Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (4) 440–460
http://journals.rudn.ru/miph

Physics and Astronomy

Research article

UDC 519.6, 530.145.67
PACS 03.67.Ac, 03.67.Lx,
DOI: 10.22363/2658-4670-2025-33-4-440-460
EDN: HYWEXV

Simulating QAOA operation using Cirq and qsim quantum frameworks

Yuri G. Palii, Alla A. Bogolubskaya, Denis A. Yanovich

Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: August 12, 2025; revised: September 2, 2025; accepted: September 10, 2025)

Abstract. The problem of finding the lowest-energy state in the Ising model with a longitudinal magnetic field is studied for two- and three-dimensional lattices of various sizes using the Quantum Approximate Optimization Algorithm (QAOA). The basis states of the quantum computer register correspond to spin configurations on a spatial lattice, and the Hamiltonian of the model is implemented using a sequence of quantum gates. The average energy value is efficiently measured using the Hadamard test. We simulate the QAOA operation on increasingly complex lattice configurations using the software libraries `Cirq` and `qsim`. The results of optimization, obtained using gradient-based and gradient-free methods, demonstrate the superiority of the latter in both modeling performance and quantum computer usage. Key arguments in favor of the advantages of quantum computation for this problem are presented.

Key words and phrases: quantum computing, QAOA, Ising model, quantum simulation, optimization, Cirq, qsim, cuStateVec

For citation: Palii, Y. G., Bogolubskaya, A. A., Yanovich, D. A. Simulating QAOA operation using Cirq and qsim quantum frameworks. *Discrete and Continuous Models and Applied Computational Science* **33** (4), 440–460. doi: 10.22363/2658-4670-2025-33-4-440-460. edn: HYWEXV (2025).

1. Introduction

The phenomenon of quantum entanglement (that is, the existence of special correlations between the states of particles in the microcosm that are not described by classical physics) makes the simulation of a system of hundreds of such particles intractable for supercomputers without significant simplifications. At the same time, for a quantum computer that uses quantum bits (qubits), entanglement serves as a resource for algorithms that are much more efficient than classical ones. An important example of this is quantum field theory on a space-time lattice, where classical computations struggle due to issues such as the Monte Carlo sign problem [1]. There are already quantum algorithms offering polynomial or even exponential speedups compared to their classical counterparts. It is widely believed that quantum computers will lead to breakthroughs in high-energy physics [2], quantum chemistry, and other fields facing computationally challenging problems [3, 4].

Due to the imperfections of quantum computers, the focus today is on hybrid quantum-classical algorithms that implement variational procedures. A quantum computer is used to construct a wave function (ansatz) depending on variational parameters and to measure observables for the modeled system. A classical computer then finds a new set of parameters for quantum gates by exploiting some optimization procedure in order to achieve the desired value of the cost function, which is one of the observables [5]. The theoretical foundation of such algorithms is the quantum mechanical Rayleigh–Ritz variational principle, which states that, for any trial wave function $|\psi(\boldsymbol{\alpha})\rangle$ with parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$, the average value of the Hamiltonian is not less than the ground-state energy: $\langle\psi(\boldsymbol{\alpha})|\, \mathcal{H}\, |\psi(\boldsymbol{\alpha})\rangle \geq E_0$.

Among hybrid algorithms, one of the most promising is considered to be the Quantum Approximate Optimization Algorithm (QAOA) [6], which is the focus in this paper. It has the property of universality due to its adaptability to a specific task, since its Hamiltonian is employed to prepare the variational wave function ansatz. Moreover, since QAOA is a problem-specific ansatz, it avoids the *barren plateau problem* (a common challenge in parameter optimization) [4]. There are arguments in favor of the algorithm's ability to demonstrate the quantum advantage.

The problem of finding the ground state energy and the corresponding state vector is crucial for the theory of phase transitions and critical phenomena in quantum systems. The Ising model, in its various formulations, along with other spin systems models, serves as a starting point for numerous applications in statistical physics, lattice field theory, as well as quantum technologies (e.g., material science and device engineering). Notably, finding the ground state for the Ising model constitutes an NP-hard computational problem. The QAOA algorithm has been widely used to study ground states in models of magnetic materials [7], with analytical approaches developed for improved parameter optimization [8].

This paper provides a concise introduction to the application of the QAOA for the aforementioned problem. Drawing upon resources from Google–Quantum AI's documentation of the `Cirq` quantum computing library [9], we present numerical simulations of spin lattice models in 2D and 3D, with the latter being non-integrable. Our results demonstrate increasing accuracy of the energy estimation when we enlarge the number of the ansatz layers. Furthermore, we implement the Hadamard test to enhance the efficiency of the average energy measurements.

