QAOA performance on MaxCut for various graphs using Rigetti's QVM

Joost Bus

j.c.p.bus@student.tudelft.nl

Technische Universiteit Delft, Nederland

April 29, 2020

Overview

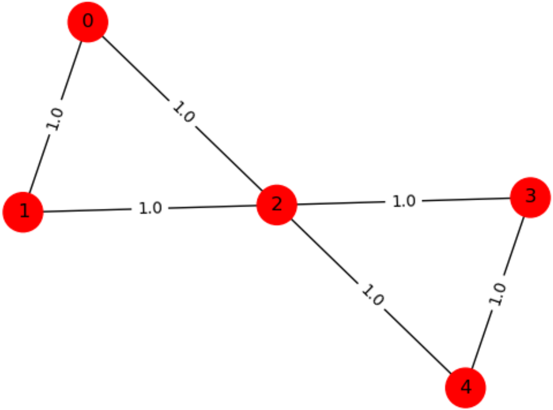
1. Graphs
   - Butterfly
   - Diamond
   - Cycle graphs
   - 3-regular graphs
2. Approximation ratio on 3-regular graphs
3. Adiabatic Theorem
4. Further investigation

TUDelft

Disclaimers

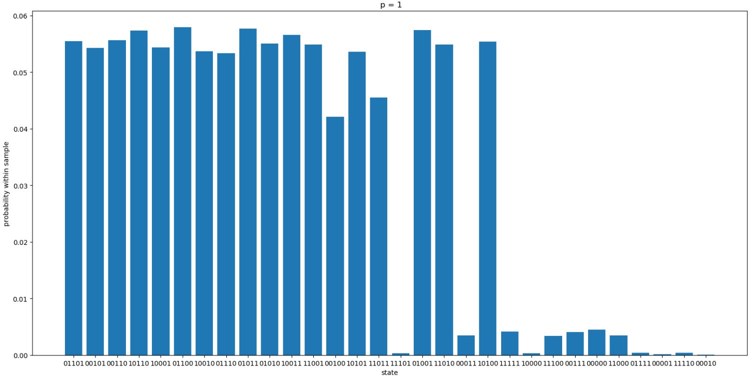
- I ran these experiments using the pyQuil QVM; these are all simulated
- These simulations were run **without noise simulation**
- The algorithm works on arbitrary graphs and for various integers p. Therefore it disregards possible restrictions on the topology of the QPU
- The angle optimisation is done using VQE, which is also simulated without noise
- For each graph I used 10000 samples. However, the angles are also chosen stochastically (using VQE) therefore the algorithm does not produce the same result with the same inputs.
- The presentation of results is not optimal, in the future I would like to present all possible bitstring (to show contrast / interference) and moreover group the bitstring together with the same cost.

