QAMP Spring 2023 – Checkpoint 2 Report

Exploring QUBOs, QAOA, and circuit knitting #11

Mentor: Travis Scholten

Mentees: Hemavathi Santhanam; Marco Antonio Barroca

As stated in Checkpoint 1, the goal of this project is to understand how circuit cutting can help near term quantum computers run the QAOA algorithm despite current constraints in size and connectivity in the hardware. We want to do so by applying this technique to the ansatz circuits created for the algorithm from the Q matrix that specifies a given QUBO problem.

Specifically, we want to understand how the sparsity of the Q matrix can affect the number of unitary factors the ansatz circuit has. Here, unitary factors refer to the non-entangled subcircuits into which the original quantum circuit can be decomposed. These are specially interesting because a circuit with a high number of unitary factors shouldn't benefit from circuit cutting as it "naturally" comes apart. We will present the results as a graph, highlighting trends for matrices of different sizes and densities.

We utilize the following procedure:

1. Generate a set number of random Q matrices for a given density and size. These can follow any random distribution we desire.
2. Build the QAOA ansatz circuit with Qiskit using QAOAAnsatz().
3. Record the count of unitary factors within the ansatz circuit using the .num_connected_components() method for the QuantumCircuit object.
4. Compute a Unitary Size Ratio (# Unitary Factors/Matrix size)

We then repeat this procedure and chart the relationship between matrix density and the number of unitary factors for specific dimension values. These plots are shown on Figure 1.

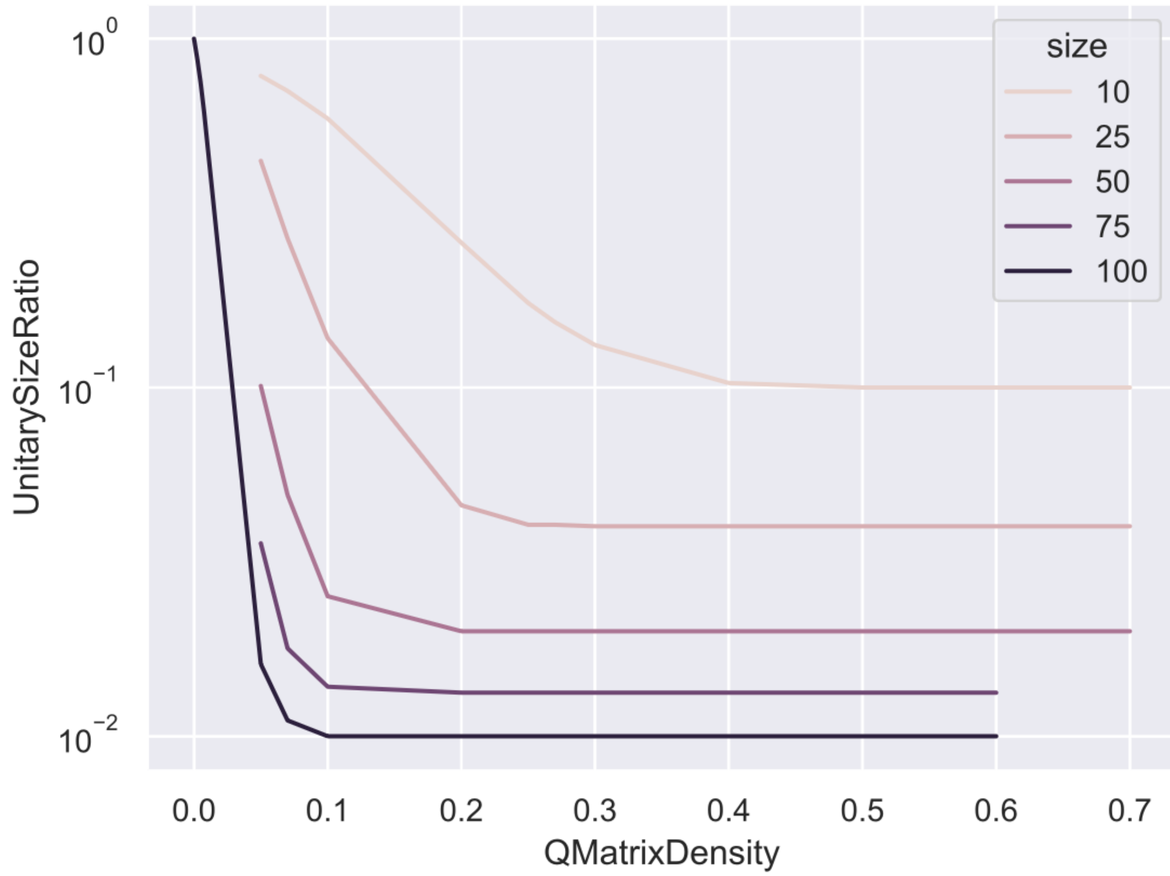

Figure 1: log 10 scaled plot of Unitary Size Ratio Vs. Q Matrix Density. Each line represents a different size matrix and all the ratio converges to one as the matrix density decreases. Note that the behavior at which the ratio increases changes as the matrix size decreases.

We see these have some of sort "critical value", related to the matrix size, after which the number of unitary factors increases. Using different distributions for matrix generation seem to have no effect on the number of unitary factors, most likely because the odds of the getting a zero element randomly are very low. All lines converge to 1 as the matrix density decreases as a circuit can't have more unitary factors than the matrix size.

For checkpoint 2 we also wanted to look at the circuit width of the subcircuits that make the unitary factors and how it relates to the Q matrix density, unfortunately due to time constraints this will have to wait until the final presentation. We already have plans to implement this, having looked at how the num_connected_components() method works in the Qiskit source code it should be possible to build on top of it to get the width of the subcircuits as well. This might also be a useful feature to include into Qiskit in the future.

Beyond that, we also need to investigate how the unitary factors relate to circuit cutting. As stated, a high number of unitary factors reduces the usefulness of circuit cutting but it's unclear where that "usefulness threshold" would lie and how it relates to density and size of the matrix.