Solving Power Grid Optimization Problems with Rydberg Atoms

Nora Bauer,[1, *] Kübra Yeter-Aydeniz,[2, †] Elias Kokkas,[1, ‡] and George Siopsis[1, §]

[1]*Department of Physics and Astronomy, The University of Tennessee, Knoxville, TN 37996-1200, USA*
[2]*Emerging Technologies and Physical Sciences Department,*
The MITRE Corporation, 7515 Colshire Drive, McLean, Virginia 22102-7539, USA
(Dated: April 18, 2024)

The rapid development of neutral atom quantum hardware provides a unique opportunity to design hardware-centered algorithms for solving real-world problems aimed at establishing quantum utility. In this work, we study the performance of two such algorithms on solving MaxCut problem for various weighted graphs. The first method uses a state-of-the-art machine learning tool to optimize the pulse shape and embedding of the graph using an adiabatic Ansatz to find the ground state. We tested the performance of this method on finding maximum power section task of the IEEE 9-bus power system and obtaining MaxCut of randomly generated problems of size up to 12 on the Aquila quantum processor. To the best of our knowledge, this work presents the first MaxCut results on Quera's Aquila quantum hardware. Our experiments run on Aquila demonstrate that even though the probability of obtaining the solution is reduced, one can still solve the MaxCut problem on cloud-accessed neutral atom quantum hardware. The second method uses local detuning, which is an emergent update on the Aquila hardware, to obtain a near exact realization of the standard QAOA Ansatz with similar performance. Finally, we study the fidelity throughout the time evolution realized in the adiabatic method as a benchmark for the IEEE 9-bus power grid graph state.

I. INTRODUCTION

Neutral atom quantum computers are a promising candidate for quantum optimization and simulation, offering large numbers of qubits, flexible connectivity, and longer coherence times. Current hardware, such as QuEra's Aquila processor [1], offer analog control, where instead of implementing circuits comprised of discrete gates as in digital quantum computers, the user programs a time-dependent Hamiltonian to evolve the quantum system. The Aquila processor offers 256 Rb-87 atoms as physical qubits that can be configured in a 2-dimensional array, and their evolution is governed by Rabi frequency and detuning pulses, which can be programmed as time-dependent functions. The atoms experience a Rydberg interaction between excited states that depends on the physical distance between them.

Neutral atom quantum computers are also capable of acting as logical processors with quantum error correction. In Ref. [2], a neutral atom quantum system with 280 physical qubits was used to encode up to 48 logical qubits and 40 color code qubits. However, this procedure involved universal digital single-qubit gates, arbitrary connectivity, and mid-circuit readout, which are not publicly available to users on current hardware. In the near term, the constraint of a fixed Hamiltonian as well as the flexibility of atom placement offer a potential for efficient solutions to certain problems in quantum simulations and optimization. The Aquila processor can be used to engineer quantum states such as Bell states, cat and Green-

berger–Horne–Zeilinger (GHZ) states [3, 4], as well as antiferromagnetic spin states with domain wall defects [5, 6]. The Aquila processor has been used to prepare and study a two-dimensional quantum spin liquid with $\mathbb{Z}_2$ topological order and anyon condensation [7]. Additionally, neutral atom quantum computers have a native realization of the maximal independent set (MIS) combinatorial optimization problem [8, 9], where the ground state solution of the quantum Hamiltonian can be realized by optimizing an adiabatic evolution. The capabilities can be expanded to non-native independent set problems which can also be solved by optimizing a non-adiabatic pulse sequence [5]. Recent work using an adiabatic approach utilized a graph embedding optimization procedure and machine learning model to train the pulse parameters for solving combinatorial graph problems on a neutral atom quantum computer [10]. They also measured the Q-score benchmark value which quantifies the performances of a given device or method in solving a specific combinatorial optimization problem, such as the Maximum Cut (MaxCut) problem. In neutral atom quantum computers, state preparation is usually achieved by pulse shaping, i.e., optimizing the time-dependence of the Hamiltonian. This process can be extremely costly, as it requires a lot of sampling of the final state in the quantum processor (or in its emulator). Recent developments in software designed for analog computers expand capacities for differentiable circuit optimization with applications in solving differential equations and optimization problems [11].

Many real-world optimization problems can be mapped to weighted MaxCut problems. Power grid optimization [12] and EV charging station placement problems [13, 14] are of particular interest for quantum utility, and quadratic unconstrained binary optimization (QUBO) formulations have been solved using the quan-

* nbauer1@vols.utk.edu
† kyeteraydeniz@mitre.org
‡ ikokkas@vols.utk.edu
§ siopsis@tennessee.edu

tum approximate optimization algorithm (QAOA) on digital hardware. Weighted MaxCut problems are particularly suitable for neutral atom hardware because the atom placements can be adjusted to encode weights efficiently.

In order to compare across platforms, such as superconducting and ion trap quantum hardware, it is necessary to develop and test benchmarks on neutral atom quantum hardware. The Quantum Economic Development Consortium (QED-C) algorithmic benchmark set has been used to quantify the performance of simulated noisy neutral atoms on problems such as the quantum Fourier transform (QFT), Deutsch-Jozsa, and phase estimation algorithms [15, 16]. There has also been a benchmark developed for the generation of GHZ states on a neutral atom quantum computer [17, 18]. However, these studies use digital quantum gates rather than analog quantum computing. Benchmarks developed for analog computing [19, 20] involve measuring the fidelity of the experimental system compared to the noiseless system as a function of time throughout the evolution of the system.