2. QAOA Ansatz for the Ising Model

2.1. Ising Model on a Quantum Computer

Each lattice site corresponds to a qubit in the quantum computer's register. An arbitrary distribution of spin values across the lattice sites can be represented as a binary string $z = z_1 z_2 \dots z_n$, where each variable z_i specifies the spin orientation at the i-th site and takes values $z_i = \pm 1$. The value of z_i corresponds to the measurement outcome of the Pauli operator $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ acting on the i-th qubit in the computational basis:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \leftrightarrow z_i = +1, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \leftrightarrow z_i = -1.$$

In this way the 2^n basis states of a quantum register are in one-to-one correspondence with the 2^n possible spin configurations on the lattice:

$$\text{bit string } z = z_1 z_2 \dots z_n \;\longleftrightarrow\; \text{quantum register } |z\rangle = |z_1 z_2 \dots z_n\rangle.$$

An arbitrary state $|\psi\rangle$ of the quantum register represents a superposition of all possible spin configurations on the lattice with different amplitudes. The Ising model Hamiltonian is constructed from multi-qubit operators $Z^{(i)}$,

$$Z^{(i)} = \mathbb{I} \otimes \cdots \otimes \underset{i\text{-th position}}{\underset{\uparrow}{Z}} \otimes \cdots \otimes \mathbb{I},$$

where the Pauli-Z operator acts only on the i-th qubit. For nearest-neighbor spin-spin interactions with the coupling constant J and interaction with an external magnetic field h, the Hamiltonian takes the form:

$$\mathcal{H}(Z) = -J \sum_{\langle i,j\rangle} Z^{(i)} Z^{(j)} - h \sum_{i} Z^{(i)}, \tag{1}$$

where $\langle i, j\rangle$ denotes the set of nearest-neighbor spin pairs, and the second term sums over all lattice sites. The expectation value of the Hamiltonian serves as the cost function in the variational minimization procedure.

2.2. Driver and Mixer Operators in the Ansatz

The variational ansatz $|\psi(\boldsymbol{\alpha})\rangle$ for the QAOA implementation ([5], Chapter 9.3) consists of multiple identical layers of quantum operations. Each layer contains two operators: a *driver* and a *mixer*, which act sequentially on the quantum register.

First, a uniform superposition state is prepared from the state $|0\rangle^{\otimes n}$ by applying Hadamard gates H to each of the n qubits in the register:

$$\left|\psi_0^{(n)}\right\rangle = \mathsf{H}^{\otimes n}\,|0\rangle^{\otimes n} = \frac{1}{2^{n/2}} \sum_{z\in\{0,1\}^n} |z\rangle, \quad \mathsf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2}$$

As a result, all possible spin configurations have equal probability amplitudes.

- The driver operator represents the evolution operator corresponding to the Hamiltonian $\mathcal{H}(Z)$ (with $J = 1$):

$$U(\gamma, \mathcal{H}) = e^{i\pi\gamma\mathcal{H}/2} = \prod_{\langle j,k\rangle} e^{-i\pi\gamma Z^{(j)} Z^{(k)}/2} \prod_{l} e^{-i\pi\gamma h Z^{(l)}/2}, \quad i = \sqrt{-1}, \tag{3}$$

where the variational parameter γ acts as an effective evolution time.

- The mixer operator is constructed from Pauli-X operators:

$$U(\beta, B) = e^{i\pi\beta B/2} = \prod_{j=1}^{n} e^{i\pi\beta X^{(j)}/2}, \quad \text{where } B = \sum_{j=1}^{n} X^{(j)}. \tag{4}$$

Here β serves as the second variational parameter in the ansatz layer. Thus, the p-layer QAOA ansatz looks as follows:

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \underbrace{U(\beta_p, B) U(\gamma_p, \mathcal{H})}_{p} \cdots \underbrace{U(\beta_1, B) U(\gamma_1, \mathcal{H})}_{1} |\psi_0^{(n)}\rangle. \tag{5}$$

From the theorem proved in [6], as the number of layers p increases, the minimal expectation value $E_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ of the Hamiltonian $\mathcal{H}$ found by the QAOA converges to the minimal value $\min_z \mathcal{H}(z)$ among all possible bit strings z:

$$\lim_{p\to\infty} \min_{\boldsymbol{\gamma},\boldsymbol{\beta}} E_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \min_{z} \mathcal{H}(z), \quad E_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \equiv \left\langle \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \right| \mathcal{H} \left| \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \right\rangle. \tag{6}$$

In practice, the number of layers p is finite, making the algorithm approximate.

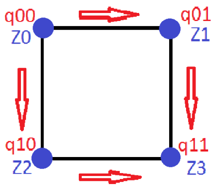


Figure 1. 2D lattice 2 × 2

3. Energy Calculation Methods

3.1. Direct State Vector Computation

This section presents the simplest approach for simulating QAOA on a quantum computer simulator. Using the quantum register's state vector $|\psi_p(\gamma,\beta)\rangle$ provided by the simulator and the diagonal elements of the the Hamiltonian (1), the per-site average energy (6) reduces to a sum:

$$E_p(\gamma,\beta) = \frac{1}{n}\sum_{i=0}^{2^n-1} \overline{\psi_p(\gamma,\beta)}_i \psi_p(\gamma,\beta)_i \mathcal{H}_{ii}, \tag{7}$$

where n is the number of lattice sites and the bar denotes complex conjugation.

