Applying a Quantum Approximate Optimization Algorithm and Grover's Algorithm to an Unweighted Max-Cut Problem

Aneesh Sreedhara

December 6, 2023

North Carolina State University

NC STATE UNIVERSITY

Table of contents

Problem

Unweighted Max-Cut

Tasked with partitioning V from graph $G = (V, E)$ into V_1, V_2 to maximize number of edges $(i, j) \in E$ where set$(i) \neq$ set(j) [2].

Define partitions with set $Y = \{y_i\}$ as follows.

$$y_i = \begin{cases} -1 & \text{if } i \in V_1 \\ 1 & \text{if } i \in V_2 \end{cases}$$

Then define the objective as follows.

$$\max \frac{1}{2} \sum_{(i,j) \in E} (1 - y_i y_j)$$

Input Graph i

Define the following graph $G = (V, E)$ where $|V| = 6$ and $|E| = 10$. Figure 1 provides a visual of our graph.

$$V = \{0 \to 5\}$$

$$E = \{(0,2),(0,3),(0,4),(1,2),(1,4),(1,5),(2,3),(3,4),(3,5),(4,5)\}$$

Input Graph ii

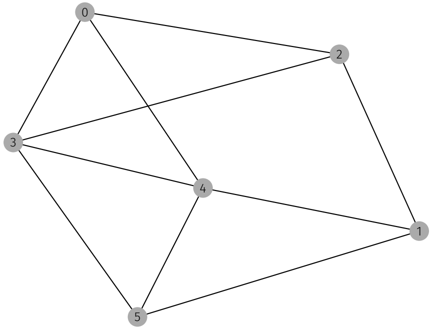

Figure 1: Input Graph $G = (V, E)$ with $|V| = 6$ Vertices and $|E| = 10$ Edges

Max-Cut of G represented with partitioning $Y = \{-1, -1, 1, -1, 1, 1\}$ and results in 8 cuts. Figure 2 provides a visual of the Max-Cut.

$$V_1 = \{0, 1, 3\}$$

$$V_2 = \{2, 4, 5\}$$

Input Graph iv

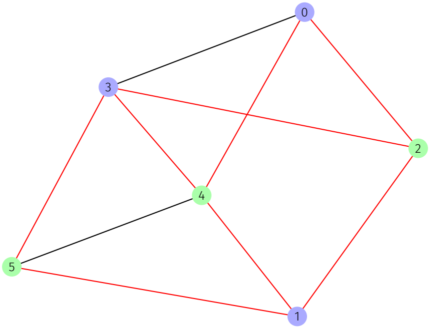

Figure 2: Max-Cut of Input Graph G

Algorithms

Recall that we aim to satisfy the following.

$$\max \frac{1}{2} \sum_{(i,j)\in E} (1 - y_i y_j)$$

We can rewrite this as a minimization problem.

$$\min \frac{1}{2} \sum_{(i,j)\in E} (1 + y_i y_j)$$

We can alter this to look more like a QUBO (expression value changes but solution set Y does not).

$$\min \sum_{(i,j)\in E} w_{ij} y_i y_j$$

Follow Ref [8] to convert out near-QUBO to an Ising Hamiltonian to obtain cost Hamilonian H_C.

$$H_C = \sum_{(i,j)\in E} w_{ij}\sigma_i^z\sigma_j^z$$

Follow Ref [3] to define mixer Hamiltonian H_B.

$$H_B = \sum_{j=1}^{n} \sigma_j^x$$

Follow Ref [3] to define cost unitary $U(C, \gamma)$ ($C = H_C$).

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha}$$

Figure 3 provides a circuit for $U(C, \gamma_0)$ for a simplified $G_s = \{\{0 \to 4\}, \{(0, 1)(0, 2)(0, 3)(0, 4)\}\}$. Figure 4 provides a decomposed version.