In this work, we studied the performance of two different methods for solving weighted MaxCut problems corresponding to the power grid and EV charging optimization problems. The first method used pulse shaping of an adiabatic Ansatz via machine learning and a graph embedding algorithm that optimized the atom placements based on graph structures. We studied problem sizes up to 13 graph nodes and find over 95% probability of measuring the ground state solution on the graphs tested, in addition to reasonable scaling with the number of steps required to find the solution. We performed numerical simulations and also tested the performance on the Aquila hardware. The second method used QAOA with local detuning control on atoms, which is a near-term update on the Aquila hardware. Using this method, the detuning is not a free parameter anymore but depends on the geometry of the graph. By doing so, we could eliminate all the linear Z terms from the Hamiltonian of the neutral atoms, which gave a native realization of the weighted MaxCut cost function. We performed the QAOA for random graphs placed on square and honeycomb lattices up to 13 vertices, and solved the MaxCut problem with probability 83% in the best case scenario. We also performed a fidelity benchmarking test on the power grid optimization problem on the hardware.

Our discussion is organized as follows. In Section II, we introduce the power grid and EV charging optimization problems and their mapping to the weighted MaxCut problem. In Section III, we discuss two methods used to solve the optimization problems on analogue computers and provide experimental and numerical results. Section IV focuses on the neutral atom benchmarking procedure where we provide experimental results solving the optimization problems on the Aquila hardware. Finally in Section V, we summarize our results and provide future directions of research.

II. USE CASES

Electric power systems are critical in industrial societies. Operation of large transmission and distribution systems requires resource intensive computational processes. Optimization is one of these computational processes which plays an essential role in power systems operation ranging from transmission and distribution optimization for optimizing the routing and scheduling of electricity through the power grid to resilience optimization to recover from various types of disruptions, from equipment failures to natural disasters and cyber attacks.

Quantum optimization tools are expected to achieve super-polynomial advantage for solving combinatorial optimization problems. Hence, quantum optimization tools have been studied in literature for solving power system problems. One example is presented in Ref. [21] where they studied a power grid optimization problem on D-Wave quantum annealer's constrained and binary quadratic model solvers. To this end, they needed to map the problem onto a QUBO problem. However, the problem involves both equality and inequality constraints. They introduced a method to translate inequality constraints with real-valued coefficients into approximate penalty functions in order to reduce the number of required slack variables. They also partitioned the problem into smaller graph partitions and demonstrated that hybrid quantum-classical methods outperformed classical methods in terms of solution quality for the problem studied.

In this work, we identified our optimization problem of interest as solving weighted MaxCut problems and left other optimization problems in power systems as future work. Determining maximum power exchange sections is one of the most interesting real-life applications of weighted MaxCut problems. The maximum power section of a power grid system is an important index for identifying a system's power delivery capability [12], and in the case of a microgrid, it determines the dependence of each microgrid on other microgrids [14].

The maximum power sections problem has been studied in the literature using QAOA. The authors of Ref. [12] studied the IEEE 24-bus system and proposed a data-driven QAOA that yielded results comparable to those by the Goemans-Williamson algorithm. Similarly, in Ref. [13], they performed a dynamic analysis of a microgrids test system with 20 buses that included 7 EV charging stations using QAOA. By changing the power demands of these EV charging stations, they studied the maximum power sections. In Ref. [14], they switched their focus to a disturbance analysis of microgrids using QAOA. Inspired by this research, we studied maximum power section identification in IEEE 9- and 14-bus systems.

Power grid optimization systems can be solved by mapping the transmission system to a weighted graph and solving the weighted MaxCut problem [22]. The edge weights of the corresponding graph are given by the line

impedances as

$$w_{ij} = \begin{cases} \frac{1}{\sqrt{R_{ij}^2 + X_{ij}^2}} \ , & \text{nodes } i \text{ and } j \text{ connected} \\ 0 \ , & \text{otherwise} \end{cases} \quad (1)$$

where R_{ij} is the line resistance and X_{ij} is the line reactance. These are parameters that are provided in the IEEE test case datasets [23].

III. SOLVING MAXCUT PROBLEM ON ANALOG QUANTUM COMPUTERS

Analogue Hamiltonian simulation is an indispensable tool for the investigation of the features and behavior of a given quantum system. It is implemented by manipulating the parameters of another quantum system that bears similarities to the one being studied and minimizing the cost function that depends on these parameters [24]. The methods studied in the literature include, but are not limited to, using stochastic optimizers, and Bayesian optimization [25]. We identified two methods of solving the MaxCut problem, i.e., finding the maximum cut value of a graph of interest, on analog quantum computers: by (a) pulse shaping using machine learning, and (b) fine tuning of the local detuning parameters. In what follows, we discuss the details of these two methods.