Consider a 2×2 spin lattice (Fig. 1). Red arrows indicate pairwise interactions in the Hamiltonian (1). All possible spin configurations are encoded by a 4-qubit register ($n = 4$). An arbitrary normalized final state vector of a register in the computational basis is given by the amplitudes $\alpha_i(\gamma,\beta)$, $i = 0,\dots,15$:

$$|\psi_1(\gamma,\beta)\rangle = \alpha_0(\gamma,\beta)|0000\rangle + \alpha_1(\gamma,\beta)|0001\rangle + \cdots + \alpha_{15}(\gamma,\beta)|1111\rangle, \tag{8}$$

where $\sum_i \bar{\alpha}_i(\gamma,\beta)\alpha_i(\gamma,\beta) = 1$. For $J = 1$ and $h = 1/2$ in (1), the diagonal elements of the Hamiltonian are:

$$[-1.5, -0.25, -0.25, 0, -0.25, 0, 1, 0.25, -0.25, 1, 0, 0.25, 0, 0.25, 0.25, -0.5].$$

The ground state $|0000\rangle$ (all spins aligned with the field) has energy -1.5. This means that during parameter variation, we aim to find the following values $\gamma_{\min}, \beta_{\min}$ that satisfy the following conditions:

$$\bar{\alpha}_0(\gamma_{\min},\beta_{\min})\alpha_0(\gamma_{\min},\beta_{\min}) = 1, \qquad \alpha_i(\gamma_{\min},\beta_{\min}) = 0,\ i \neq 0.$$

The quantum circuit implementing the single-layer QAOA ansatz (5) (Fig. 2) uses four qubits:

$$q_{(0,0)},\ q_{(0,1)},\ q_{(1,0)},\ q_{(1,1)}$$

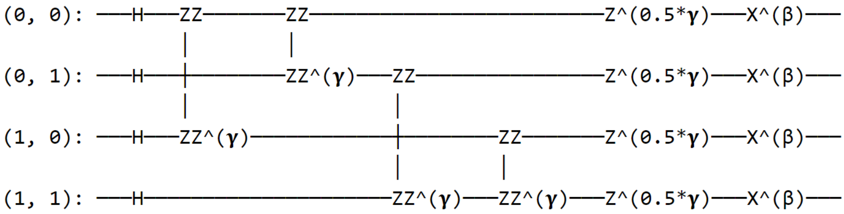
which represent spins at lattice sites and form the quantum register.

The horizontal lines represent qubits, and the symbols on them show quantum gates acting sequentially from left to right. In the first (leftmost) column, Hadamard gates H prepare an equal superposition state (2). Subsequent columns contain gates that implement the driver operator $U(\gamma,\mathcal{H})$ and the mixer operator $U(\beta,B)$. The spin-spin interaction part of the driver operator $U(\gamma,\mathcal{H})$ is implemented by ZZ gates (vertical connections in the circuit diagram, labeled ZZ at the top and ZZ^() at the bottom).

For two qubits, the ZZ gate (Fig. 3) corresponds to a rotation operator with variational parameter γ (rotation angle)[1]:

---

[1] In Cirq, this gate's matrix differs by a factor of $i = \sqrt{-1}$.



```
(0, 0): ───H───ZZ────────ZZ────────────────────────Z^(0.5*γ)───X^(β)───
               │         │
(0, 1): ───H───┼─────────ZZ^(γ)───ZZ───────────────Z^(0.5*γ)───X^(β)───
               │                  │
(1, 0): ───H───ZZ^(γ)─────────────┼────────ZZ──────Z^(0.5*γ)───X^(β)───
                                  │        │
(1, 1): ───H──────────────────────ZZ^(γ)───ZZ^(γ)───Z^(0.5*γ)───X^(β)───
```

Figure 2. Circuit diagram of the $p = 1$ QAOA ansatz (5) for a 2×2 Ising lattice ($J = 1$, $h = 0.5$), constructed using Cirq [9]

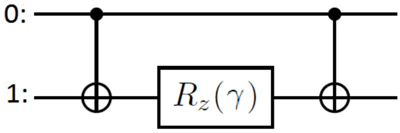


Figure 3. Quantum circuit for the ZZ gate (9)

$$R_{ZZ}(\pi\gamma) \equiv \exp\left(-i\pi\gamma\, Z \otimes Z/2\right) = \begin{bmatrix} e^{-i\pi\gamma/2} & 0 & 0 & 0 \\ 0 & e^{i\pi\gamma/2} & 0 & 0 \\ 0 & 0 & e^{i\pi\gamma/2} & 0 \\ 0 & 0 & 0 & e^{-i\pi\gamma/2} \end{bmatrix}, \tag{9}$$

which can be constructed using CNOT and $R_Z(\gamma)$ gates:

$$\mathrm{CNOT} \cdot (\mathbb{I} \otimes R_Z(\gamma)) \cdot \mathrm{CNOT} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e^{-i\pi\gamma/2} & 0 & 0 & 0 \\ 0 & e^{i\pi\gamma/2} & 0 & 0 \\ 0 & 0 & e^{-i\pi\gamma/2} & 0 \\ 0 & 0 & 0 & e^{i\pi\gamma/2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

An important property of the ZZ gate is its ability to create entangled qubit states [10], which gives quantum computations a significant advantage over classical approaches for studying quantum systems. The wave function of an entangled system cannot be factorized into a product of subsystem wave functions and represents a coherent superposition of combined subsystem states.