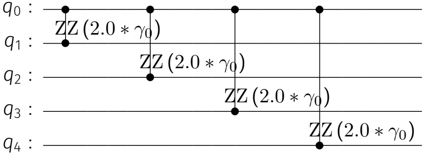

Figure 3: Cost Unitary Operator for G_s

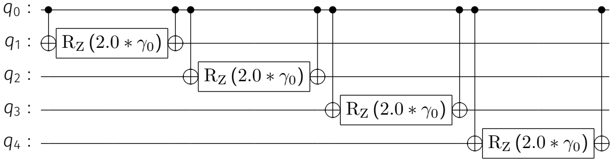

Figure 4: Decomposed Cost Unitary Operator for G_s

Follow Ref [3] to define mixer unitary $U(B, \beta)$ ($B = H_B$).

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x}$$

Figure 5 provides a circuit for $U(B, \beta_0)$ for a simplified $G_s = \{\{0 \rightarrow 4\}, \{(0, 1)(0, 2)(0, 3)(0, 4)\}\}$.

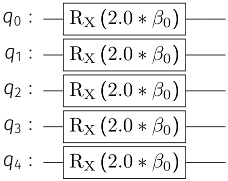

Figure 5: Mixer Unitary Operator for G_s

Quantum Approximate Optimization Algorithm viii

Follow Ref [3] to define our ansatz $|\gamma, \beta\rangle$ dependent on parameters $\theta = (\gamma, \beta)$.

$$|\gamma, \beta\rangle = U(B, \beta_p)U(C, \gamma_p)...U(B, \beta_1)U(C, \gamma_1)|s\rangle$$

Figure 6 provides a circuit for QAOA for our input graph G.

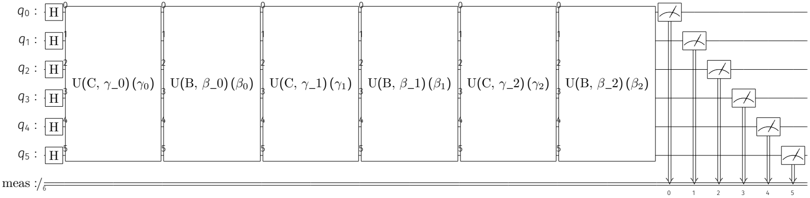

Figure 6: Circuit for QAOA for Max-Cut for G

Quantum Approximate Optimization Algorithm x

Utilize a classical optimizer with the ansatz to optimize θ to minimize the expectation value of H_C.

Gradient descent is a common approach for minimization via a function's gradient [7].

Performing the numerous, precise measurements required to evaluate the gradient $\nabla\mathcal{L}$ presents a challenge.

Simultaneous Perturbation Stochastic Approximation (SPSA) provides a more efficient alternative so we use that instead [6].

Grover's Algorithm i

We will not be "solving" a Max-Cut problem with Grover's algorithm in the traditional sense.

Grover's algorithm is meant for *searching* not *optimizing*.

Given the correct number of *cuts* x for a Max-Cut problem we will use Grover's algorithm to find the correct *partitioning*.

Must define oracle gate O which flips the phase of states with x cut edges [1].

How can we tell if a state has x cut edges?

We can do so with three steps:

1. Edge Processing: for each edge, check if it has been cut.
2. Cut Summing: sum the number of cut edges.
3. Phase Flip: if the sum equals x then flip.

Grover's Algorithm iii

For each edge $(i, j) \in E$ we define ancilla q_{ij} initialized to $|0\rangle$. We refer to these as edge ancillas.

We apply a Controlled-NOT (C-NOT) gate for control q_i and target q_{ij} and then another C-NOT for control q_j and target q_{ij}.

Every $(i, j) \in E$ has ancilla q_{ij} where $\text{set}(i) = \text{set}(j) \implies q_{ij} = |0\rangle$ and $\text{set}(i) \neq \text{set}(j) \implies q_{ij} = |1\rangle$.