The MaxCut problem is defined as follows. Given a graph $G = (\mathcal{V}, \mathcal{E})$, the graph is cut into two disjoint sets $\tilde{\mathcal{V}}$ and $\mathcal{V}'$ such that $\tilde{\mathcal{V}} \subset \mathcal{V}$ and $\mathcal{V}' = \mathcal{V} \backslash \tilde{\mathcal{V}}$. Then the subset of edges that correspond to these set of vertices are $\tilde{\mathcal{E}}$ and $\mathcal{E}'$. Similarly, the subset of edges are connected to the vertices as follows.

$$\tilde{\mathcal{E}} = \{\{i, j\} \in \mathcal{E} | i \in \tilde{\mathcal{V}} \oplus j \in \tilde{\mathcal{V}}\} \ , \quad (2)$$

where $\oplus$ is the logic operator exclusive or. Then the Max-Cut problem can be defined as a combinatorial optimization problem that aims to split the graph into two disjoint subsets such that the number of edges spanning the two subsets is maximized. This problem can be generalized to include weights w_{ij} on the edges $((i, j) \in \mathcal{E})$ and represented as a QUBO problem with the cost function defined as

$$\mathcal{C} = \sum_{(i,j) \in \mathcal{E}} w_{ij} Z_i Z_j \ , \quad (3)$$

for given weights w_{ij}. We will minimize this cost function by utilizing the evolution of a two-dimensional array of Rydberg atoms which is governed by the time-dependent Hamiltonian

$$\begin{aligned} H(t) &= \frac{\Delta_{\text{global}}}{2} \sum_{i=1}^{n} Z_i + \frac{\Omega}{2} \sum_{i=1}^{n} [X_i \cos \phi - Y_i \sin \phi] \\ &+ \frac{C_6}{4} \sum_{i<j} \frac{(Z_i - \mathbb{I})(Z_j - \mathbb{I})}{[(x_i - x_j)^2 + (y_i - y_j)^2]^3} \ , \quad (4) \end{aligned}$$

where $\Delta_{\text{global}}, \Omega, \phi$ are, in general, time-dependent parameters whose time evolution is controllable.

A. Pulse Shaping

The pulse shaping procedure finds the ground state by first finding an optimal register layout, then optimizing the amplitude and detuning pulses using the best layout. An adiabatic-type Ansatz [26] is chosen for the amplitude and detuning parameters as functions of time, where

$$\begin{aligned} \Omega(t, \boldsymbol{p}) &= p_0 \left(1 - \left[1 - \sin\left(\frac{\pi t}{t_{\max}}\right)^2\right]^{\frac{p_1}{2}}\right) \ , \\ \Delta_{\text{global}}(t, \boldsymbol{p}) &= \frac{2}{\pi} p_2 \tan^{-1}\left(p_1 \left[t - \frac{t_{\max}}{2}\right]\right) \ , \quad (5) \end{aligned}$$

and $\boldsymbol{p} = (p_0, p_1, p_2)$ is a vector of hyperparameters for the optimization. We set $t_{\max} = 4 \ \mu$s. The amplitude $\Omega(t, \boldsymbol{p})$ is the hopping parameter, and must start and end at 0 MHz on the Aquila hardware. The detuning $\Delta(t, \boldsymbol{p})$ is the onsite operator, and is turned on gradually to slowly evolve the system from $|0\rangle^{\otimes N}$ to the MaxCut ground state. These functions before and after optimization for the IEEE 9-bus problem are shown in Fig. 1. Note that there is also a maximum range of the detuning and amplitude, where $0 \leq \Omega \leq 2.51$ MHz, and $|\Delta| \leq 19.89$ MHz, and these bounds are enforced in the optimization.
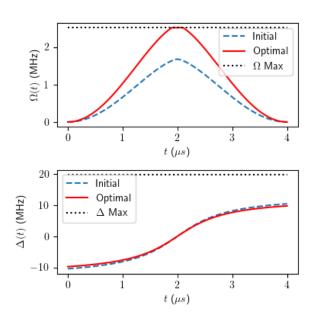

FIG. 1: Amplitude (Ω) and detuning (Δ) as functions of time with default parameters (blue dashed line) and optimized parameters (red solid line) for the IEEE 9-bus problem. The black dotted line denotes the maximal amplitude and detuning values currently allowed on QuEra's Aquila processor.

After choosing a set of default hyperparameters, N_R random layouts of graph vertex positions are chosen and the initial cost function is measured after evolving with the adiabatic evolution using the default parameters.

Then, the layout that produces the lowest cost function is chosen for the pulse shape optimization step. The random layouts are found using an adaptation of the Fruchterman-Reingold (FR) layout [27].

The graph embedding into the two-dimensional plane heavily influences the ability to find the ground state. Since the quantum Hamiltonian corresponding to the MaxCut problem is minimized when the atoms with edges between them are in opposite basis states, the graph embedding should place atoms with edges between them close together (based on edge weight) and atoms without edges between them far apart. The FR force directed algorithm [27] finds the optimal node placement by relaxing a classical system with a repulsive force between nodes, f_r, and an attractive force between nodes with edges, f_a, given, respectively, by

$$f_{ij}^r(d_{ij}) = -k^2/d_{ij} \ , \ f_{ij}^a(d_{ij}) = d_{ij}^2/k \ , \qquad (6)$$

where d_{ij} is the distance between nodes i and j, and k is the optimal distance between nodes, with $k \propto \sqrt{\text{Area}/\#\text{vertices}}$. In this work, we introduce a variant of the FR algorithm that mimics the r^{-6} scaling of the Rydberg interaction and also considers graph structures, where