In the case of bipartite quantum systems — specifically two-qubit systems — the degree of entanglement can be quantified by the *concurrence* measure. As an illustration, consider the entangled state $|\psi_{ZZ}(\gamma)\rangle$ prepared by applying the ZZ interaction to the equal superposition state (Eq. 2) for $n = 2$ qubits:

$$|\psi_{ZZ}(\gamma)\rangle = R_{ZZ}(\pi\gamma)\, |\psi_0^{(2)}\rangle = \frac{1}{2}\Big(e^{-i\pi\gamma/2}|00\rangle + e^{i\pi\gamma/2}|01\rangle + e^{i\pi\gamma/2}|10\rangle + e^{-i\pi\gamma/2}|11\rangle\Big).$$

The density matrix of the pure state $|\psi_{ZZ}(\gamma)\rangle$ is the following one

$$\rho(\gamma) = |\psi_{ZZ}(\gamma)\rangle\langle\psi_{ZZ}(\gamma)| = \frac{1}{4}\begin{pmatrix} e^{-i\pi\gamma} & 1 & 1 & e^{-i\pi\gamma} \\ 1 & e^{i\pi\gamma} & e^{i\pi\gamma} & 1 \\ 1 & e^{i\pi\gamma} & e^{i\pi\gamma} & 1 \\ e^{-i\pi\gamma} & 1 & 1 & e^{-i\pi\gamma} \end{pmatrix},$$

By taking the partial trace over the second qubit (subsystem B of the two-qubit system), we obtain the reduced density matrix for the first qubit (subsystem A), which depends on the parameter γ:

$$\rho_A(\gamma) = \mathrm{Tr}_B\left(\rho(\gamma)\right) = \frac{1}{2}\begin{pmatrix} \cos\pi\gamma & 1 \\ 1 & \cos\pi\gamma \end{pmatrix}.$$

The concurrence [10] is then calculated as:

$$\mathcal{C}(|\psi_{ZZ}(\gamma)\rangle) = \sqrt{2\left(1 - \mathrm{Tr}\left(\rho_A^2(\gamma)\right)\right)} = \sin(\pi\gamma).$$

As evident, the *concurrence* $\mathcal{C}(|\psi_{ZZ}(\gamma)\rangle)$ is non-zero, indicating that the ZZ gate entangles qubit states for $\gamma \notin \mathbb{Z}$. Maximal entanglement occurs at $\gamma = 1/2$, where the operator assumes the form:

$$R_{ZZ}(1/2) = \frac{1}{\sqrt{2}}\left(\mathbb{I}_4 - iZ \otimes Z\right),$$

and the two-qubit state vector $|\psi_{ZZ}(\gamma = 1/2)\rangle$ decomposes into the Bell states:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad \text{and} \quad |\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle),$$

$$|\psi_{\mathrm{ZZ}}(1/2)\rangle = \frac{1}{2}\left[(1 - i)|\Phi^+\rangle + (1 + i)|\Psi^+\rangle\right].$$

The ZZ gate implementing the interaction between spins at $(0,0)$ and $(1,0)$ in Fig. 1 (qubits $q_{(0,0)}$ and $q_{(1,0)}$) acts via the Pauli operators $Z^{(0)}$ and $Z^{(2)}$:

$$\exp\left(-i\pi\gamma Z^{(0)}Z^{(2)}/2\right), \qquad \begin{cases} Z^{(0)} = Z \otimes \mathbb{I}^{\otimes 3}, \\ Z^{(2)} = \mathbb{I}^{\otimes 2} \otimes Z \otimes \mathbb{I}, \end{cases}$$

and is applied right after the initial Hadamard gates. The spin-spin interaction part of the driver operator $U(\gamma, \mathcal{H})$ consists of four analogous factors,

$$U_{ZZ}(\gamma, \mathcal{H}) = e^{-i\pi\gamma Z^{(2)}\cdot Z^{(3)}/2} e^{-i\pi\gamma Z^{(1)}\cdot Z^{(3)}/2} e^{-i\pi\gamma Z^{(0)}\cdot Z^{(1)}/2} e^{-i\pi\gamma Z^{(0)}\cdot Z^{(2)}/2},$$

and is the diagonal 16×16 matrix with elements:

$$\mathrm{diag}\left[e^{-2\pi i\gamma}, 1, 1, 1, 1, 1, e^{2\pi i\gamma}, 1, 1, e^{2\pi i\gamma}, 1, 1, 1, 1, 1, e^{-2\pi i\gamma}\right].$$

The magnetic field interaction in the driver operator (3) is implemented via Z-axis rotations acting on each qubit:

$$R_z(\pi\gamma h) = e^{-i\pi\gamma h Z/2} = \cos\left(\frac{\pi\gamma h}{2}\right)\mathbb{I} - i\sin\left(\frac{\pi\gamma h}{2}\right)Z = \begin{bmatrix} e^{-i\pi\gamma h/2} & 0 \\ 0 & e^{i\pi\gamma h/2} \end{bmatrix},$$