Figure 7 provides a circuit for the previous steps for a simplified $G_s = \{\{0 \rightarrow 4\}, \{(0, 1)(0, 2)(0, 3)(0, 4)\}\}$.

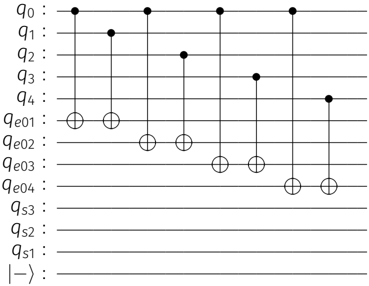

Figure 7: Oracle Edge Processing Step for G_s

We define $\lceil \log_2(|E| + 1) \rceil$ ancillas initialized to $|0\rangle$ to hold the sum of cut edges. We refer to these as summation ancillas.

For each edge ancilla q_{ij} where $q_{ij} = |1\rangle$, we increment the value stored by the summation ancillas by 1.

To perform an increment, going in descending order of significance, we flip a summation ancilla if all less-significant qubits are $|1\rangle$.

Figure 8 provides a circuit for the previous steps for a simplified $G_s = \{\{0 \to 4\}, \{(0,1)(0,2)(0,3)(0,4)\}\}$.

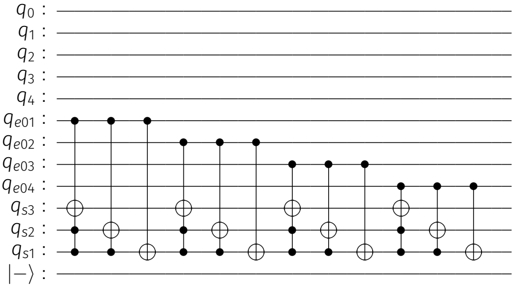

Figure 8: Oracle Cut Summing Step for G_s

Flip the phase of states which meet our condition by flipping the phase of summation ancillas if they represent x.

Figure 9 provides a circuit for the previous steps for a simplified $G_s = \{\{0 \rightarrow 4\}, \{(0,1)(0,2)(0,3)(0,4)\}\}$.

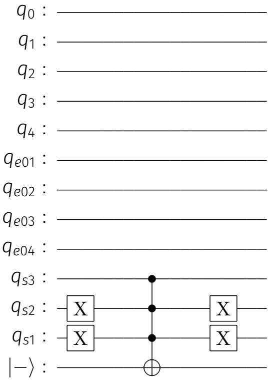

Figure 9: Oracle Phase Flip Step for G_s for $x = 4$

For cleanup we simply do all the operations leading up to the oracle phase flip in reverse.

This completes our O operator.

Follow Ref [4] to define the Grover Diffusion operator D.

Figure 10 provides a circuit for the previous steps for a simplified $G_s = \{\{0 \to 4\}, \{(0,1)(0,2)(0,3)(0,4)\}\}$.

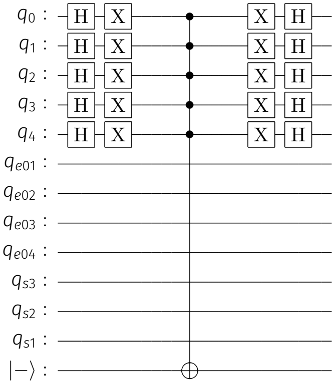

Figure 10: Diffusion Operator for G_s

We initialize our state qubits to a uniform superposition.

Following Ref [5] we apply OD $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ times.

We finally measure the resulting states.

Figure 11 provides a circuit for Grover's Algorithm for input graph G.