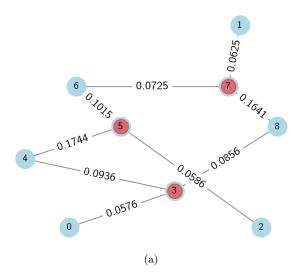$$f_{ij}^r(d_{ij}) = -k_{ij}^6/d_{ij}^6 \ , \ f_{ij}^a(d_{ij}) = d_{ij}^2/k_{ij} \ , \qquad (7)$$

Here, k_{ij} is now local and dependent on the vertices i and j, and can be determined by graph structure parameters such as connectivity, diameter, and cliques. We obtained the best results by determining k_{ij} at each pair (i, j) based on whether the pair (i, j) is part of a 3-node clique, or triangle. We used the form

$$k_{ij} \propto \left(\frac{1}{N_\Delta} \sum_\Delta \{1 | i, j \in \Delta\} \right)^\rho \ , \qquad (8)$$

with the exponent ρ a hyperparameter that was optimized.

The pulse optimization was performed by optimizing the hyperparameters $\boldsymbol{p}$ using Pennylane's Nadam optimizer [28]. The optimization steps were simulated using Pennylane's Jax interface, and the optimized pulse shapes were then run on QuEra's Aquila device with 1000 shots [26].

For the IEEE dataset maximum power section optimization problems, the problem can be mapped to a weighted MaxCut problem. The weighted graph corresponding to the IEEE 9-bus problem is shown in Fig. 2(a). The solution is given by the node color, where the red and blue nodes correspond to different sets. The experiment is performed using the simulator as well as the Aquila hardware with $N_R = 50$, and the histograms are shown in Fig. 2(b). We obtained the probability of measuring the maximum cut (overlap with the ground state), $P(GS) = 0.939$ for the noiseless simulation, and $P(GS) = 0.639$ using quantum hardware. Thus, despite degradation of performance, it was still possible to find
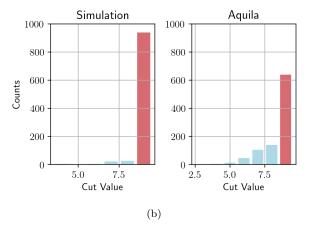

(a)

(b)

FIG. 2: (a) IEEE 9-bus dataset graph where each node represents a bus, weights correspond to the impedance calculated from dataset using $Z = \sqrt{X^2 + R^2}$. The Max-Cut solution for the dataset presented in left panel, where the red and blue nodes correspond to different sets (b) Simulated and experimental histograms for the IEEE 9-bus problem after performing the optimization procedures.

the solution to the problem using the Aquila quantum hardware.

In order to study the scaling of the performance with graph size, we also tested our method on Erdos-Rényi graphs with $p = 0.25$ and weights selected from the uniform distribution $[0, 0.25]$, which we found to be similar in complexity to the graphs corresponding to the IEEE bus problems. We chose $N_R = 50$ optimizer steps and computed the average probability of measuring the maximum cut state $P(GS)$ for graphs with $8 \leq N \leq 12$ vertices. Results are given for the ideal case and experimental results with standard error bars, shown in Fig. 3(a). We find the probability of measuring the solution $> 95\%$ for the noiseless simulations, and the average probability of measuring the solution is 60% on Aquila quantum hard-

ware.

We also computed the average step to solution S, given as

$$S = \frac{\log(0.01)}{\log(1 - \bar{P}(GS))} \, , \quad (9)$$

which gives the average number of register layout optimization steps required to obtain the solution with 99% certainty, shown in Fig. 3(b). For the graphs tested in the range $10 \leq N \leq 13$, we see an approximately linear scaling in the amount of register layout steps required to find the solution.

B. Local Detuning

Our second approach to the MaxCut problem was by using the QAOA. In the analog quantum computer of neutral atoms, we define the cost and mixer time evolution operators by fine-tuning the detuning, the Rabi frequency and the laser phase. The solution to the problem can be improved by assuming access to local detuning. In this case, we can re-write the Hamiltonian, Eq. (4), as

$$H = \sum_{i=1}^{n} \left(\frac{\Delta_i}{2} - \sum_{j \neq i} w_{ij} \right) Z_i + \sum_{i<j} w_{ij} Z_i Z_j$$
$$+ \frac{\Omega(t)}{2} \sum_{i=1}^{n} [X_i \cos \phi(t) - Y_i \sin \phi(t)] \, , \quad (10)$$

with distance-dependent weights given by

$$w_{ij} = \frac{C_6}{4 \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right]^3} \, . \quad (11)$$

Notice that by setting the local detuning parameters to $\Delta_i = 2 \sum_{j \neq i} w_{ij}$, we can completely eliminate the linear Z terms from Eq. (10). These values of local detuning parameters will be assumed in the following discussion.
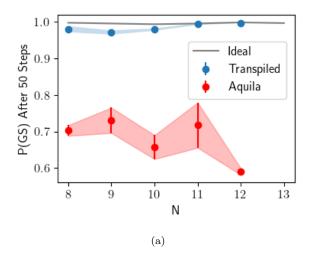
To apply the QAOA, we prepared the atoms in the ground state $|0\rangle^{\otimes n}$ and let the system evolve for time $t_0 = \frac{\pi}{2\Omega_0}$ while fixing $\phi(t) = -\frac{\pi}{2}$ and $\Omega(t) = \Omega_0$. With these choices, the system's Hamiltonian became

$$H_0 = \frac{\Omega_0}{2} \sum_{i=1}^{n} Y_i + \sum_{i<j} w_{ij} Z_i Z_j \, . \quad (12)$$

After time $t = t_0$, the state of the neutral atoms evolved from the product state $|0\rangle^{\otimes n}$, to the uniform superposition $|+\rangle^{\otimes}$, to a good approximation. This is because the amplitude was chosen to be $\Omega_0 = 5\pi$ MHz, whereas the maximum weight was $w = 0.45$ MHz, corresponding to minimum distance between two vertices (nearest neighbors) $\alpha = 12 \ \mu$m, therefore $w_{ij} \ll \Omega_0$.

