Arttu Aleksi Huikuri arttu.huikuri@gmail.com
Konstantin Sakharovskiy

FYSS5403: Solving the combinatorial exact cover problem using quantum approximate optimization algorithm

Report returned: 17.12.2020

Abstract:

IMBQ:s physical qubits and qiskit simulations were used to apply the quantum approximate optimization algorithm to the exact cover problem with 2 qubits. The results show that IBMQ:s qubits can be used to solve this problem.

1 Introduction

The exact cover problem is: given a set U and a set of its subsets V_i, is there a subset of the sets V_i R so that the elements of R are disjoint sets and U is a union of all of the elements of R? [1]. The Quantum approximate optimization algorithm (QAOA) [2] can be used to solve the exact cover problem by encoding the exact cover problem into an Ising hamiltonian [1] and then solving for the ground state of this hamiltonian by conventionally going through the possible states with variational angles. [2]

2 Theory

The cost function for the exact cover problem shown by Bengtsson et al. [2] is

$$\hat{C} = h_1 Z_1 + h_2 Z_2 + J Z_1 Z_2 \tag{1}$$

and the mixing Hamiltonian is

$$\hat{B} = X_1 + X_2 \tag{2}$$

for a two qubit circuit. Finding the ground state of the cost function requires applying these two Hamiltonians with real variational angles γ and β to an equal superposition of 2 qubits, achieved by applying a Hadamard gate to both of the qubits. This prepares the state

$$|\gamma, \beta\rangle = e^{-i\beta\hat{B}} e^{-i\gamma\hat{C}} \left(\frac{|0\rangle + |1\rangle}{2} \right)^{\otimes 2}. \tag{3}$$

Measuring this state gives the cost function. By classically varying β and γ, the minimum value of the cost function can be found. [2]

Bengtsson et al. [2] show an implementation of the (QAOA) with p levels, where level i is the quantum circuit is shown in figure 1. In order to represent this circuit as an array, the rotation gates R_x and R_z can be represented as

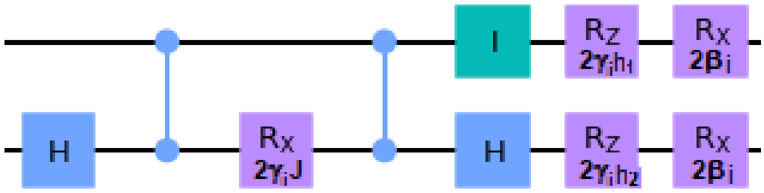

Figure 1: Quantum circuit for level i of the QAOA algorithm

[3]

$$\begin{aligned} R_z &= \exp(-i\frac{\lambda}{2}Z) \\ R_x &= \exp(-i\frac{\theta}{2}X), \end{aligned} \tag{4}$$

are used, where λ, θ are the rotation angles and Z and X are the Z and X gates. It is easy to see that the two last R_x gates apply to the state using the identity in (4) yields

$$\exp(-i\beta X_2) \otimes \exp(-i\beta X_1) = \exp(-i\beta(X_1 + X_2)) = e^{-i\beta\hat{B}}. \tag{5}$$

Similarly, the parallel R_z gates apply

$$\exp(-i\gamma h_2 Z_2) \otimes \exp(-i\gamma H_1 Z_1) = \exp(-i\gamma(h_1 Z_1 + h_2 Z_2)). \tag{6}$$

The gates before them can be represented with

$$\big(H \otimes I\big) Z_{NOT} \big(R_x(2\gamma J) \otimes I\big) Z_{NOT} \big(H \otimes I\big), \tag{7}$$

where

$$\big(H \otimes I\big) = \frac{1}{\sqrt{2}} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \tag{8}$$

$$Z_{NOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \tag{9}$$

$$\big(R_x(2\gamma J) \otimes I\big) = \begin{bmatrix} I\cos\gamma J & -iI\sin\gamma J \\ -iI\sin\gamma J & I\cos\gamma J \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} \cos\gamma J & 0 & -i\sin\gamma J & 0 \\ 0 & \cos\gamma J & 0 & -i\sin\gamma J \\ -i\sin\gamma J & 0 & \cos\gamma J & 0 \\ 0 & -i\sin\gamma J & 0 & \cos\gamma J \end{bmatrix},$$

Multiplying these, we get

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{11}$$

and

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{12}$$

Now equation (7) can be written as

$$\frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma J & 0 & -i\sin\gamma J & 0 \\ 0 & \cos\gamma J & 0 & -i\sin\gamma J \\ -i\sin\gamma J & 0 & \cos\gamma J & 0 \\ 0 & -i\sin\gamma J & 0 & \cos\gamma J \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \tag{13}$$

