Quantum-Enhanced Coordination in Distributed Systems

QIA 2025 Hackathon Challenge

Overview

In distributed systems, parties often need to coordinate their actions based on local observations. Optimal action often requires that they share their local observations, but this might be impossible or impractical due to e.g., latency associated with communication.

Quantum entanglement provides a coordination resource that can outperform non-communicating classical strategies. By sharing entangled quantum states beforehand and performing local measurements conditioned on their observations, two parties can achieve coordinated decisions that are impossible classically. This challenge explores how to use entanglement in real-world coordination problems.

Example 1: High-Frequency Trading Coordination

This example is based on [1].

Scenario: Two trading servers at different exchanges (e.g., NYSE and NASDAQ) must coordinate trading decisions for correlated stocks X and Y. The servers cannot communicate due to microsecond decision timescales, but speed-of-light delays between exchanges are hundreds of microseconds.

Each server acts as a market maker, providing liquidity by issuing pairs of orders (one bid to buy, one ask to sell) for their respective stocks. Since orders are processed sequentially, the first order issued has higher execution probability due to arriving earlier with more current information. This creates a directional bias: issuing the bid first creates slight long exposure, while issuing the ask first creates slight short exposure.

The challenge lies in coordinating these biases across correlated assets. When stocks X and Y are positively correlated (typical case), optimal hedging requires *opposite* biases: if server A goes slightly long on stock X (bid first), server B should go slightly short on stock Y (ask first). This creates a balanced portfolio that reduces overall risk.

However, market conditions sometimes signal that correlation may turn negative. Technical indicators like diverging price movements, unusual volume patterns, or macroeconomic events can suggest the stocks might move in opposite directions. In such cases, *same* biases are preferable: both servers should take similar directional positions to benefit from the anticipated correlation breakdown.

Mapping to coordination game:

- **Inputs:** Each server observes a binary indicator $x, y \in \{0,1\}$ where 1 means "indicator of negative correlation detected" and 0 means "no indicator"
- **Outputs:** Each server makes a binary decision $a, b \in \{0,1\}$ where 0 = "ask first" and 1 = "bid first"
- **Win condition:** Servers win if $a \oplus b = x \wedge y$ (XOR of outputs equals AND of inputs)
- **Interpretation:** The binary inputs represent each server's local assessment: $x = 1$ means "I detect signals suggesting negative correlation," $x = 0$ means "normal positive correlation expected." The win condition $a \oplus b = x \wedge y$ implements the optimal hedging strategy:
  - When both servers detect negative correlation signals ($x \wedge y = 1$), they should make the *same* decision ($a \oplus b = 1$, meaning $a \neq b$)
  - In all other cases ($x \wedge y = 0$), they should make *opposite* decisions ($a \oplus b = 0$, meaning $a = b$) for normal hedging

Coordination performance:

- Classical strategies: at most 75% correct decisions
- Quantum strategies: ∼85.4% correct decisions

Example 2: Distributed Load Balancing

This example is based on [2].

Scenario: Two servers process both a continuously available baseline task and customer requests that arrive simultaneously in pairs. The system aims to maximize baseline task throughput while meeting quality of service constraints on customer waiting times.

When customer pairs arrive, routers must decide whether to "bunch" them on the same server or "split" them across both servers. The optimal strategy depends on the total service time $X_1 + X_2$: pairs requiring long total service time should be split to avoid excessive waiting times, while pairs with short total service time should be bunched to minimize interruption of the baseline task.

The challenge is that each router only observes its local customer's service time (X_1 or X_2) but the optimal decision requires knowledge of the sum $X_1 + X_2$. The optimal threshold strategy is: bunch if $X_1 + X_2 \leq t^*$, split if $X_1 + X_2 > t^*$, where t^* is chosen to satisfy the QoS constraint.

Mapping to coordination game:

- **Inputs:** Each router observes service time $X_i \sim \text{Exp}(\mu)$ and transforms it to $a, b \in [0, 1)$ using the cumulative distribution function: $a = 1 - e^{-\mu X_1}$, $b = 1 - e^{-\mu X_2}$
- **Outputs:** Each router makes decision $o_A, o_B \in \{+1, -1\}$ where $+1$ corresponds to "send to server 1" and -1 to "send to server 2"
- **Win condition:** Players win if their outputs correctly implement the threshold strategy:

$$o_A \cdot o_B = \begin{cases} +1 & \text{if } (1-a)(1-b) \geq C_t \text{ (bunch)} \\ -1 & \text{if } (1-a)(1-b) < C_t \text{ (split)} \end{cases} \tag{1}$$

  where $C_t = e^{-\mu t^*}$ is determined by the threshold t^*
- **Interpretation:** The condition $(1-a)(1-b) \geq C_t$ corresponds to $X_1 + X_2 \leq t^*$ under the exponential distribution transformation

Coordination resource:

You have access to two coordination primitives. Each takes continuous inputs $a, b \in [0, 1)$ (derived from the observed service times) from the two parties and produces binary outputs $o_A, o_B \in \{+1, -1\}$:

- *Classical:* Outputs correctly implement the threshold strategy with probability ∼0.76
- *Quantum:* Outputs correctly implement the threshold strategy with probability ∼0.81

Your Task

Choose one of these applications and implement both classical and quantum coordination strategies:

1. **Set up the scenario:** For HFT, simulate two trading servers with correlated stock price feeds. For load balancing, create a queueing system with arrival times and service requirements.
2. **Classical baseline:** Implement strategies where each party uses only local information (e.g., "always bid first if local indicator > threshold").
3. **Quantum strategy:** Implement the quantum coordination protocol that provides the performance listed above.

4. **Compare performance:** Track relevant metrics over many trials (profit/loss, throughput, waiting times) and demonstrate the quantum advantage.

5. **Robustness:** Test with noise, measurement errors, or communication delays.

Alternatively, identify your own coordination problem and develop the mapping to a coordination game structure. Look for scenarios where two parties need coordination but can't communicate. Key ingredients: (1) local observations that together determine optimal joint action, (2) cost of suboptimal coordination, (3) clear performance metric.

References

[1] Dawei Ding and Liang Jiang. Coordinating decisions via quantum telepathy. *arXiv preprint arXiv:2407.21723*, 2024.

[2] Francisco Ferreira da Silva and Stephanie Wehner. Entanglement improves coordination in distributed systems. In *Proceedings of the 2nd Workshop on Quantum Networks and Distributed Quantum Computing*, pages 14–20, 2025.