

Logarithmic encoding of Hamiltonians of NP-Hard Problems on a Quantum Computer

Yagnik Chatterjee^{1,2}, Marko J. Rančić¹, Eric Bourreau²

¹ TotalEnergies, 2 place Jean Millier 92078 Paris la Défense cedex, France
{yagnik.chatterjee,marko.rancic}@totalenergies.com

² LIRMM, Université de Montpellier, CNRS, 161 rue Ada, 34392 Montpellier Cedex 5
eric.bourreau@lirmm.fr

Keywords : *Combinatorial optimization, Variational quantum algorithms, Logarithmic Encoding*

1 Introduction

While finding exact solutions to NP-hard problems is difficult, many approximation algorithms exist. A huge amount of research has been carried out on hybrid quantum classical algorithms, where quantum measurements are used together with a classical optimization loop to obtain a solution.

The Quantum Approximate Optimization Algorithm (QAOA) [1] is one of the most commonly used hybrid algorithms. It scales linearly with problem size. This means that a graph of n nodes would require an n -qubit quantum computer (QC). Current QCs are relatively small and to tackle real-life problems, it is important to improve how the algorithms scale.

Taking this into consideration, an algorithm to encode an n -node MaxCut problem on a QC using $\lceil \log n \rceil$ qubits was developed [2]. This encoding allows us to represent much larger problems using a fairly small number of qubits. The number of CNOT gates required for the QAOA ansatz is $p|E|$, where p is the depth of the algorithm and $|E|$ is the number of edges in the graph. In our algorithm the number of CNOTs is equal to $|V| - 1$, $|V|$ being the number of vertices. Generally, and especially at higher densities, it is easy to see that $p|E| \gg |V|$. Thus our circuit is much shallower than that of QAOA.

In this work, we study the performance of the MaxCut algorithm and show ways in which we can extend this algorithm to a plethora of problems. The algorithms are tested on a quantum simulator with graph sizes of over a hundred nodes and on real QCs up to a graph size of 256.

2 A qubit-efficient algorithm

Given a graph $G(V, E)$, the MaxCut can be represented using the graph Laplacian matrix. The graph Laplacian is defined as follows :

$$L_{ij} = \begin{cases} \text{degree}(i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The MaxCut is given by the following equation :

$$C = \frac{1}{4} x^T L x \quad (2)$$

where L is the Laplacian matrix and $x \in \{1, -1\}^{|V|}$ is the bi-partition vector.

The quantum analog of equation (2) is

$$C(\theta_1 \dots \theta_n) = 2^{n-2} \langle \Psi(\theta_1 \dots \theta_n) | L | \Psi(\theta_1 \dots \theta_n) \rangle \quad (3)$$

where L is the Laplacian matrix of the graph, $|\Psi\rangle$ is the parameterized ansatz, n is the size of the graph, and θ_i are the parameters to be optimized. 2^{n-2} is the normalization constant. Here $|\Psi\rangle$ is a $N = \lceil \log n \rceil$ qubit state. The Laplacian acts as our problem Hamiltonian.

3 Approaches and Sample Results

We use the inter-convertibility of NP hard problems [3] and QUBO [4] to model the problem Hamiltonians of a number of problems.

Table 1 shows the results for the MaxCut problem and Table 2 shows the results of the Minimum Partition problem on a QC (IBM Quantum’s *ibmq_mumbai*) and using a quantum simulator.

The MaxCut instance generated is a random graph using the *networkx* package of Python. The instance shown has a graph density of 0.5 and a random seed of 0. The Minimum Partition instances consist of random integers between 1 and 100.

Note that these results are only representative and the work includes tests on many more instances of different problem sizes.

Size	Normalized Cut Simulator(%)	Normalized Cut QC(%)
256	84.6	81.7

TAB. 1: Result of **MaxCut** Problem Normalized using upper bound obtained from the Goemans Williamson algorithm. The simulator data is based on an average of 50 runs while the QC data is based on a single run.

Problem : Minimum Partition

Definition : Given a set $S = \{w : w \in \mathbb{Z}^+\}$, find $A \subseteq S$ that minimizes $|\sum_{w_i \in A} w_i - \sum_{w_i \notin A} w_i|$.

Size	Normalized Results Simulator(%)	Normalized Results QC(%)
32	99.3	92.5
64	90.7	92.5
128	98.9	–

TAB. 2: Results of **Minimum Partition** Problem Normalized using optimal value obtained from an Integer Linear Program. The simulator data is based on 100 runs for the problem of size 32, 10 runs for size 64 and 4 runs for size 128. For all cases the QC runs are based on a single run.

References

- [1] E. Farhi and J. Goldstone and S. Gutmann, *A Quantum Approximate Optimization Algorithm*. arXiv:1411.4028, 2014.
- [2] Marko J. Rančić, *An exponentially more efficient optimization algorithm for noisy quantum computers*. arXiv:2110.10788, 2021.
- [3] R. M. Karp, *Reducibility among Combinatorial Problems*. In: Miller, R.E., Thatcher, J.W., Bohlinger, J.D. (eds) Complexity of Computer Computations. The IBM Research Symposia Series. Springer, Boston, MA. 1972.
- [4] F. Glover, G. Kochenberger and Yu Du, *Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models* 4OR, Springer **17(4)**, 2019.