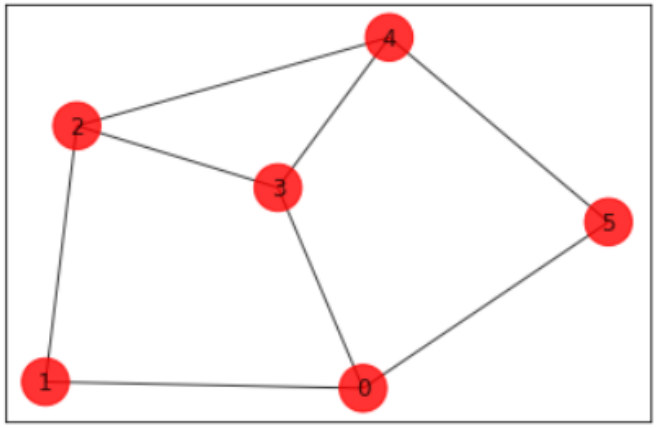
CSCI 4140 Laboratory Ten

Quantum Optimization

Introduction

In this laboratory you will follow the same steps as in the lecture notes to optimize a problem using the VQE quantum optimization solver. This will again be a graph based optimization problem, but with a different graph. The graph that we will use for this lab is:



Note that there are 6 vertices in this graph, instead of the 5 vertices in the graph used in the lecture. Also change the constraint to have 3 vertices in one of the subgraphs. The other difference is that we want to minimize instead of maximize.

You can start off with some import statements:

```
# Some standard code imports
import matplotlib.pyplot as plt
import matplotlib.axes as axes
import numpy as np

# For drawing graphs
import networkx as nx

# Qiskit imports
from qiskit import Aer, execute, QuantumCircuit
from qiskit.quantum_info import Statevector
from qiskit.optimization import QuadraticProgram
from qiskit.aqua.algorithms import NumPyMinimumEigensolver
from qiskit.optimization.algorithms import MinimumEigenOptimizer
from qiskit.optimization.converters import LinearEqualityToPenalty
from qiskit.optimization.converters import QuadraticProgramToQubo
from qiskit.aqua.algorithms import VQE
from qiskit.circuit.library import RealAmplitudes

# auxilliary function to plot graphs
def plot_result(G, x):
    colors = ['r' if x[i] == 0 else 'b' for i in range(n)]
    pos, default_axes = nx.spring_layout(G), plt.axes(frameon=True)
    nx.draw_networkx(G, node_color=colors, node_size=600, alpha=.8, pos=pos)
```

Procedure

The first thing that you need to do is create the graph data structure. The process for doing this is shown on slide 11 of the quantum optimization lecture.

Once you have the graph constructed you need to build the DOcplex description of the optimization. Slide 15 shows the code for doing this. Remember, we are doing a minimization and not a maximization.

Next convert the DOcplex description into a quadratic problem as shown on slide 17. At each step you should be checking your results. They will be different from the results in the lecture slides, due to the different problem, but they should look similar.

Now that we have a quadratic problem, we can use a classical solver to determine the correct answer to our optimization problem. The process for doing this is shown on slide 20.

With the answer in hand the next step is to remove the constraint from the quadratic problem to produce a QUBO. Slide 23 shows how this can be done.

Now we can convert our optimization problem into an Ising model. The steps for doing this are shown on slide 32. Remember, that the offset value is important.

Slide 34 shows how we can quickly verify the Ising model. You need to make one important change to this code. On the call to plt.xticks() you need to change the format from {0:05b} to {0:06b} since we now have 6 vertices in the graph.

Finally create the ansatz as shown on slide 37 and call VQE as shown on slide 38. Note, when constructing the ansatz you will need to change the first parameter to RealAmplitudes() from 5 to 7. This will give you the optimal value. Next plot the results as shown on slide 39. Remember, to change the format in the call to plt.xticks(). You are now done.

Laboratory Report

The laboratory report should only include the results of the last step of the above procedure. Format this as a PDF or PNG file and submit it through Canvas.