Random Walks with Quantum Galton Board

State preparation is fundamental in Quantum Computing. In this work, we are using the technique suggested by the Universal Statistical Simulator paper to create variety of distributions in quantum state. The authors design a quantum version of Galton Board and demonstrate symmetric Gaussian as well as some biased distributions. Their approach results in a superposition of Hamming Weight (HW) 1 states. In case of the uniform distribution, they are called W states or a private case of Dicke states with $HW = 1$. Those states are entangled and can be a resource in different algorithms (quantum cryptography, differential equations, optimization problems). Our solution allows any distribution to be encoded with Hamming Weight 1 states.

We will generalize the model to the n-level Galton Board and demonstrate Gaussian on noiseless simulator. We will then show that quantum pegs and a coin can be modified to output different distributions, such as Exponential or Hadamard Random Walk.

Generalizing to n levels

Galton Board can be viewed as a decision tree (though technically it is a Directed Acyclic Graph), where each node (peg) is associated with probabilities to go left or right. One can also say that a node is acting as a random coin deciding what the next step should be.

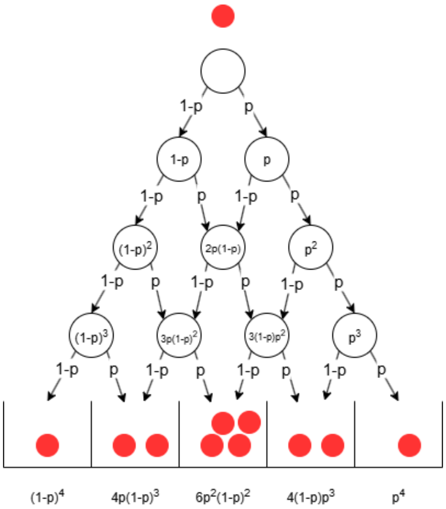

Figure 1

In this figure we are showing a Galton board with probabilities *p* and *(1-p)* at each peg. If *p=1/2*, we will get a symmetric normal distribution in the tally bins. Modifying p will skew the distribution either left or right. In the quantum circuit, we control the probabilities by applying rotations to the "coin" qubit, which is later used as control qubit for the CSWAP gate, which in turn moves the quantum ball left or right with the desired probability. Moving from level to level, we reach the required distribution in the tally bins.

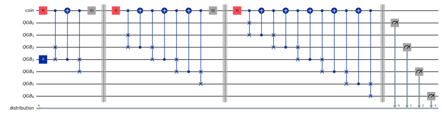

Figure 2

Generalizing the quantum circuit suggested by the scientific paper to n levels is fairly straightforward. Figure 2 shows the 3 levels Quantum Galton Board circuit. We are applying Hadamard to the coin qubit; thus, generating Gaussian.

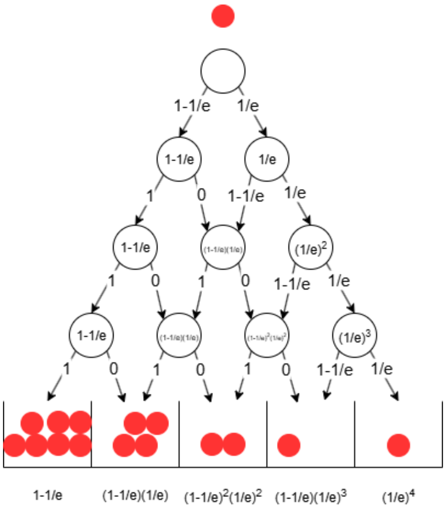

Figure 3

To obtain a different distribution, we should modify the probabilities depending on the level we are at. The figure shows the configuration of probabilities to achieve exponential distribution. As we can see, on each level, the right most quantum peg is assigned with *(1-1/e, 1/e)* probabilities while the rest are *(1, 0)*. This will require to either rotate the "coin" qubit by arccos(1/e) for the right most pegs or simply flip the coin qubit making it |*1*>. In terms of Quantum Circuit, it is just a matter of modifying the `quantum_coin` function to adjust the probabilities. We will use controlled RY rotations to bias the "coin" appropriately.