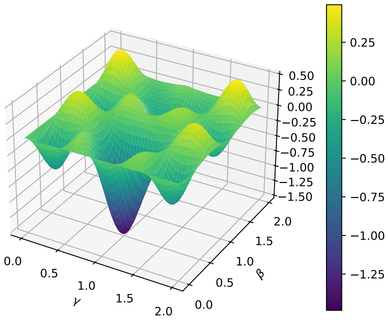


Figure 4. Energy landscape: $E_1(\gamma, \beta)$ computed via the circuit in Fig. 2 with a grid step of 1/100. Lattice of size 2×2. One layer ansatz

denoted as `Z^(0.5 * )` in the circuit diagram. The complete magnetic part of the driver operator is a product of exponentials of Pauli Z-strings performing single-qubit Z rotations and can be represented by a diagonal matrix

$$U_{hZ}(\gamma, \mathcal{H}) = \prod_{j=0}^{3} \exp\left(-i\pi h\gamma Z^{(j)}/2\right),$$

where $Z^{(j)}$ rotates the jth qubit, with diagonal entries:

$$\mathrm{diag}[e^{-2\pi ih\gamma}, e^{-\pi ih\gamma}, e^{-\pi ih\gamma}, 1, e^{-\pi ih\gamma}, 1, 1, e^{\pi ih\gamma}, e^{-\pi ih\gamma}, 1, 1, e^{\pi ih\gamma}, 1, e^{\pi ih\gamma}, e^{\pi ih\gamma}, e^{2\pi ih\gamma}].$$

The mixing operator $U(\beta, B)$ (4) consists of single-qubit X-rotations:

$$R_x(\pi\beta) = e^{-i\pi\beta X/2} = \cos\left(\frac{\pi\beta}{2}\right)\mathbb{I} - i\sin\left(\frac{\pi\beta}{2}\right)X = \begin{bmatrix} \cos\frac{\pi\beta}{2} & -i\sin\frac{\pi\beta}{2} \\ -i\sin\frac{\pi\beta}{2} & \cos\frac{\pi\beta}{2} \end{bmatrix},$$

where X is the Pauli-X gate. In the quantum circuit (Fig. 2), they are denoted as `X^()`. Thus, the mixing operator is represented by the matrix:

$$U_{Mix}(\beta, B) = e^{-i\pi\beta X^{(3)}/2} e^{-i\pi\beta X^{(2)}/2} e^{-i\pi\beta X^{(1)}/2} e^{-i\pi\beta X^{(0)}/2},$$

which is a product of exponentials of Pauli-X strings $X^{(i)}$ defined as

$$X^{(i)} = \mathbb{I}^{\otimes n} \otimes \underset{\substack{i\text{-th} \\ \text{position}}}{X} \otimes \mathbb{I}^{\otimes(n-i-1)},$$

where the Pauli-X operator acts on the i-th qubit.

Figure 4 shows the energy landscape $E_1(\gamma, \beta)$ (7) for $p = 1$, computed over a two-dimensional grid with a step of 1/100, where the parameters γ and β range from 0 to 2. The characteristic “egg tray” surface exhibits a global minimum (dark blue region) at:

$$\gamma_{\min} = 1.0, \quad \beta_{\min} = 0.5. \tag{10}$$

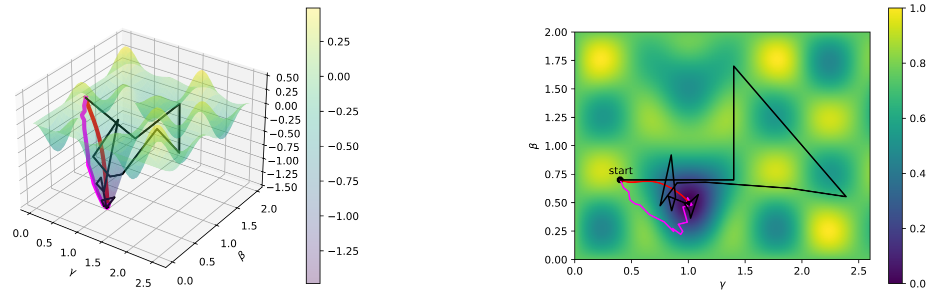


Figure 5. Optimization trajectories

For these parameters, the ground state amplitude α_0 in (8) satisfies $|\alpha_0|^2 \approx 1$ while the amplitudes of all other states vanish within numerical precision.

The energy functional $E_1(\gamma, \beta)$, obtained via the QAOA ansatz, can be minimized using optimization methods implemented on the classical part of the algorithm. Figure 5 shows the optimization trajectories — paths starting from $(\gamma_0, \beta_0) = (0.4, 0.7)$ near a local maximum — on the energy landscape and its 2D projection (contour plot) for three methods[2]:

- Gradient descent method — red trajectory,
- Nelder–Mead method (1965) — magenta trajectory,
- COBYLA method (Powell 1994) — black/white trajectory.

All methods converge to the global minimum (10) within 100 iterations, achieving an error below 1%. Notably, gradient-free methods reach the target more efficiently due to fewer quantum calls — i.e., quantum circuit executions — for the same accuracy. Thus, the COBYLA method achieves an energy minimization precision of 10^{-8} within 40 quantum calls, Nelder–Mead within 90, and gradient descent within 1800.

