Graph Coloring

Hugo WEISS
Basile BUFFIN
Dimitri BERHAULT

Contents

1 Graph Coloring **3**
1.1 Classical Implementation 3
1.2 Pulser Implementation 3
1.3 Conclusion 4

1 Graph Coloring

1.1 Classical Implementation

To model the problem of antenna frequency on unit-disk graphs, we check each coordinate to see if there is another coordinate within the specified distance. If there is, we add the relation to an adjacency list. Then, we construct the graph using the NetworkX library. (step1.py)

In the graph coloring implementation, we recursively use the MIS. To find the MIS, we compute all cost configurations and select the one with the minimal cost. The graph is modified after each MIS iteration. (step2.py)

1.2 Pulser Implementation

For each antenna's coordinate, we place a qubit at the exact same position in a register. Then, we utilize the adiabatic theorem to 'slowly' transition from the ground state of a known Hamiltonian to the ground state of the Hamiltonian that encodes the MIS problem. We employ an adiabatic pulse with a duration of 10^{-2} ms and a non-zero amplitude to attain the ground state of the Hamiltonian representing our MIS problem. (step3.py)

In the section involving realistic hardware and error sources, we create a list of trap coordinates, some of which correspond to the antenna's coordinates. This list is used as input for our layout Register. We then add our adiabatic pulse as previously explained. Before simulating, we specify our intention to introduce SPAM and Doppler shift noise, and we set the characteristics of these noises. (step5.py)

When comparing the results with the previous implementation, we observe a drastic decrease in the number of samples displaying the correct answer. Additionally, states that were not present in the previous implementation now emerge. This behavior aligns with our expectations. (Note that due to the noise, the MIS is occasionally not returned) However, there is a limitation of 7 qubits for our noise configuration. When exceeding this limit, the Qutip emulator outputs an error message stating: "Excess work done on this call".

1.3 Conclusion

In summary, this project effectively models the antenna frequency problem using both classical computing methods and a quantum computing emulator. The classical approach laid a foundational understanding through graph coloring and MIS algorithms. In the quantum emulation phase, the adiabatic theorem and an adiabatic pulse were employed to simulate quantum solutions for complex problems. While we faced limitations in qubit scalability and noise simulation, the project provides valuable insights into the capabilities and current boundaries of quantum computing emulation.