For the Hadamard case, we have to perform the walk in a superposition and preserve the effect of interference that gives birth to a unique form of the distribution that favors tally bins close to the tail ends and has a valley in the middle. To achieve that we need to:

- alternate the control value for each CSWAP gate (simulates stepping in superposition)
- avoid resetting the coin state (preserves interference)
- perform a final swap after all levels applied to achieve a symmetric distribution.

Optimization

The straightforward implementation of n-level Quantum Galton Board (as described in the scientific paper) requires $2n+3$ qubits. We measure $n+1$ tally bins' qubits to get the distribution. When looking at the Exponential distribution configuration, it can be seen that the pegs with the $(p_{left}=1, p_{right}=0)$ are no longer needed. Their parents nodes can be considered leafs of the decision tree / DAG and the appropriate qubits can be measured instead of performing additional CSWAPs that don't change the probabilities anymore.

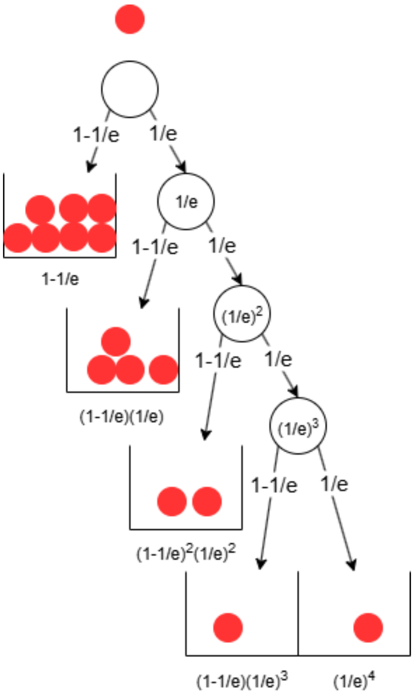

Figure 4

Cutting the extra pegs produces much leaner Quantum Circuit requiring only $n+3$ qubits. As a result, transpiled circuits are of lower depth and perform better under noise. Figure 4 shows the updated Galton Board for exponential distribution.

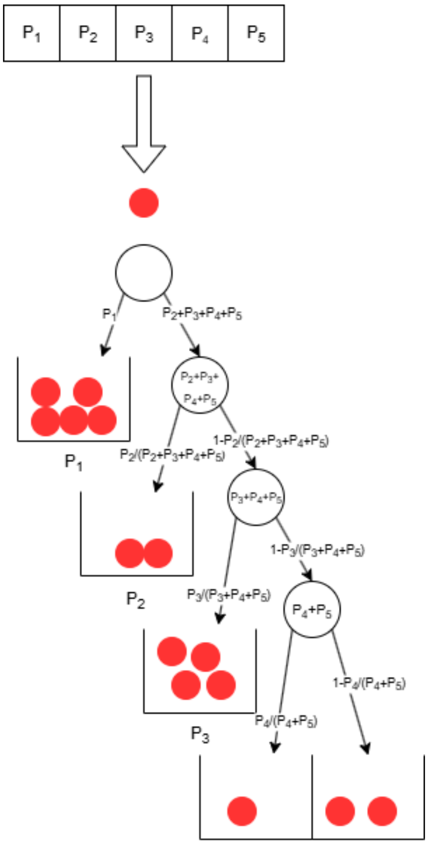

Figure 5

In fact, we can use very similar approach to obtain any distribution. Figure 5 shows the configuration for an arbitrary array of probabilities. This way we can create a universal n-level quantum circuit only using *n+3* qubits.

Additional optimization is achieved through AI passes of Qiskit transpiler. Noise mitigation is done using the `mthree` package.

Distribution Analysis

We have used a combination of Weisersteing Statistical distance and Mean Square Errors to understand the accuracy of generated distribution under noise.