3.2. Sampling: Estimating Spin Configuration Probabilities

In Sec. 3.1, the energy $E_p(\gamma, \beta)$ was calculated using the state vector with amplitudes α_i (8) prepared by quantum circuits in Fig. 2. However, since the amplitudes α_i cannot be measured directly, the register state is determined through repeated measurements (shots) of the qubits after applying the p-layer QAOA ansatz with fixed values of all $2p$ parameters $\vec{\gamma}, \vec{\beta}$. Then the energy can be approximated by the sum:

$$E_p(\gamma, \beta) \approx \sum_{i=0}^{2^n-1} \mathcal{P}_{i,p}(\gamma, \beta)\mathcal{H}_{ii}, \tag{11}$$

where $\mathcal{P}_{i,p}$ is the empirical probability of finding the register in the i-th basis state, which occurs $n_{ith\ state}$ times in a series of N_{meas} measurements (trials, shots),

$$\mathcal{P}_{i,p}(\gamma, \beta) = n_i / N_{\text{meas}}, \qquad \sum_{i=0}^{2^n-1} \mathcal{P}_{i,p} = 1.$$

Convergence to the true energy (7) is achieved as $N_{\text{meas}} \to \infty$.

---

[2] For references and extended comparison of methods, see section 4.

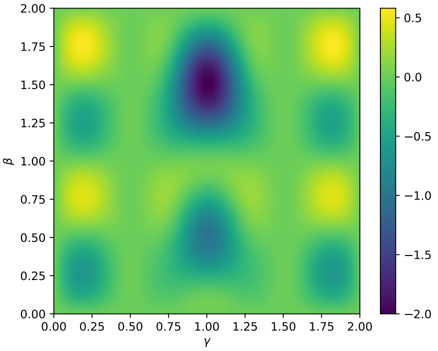


Figure 6. Energy landscape: $E_1(\gamma, \beta)$ computed with a grid step of 1/100. Lattice of size 2 × 2 × 2. One layer ansatz

Table 1

Histogram data of measurement outcomes for the register state $|\psi_p(0.8, 1.2)\rangle$, $N_{\text{meas}} = 10000$

Histogram bin numbers	0	1	2	...	248
Basis state corresponding to the bin	$\|00000000\rangle$	$\|00001000\rangle$	$\|10000000\rangle$	...	$\|10010011\rangle$
Frequency of the basis state	447	301	269	...	1
Energy of the basis state	−2.0	−1.125	−1.125	...	0.5

In this section, we analyze the sampling method for the single-layer QAOA applied to a 2×2×2 Ising model. Figure 6 shows the landscape of the energy $E_1(\gamma, \beta)$ (7), computed via the state vector method from Sec. 3.1 with $J = 1$, $h = 1/2$ in (1). The global minimum $E_1 = -2.0$ occurs at $(\gamma, \beta) = (1.0, 1.5)$, where the amplitude of the state $|00000000\rangle$ equals 1, and the amplitudes of all other states vanish (within numerical precision).

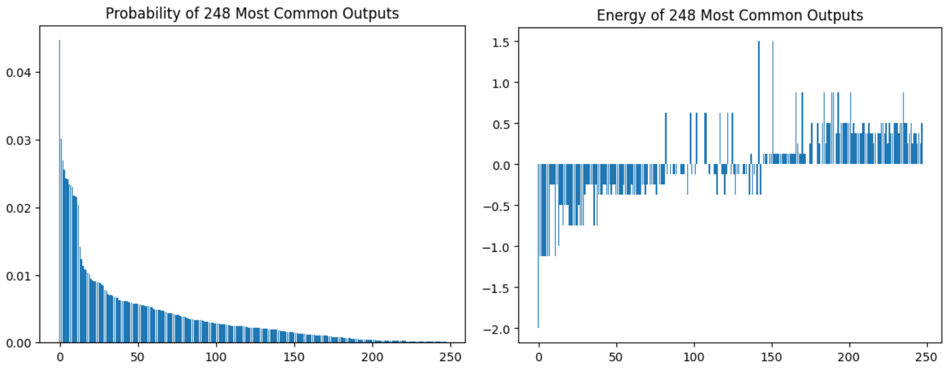
To demonstrate the sampling method, we perform a series of measurements with fixed parameters $(\gamma, \beta) = (0.8, 1.2)$. The computational basis states are enumerated from 0 to 255:

$$|0\rangle \equiv |00000000\rangle,\ |1\rangle \equiv |00000001\rangle,\ \ldots,\ |255\rangle \equiv |11111111\rangle.$$

Table 1 shows the histogram data obtained from $N_{\text{meas}} = 10000$ measurements (shots) of the register state $|\psi_p(0.8, 1.2)\rangle$. The state probabilities and corresponding energies are presented by histograms on Fig. 7. The sampled energy E_1 computed via Eq. (11) agrees with the state vector result from Eq. (7) at $(\gamma, \beta) = (0.8, 1.2)$ with high precision:

$$-0.5187 \text{ (Sampling)} \quad \text{vs.} \quad -0.5184 \text{ (State vector)},$$

with a relative difference of only 3×10^{-4}, which validates the sampling approach.



