Solving the Max-cut problem with neutral atoms

Victor Onofre*

***Abstract*— In this work, I solved the Max-cut problem using the Quantum Approximate Optimization Algorithm (QAOA) and the Variational Quantum Adiabatic Algorithm (VQAA) with neutral atoms. Also, it explained the encoding of the problem into the atomic register. All simulations were done with Pulser and found in this GitHub Repository.**

I. Max-cut problem

The goal of the Maximum cut (Max-cut) problem is to have as many cut edges as possible. Where an edge is cut if its endpoints have different colors. The output is a "2-coloring" (two subsets) of V: Each vertex is designated a color (examples: blue or red). The maximum cut can be formulated as an optimization problem:

$$\max_s \frac{1}{2}\sum_{ij\in E}(1-s_i s_j), \quad s_i \in \{-1,+1\} \tag{1}$$

If we have the same sign, no edge is cut (no contribution to the objective function), $\frac{1}{2}(1-s_i s_j)=0$. With different sign, an edge is cut (contribution to the objective function), $\frac{1}{2}(1-s_i s_j)=1$.

The Max-cut problem can be naturally transformed into the Quadratic Unconstrained Binary Optimization (QUBO) model [1]. If we look for a maximum cut in our graph, then we are looking for a way to split the nodes into two groups so that there are as many edges as possible between the groups, we can define a binary variable x_i for each node,

$$x_i = 0 \rightarrow \text{The node is in the first subset} \tag{2}$$

$$x_i = 1 \rightarrow \text{The node is in the second subset} \tag{3}$$

Defining the $EdgeScore(i,j)$ as 1 if the two nodes are in a different subset, 0 otherwise, we can build the following table:

x_i	x_j	$EdgeScore(i,j)$
0	0	0
0	1	1
1	0	1
1	1	0

TABLE I: Different combinations of the nodes to compute the edge score.

The $EdgeScore(i,j)$ is define as $x_i + x_j - 2x_i x_j$. Then for the entire graph, the objective function can be written as $\max\sum_{(i,j)\in E}(x_i + x_j - 2x_i x_j)$. Multiplying by -1 to the convert to a minimization problem:

$$\min \sum_{(i,j)\in E}(-x_i - x_j + 2x_i x_j) \tag{4}$$

If x is one of the binary solutions, the QUBO matrix will give us the total cuts as follows:

$$f_Q(x) = x^T Q x \tag{5}$$

where Q is defined by the matrix create with equation 4.

II. Neutral Atoms Device

Neutral atom devices are made up of two main components: the Register and the channels [2]. The Register is a group of trapped atoms arranged in a defined but changeable configuration. Each atom holds a specific quantum state that is encoded in a particular electronic level. The channels are responsible for manipulating the state of the atoms by addressing specific electronic transitions. These channels consist mostly of lasers. Each channel is tuned such that each of its pulses coherently drives a specific electronic transition between two energy levels of an atom.

In this system, a pulse acting on a atom i, with Rabi fequency $\Omega(t)$, detuning $\delta(t)$ and a fixed phase ϕ, will have the Hamiltonian terms:

$$\frac{\hbar\Omega(t)}{2}(\cos(\phi))\sigma_i^x - \sin(\phi)\sigma_i^y) - \frac{\hbar}{2}\delta(t)\sigma_i^z \tag{6}$$

where σ^α for $\alpha = x, y, z$ are the Pauli matrices.

Atoms in neutral atom devices can be driven to Rydberg states to enable them to interact over large distances. The Van der Waals force describes the interaction between two atoms at the same Rydberg level and at a distance R, which scales as R^{-6}.

The Rydberg blockade is a phenomenon where an atom cannot be excited to the Rydberg level if there is already another atom nearby in such state. In order to represent this interaction, we consider it as a penalty term in the Hamiltonian that describes the state where both the atoms are excited: $U_{ij}n_i n_j$,, where $n = (1+\sigma^z)/2$ is the projector on the Rydberg state, $U_{ij} \propto R_{ij}^{-6}$ and R_{ij} is the distance between the atoms i and j.

The proportionality constant is set by the chosen Rydeberg level. If the atoms are excited simultaneously, only the entangled state $(|gr\rangle + |rg\rangle)/\sqrt{2}$ is obtained.