Butterfly

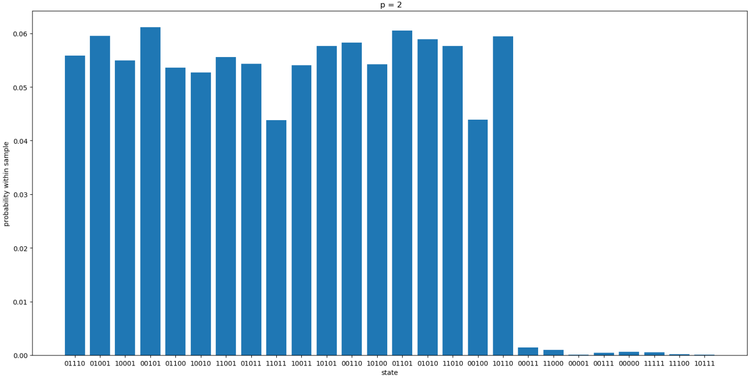


Observe that every $2-3$ cut is optimal, as is the cut $00100 \hat{=} 11011$

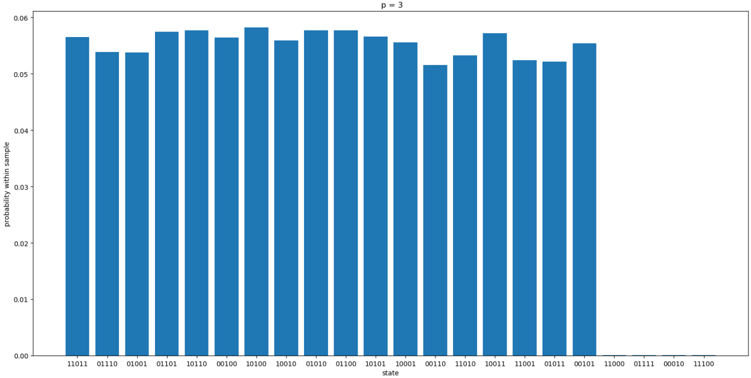
Butterfly, $p = 1$



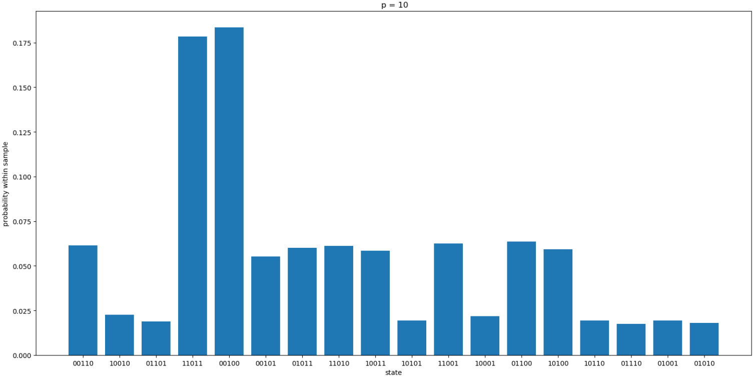
Butterfly, $p = 2$



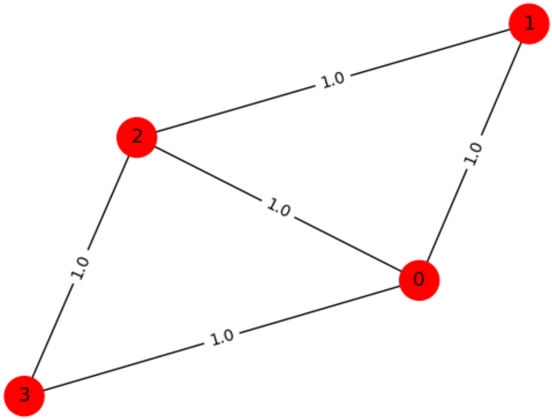
Butterfly, $p = 3$



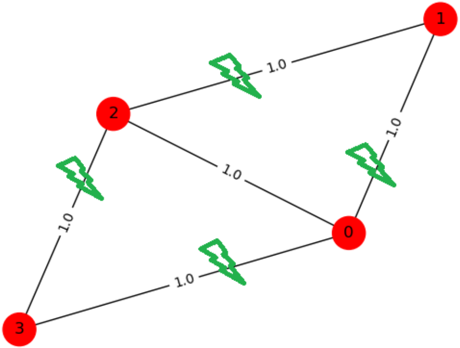
Butterfly, pushing it to p = 10



Diamond

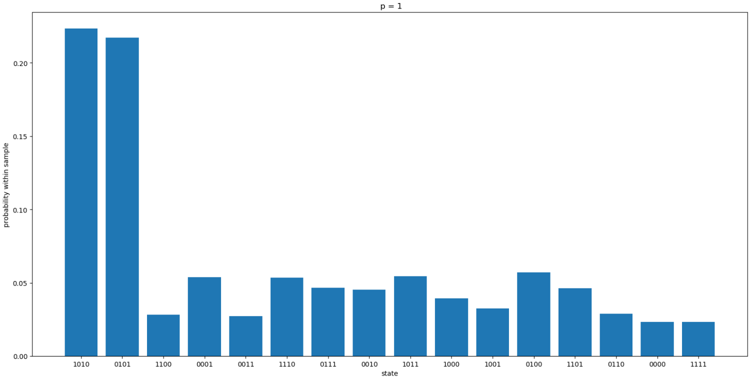


Diamond



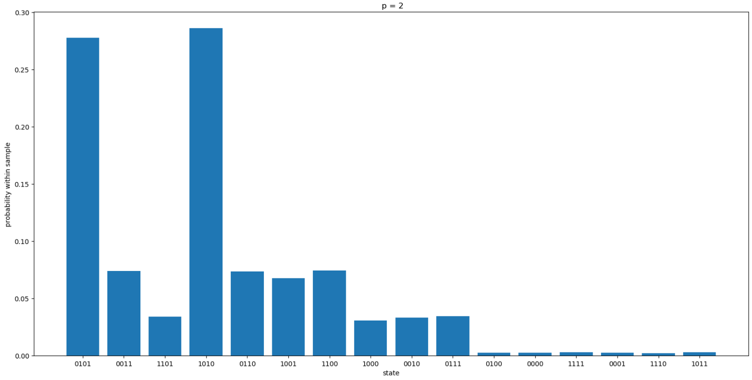
Observe that there are two optimal cuts: 1010 and 0101