Next, we introduce the cost Hamiltonian by setting $\Omega(t) = 0$

$$H_c = \sum_{i<j} w_{ij} Z_i Z_j \, , \quad (13)$$
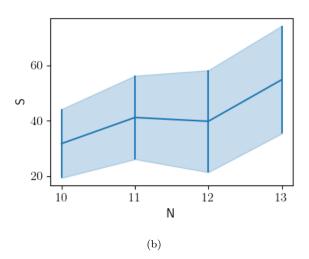

(a)

(b)

FIG. 3: (a) Probability of measuring the ground state after $N_R = 50$ steps for $8 \leq N \leq 12$ vertex graphs, where the grey solid line represents the ideal result, the blue dots represent the result after the ideal evolution is fit within the hardware parameters, and the red dots represent the experimental results measured on the Aquila processor. The bars denote standard error from averaging over random Erdos-Rényi graphs. (b) Step to solution S as defined in Eq. (9) for $10 \leq N \leq 13$ vertex graphs. The bars denote standard error from averaging over random Erdos-Rényi graphs.

and the mixer Hamiltonian by choosing $\Omega(t) = 2$ MHz (which was the best performing value for smaller graphs) and $\phi(t) = 0$

$$H_b = \sum_{i=1}^{n} X_i + \sum_{i<j} w_{ij} Z_i Z_j \, . \quad (14)$$

A single QAOA layer is implemented by time evolution under the cost Hamiltonian for $t_1 = \gamma_1$ and mixer Hamiltonian for time $t_2 = \beta_1$. After applying p layers we obtain

the state

$$|\psi_p\rangle = e^{-iH_b\beta_p}e^{-iH_c\gamma_p}\cdots e^{-iH_b\beta_1}e^{-iH_c\gamma_1}|\psi_0\rangle \ . \quad (15)$$

At the end, we minimize the value of the cost function,

$$\mathcal{C}_p = \langle\psi_p|\mathcal{C}|\psi_p\rangle \ , \quad (16)$$

where the cost function is defined in Eq. (3). It is important to understand that there is a difference between the cost function (Eq. (3)) we are trying to minimize and a similar set of terms in the Hamiltonian (10) we are utilizing. The latter includes all the interactions of the n body quantum system which fall off rapidly according to the van der Waals law, $\sim r^{-6}$. The sum in the cost function has only terms that correspond to an edge on the graph we are interested in. Because of this, the performance of QAOA depends significantly on the position of each atom and how well they represent the graph, including weights, on which the cost function is defined.

Instead of trying to find the optimal position of atoms based on the FR algorithm, as indicated above, we considered the placement of the atoms at fixed lattice points. Using this technique, we solved the MaxCut problem for random graphs with a number of vertices $\in [10, 13]$ generated from square and honeycomb lattices, and for the IEEE 9-bus dataset. Specifically, for the random graphs, we considered two graphs for each lattice geometry.

For numerical results, we calculated the cost function following the QAOA protocol and then used the Adam Optimizer to find the optimal values of β_i and γ_i in (15). For each graph, we considered 20 different initial seeds on these parameters and up to 2000 optimization steps. For the IEEE 9-bus graph, we find an average probability of measuring the ground state (which corresponds to the MaxCut) equal to $P(GS) = 0.984$. We also evaluated the average probability to find the MaxCut for the vanilla QAOA [29], in which there is no distance dependence and found $P(GS) = 0.99$. These results are illustrated in Fig. 4. Finally, for the random graphs on the lattice found on average probability of measuring the ground state that decrease from $P(GS) = 0.67$ to $P(GS) = 0.5$ as we increase the number of vertices. However, on larger graphs these results depend stronger on the initial random seed. Because of this we have outliers where the optimization fails to converge for specific seeds. It is worth mentioning that in the best case scenario, there are initial seeds that give probability which decreases from $P(GS) = 0.956$ to $P(GS) = 0.838$. These results are summarized in Fig. 5.