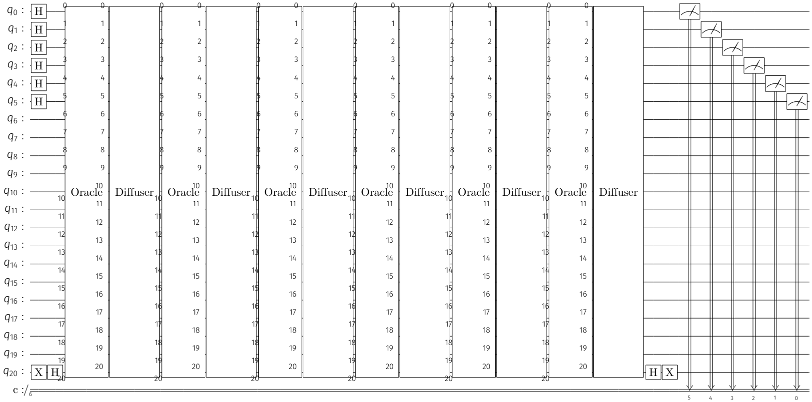

Figure 11: Circuit for Grover's Algorithm for Max-Cut for G

Results

Environment

We utilize the Aer simulator for Qiskit to simulate the execution of our circuits in noisy and ideal environments [1].

- For an ideal environment we utilize Aer's default ideal simulator.
- For a noisy environment we utilize Aer with Qiskit's "FakeCairoV2" backend [1].

Simulations were run on an AMD Ryzen 7 5800X 8-Core Processor (3.80 GHz).

Figures 12 and 13 provide the optimizer convergence and the output distribution respectively after running QAOA with SPSA for G for $p = 3$ and SPSA Iterations $= 300$ on an ideal simulator.

The Max-Cut partition is correctly determined.

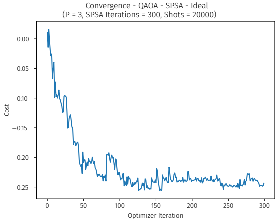

Figure 12: Optimizer Convergence for QAOA with SPSA on Ideal Simulator for $P = 3$, SPSA Iterations $= 300$

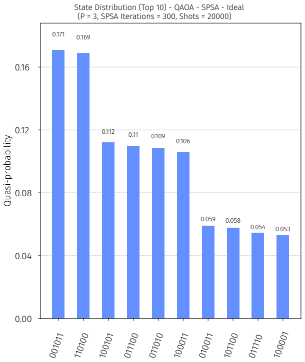

Figure 13: Top 10 Quantum State Distribution for QAOA with SPSA on Ideal Simulator for $P = 3$, SPSA Iterations $= 300$

Quantum Approximate Optimization Algorithm iii

Rapid approach to local minimum followed by gradual approach to global minimum.

Stochasticity introduced by SPSA helps escape local minima.

We observe relatively smooth convergence.

Suboptimal solutions have middling probabilities.

Quantum Approximate Optimization Algorithm iv

Figures 12 and 13 provide the optimizer convergence and the output distribution respectively after running QAOA with SPSA for G for $P = 3$ and SPSA Iterations $= 300$ on a noisy simulator.

The Max-Cut partition is correctly determined.

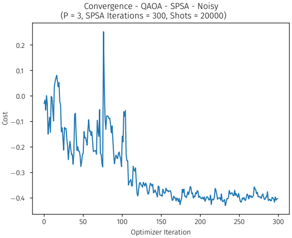

Figure 14: Optimizer Convergence for QAOA with SPSA on Noisy Simulator for $P = 3$, SPSA Iterations $= 300$

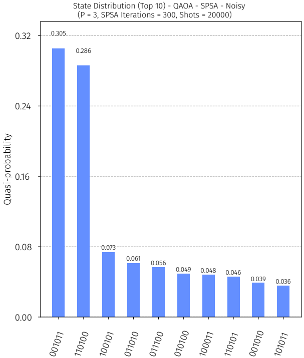

Figure 15: Top 10 Quantum State Distribution for QAOA with SPSA on Noisy Simulator for $P = 3$, SPSA Iterations $= 300$

Quantum Approximate Optimization Algorithm vi

Volatility during optimization process due to noise introduced by environment.