Diamond, $p = 1$



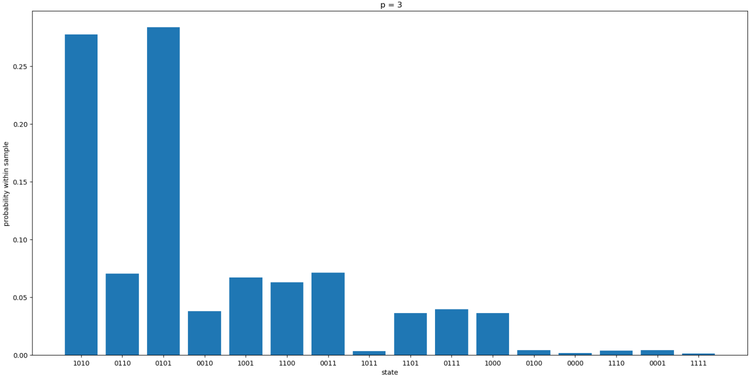
p = 1
probability within sample
0.00
0.05
0.10
0.15
0.20
1010
0101
1100
0001
0011
1110
0111
0010
1011
1000
1001
0100
1101
0110
0000
1111
state

Diamond, $p = 2$



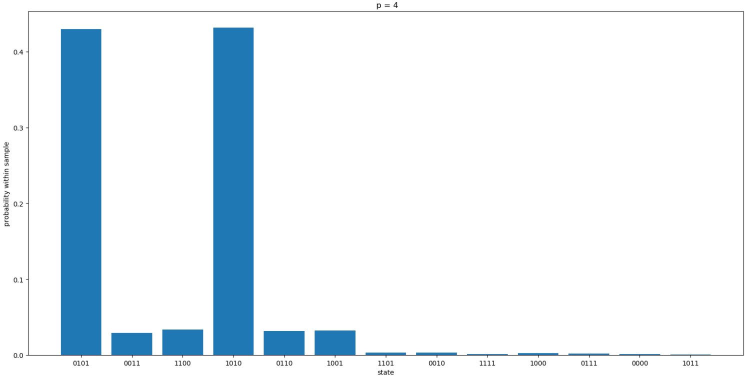
p = 2
probability within sample
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0101
0011
1101
1010
0110
1001
1100
1000
0010
0111
0100
0000
1111
0001
1110
1011
state

Diamond, $p = 3$



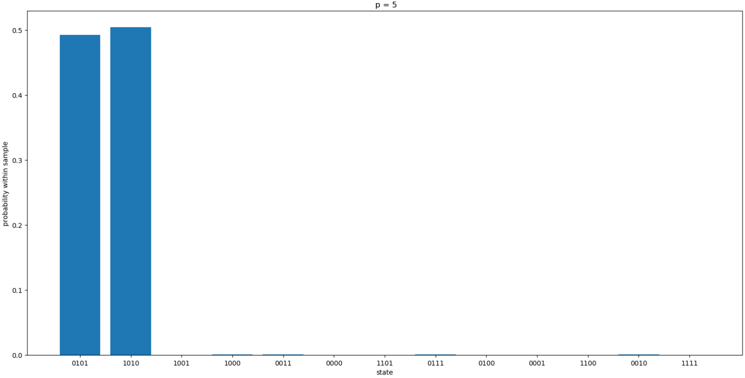
p = 3
probability within sample
0.00
0.05
0.10
0.15
0.20
0.25
1010
0110
0101
0010
1001
1100
0011
1011
1101
0111
1000
0100
0000
1110
0001
1111
state