IV. FIDELITY ESTIMATION ON QUERA

In this section, we present the fidelity estimation of the graph states that are studied in Section II. Fidelity can be utilized as an important benchmark for studying quantum systems and can be defined as

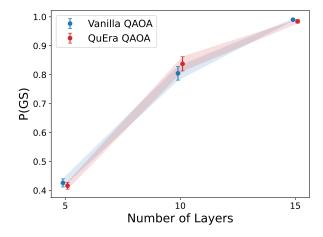$$F = \langle\psi|\rho_{\exp}|\psi\rangle \ , \quad (17)$$

FIG. 4: Probability of measuring the ground state for the IEEE 9-bus graph. For both simulations (Vanilla QAOA and QuEra) we calculated the probability by applying 5, 10 and 15 layers. In each case, we started with 1000 optimization steps and increased then up to 2000.
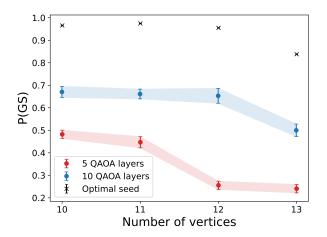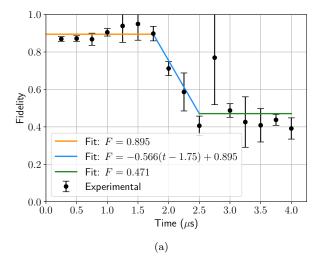

FIG. 5: Probability of measuring the ground state for $10 < N < 13$ vertex graph's by simulating QuEra's hardware. For 5 layers optimized the cost function up to 1000 steps using the Adam Optimizer, whereas for 10 layers we considered up to 2000 steps.

where $\rho_{\exp}$ is the experimental mixed state. On digital quantum computers, there is more than one way of computing the fidelity, i.e., using quantum state tomography, quantum process tomography, randomized benchmarking [30], or cross entropy benchmarking [31]. On the other hand, there are also various protocols developed for fidelity estimation on analogue quantum computers, i.e., quantum simulators [19, 20, 32].

We conducted a fidelity estimation using the following equation from Ref. [32]:

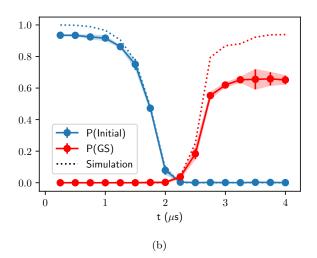$$F = 2\frac{\sum_z p_0(z)p(z)}{\sum_z p_0^2(z)} - 1 \ , \quad (18)$$

(a)

(b)

FIG. 6: (a) Fidelity values calculated in 16 time cycles where each time cycle is 0.25 μs. The experiments were conducted three times to obtain the error bars. (b) Probability of measuring the initial state $|0\rangle^{\otimes N}$ (blue points) and probability of measuring the ground state (red points) as a function of time, as measured on hardware. The dotted lines correspond to the values measured in simulation. Error bars represent 3σ at each time step.

where $p_0(z)$ is the theoretical and $p(z)$ the experimental probability values for each bit string. For experimental probability values we followed the protocol in Ref. [20] which can be summarized as follows:

Experiment:

1. Prepare an initial step ρ_0, which approximates a pure state $|\psi_0\rangle \langle\psi_0|$.

2. Evolve the system under its natural Hamiltonian H for a time t.

3. Measure the evolved state $\rho(t)$ in a natural basis, obtaining configurations $\{z_1, \ldots, z_M\}$.

In order to obtain the experimental probability values, we prepared the IEEE 9-bus dataset graph in the initial state $|\psi_0\rangle = |0\rangle^{\otimes N}$, as seen in Fig. 2, and time evolved it in 16 time cycles where each time cycle was $t_{\text{cycle}} = 0.25$ μs. We measured the time evolved state in the computational basis at each time cycle. We used $n_{\text{shots}} = 1000$ for our experiments run on QuEra's Aquila neutral atom quantum computer. On the other hand, for theoretical probability values, $p_0(z)$, we ran a noiseless quantum simulator with $n_{\text{shots}} = 100,000$. We obtained the fidelity estimation values depicted in Fig. 6(a). Evidently, the fidelity remains relatively constant in the first ≈ 1.75 μs, then decays and settles into another nearly constant value after $t \approx 2.5$ μs. This matches the behavior of the detuning and amplitude (see Fig. 1) as a function of time: for the first 2 μs, the detuning has a negative value and the amplitude increases slowly whereas for $t > 2$ μs, the detuning has positive value and the amplitude decreases down to zero at $t = 4$ μs. Correspondingly, the probability of overlap with the initial and ground states, shown in Fig. 6(b), changes around $t = 2$ μs. A fit of the experimental data on the fidelity (Fig. 6(a)) reveals that for $t < 2$ μs, the fidelity stays fairly constant at 89.5%. During this time the system remains approximately in the initial state which is not entangled. As it undergoes entanglement, the fidelity decreases down to 47.1% and remains fairly constant for $t > 2$ μs. During this time, the system has large overlap with the ground state which is also not entangled. Thus, we observe a degradation in fidelity during the evolution of the system that involves an increase in entanglement. This result is an important benchmark of the neutral atom hardware. It should be pointed out that despite degradation, we still attain our goal, as we obtain the solution to the MaxCut problem with 60% probability.

V. CONCLUSION

In this work, we investigated the performance of two methods we introduced for solving power grid optimization problems mapped onto weighted MaxCut problems using neutral atom quantum hardware.

In the first method, we used a Rydberg interaction inspired graph embedding algorithm and Pennylane's Jax machine learning interface to optimize the pulse shape of an adiabatic Ansatz. We found that with under 100 embedding steps, the adiabatic algorithm was able to find the ground state with over 95% overlap. The optimized pulse shapes were also tested on QuEra's Aquila processor, and we found an average 60% overlap with the ground state experimentally. Although far from ideal, the performance of the QuEra processor was adequate to yield the solution to the problem at hand.