QAOA with SPSA handles noise well and finds the global minimum (usually).

Performance less consistent than on the ideal simulator and probabilities for non-solution states vary from run to run.

Figure 16 provides the output distribution after running Grover's Algorithm for G and Grover Iterations $= 6$ on an ideal simulator.

The partition which provides the given target number of cuts $x = 8$ is correctly identified.

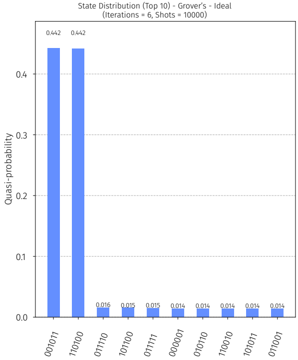

Figure 16: Top 10 Quantum State Distribution for Grover's Algorithm on Ideal Simulator

The algorithm performs well in an ideal environment, but does it perform well in a noisy environment?

Absolutely not, as demonstrated by Figure 17.

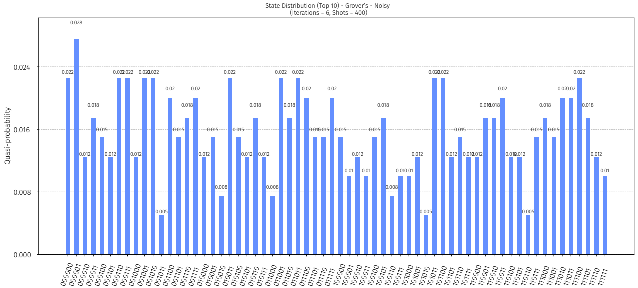

Figure 17: Full Quantum State Distribution for Grover's Algorithm on Noisy Simulator

Closing

Recap

We applied QAOA and Grover's Algorithm to an unweighted Max-Cut problem with an input graph $G = (V, E)$ with $|V| = 6$ vertices and $|E| = 10$ edges and tested our algorithms in both ideal and noisy environments.

With both algorithms, we observed favorable performance in ideal conditions.

QAOA with SPSA fared fine in a noisy environment, as expected.

Grover's algorithm did not fare well in a noisy environment, also as expected.

Closing

Thank you!

A. Abbas, S. Andersson, A. Asfaw, A. Corcoles, L. Bello, Y. Ben-Haim, M. Bozzo-Rey, S. Bravyi, N. Bronn, L. Capelluto, A. C. Vazquez, J. Ceroni, R. Chen, A. Frisch, J. Gambetta, S. Garion, L. Gil, S. D. L. P. Gonzalez, F. Harkins, T. Imamichi, P. Jayasinha, H. Kang, A. h. Karamlou, R. Loredo, D. McKay, A. Maldonado, A. Macaluso, A. Mezzacapo, Z. Minev, R. Movassagh, G. Nannicini, P. Nation, A. Phan, M. Pistoia, A. Rattew, J. Schaefer, J. Shabani, J. Smolin, J. Stenger, K. Temme, M. Tod, E. Wanzambi, S. Wood, and J. Wootton.
Learn quantum computation using qiskit, 2020.

C. Commander.
Maximum cut problem, max-cut.
01 2008.

E. Farhi, J. Goldstone, and S. Gutmann.
A quantum approximate optimization algorithm, 2014.

L. K. Grover.
A fast quantum mechanical algorithm for database search.
In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

P. Kaye, R. Laflamme, and M. Mosca.
An introduction to quantum computing.
OUP Oxford, 2006.

J. C. Spall.
A one-measurement form of simultaneous perturbation stochastic approximation.
Automatica, 33(1):109–112, 1997.

R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert.
Stochastic gradient descent for hybrid quantum-classical optimization.
Quantum, 4:314, 2020.

Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel.
Quantum approximate optimization algorithm for maxcut: A fermionic view.
Physical Review A, 97(2), Feb. 2018.