An entire array of interacting atoms acted on by the same pulse can be represented as an Ising-like Hamiltoniam:

$$H = \frac{\hbar}{2}\sum_i \Omega(t)\sigma_i^x - \frac{\hbar}{2}\sum_i \delta(t)\sigma_i^z + \sum_{i<j} U_{ij}n_i n_j, \tag{7}$$

*Quantum Open Source Foundation; email: vonofre68@gmail.com

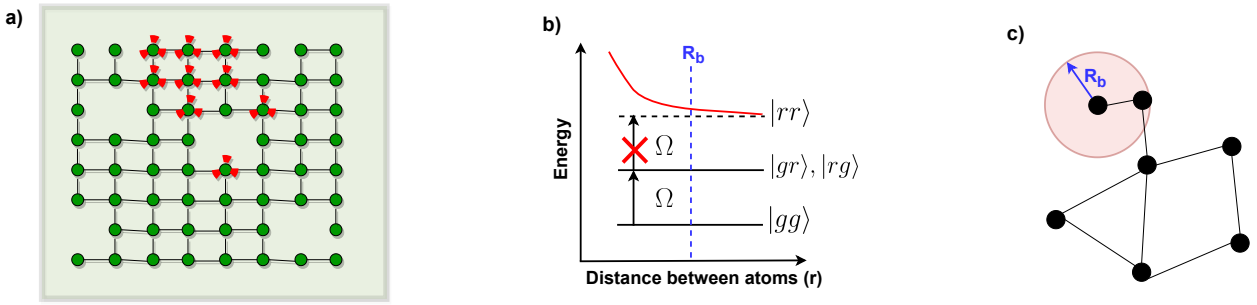

Fig. 1: a) Arrays of optical tweezers are used to prepare a register made of neutral atoms . b) Rydberg Blockade Effect, an atom cannot be excited to the Rydberg level if a nearby atom is already in such state. c) Rydberg atoms correspond to the nodes of a Unit Disk graph.Due to the Rydberg blockade effect.

With *Pulser* we can simulate the behavior of neutral atom devices. *Pulser* is an open-source Python software package from Pasqal. It provides easy-to-use libraries for designing and simulating pulse sequences that act on programmable arrays of neutral atoms [2].

III. Graph encoding in neutral atoms

We can use a unit-disk graph since they are in one-to-one correspondence with atom arrangements in 2D. Specifically, each atom represents a vertex, and we identify the blockade radius with the unit-disk radius of the graph. A unit-disk graph is a graph $G = (V, E)$ with vertices V and edges E that can be embedded in the 2D Euclidean plane such that has an edge between any two vertices whose Euclidean distance is less than 1. We can encode the off-diagonal terms of the adjacency matrix by using the Rydberg interaction between atoms. In this case we used the QUBO matrix to the encoding, we can see that both matrices shown the same information about whether pairs of vertices are adjacent or not in the graph. As an example consider the graph shown in figure 2.

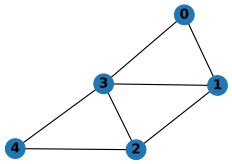

Fig. 2: Example graph to encode in a neutral atom device.

The QUBO matrix Q for our example is define using the equation 4 as,

$$Q = \begin{pmatrix} -2 & 2 & 0 & 2 & 0 \\ 0 & -3 & 2 & 2 & 0 \\ 0 & 0 & -3 & 2 & 2 \\ 0 & 0 & 0 & -4 & 2 \\ 0 & 0 & 0 & 0 & -2 \end{pmatrix} \tag{8}$$

And, the adjacency matrix is defined as,

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{9}$$

We can see that the same information in convey in both matrices related to the adjacent pairs of vertices.

As the interaction U depends on the pairwise distance (C_6/r_{ij}^6) between atoms i and j, it is possible to approximate the optimal positions of the atoms in the Register that replicate best the off-diagonal terms of the matrix [2]. An optimization process was done using the minimizer from Scipy. Given that the QUBO matrix has small values, is necessary to scale the values, in this work with a scale factor of 3 was enough for a sufficient encoding. This method fails with graphs with to many connections, which is consisting with the results in [3] where it is stated that finding a unit-disk realization for a given graph is proven to be NP-hard. A more advanced algorithm for encoding the graphs is necessary.