In the second method, we applied the QAOA on graphs which can be embedded into square or honeycomb lattices. Using the Adam Optimizer for the classical optimization and up to 10 QAOA layers we were able to find

the MaxCut on average with probability ranging from 67% (for 10 vertices) to 50% (for 13 vertices).

We also measured the experimental fidelity of the adiabatic evolution to the ground state for the IEEE 9-bus problem on the Aquila processor to be utilized as a benchmark.

With the improvements in neutral atom hardware, it is important to benchmark the algorithmic performance in order to compare with other quantum hardware. Schemes have been devised for quantum reservoir computing with neutral atoms [33], using the native Hamiltonian for computation in a quantum recurrent neural network. Utilizing neutral atom hardware is a promising avenue for reservoir computing because the Rydberg interaction allows for a high level of entanglement, which in turn allows for more complex reservoir dynamics and a longer short-term memory [34]. There have also been proposals for the quantum kernel method benchmarking on neutral atom quantum computers [35], which involves measuring the similarity between graph-structured data.

Since our optimization was performed on a simulator instead of the Aquila hardware, it is important to study the performance when the optimization is performed on actual hardware. Bayesian optimization has been studied in optimizing pulse shapes in neutral atom hardware for the MIS problem [36], so it would be interesting to compare the performance of Bayesian optimization with the gradient descent NAdam algorithm on MaxCut problems. Moreover, it is important to study error mitigation techniques specialized for analog processors, such as dual-state purification [37].

Additionally, the emergence of local detuning on analog neutral atoms systems increases the possibilities for quantum algorithms on neutral atom quantum computers. Local detuning has been shown to allow for single and two-qubit digital gates, SWAP networks, and analog variational quantum eigensolver (VQE) algorithms [38]. Using digital gates, there have also been proposals for quantum machine learning on neutral atom hardware, where the trainable Ansatz uses both digital and analog evolution, and numerical experiments are performed on handwritten digit classification and quantum phase boundary learning. It was found that the digital-analog schemes require shorter depth and are more robust to noise [39]. It is important to apply these tools to real-life applications, such as power grid optimization and placement of EV charging stations that were discussed here, in search of quantum utility.

[1] J. Wurtz, A. Bylinskii, B. Braverman, J. Amato-Grill, S. H. Cantu, F. Huber, A. Lukin, F. Liu, P. Weinberg, J. Long, S.-T. Wang, N. Gemelke, and A. Keesling, Aquila: Quera's 256-qubit neutral-atom quantum computer (2023), arXiv:2306.11727 [quant-ph].

[2] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).

[3] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, *et al.*, Generation and manipulation of schrödinger cat states in rydberg atom arrays, Science **365**, 570 (2019).

[4] Generating highly-entangled states in large Rydberg-atom arrays, https://docs.q-ctrl.com/boulder-opal/application-notes/generating-highly-entangled-states-in-large-rydberg-atom-arrays (2024).

[5] J. Balewski, M. Kornjaca, K. Klymko, S. Darbha, M. R. Hirsbrunner, P. Lopes, F. Liu, and D. Camps, Engineering quantum states with neutral atoms (2024), arXiv:2404.04411 [quant-ph].

[6] Improving Z2 state generation by 3X on QuEra's Aquila QPU, https://docs.q-ctrl.com/boulder-opal/application-notes/improving-z2-state-generation-by-3x-on-queras-aquila-qpu (2024).

[7] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, A. Omran, S. Sachdev, A. Vishwanath, M. Greiner, V. Vuletić, and M. D. Lukin, Probing topological spin liquids on a programmable quantum simulator, Science **374**, 1242 (2021).

[8] 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 **376**, 1209 (2022).

[9] M. Kim, K. Kim, J. Hwang, E.-G. Moon, and J. Ahn, Rydberg quantum wires for maximum independent set problems, Nature Physics **18**, 755 (2022).

[10] W. da Silva Coelho, M. D'Arcangelo, and L.-P.

Henry, Efficient protocol for solving combinatorial graph problems on neutral-atom quantum processors (2022), arXiv:2207.13030 [quant-ph].

[11] D. Seitz, N. Heim, J. P. Moutinho, R. Guichard, V. Abramavicius, A. Wennersteen, G.-J. Both, A. Quelle, C. de Groot, G. V. Velikova, V. E. Elfving, and M. Dagrada, Qadence: a differentiable interface for digital-analog programs (2024), arXiv:2401.09915 [quant-ph].

[12] H. Jing, Y. Wang, and Y. Li, Data-driven quantum approximate optimization algorithm for power systems, Communications Engineering **2**, 12 (2023).

[13] H. Jing, Y. Wang, Y. Li, L. Du, and Z. Wu, Dynamics analysis of microgrids integrated with ev charging stations based on quantum approximate optimization algorithm, in *2022 IEEE Transportation Electrification Conference & Expo (ITEC)* (IEEE, 2022) pp. 574–578.

[14] H. Jing, Y. Wang, Y. Li, L. Du, and Z. Wu, Quantum approximate optimization algorithm-enabled der disturbance analysis of networked microgrids, in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)* (IEEE, 2022) pp. 1–5.

[15] N. Wagner, C. Poole, T. M. Graham, and M. Saffman, Benchmarking a neutral-atom quantum computer (2023), arXiv:2311.04141 [quant-ph].

