution schedule. Plotting its expected cost against its risk traces the efficient frontier.

The Naïve benchmark (λ = 0): With λ = 0, the model spreads sales evenly: n□ = X/N. This keeps expected impact low, but leaves more inventory exposed.

First-order vs. second-order trade-off: Starting from equal-sized trades, we can often cut risk substantially with only a small increase in expected cost.

Trade half-life (θ = 1/κ): The scale θ = 1/κ tells us how quickly the position is sold; it depends on volatility, risk aversion, and temporary impact.

$$\kappa \approx \sqrt{\frac{\lambda\sigma^2}{\eta}}, \quad x_j = \frac{\sinh(\kappa(T - t_j))}{\sinh(\kappa T)} X, \quad \theta = \frac{1}{\kappa}$$

Septoreshmi Bandyopadhyay

In-Class Problem

Saptarashmi Bandyopadhyay

Problem Setup

Suppose our trading agent must sell Q = 8 shares over two periods. It sells x now and the remaining 8 - x next period.

The Two Penalties:

- *Market impact penalty*: Trading x shares now and 8 - x later costs $x^2 + (8 - x)^2$ in market impact.
- *Overnight price risk*: Holding the unsold 8 - x shares overnight adds a risk penalty of $2(8 - x)^2$.

Your Task: What value of x minimizes C(x) = Impact + Risk?

Hints

Sanity check: If risk were zero, you would split evenly: x = 4 in each period (TWAP).

Calculus tip: Combine like terms to get $C(x) = x^2 + 3(8 - x)^2$, take the derivative dC/dx, set to 0, and solve for x.

Santarshmi Bandyopadhyay

Solution

1. **Differentiate**: Start from $C(x) = x^2 + 3(8 - x)^2$ and take dC/dx.
2. **Solve the linear equation**: Set the derivative to zero. This gives $8x - 48 = 0$.
3. **Optimal trade schedule**: So we sell $x^* = 6$ shares now and the final 2 shares in Period 2.

The Core Intuition: Without overnight risk, we would sell 4 and 4. Once we account for risk, the optimal schedule moves 75% of the position into the first period.

$$\frac{dC}{dx} = 2x - 6(8 - x) = 2x - 48 + 6x = 8x - 48 = 0$$

$$8x = 48 \implies x^* = \mathbf{6} \text{ shares (Period 1)}, \quad 8 - x^* = \mathbf{2} \text{ shares (Period 2)}$$

Saptarashmi Bandyopadhyay

Questions?

Saptarashmi Bandyopadhyay