(a) State Probabilities

(b) State Energies

Figure 7. Sampling Histograms

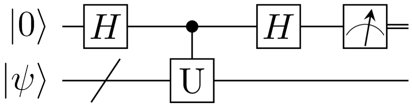


Figure 8. Hadamard test circuit (source: Wikipedia)

3.3. The Hadamard Test: Evaluation of Quantum Averages of Operators

In the examples presented above, the role of the quantum computer was limited to preparing the ansatz state $|\psi_p(\gamma,\beta)\rangle$ (5), while classical computation handled two key tasks. Firstly, it computes energies using either the state vector method (7) or the measurement sampling approach (11). Secondly, it optimizes the parameters (γ,β) based on these energy evaluations.

A more efficient approach involves direct measuring the energy $E_p(\gamma,\beta)$ on the quantum computer exploiting the Hadamard test (Fig. 8) which requires estimating only the final state of a single ancillary qubit. In addition to the QAOA scheme preparing the state $|\psi\rangle \equiv |\psi_p(\gamma,\beta)\rangle$ the following transformations are fulfilled:

$$|0\rangle \otimes |\psi\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle) \xrightarrow{contr-U} \frac{1}{\sqrt{2}}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U|\psi\rangle) \xrightarrow{H}$$
$$\xrightarrow{H} \frac{1}{2}\Big((|0\rangle + |1\rangle) \otimes |\psi\rangle + (|0\rangle - |1\rangle) \otimes U|\psi\rangle\Big) == \frac{1}{2}\Big((|0\rangle \otimes (\mathbb{I} + U)|\psi\rangle + |1\rangle \otimes (\mathbb{I} - U)\psi\rangle\Big).$$

The measurement probabilities of the ancilla qubit's output states $|0\rangle$ and $|1\rangle$ depend on the real part of the operator U:

$$P(0) = \frac{1}{2}\langle\psi|(\mathbb{I} + \mathrm{Re}\,U)|\psi\rangle, \quad P(1) = \frac{1}{2}\langle\psi|(\mathbb{I} - \mathrm{Re}\,U)|\psi\rangle.$$

Given the state normalization $\langle\psi|\psi\rangle = 1$, the real part of the expectation value of the operator equals:

$$\mathrm{Re}\langle\psi|U|\psi\rangle = P(0) - P(1).$$

Using the identity $P(0) + P(1) = 1$, the expectation value can also be expressed as:

$$\mathrm{Re}\langle\psi|U|\psi\rangle = 1 - 2P(1) \quad \text{or} \quad \mathrm{Re}\langle\psi|U|\psi\rangle = 2P(0) - 1.$$

Thus the energy $E(\gamma, \beta)$ (Eq. (6)) can be measured as a sum of expectation values of all Hamiltonian terms:

$$E(\gamma,\beta) = -J \sum_{\langle i,j\rangle} \langle\psi(\gamma,\beta)|Z^{(i)}Z^{(j)}|\psi(\gamma,\beta)\rangle - h\sum_{i}\langle\psi(\gamma,\beta)|Z^{(i)}|\psi(\gamma,\beta)\rangle. \tag{12}$$

4. Optimization

4.1. Computational Setup: Software and Hardware

Initially, the problem was solved interactively on a quantum testbed [11, 12] using `Cirq` [9], a Python library for quantum circuit simulation (supporting up to 30 qubits). However, as the lattice dimension increases, the problem complexity grows exponentially (see Appendix), requiring substantial RAM and high-performance CPU/GPU resources. Even for the $3 \times 3 \times 3$ lattice case, we had to employ the `qsim` simulator, a C++ backend integrated with `Cirq`. It uses SIMD (Single Instruction, Multiple Data) for vectorization and `OpenMP` for CPU multithreading, and supports GPU acceleration via NVIDIA's `cuStateVec` [13], a high-performance, GPU-accelerated library for quantum state vector operations. This hybrid configuration enables simulation of large-scale quantum circuits (up to approximately 40 qubits) while providing significant computational acceleration.

The results presented below were obtained employing the `Cirq+qsim` framework, with computations performed on both a central processing unit (CPU) and a graphics processing unit (GPU) in two configurations:

1. a basic CUDA implementation;
2. the `cuStateVec` library from the NVIDIA `cuQuantum` SDK [14], which delivered superior performance for quantum circuit simulations, especially for sampling-based methods.

We performed simulations on the workstation equipped with CPU 8-core AMD Ryzen 7 3700X, DDR4 RAM 128 GB, and the GPU NVIDIA GeForce RTX 4070 Ti SUPER.

4.2. Computational Resource Requirements for Simulation

Energy minimization was done for six spin lattice configurations, as shown in Table 2. The table contains key parameters that determine computational complexity: the number of qubits in the quantum circuit, the number of terms in the Hamiltonian, and the dimension of the state vector of the register (i.e., the number of spin configurations). For each lattice configuration, we employed a three-layer QAOA ansatz with six variational parameters: $(\gamma_1, \beta_1, \gamma_2, \beta_2, \gamma_3, \beta_3)$.

