NEUTRAL ATOM COMPUTING

Graph Coloring

Authors
Guichard Benjamin
Bouyjou Maximilien

Contents

1 Graph Coloring

1.1 Goal

Given a list of antennas with their respective position and an interference radius. We would like to give each antenna a specific frequency so that the final solution doesn't present any interference. We are going to picture the problem with a colored graph problem where each color is one frequency.

1.2 Introduction

In This Report we are going to discuss how we can implement the classical and Pulser solutions to a given graph coloring problem. After that we are going to analyze the results and compare the solutions objectively.

1.3 Classical Resolution

Init: To initialize our project, we decided to build a simple Graph management class which has methods to make the worflow simplier and easier to read. These methods include :

- An algorithm which creates the edges if two nodes distance is under the threshold.
- Generating and updating the graph.
- Add new edges to the graph.
- Add new colors to the graph by reading a bitstring. Returning a new graph with the uncolored nodes only.
- A method to visualize the graph.

After creating the class, we can now instantiate our graph with the data of the problem and the given threshold. After doing so, this is the graph we get (Figure 1)

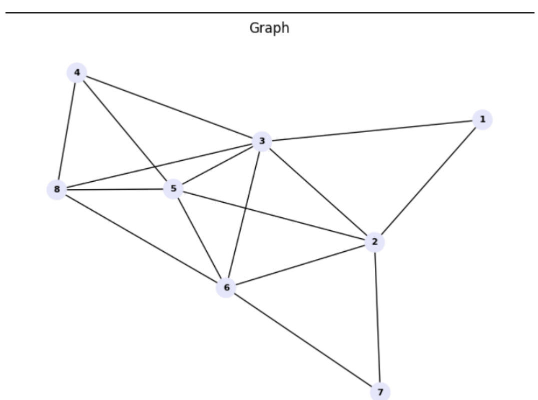

Figure 1: uncolored graph of the problem

Resolution with MIS:

As we have seen in the lectures, we are going to color the graph by applying the MIS algorithm which will return the Bitstring corresponding to the indexes of the nodes we have to color and iterate on the smaller and uncolored graphs till all nodes are correctly colored. We get the following graph: (Figure 2)

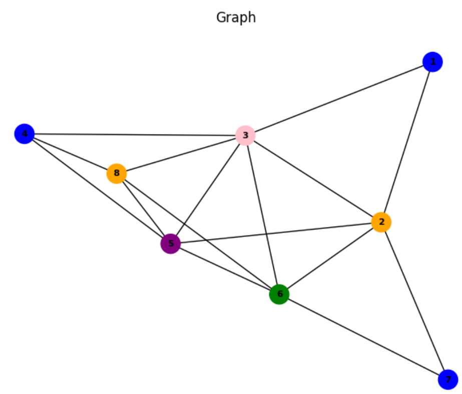

Figure 2: colored graph using classical resolution

As we can see, the graph is now correctly colored and we can define the number of iteration by counting the number of different colors (in our case, 5)

1.4 Quantum Resolution

Init: This time we are going to use the Pulser solution to our problem. We start by mapping the qubit coordinates to the exact position of the antennas, and defining the maximum Rabi frequence according to the antennas interference value. So far, we only have defined our grid of atoms mapped to our antenna coordinates. Now, we need to declare a sequence which will be holding the pulses that will be sent to the atoms.

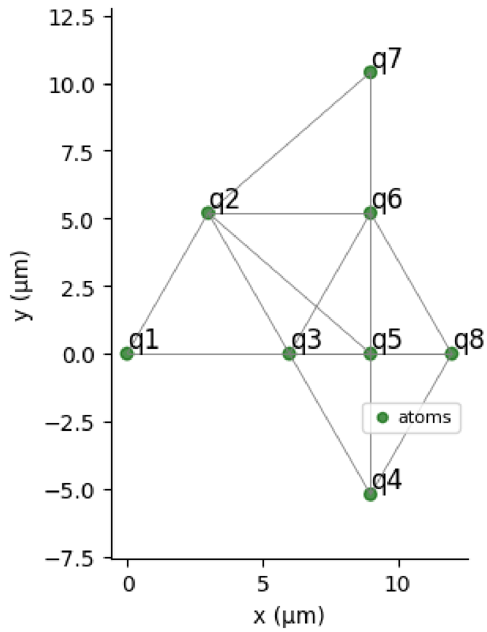

Figure 3: qubits at their respective position

To define these pulses, we refer to the adiabatic theorem which says that if our system starts in a ground state of an initial hamiltonian and the environment conditions are gradually changing, then the system will end up in the ground state of another corresponding hamiltonian. Thus, we specifically choosed the duration of the pulses, while trying to make them as short as possible. After that, we run the simulation and recover the most probable state which is the MIS of our graph.

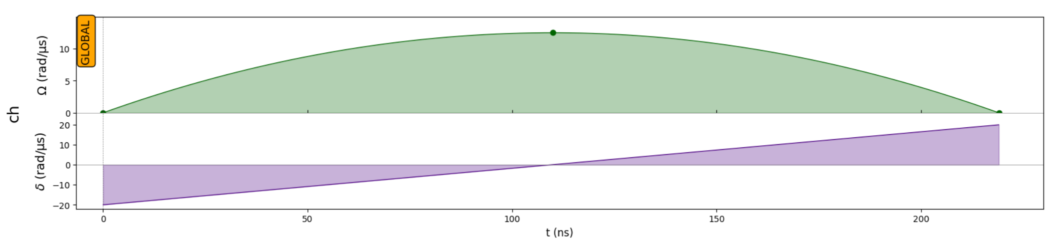

Figure 4: sequence of the pulse

Graph coloring Resolution

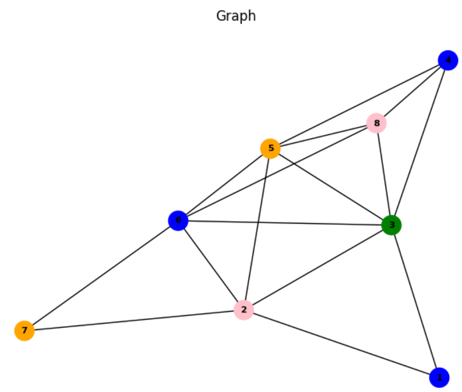

Figure 5: Pulser's new colored graph

To color our graph, as in classical, we need to loop by applying our neutral atom MIS solver in each iteration and remove the atoms corresponding to the solutions of each step. As we can see through the results, the neutral atom graph coloring algorithm is giving us an improvement as we obtain a total number of 4 colours, comparing to 5 in the classical resolution.

1.5 Conclusion

As discussed earlier, the Pulser solution demonstrated marginally more pertinent results, suggesting the potential for achieving a quantum advantage in addressing this particular problem. Despite the fact that the computations were conducted on an emulator, the findings underscore a genuine interest in leveraging quantum solutions for such applications.