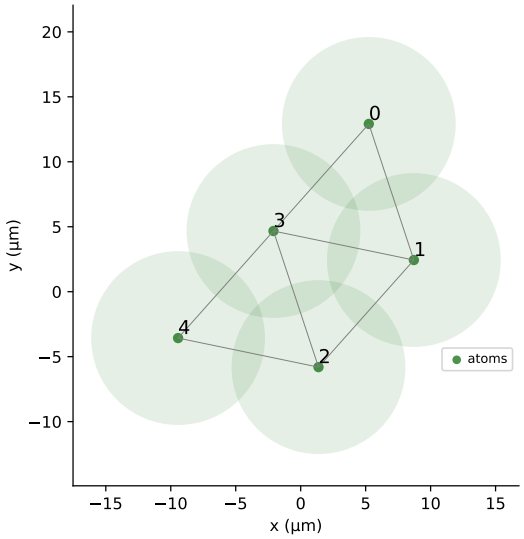

Fig. 3: Encoding into a atomic register of the graph shown in figure 2.

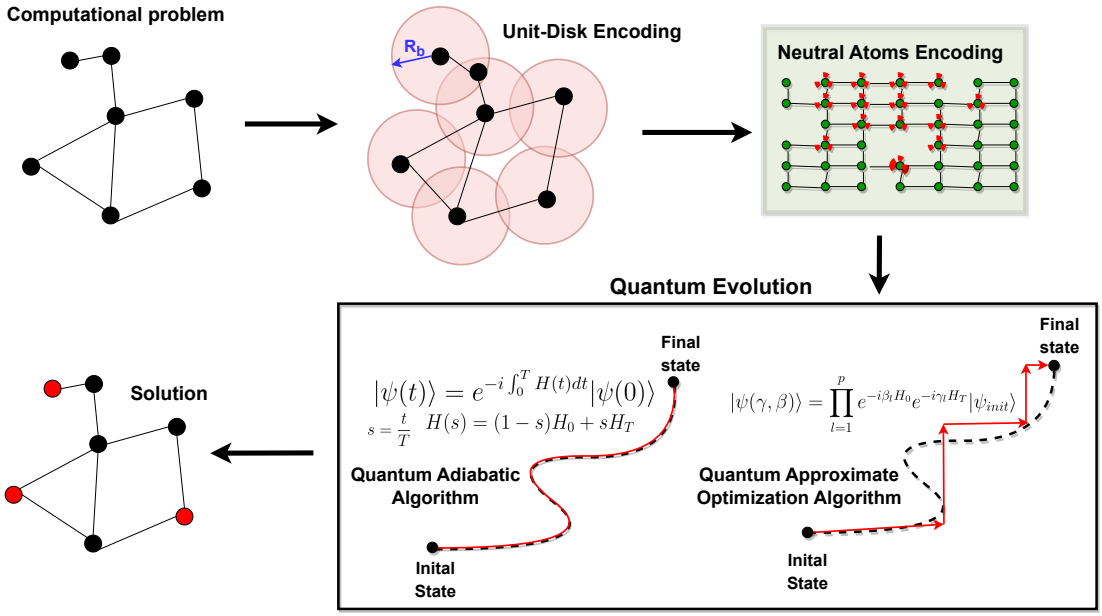

Fig. 5: Given the computational problem (in our case the Max-Cut problem) it is necessary to map it to a unit-disk graph for the encoding. The most natural quantum evolution of the system in a neutral atoms device is the QAA or QAOA. We can observe a comparison of the two methods, the QAOA can be seen as a discretization of the QAA.

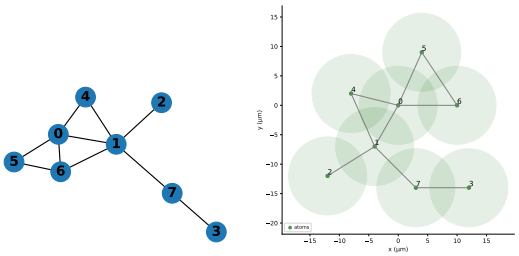

Fig. 4: Example of graph which is hard to encoded with the method used in this work.

Examples of the encoding with different graph instances can be found here.