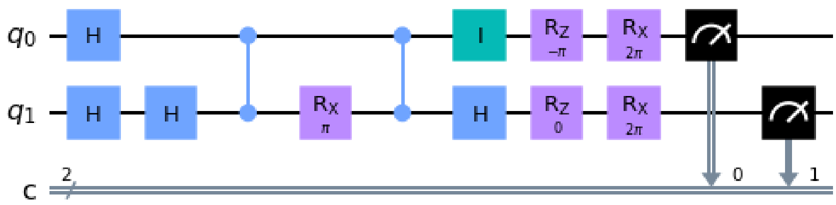

Figure 2: Quantum circuit for the QAOA algorithm

Which equals

$$\begin{bmatrix} e^{-i\gamma J} & 0 & 0 & 0 \\ 0 & e^{i\gamma J} & 0 & 0 \\ 0 & 0 & e^{i\gamma J} & 0 \\ 0 & 0 & 0 & e^{-i\gamma J} \end{bmatrix} = \exp\left(i\gamma J Z_1 Z_2\right). \tag{14}$$

The final quantum circuit can then be represented by combining equations (5), (6) and (14) by multiplying the results. We get

$$\exp\left(-i\gamma J Z_1 Z_2\right)\exp(-i\gamma(h_1 Z_1 + h_2 Z_2))e^{-i\beta\hat{B}} = e^{-i\gamma\hat{C}}e^{-i\beta\hat{B}} \tag{15}$$

and by first initializing the qubits into an equal superposition by applying hadamard gates to both, we gain the state (3). Adding a measurement to the qubits, the complete circuit with p = 1 is shown in figure 2.

3 Simulations and runs on IMBQ:s qubits

The four exact cover problems presented by Bengtsson et al. [2] were solved using qiskit with both simulations and by solving them in IMBQ:s physical qubits. Table 1 shows the problems and the values of h_1, h_2 and J used to encode each problem. [2]

Table 1: The four exact cover problems with two subsets, as shown in table 1 in reference [2] Note that the order of the qubits is reversed in qiskit compared to the ones in [2].

#	Subsets	h_1	h_2	J	Solution
A	$\{x_1, x_2\}, \{x_1\}$	-1/2	0	1/2	$\|01\rangle$
B	$\{x_1, x_2\}, \{\}$	-1	0	0	$\|01\rangle$ or $\|11\rangle$
C	$\{x_1\}, \{x_2\}$	-1/2	-1/2	0	$\|11\rangle$
D	$\{x_1, x_2\}, \{x_1, x_2\}$	0	0	1	$\|10\rangle$ or $\|01\rangle$

3.1 Qiskit simulation

Simulation of the problems using qiskit's qasm simulator [3] yields a result for each point in the (β, γ) space, with $\beta, \gamma \in [0, \pi]$. Similarly to Bengtsson et al. [2] 61x61 points were simulated. 1000 counts were simulated for each point. the results are shown in figure 3.

The circuit shown in figure 2 was shown to work in the simulations and the results were quite close to those gained by Bengtsson et al. [2] A python code for running the simulations and plotting the results is shown in `https://github.com/kosakhro/QAOA-for-exact-cover/blob/main/qaoa_test.ipynb`.

3.2 Running QAOA on IBMQ:s qubits

The solution for the QAOA on IBMQ:s physical qubits are shown for each problem in the following links. Note that the quantum circuits were transpiled to improve performance.

Problem A: `https://github.com/kosakhro/QAOA-for-exact-cover/blob/main/A-test-on-real-QC.ipynb`

Problem B: `https://github.com/kosakhro/QAOA-for-exact-cover/blob/main/B-test-on-real-QC.ipynb`

Problem C: `https://github.com/kosakhro/QAOA-for-exact-cover/blob/main/C-test-on-real-QC.ipynb`

Problem D: `https://github.com/kosakhro/QAOA-for-exact-cover/blob/main/D-test-on-real-QC.ipynb`.

