Quantum Galton Board Simulation - Summary

WISER 2025 x NNL Challenge — Participant: Adele Mehrafrouz

Introduction

This project explores how quantum circuits can simulate complex systems through a Galton Box-style Monte Carlo problem. The Quantum Galton Board (QGB) serves as a quantum analog of the classical Galton board, demonstrating how quantum superposition and interference can model statistical distributions. This approach is relevant to high-dimensional problems such as particle transport and quantum systems. The implementation builds on concepts from the Quantum Fourier Transform (QFT) and highlights potential exponential speed-ups over classical simulation methods.

Relation to the Paper's QGB Approach

The reference paper (arXiv:2202.01735) introduces a QGB model using a 'ball' qubit and output registers, where each peg splits the amplitude between left and right paths using Hadamard or Rx(θ) gates, followed by CSWAP operations. This design yields a binomial (Gaussian-like) distribution for $\theta = \pi/2$.

My implementation follows the same principles but includes generalizations and optimizations:

1. **Generalization to arbitrary layers: I generate circuits for any number of layers using a single function.**
2. Biasing control: Rx(θ) is applied uniformly across layers to produce exponential-like distributions.
3. Hadamard quantum walk: I extend beyond the Galton board to include quantum walks.
4. Circuit depth optimization: I developed binary-encoded QGB circuits with fewer qubits and reduced depth.

Despite structural differences, the resulting distributions match the expected behavior (Gaussian for $\theta = \pi/2$, exponential for $\theta \neq \pi/2$).

Methods and Implementation

The implementation consists of the following components:

- circuits/quantum_peg_multi.py: General QGB circuit generator supporting arbitrary layers.
- circuits/biased_distribution_multi.py: Generates biased (exponential-like) distributions using Rx(θ).

- circuits/hadamard_walk_qiskit.py: Implements a Hadamard quantum walk.
- metrics/distance_metrics_multi.py: Provides Total Variation Distance (TVD) and KL divergence metrics.
- circuits/verify_gaussian.py: Validates Gaussian-like output distributions for $\theta = \pi/2$.
- circuits/noise_runs.py: Simulates noisy QGBs with real hardware noise models.
- optimization/: Contains circuit depth and gate optimizations.

I use Qiskit as the primary SDK and matplotlib for visualization.

Results and Analysis

I verified that the QGB output approximates a Gaussian distribution for $\theta = \pi/2$ using TVD and KL divergence against the ideal binomial distribution. For $\theta = 2\pi/3$, the output matches an exponential-like shape. I also implemented a Hadamard quantum walk that produces characteristic quantum interference patterns.

Noise modeling with Qiskit AerSimulator and depolarizing + readout errors shows gradual divergence from the ideal distribution, quantified using TVD and KL metrics. Circuit depth optimizations significantly reduce gate counts, allowing deeper QGBs to be simulated efficiently.

Quantum Walks and Monte Carlo in This Project

Monte Carlo methods rely on random sampling to estimate statistical distributions or solve complex problems. In this project, the Quantum Galton Board (QGB) and Hadamard quantum walks demonstrate quantum-enhanced Monte Carlo principles:

1. Quantum Galton Board (QGB):

- A classical Galton board uses randomness at each peg to build a binomial distribution.
- The QGB replicates this behavior using quantum superposition, where the 'ball' qubit exists in all possible paths simultaneously.
- Measurements of the QGB after multiple shots (e.g., 2048) sample the output distribution, analogous to classical Monte Carlo trials.
- For $\theta = \pi/2$, this produces a Gaussian-like distribution, while for $\theta \neq \pi/2$, a skewed (exponential-like) distribution appears.

2. Hadamard Quantum Walk:

- A quantum walk generalizes the concept of a classical random walk.

- The Hadamard gate acts as a quantum coin flip, while controlled operations shift the walker's position.
- Interference between different paths leads to unique, non-classical probability distributions.
- By executing the quantum walk circuit multiple times, I reconstruct the probability distribution of the walker's position using Monte Carlo sampling.

Together, these approaches illustrate how quantum circuits can leverage quantum superposition and measurement to simulate classical Monte Carlo processes, while potentially offering computational advantages by exploring multiple paths in parallel.

Conclusion

My implementation meets the challenge requirements by:

- Extending 1- and 2-layer Galton box codes to arbitrary layers.
- Producing Gaussian and exponential distributions.
- Implementing a Hadamard quantum walk.
- Analyzing noise effects and optimizing circuit depth.

The approach demonstrates how quantum circuits can simulate classical statistical problems while highlighting opportunities for optimization and scalability.