IV. Methods

A. Quantum Adiabatic Algorithm (QAA)

A method to solve combinatorial optimization problems is the Quantum Adiabatic Algorithm (QAA) [4]. We consider a Hamiltonian of the following form:

$$H(t) = u(t)H_M + (1 - u(t))H_C, \tag{10}$$

where H_C is the problem (cost) Hamiltonian, it encodes the optimization task that we are trying to solve, H_M is the mixer Hamiltonian, it encodes quantum mixing (a uniform transverse field on qubits), and $u(t)$ the control function. The idea behind the adiabatic algorithm is to slowly evolve the system from an easy-to-prepare ground state to the ground state of H. If done slowly enough, the system of atoms stays in the instantaneous ground-state. An example of an Pulse applying the adiabatic method with *Pulser* is showed in figure 6

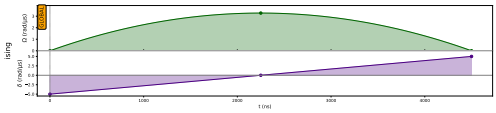

Fig. 6: Example of an adiabatic pulse using *Pulser*.

In figure 7 we can see an example of time evolution on the adiabatic algorithm. Slower times gives better solutions.

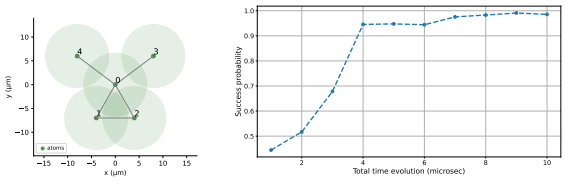

Fig. 7: (Rigth) Graph encoded to solved the Maximum Independent Set (MIS) problem. (Left) Success probability of solving the MIS problem with different time evolution using the pulse showed in figure 6. All the figures were obtain using *Pulser*.

B. Quantum Approximate Optimization Algorithm (QAOA)

We can use the Quantum Approximation Optimization Algorithm (QAOA) to solve the Max-cut problem [5]. The QAOA is a hybrid quantum-classical algorithm. The classical part of the algorithm is an optimizer. For our specific problem, the corresponding quantum part is a neutral-atom quantum computer first evolving under the

Rydberg Hamiltonian with parameterized pulse sequences and then being measured.

The standard definition of QAOA involves applying the problem (expressed as a cost function) Hamiltonian C and the transverse field Hamiltonian, or mixer, B alternately. In our case, the QAOA is define by a sequential layers of constant pulses with a variable duration time [6, 7]. The mixer B is applied by a pulse with a frequency 1 $rad/\mu s$ and detuning 0. The Hamiltonian C, in this case will not be definied by the cost function but instead by a pulse with frequency 0 and detuning 1 $rad/\mu s$. Each pulse has a parametrized time (beta and gamma) changing depending on the results of the classcal optimizer. It is important parameter to take into account the number of layers (p), meaning the number of pulses used to define the sequence.

The QAOA has the key feature that as p increases the approximation improves. It's not guaranteed to give the lowest possible value of the cost function (hence "approximate" in the name). Even though at $p \to \infty$, the QAOA reaches the same behaviour as with the Quantum Adiabatic Algorithm.

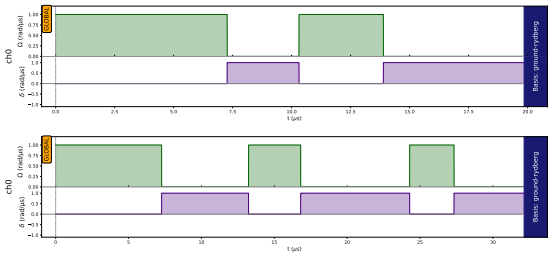

Fig. 8: Sequence for the QAOA with random parameters for $p = 2$ (Top) and $p = 3$ (Bottom).

C. Variational Quantum Adiabatic Algorithm (VQAA)

The second approach is a variational quantum adiabatic algorithm (VQAA) based on [6]. Similar to QAOA, the times are treated as the variational parameters. However, the evolution in the different chunks is performed adiabatically. Here, the VQAA differs from the fixed Hamiltonians found in the QAOA. The adiabatic algorithm is known to return the ground state for a sufficiently long time T. But, due to the limited decoherence times of current NISQ devices and analog quantum simulators, the time T for an optimal solution is not possible to implement. Finding a path with a time possible for implementation is of great relevance for the feasibility of the adiabatic approach. For example, the time evolution cannot be of 20 μs because at the moment it is not possible to implement that time in real hardware. Normally, you can do a sweep over the parameters, but this is computationally expensive. The idea of the VQAA is to find the correct parameters for the adiabatic evolution using a classical optimizer (in this case the ones from Scipy).