The results are represented as heatmaps in figure 4. As expected, in B and

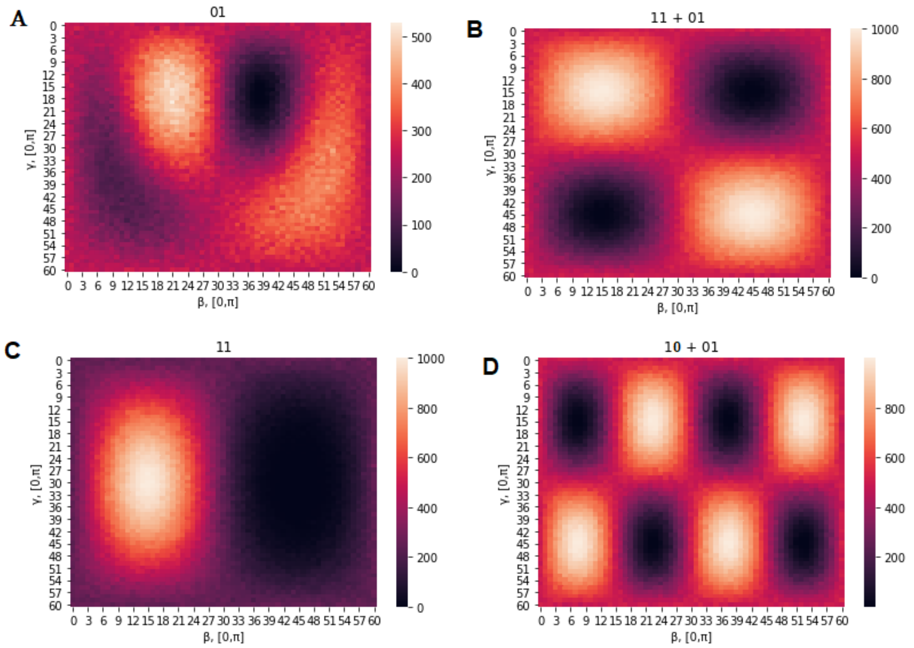

Figure 3: Simulated results results with Qiskit's QASM simulator

C the ground states are degenerate [1, 2] and in C the ground state is found in $|11\rangle$. Similarly to Bengtsson et al. [2] problem A gave the least certain solution, which could possibly be improved by making the circuit have a second application of the hamiltonians $\hat{B}, \hat{C}$ (p=2). [2].

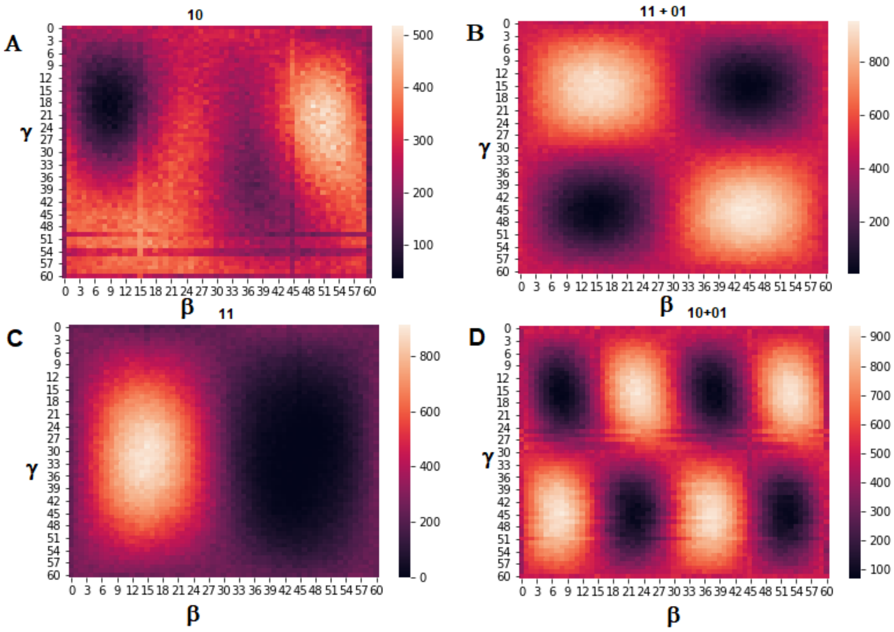

Figure 4: Measured results on IMBQ:s qubits. Note that for problem A, the correct heatmap would be 01, due to the different ordering of qubits in qiskit and in [2]

References

[1] A. Lucas, *Ising formulations of many NP complete problems*, Frontiers in physics volume 2 article 5, 2014

[2] A. Bengtsson et al., *Improved success probability with greater circuit depth for the quantum approximate optimization algorithm*, Cornell University Library, arXiv.org 2020

[3] Qiskit Development Team, *Qiskit documentation*, Qiskit.org, 2020