Diamond, $p = 4$

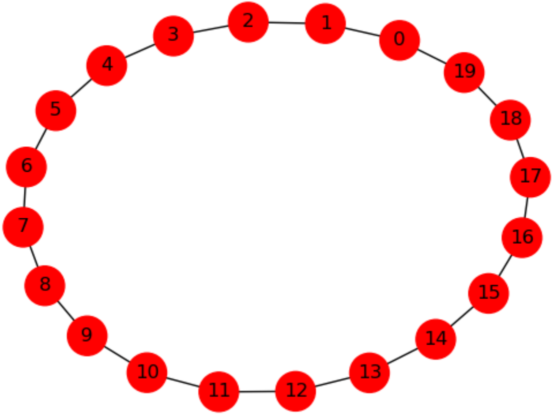


p = 4
probability within sample
0.0
0.1
0.2
0.3
0.4
0101
0011
1100
1010
0110
1001
1101
0010
1111
1000
0111
0000
1011
state

Diamond, $p = 5$


p = 5
probability within sample
0.0
0.1
0.2
0.3
0.4
0.5
0101
1010
1001
1000
0011
0000
1101
0111
0100
0001
1100
0010
1111
state

Cyclic graphs (connected 2-regular graphs)



The optimal cut for these graphs is $\lfloor \frac{N}{2} \rfloor$, i.e. just alternating 1 and 0 (for odd degree there has to be one neighbour pair in the same set)

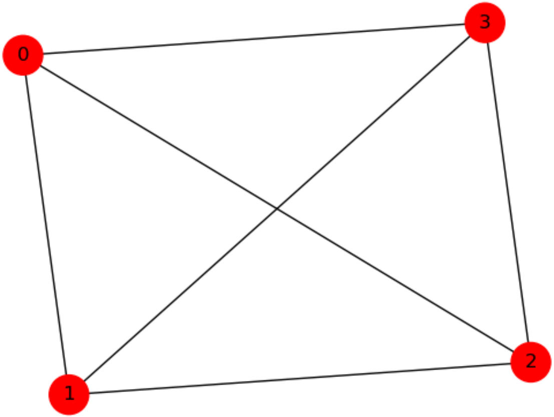
Cycle-20, $p = 1$

On a graph with 20 nodes it takes quite a while before the VQE (simulator) determined the optimal angles... (~3 min)

The histogram is not very insightful as it is very wide. However, we did find the right solution, namely the alternating one

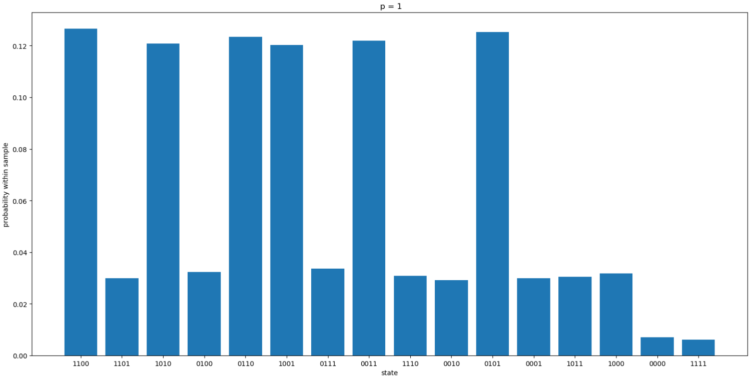
```
Values of betas: [0.39168088]
Values of gammas: [0.78795143]


And the most common measurement is...
(1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19)
```

3-regular graphs - 4 nodes

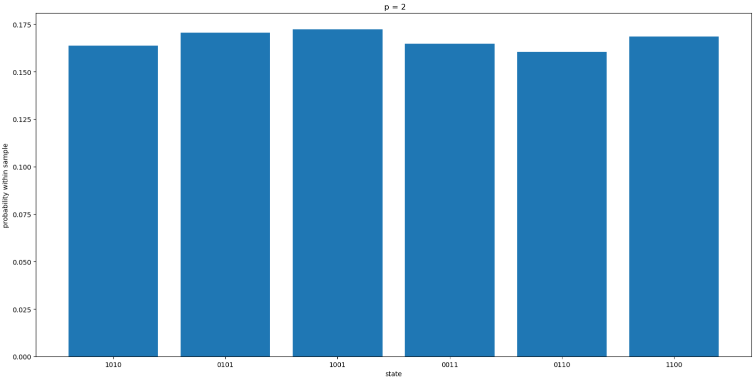


Observe that every $2-2$ cut is optimal, therefore there are $\binom{4}{2} = 6$ optimal cuts with value 4