[16] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen, N. Franco, F. Fratini, B. Fuller, J. Gacon, C. Gonciulea, S. Gribling, S. Gupta, S. Hadfield, R. Heese, G. Kircher, T. Kleinert, T. Koch, G. Korpas, S. Lenk, J. Marecek, V. Markov, G. Mazzola, S. Mensa, N. Mohseni, G. Nannicini, C. O'Meara, E. P. Tapia, S. Pokutta, M. Proissl, P. Rebentrost, E. Sahin, B. C. B. Symons, S. Tornow, V. Valls, S. Woerner, M. L. Wolf-Bauwens, J. Yard, S. Yarkoni, D. Zechiel, S. Zhuk, and C. Zoufal, Quantum optimization: Potential, challenges, and the path forward (2023), arXiv:2312.02279 [quant-ph].

[17] T. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemeyer, *et al.*, Multi-qubit entanglement and algorithms on a neutral-atom quantum computer, Nature **604**, 457 (2022).

[18] A. L. Shaw, Z. Chen, J. Choi, D. K. Mark, P. Scholl, R. Finkelstein, A. Elben, S. Choi, and M. Endres, Benchmarking highly entangled states on a 60-atom analogue quantum simulator, Nature **628**, 71 (2024).

[19] A. L. Shaw, Z. Chen, J. Choi, D. K. Mark, P. Scholl, R. Finkelstein, A. Elben, S. Choi, and M. Endres, Benchmarking highly entangled states on a 60-atom analog quantum simulator, arXiv preprint arXiv:2308.07914 (2023).

[20] D. K. Mark, J. Choi, A. L. Shaw, M. Endres, and S. Choi, Benchmarking quantum simulators using ergodic quantum dynamics, Physical Review Letters **131**, 110601 (2023).

[21] G. Colucci, S. van der Linde, and F. Phillipson, Power network optimization: a quantum approach, IEEE Access (2023).

[22] H. Tong, R. C. Qiu, D. Zhang, H. Yang, Q. Ding, and X. Shi, Detection and classification of transmission line transient faults based on graph convolutional neural network, CSEE Journal of Power and Energy Systems **7**, 456 (2021).

[23] MATPOWER Index, https://matpower.org/docs/ref/menu5.0.html (2015).

[24] AWS, Analog hamiltonian simulation with pennylane, https://github.com/amazon-braket/amazon-braket-examples/blob/main/examples/analog_hamiltonian_simulation/06_Analog_Hamiltonian_simulation_with_PennyLane.ipynb (2024).

[25] J. R. Finžgar, M. J. Schuetz, J. K. Brubaker, H. Nishimori, and H. G. Katzgraber, Designing quantum annealing schedules using bayesian optimization, arXiv preprint arXiv:2305.13365 (2023).

[26] Introduction to Aquila, https://github.com/amazon-braket/amazon-braket-examples/blob/main/examples/analog_hamiltonian_simulation/01_Introduction_to_Aquila.ipynb (2024).

[27] T. M. J. Fruchterman and E. M. Reingold, Graph drawing by force-directed placement, Software: Practice and Experience **21**, 1129 (1991).

[28] Analog Hamiltonian simulation with PennyLane , https://github.com/amazon-braket/amazon-braket-examples/blob/main/examples/analog_hamiltonian_simulation/06_Analog_Hamiltonian_simulation_with_PennyLane.ipynb (2023).

[29] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).

[30] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Randomized benchmarking of quantum gates, Phys. Rev. A **77**, 012307 (2008).

[31] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[32] J. Choi, A. L. Shaw, I. S. Madjarov, X. Xie, R. Finkelstein, J. P. Covey, J. S. Cotler, D. K. Mark, H.-Y. Huang, A. Kale, *et al.*, Preparing random states and benchmarking with many-body quantum chaos, Nature **613**, 468 (2023).

[33] R. A. Bravo, K. Najafi, X. Gao, and S. F. Yelin, Quantum reservoir computing using arrays of rydberg atoms, PRX Quantum **3**, 030325 (2022).

[34] N. Götting, F. Lohof, and C. Gies, Exploring quantumness in quantum reservoir computing, Phys. Rev. A **108**, 052427 (2023).

[35] L.-P. Henry, S. Thabet, C. Dalyac, and L. Henriet, Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits, Phys. Rev. A **104**, 032416 (2021).

[36] J. R. Finžgar, M. J. A. Schuetz, J. K. Brubaker, H. Nishimori, and H. G. Katzgraber, Designing quantum annealing schedules using bayesian optimization (2023), arXiv:2305.13365 [quant-ph].

[37] Y. Shingu, T. Nikuni, S. Kawabata, and Y. Matsuzaki, Quantum annealing with error mitigation (2022), arXiv:2210.08862 [quant-ph].

[38] C. Chevallier, J. Vovrosh, J. de Hond, M. Dagrada, A. Dauphin, and V. E. Elfving, Variational protocols for emulating digital gates using analog control with always-on interactions (2024), arXiv:2402.07653 [quant-ph].

[39] J. Z. Lu, L. Jiao, K. Wolinski, M. Kornjača, H.-Y. Hu, S. Cantu, F. Liu, S. F. Yelin, and S.-T. Wang, Digital-analog quantum learning on rydberg atom arrays (2024), arXiv:2401.02940 [quant-ph].