Quantum Computing HW 5
Carl Klier and Nachiket Patel
10/22/2020

Completed Tasks	**Owner**
Reach out to professor with clarification questions	Carl
Dwave 13 provinces of Canada Coloring • See code on github	Carl
Qiskit 2-coloring with circuits • See code on github ◦ (code/2-coloring.ipynb)	Nachiket
Set up IBM Q Hub account	Nachiket
Research Dwave Annealing Process • See included notes	Carl
Find new sources	Carl, Nachiket
Read papers	Carl, Nachiket
Narrow down our problem/focus	Carl, Nachiket

Tasks Still Left	**Owner**	**Expected Due Date**
Read more about QAOA	Nachiket	10/27
Implement 3-coloring for one specific graph	Nachiket	10/29
Finalize set up for IBM Q Hub Account	Carl	10/26
Implement 3-coloring on Dwave	Carl	10/29
Understand how annealing code might be generalized to k-coloring and finding chromatic number	Carl	10/29
Implement generalized 3-coloring on Qiskit	Nachiket	11/5
Learn about ways to improve	Carl	11/5

performance of annealing to better solve the graph coloring problem		
Analysis of results - compare and contrast Dwave and Qiskit	Carl, Nachiket	11/5
First Draft Due for Review	Carl, Nachiket	11/5
Make necessary edits	Carl, Nachiket	11/12
Finalize Project	Carl, Nachiket	11/16
Turn in Final Project	Carl, Nachiket	11/17

General

We had a few questions for Dr. Zhou which helped us to clarify the problem and provide some direction on how to get started. We were wondering if we needed to write code to be able to handle arbitrary graph coloring problems and Dr. Zhou stated that we only needed to solve a specific graph coloring problem of our choosing. We could start with a small, simple problem (which is what we did) and could work our way up towards bigger, more complicated problems. We were also wondering if we needed to code an explicit embedding or implicit embedding when working on the Dwave system and Dr. Zhou said it would be good to try both ways but really only needed code for one.

One of our tasks this week was to narrow down the focus for our problem to something more concrete and manageable. While we might make changes later depending how the work goes, we are deciding to write code to implement a 3-coloring on an undirected graph.

Dwave Annealing

In order to better understand how Dwave and the annealing process worked, I started off by reading the Dwave System documentation. You can read my notes on this reading in the pushed pdf. There were a couple main things that I learned. First, I better understood the simple transformation from a QUBO to the Ising model. Secondly, I learned about the quantum annealing controls that are available that could help improve performance later on in this project. I believe better understanding these controls will be beneficial when comparing and contrasting annealing to a gate based system. Lastly, I became better aware of how problems can be reformulated to be better suited to quantum computers. For example, it is common to reduce a 3-SAT problem to a MAX-2-SAT problem in order to decrease the number of ancilla bits required.

I was able to understand how to code a 4-coloring graph for the 13 provinces of Canada by following the tutorial that Dwave has on their site. It essentially boils down to coding the two constraints: that a node can only be colored one color and that a node needs to have a different color than all its neighbors. With these constraints, we can create a constraint satisfaction problem and convert that to a BQM which the DWaveSampler is able to solve. From this, it should be simple to change this code to solve our desired problem of a 3-coloring for an arbitrary graph.

Going forward, I would like to better understand how this code can be scaled to handle any k-coloring and to solve the chromatic number problem. I also want to explore the limitations of an implementation like this and determine if any of the quantum annealing controls can benefit us. This will involve doing some more research and reading about the problems, playing around with some smaller graphs to try and implement working code, and then working to implement annealing controls to determine if they have much of an effect.

Qiskit

To solve the graph coloring problem, the 2-color graph coloring problem was attempted on specific predefined graphs. The simple problem of two vertices with one edge between them was solved on qiskit using ccx gates to represent the edge between the vertex. The input quibts were measured to create a superposition state which can be viewed as the coloring for the graph. For example if the state measured is |10> then, according to IBM ordering the first vertex is color0 and the second vertex is color1. Due to the symmetry of the 2-color graph coloring problem the algorithm for any arbitrary graph can be done systematically by replicating the 2 vertex and 1 edge circuit to the appropriate edges in any graph. The code was expanded to a larger number of nodes and vertices to create a generalized algorithm for the 2-color graph coloring. Even on a graph which cannot be colored, the algorithm returns an output, which will have to be verified using a classical algorithm.

With this initial step solved the next step will be to research more about the QAOA by studying the Qiskit QAOA example for the max cut problem. Using the example along with the algorithm and figuring out how to implement QAOA for the graph coloring problem. More specifically the next problem to approach is the 3-color graph coloring problem and creating a qiskit circuit to find a solution to a specific graph. From this position an attempt to generalize the 3-color graph coloring problem can be approached. To implement k-graph coloring with QAOA the cost function and Hamiltonian cost function will have to be computed for any graph for which we intend to perform coloring on.

$$f_c = m - \sum_{c=1}^{k} \sum_{\{u,v\} \in E} X_{u,c} X_{v,c} \qquad H_c = \sum_{c=1}^{k} \sum_{\{u,v\} \in E} \sigma^z_{u,c} \sigma^z_{v,c}$$

Another major problem to solve for a large number of colors is to encode the solution in the superposition states which can then be used to color the graph. For 2-coloring we can use 0 to represent one color and 1 as another color, but for larger numbers a different technique has to be used.

Papers/Sources Read

https://arxiv.org/pdf/1911.00595.pdf
https://www.dwavesys.com/sites/default/files/Map%20Coloring%20WP2.pdf
https://docs.ocean.dwavesys.com/en/latest/examples/map_coloring.html
https://docs.dwavesys.com/docs/
https://medium.com/qiskit/the-variational-quantum-eigensolver-43f7718c2747

https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/optimization/3_minimum_eigen_optimizer.ipynb

New Sources

https://docs.dwavesys.com/docs/

https://core.ac.uk/download/pdf/227725576.pdf

https://qiskit.org/textbook/ch-applications/qaoa.html#5.3-Quantum-circuit

https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/optimization/3_minimum_eigen_optimizer.ipynb - tutorial on converting a quadratic to ising hamiltonian and running on qaoa

https://medium.com/qiskit/the-variational-quantum-eigensolver-43f7718c2747 - Graph coloring for VQE

https://scholar.afit.edu/cgi/viewcontent.cgi?article=4264&context=etd - Solving Combinatorial problems using QAOA

https://lucaman99.github.io/new_blog/2020/mar16.html

https://qiskit.org/textbook/ch-applications/qaoa.html

https://core.ac.uk/download/pdf/227725576.pdf