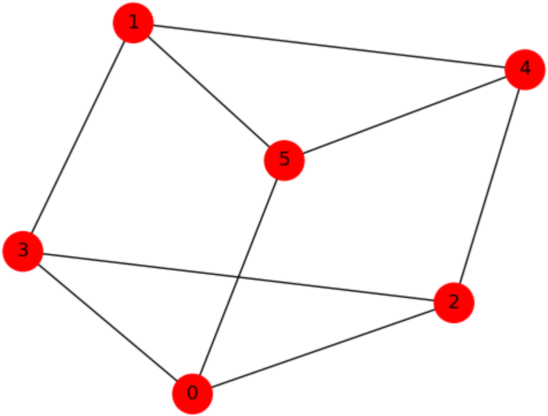
(3,4)-regular, $p = 1$



(3,4)-regular, $p = 2$

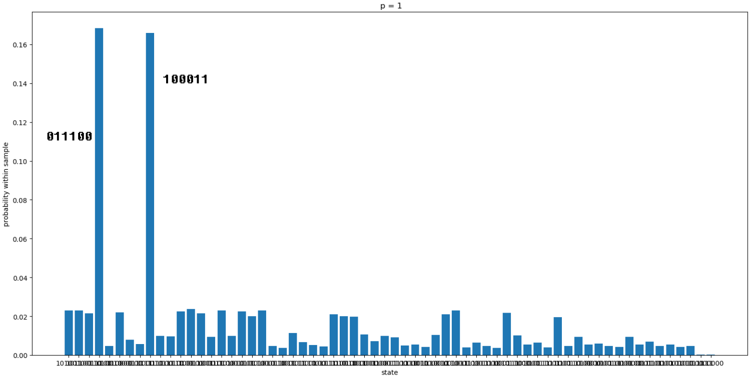


3-regular graphs - 6 nodes

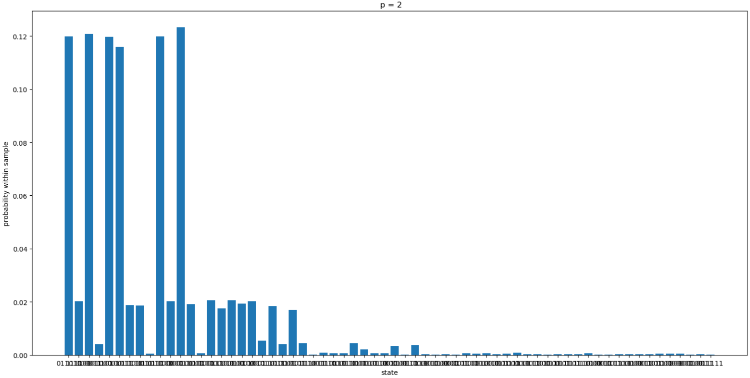


Optimal cuts are the ones with one neighbour pair and their opposite, for example $\{2, 3, 5\}$. You can form 6 of those. **labelling wrong in figure?**

(3,6)-regular, $p = 1$



(3,6)-regular, $p = 2$



Approximation ratio on 3-regular graphs

In the original QAOA paper from 2014 it was proven that for $p = 1$, QAOA has a (worst case) approximation ratio $\rho = 0.6924$ meaning

$$\max_{\mathbf{z}} C(\mathbf{z}) \geq C(\mathbf{z}^*) \geq \rho \max_{\mathbf{z}} C(\mathbf{z}) \tag{1}$$

meaning that the optimal circuit produced a distribution of states with a Hamiltonian expectation value of 0.6924 of the true maximum cut for 3-regular graphs.

Approximation ratio on 3-regular graphs

For the MaxCut problem there exists an approximate algorithm 1995 by Goemans and Williamson. This algorithm has an approximation ratio of $\rho \approx 0.87856$. This approximation ratio is believed to optimal so it is not expected to see an improvement by using a quantum algorithm. (I don't know exactly why?)

Larger p and the Adiabatic theorem

Using the adiabatic theorem it can be proven that

$$\lim_{p\to\infty} M_p = \max_z C(z) \tag{2}$$

where

$$M_p = \max_{\vec{\gamma},\vec{\beta}} F_p(\vec{\gamma},\vec{\beta}) \tag{3}$$

However, the scaling of the algorithm with p is (possibly?) not efficient. This depends on your method of determining these angles and I am not sure yet how VQE scales.

Further investigation

- larger graphs
- regular graphs analytically
- scaling of p
- other optimization methods such as pyswarm
- compare with Goemans-Williamson
- compare with Qiskit
- run on actual quantum chips
- use qaoa on other problems $\implies$ other cost Hamiltonian
- use other mixers