We aim to find an optimized profile for the adiabatic evolution determined by the parameters Ω, δ y t.The VQAA allows for a significant acceleration compared to the QAA with a linear adiabatic path [8], yet requires fewer parameters and measurements than the QAOA. As we can see in figure 9, different parameters give us different adiabatic paths, that may or not may be possible in real hardware. We aim to find the correct path that is possible to implement in real quantum hardware.

V. Results

A. QAOA

In table II we have the different combinations that can give us the maximum cut of the graph (figure 3), in this case 5 cuts.

Solution	Number of cuts
00110	5
01001	5
01010	5
01011	5
10100	5
10101	5
10110	5
11001	5

TABLE II: Different combinations of solutions given the max cut for the graph shown in figure 6 and 12

In figure 11 we can observe the results of the QAOA to solve the graph instance shown in figure 10. The optimal parameters obtained with $p = 4$ using the COBYLA optimizer are: [10.16, 3.60, 6.754, 3.741, 3.20, 5.99, 10.64, 8.57] in μs units. The results with higher counts are shown in figure 12, with 5 cuts in total. The notebooks with the full solution can be found here.

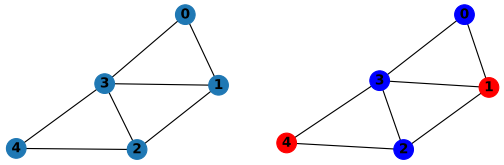

Fig. 10: Right)Problem graph.(Left) Solution using QAOA with 4 p layers with 5 cuts.

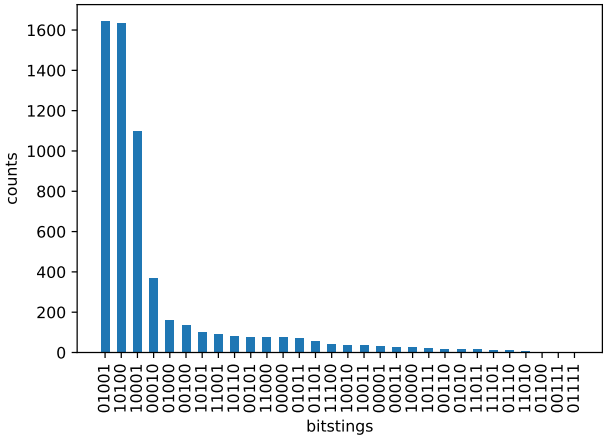

Fig. 11: Histogram output of the QAOA with 4 p layers for the graph shown in figure 10

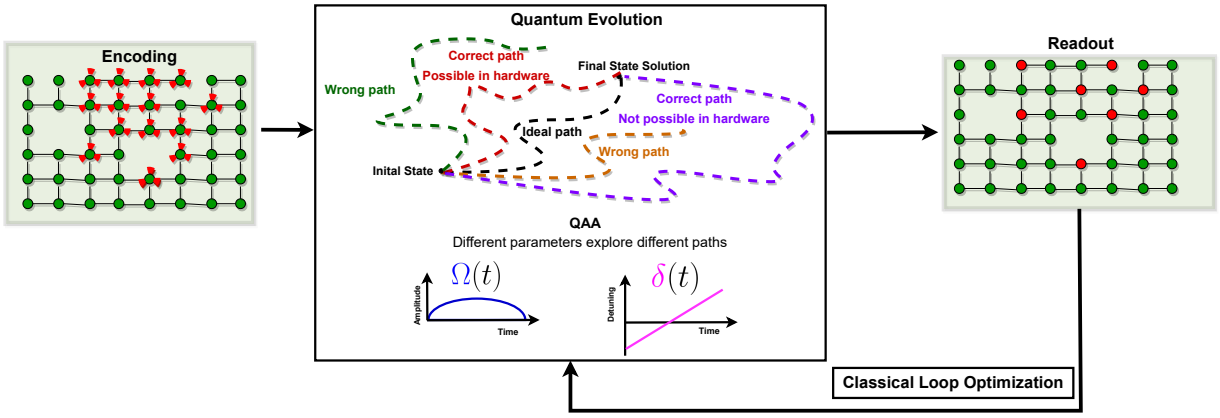

Fig. 9: The VQAA allows for a significant acceleration compared to the QAA with a linear adiabatic path [8], yet requires fewer parameters and measurements than the QAOA. We aim at finding an optimized profile for the adiabatic evolution.

