CMSC657 - Coding Assignment 2

A Quantum Approximate Optimization Algorithm applied on the MaxCut Problem

Sandesh Kalantre

December 2, 2018

1 Introduction

Quantum Approxmiate Optimization Algorithm (QAOA) was introduced by Farhi, Goldstone and Gutmann [1] to find approximate solutions for combinatorial optimization problems. I will use the naive implementation of this algorithm on the MaxCut problem as described in the paper. Combinatorial optimization problems are specified on n bits with m clauses. Let's denote the n bit string as z and the clauses as $C_\alpha(z)$ where $C_\alpha(z) = 1$ if the clause is satisfied or 0 else. The objective function $C(z)$ is defined as:

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z)$$

We want to find the maximum value of $C(z)$ over all possible n-bit strings z. A typical brute force algorithm classicaly will need to check all possible 2^n strings. Quantumly, we hope to solve this problem with n qubits.

We regard the clauses C_α as diagonal operators in the computational basis. Define operators and the initial state $|s\rangle$

$$C = \sum_{\alpha=1}^{m} C_\alpha$$
$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$
$$B = \sum_{j=1}^{n} \sigma_j^x$$
$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x}$$
$$|s\rangle = H^{\otimes n}|0\rangle$$

where H is the Hadamard gate. Since the eigenvalues of C_α and B are integers, we can restrict γ and β in $[0, 2\pi]$.

For a integer $p \geq 1$ and $2p$ angles $\gamma_1 \dots \gamma_p = \boldsymbol{\gamma}$ and $\beta_1 \dots \beta_p = \boldsymbol{\beta}$, we can define the QAOA ansatz state,

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B,\beta_p)U(C,\gamma_p)\dots U(B,\beta_1)U(C,\gamma_1)|s\rangle$$

This state can be produced by a circuit of depth at most $(m+1)p$ which is independent of n.

Let $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta}|C|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ and $M_p = \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$. One can see that M_p approximates the maximum value of $C(z)$ and the theoretical result [1] is that $\lim_{p\to\infty} M_p = \max_z C(z)$.

2 MaxCut Problem

I will apply the QAOA ansatz on a specfic problem, i.e. a particular choice of C, namely for the MaxCut problem [5]. The input is a graph with n vertices and an edge set (j,k) of size m. Here, $C_{i,j} = \frac{1}{2}(-\sigma_j^z\sigma_k^z + 1)$.

For simplicity, I will use $p = 1$ instance of the QAOA ansatz. The algorithm that is used is as follows:

Result: Maximum value of C(z)
for $(\gamma, \beta) \in [0, 2\pi] \times [0, 2\pi]$ *(discrete grid)* **do**
  for *num_shots iterations* **do**
    Prepare $|\gamma, \beta\rangle$;
    Measure $|\gamma, \beta\rangle$ and evaluate $C(z)$;
  end
  Estimate $F_p(\gamma, \beta) = \langle\gamma, \beta|C|\gamma, \beta\rangle$ from the num_shots measurements ;
end
Return maximum value of $F_p(\gamma, \beta)$ as maximum value of $C(z)$;

It is possible to use numerical optimization strategies for γ, β instead of a grid search as above, but I have not used them here.

3 Implementation using Cirq and QISKIT

The operators $U(B, \beta)$ and $U(C, \gamma)$ for the MaxCut problem can be implemeted using CNOT gates and X and Z rotations. See the codes for exact details. [2]. The optimization landscape as a function of γ, β for Cirq [4] and Qiskit [3] implementations is shown in Figure 1. A ring graph with 4 vertices was used as an example.

The results of the algorithm were as follows:

	Cirq	Qiskit
Simulation Time	700 s	3000 s
Approximate Maximum	3.26	3.28

The exact maximum value of $C(z)$ for the ring graph on 4 vertices is 4. A finer grid search or a numerical optimization over γ, β can be used to improve the results.

4 Conclusions

- Cirq is almost 5 times faster than Qiskit for discrete scans over parameters.
- Both implementations do not converge to the exact maximum without numerical optimization or very fine parameter grids. This can be remedied by increasing the number of parameters in QAOA ($p > 1$) which has not been studied in this report.

References

[1] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A quantum approximate optimization algorithm”. In: *arXiv preprint arXiv:1411.4028* (2014).

[2] *GitHub Repository*. `https://github.com/sandeshkalantre/cmsc657-quantum-computation`. (Accessed on 12/02/2018).

[3] *Qiskit*. `https://github.com/QISKit`. (Accessed on 12/02/2018).

[4] *quantumlib/Cirq: A python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits.* `https://github.com/quantumlib/Cirq`. (Accessed on 12/02/2018).

[5] Sivaramakrishnan Scribe and Sahar Karimi. “Max-cut Problem”. In: (2007). URL: `http://www4.ncsu.edu/~kksivara/ma796s/projects/sahar_report.pdf`.

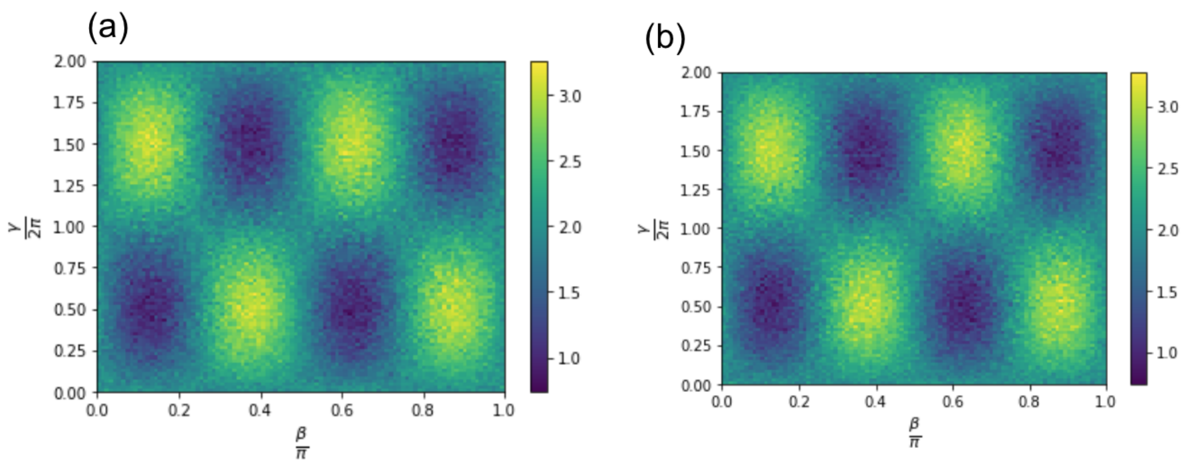

Figure 1: Comparison of the optimization landscape simulated using Cirq (a) and Qiskit (b). The estimated $F_1(\gamma, \beta)$ are plotted in the color scale as a function of γ and β.