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

Yagnik Chatterjee[1,2], Marko J. Rančić[1], Eric Bourreau[2]

[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

The MaxCut problem is defined as follows : Given a weighted graph $G(V,E,w)$, find $x \in \{1,-1\}^{|V|}$ that maximizes $\sum_{ij} w_{ij} \dfrac{(1-x_i)(1+x_j)}{4} \forall \{(i,j) \in E\}$, where w_{ij} are the weights on the edges.

The MaxCut can be represented using the graph Laplacian matrix. The graph Laplacian is defined as follows :

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

The MaxCut is given by the following equation :

$$C = \frac{1}{4} x^T L x \tag{2}$$

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

The quantum analog of equation (2) is

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

where $|\Psi\rangle$ 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 and $\langle \Psi(\theta_1...\theta_n)| L |\Psi(\theta_1...\theta_n)\rangle$ is the expectation value or energy of the state with parameters $\theta_1...\theta_n$. The Laplacian acts as our problem Hamiltonian.

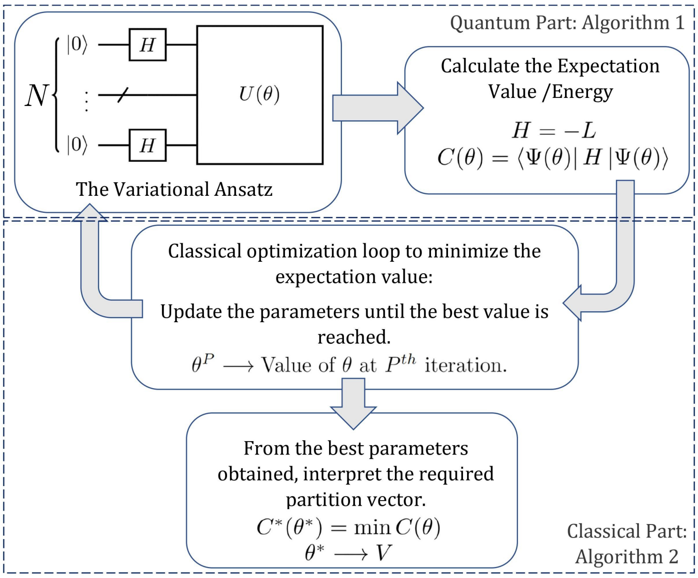

FIG. 1: Diagrammatic representation of the hybrid quantum-classical algorithm.

3 Approaches

One way to solve other problems using the alogrithm is to directly or indirectly convert them to the MaxCut problem. The inter-convertibility of NP hard problems was shown by Karp [3]. We test the Minimum Partition and the Maximum Clique problems using this method.

A second way to do this would be to use the Quadratic Unconstrained Binary Optimization (QUBO) matrix [4] of the specific problem instead of the graph Laplacian.

To obtain the QUBO matrix the objective function must be of the following form:

$$P = \sum_i a_i x_i^2 + \sum_{ij} a_{ij} x_i x_j \tag{4}$$

It can be rewritten as:

$$P = \begin{pmatrix} x_1 & x_2 & ... & x_n \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix} \tag{5}$$

$$P = x^T Q x \tag{6}$$

$$Q = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix} \tag{7}$$

Q is the required QUBO matrix.

We test the Maximum Weighted Independent Set Problem using this method.

4 Sample Results

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. In all cases the classical optimizer used is the genetic algorithm.

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. The Goemans-Williamson algorithm is used as a benchmark.

Note that these results are only representative and the complete work [5] includes tests on many more instances of different problem sizes.

Instance	Solution	GW Range	% Diff.
Size=128, Density=0.4	1538	1796 - 1864	82.5 - 85.6
Size=128, Density=0.5	2022	2186 - 2271	89.0 - 92.5
Size=256, Density=0.5	8079	8701 - 8880	90.9 - 92.8

TAB. 1: 128 and 256-Node **MaxCut** results using QC.

Problem : Minimum Partition

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

Approach used : Reformulation

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.

Problem : Maximum Weighted Independent Set (MWIS)

Definition : *Given a graph* $G(V, E)$ *with node weights* w_i*, find* $x \in \{0,1\}^{|V|}$ *that maximizes* $\sum_i w_i x_i$ *such that* $x_i + x_j \leq 1 \; \forall \; (i,j) \in E$.

Approach used : QUBO

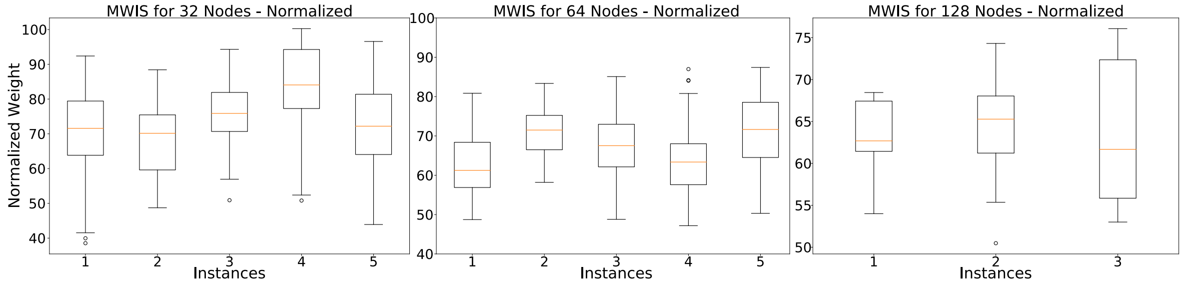

FIG. 2: **MWIS** problem for 32, 64 and 128-node graphs using a quantum simulator. Each instance was run on a quantum simulator with GA 50 times for graph sizes of 32 and 64 and 10 times for graph size of 128.

Figure 2 shows the results of the MWIS problem on a quantum simulator. The optimal solution was obtained using CPLEX solver and was used to normalise the data.

Table 3 shows results of the MWIS problem using a quantum computer.

Size	Solution	Optimal Solution	% Diff.
32	96.56	140.95	68.5
64	149.42	231.18	64.6
128	321.69	491.67	65.4

TAB. 3: **MWIS** results using QC.

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, accepted in Phy. Rev. Res., 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.

[5] Y. Chatterjee, E. Bourreau and M. J. Rančić, *Solving various NP-Hard problems using exponentially fewer qubits on a Quantum Computer* arXiv:2301.06978, 2023.