More results with QAOA with different graph instances can be found here.

B. VQAA

In figure 13 we can observe the results of the VQAA to solve the graph instance shown in figure 3. The optimal parameters obtained with the Nelder-Mead optimizer are: a time evolution of 23045.76 ns, a $\Omega = 4.27$ and $\delta = 4.51$. The results with higher counts are shown in figure 12. The notebooks with the full solution can be found here

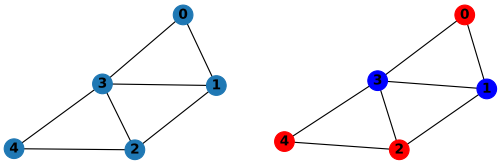

Fig. 12: (Right)Problem graph. (Left) Result with higher counts using VQAA with the Nelder-Mead optimizer with 5 cuts.

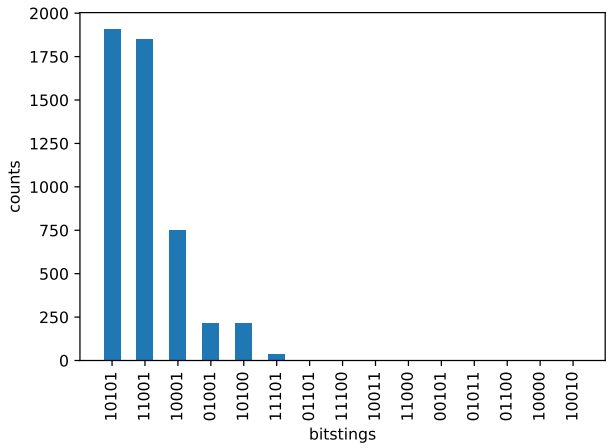

Fig. 13: Histogram output of the VQAA for the graph shown in figure 12.

More results with VQAA with different graph instances can be found here.

VI. Future upgrades

A. Weighted Max-cut

The more general version of the Max-cut problem is the weighted max-cut, where each edge is associated with its weight, and the objective is to maximize the total weight of the edges between the two subsets than the number of the edges. This problem can be encoded into a neutral atom device using different local detuning [9]. Although this is not yet available on PASQAL devices.

B. Register size

Bigger register size would enable to work on bigger graphs for real world applications of the max cut problem. Although this would come with a longer pre-processing time as the embedding into a unitary disk graph would be more complex.

C. Parameters estimation with Machine Learning

As shown in [3], we can use machine learning to learn the correct parameters for the pulses in the quantum evolution of the problem.

VII. Conclusions

The VQAA, compared with the QAOA, offers a correct solution every time it is run, but the problem is in the optimal parameters chosen by the optimizer; sometimes, the time evolution is too big. It is necessary to add a restriction for feasible parameters in real hardware. The QAOA obtains the correct answers with p = 4 layers, with 3 layers, sometimes a wrong answer is obtained. In both cases, we can observe in the histograms of the solutions that different combinations of the max cut in the graph can be seen with the higher counts.

REFERENCES

[1] dwave examples, "maximum-cut," 2023.

[2] Pasqal, "Pulser documentation," 2023.

[3] W. d. S. Coelho, M. D'Arcangelo, and L.-P. Henry, "Efficient protocol for solving combinatorial graph problems on neutral-atom quantum processors," *arXiv preprint arXiv:2207.13030*, 2022.

[4] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[5] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[6] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, *et al.*, "Quantum optimization of maximum independent set using rydberg atom arrays," *Science*, vol. 376, no. 6598, pp. 1209–1215, 2022.

[7] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, "Quantum optimization for maximum independent set using rydberg atom arrays," *arXiv preprint arXiv:1808.10816*, 2018.

[8] B. F. Schiffer, J. Tura, and J. I. Cirac, "Adiabatic spectroscopy and a variational quantum adiabatic algorithm," *PRX Quantum*, vol. 3, no. 2, p. 020347, 2022.

[9] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-T. Wang, and H. Pichler, "Quantum optimization with arbitrary connectivity using rydberg atom arrays," *PRX Quantum*, vol. 4, no. 1, p. 010316, 2023.