The energy was computed using four distinct approaches to compare them in terms of numerical accuracy, runtime performance, and memory requirements:

1. QAOA–State vector method: uses Equation (7),
2. QAOA–Sampling method: uses Equation (11),
3. Hadamard–State vector method: uses the Hadamard test circuit (Figure 8) with Equation (7),
4. Hadamard–Sampling method: uses the Hadamard test circuit with ancilla qubit measurement and Equation (12).

Table 2

Spin Lattices. Calculations with the `cuStateVec` library

Lattice Data		Energy Operator			Memory required for state vector [3]	Runtime ratio (Hadamard/QAOA)	
Lattice	qubits, n	ZZ terms	μZ terms	Total		State vector	Sampling 1000 shots
3×3	9	12	9	21	512 B	22.0	20.2
$3 \times 2 \times 2$	12	20	12	32	4 KiB	35.1	29.4
4×4	16	24	16	40	64 KiB	76.0	37.9
$3 \times 3 \times 2$	18	33	18	51	256 KiB	204.0	48.0
5×5	25	40	25	65	32 MiB	566.3	120.3
$3 \times 3 \times 3$	27	54	27	81	128 MiB	997.3	160.9

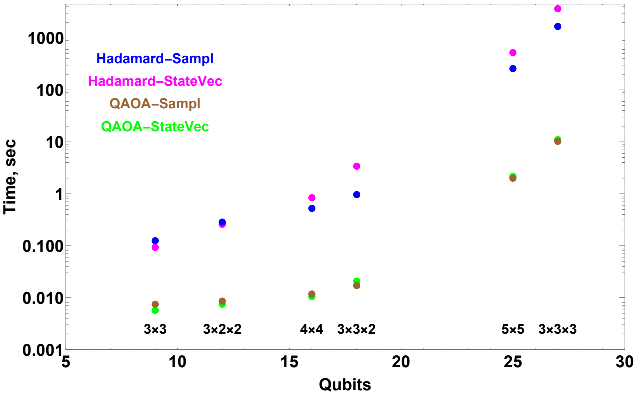


Figure 9. CPU runtime versus number of qubits for energy computations in different quantum simulation approaches

A single energy evaluation reveals significant performance differences among the methods. Figure 9 shows the CPU execution time as a function of the number of qubits. Clearly, the Hadamard-based methods exhibit significantly longer runtimes than the QAOA-based approaches. The time ratio Hadamard/QAOA approaches using the state-vector and sampling methods are shown in the last two columns of Table 2. Timing and memory analysis of all methods — illustrated for the 5 × 5 2D lattice in the Appendix — demonstrates that the `cuStateVec` library is preferable for the energy minimization procedure.

4.3. Used Local and Global Optimization Methods

For local optimization, we compare a simple gradient-based method with the following gradient-free methods [4]:

1. COBYLA (Constrained Optimization BY Linear Approximation) (Powell, 1994 [15]): a simplex method utilizing trust regions and linear approximation of the objective function.
2. Modified Powell Method (Powell, 1964 [16]): a conjugate direction algorithm.
3. Nelder–Mead Simplex (Nelder and Mead, 1965 [17]): the classical simplex direct search method, known for its simplicity and robustness.

All methods used the same initial point:

$$(\gamma_1, \beta_1, \gamma_2, \beta_2, \gamma_3, \beta_3) = (0.1, 0.1, 0.5, 1.0, 1.5, 0.5).$$

The components of the energy gradient vector were computed numerically using central finite differences. During minimization, the parameters at each iteration were updated in proportion to the gradient components, and the algorithm terminated after 100 iterations.

We tested and compared all energy computation methods described in section 3. Gradient-based optimization succeeded for both QAOA and Hadamard circuits on lattices up to size 5×5 (25 qubits), but only when using state vector simulation. With the sampling approach, gradient-based minimization failed to produce reliable values of the gradients even for small lattices. This suggests that either more advanced algorithms are required or the number of measurements must be significantly increased (well beyond 1000 shots).

Only gradient-free methods were used in conjunction with sampling (1000 shots per parameter set), for three reasons:

1. gradient-based minimization consistently failed under sampling procedure,
2. it is the sampling that is relevant approach for real quantum computers,
3. sampling is significantly faster and less memory intensive for large lattices (5×5, $3 \times 3 \times 3$), as demonstrated in the Appendix.

The derivative-free methods successfully optimized both QAOA and Hadamard circuits for all lattice sizes. Since the energy is evaluated only once per minimization step, gradient-free methods are less sensitive to energy estimation errors than gradient-based methods, especially in high-dimensional parameter spaces.

To compare the gradient-free methods, we present the results for the 5×5 lattice test (Table 3). The number of energy evaluations (NEEs) was independent of lattice size across all methods and simulation schemes. The following conclusions can be drawn from this table:

- COBYLA outperformed other methods in terms of NEEs. The Nelder–Mead method showed the worst performance, requiring the highest number of NEEs, resulting in longer runtime and less accurate energy estimates. In general, QAOA scheme required 5–10 percent fewer NEEs than the Hadamard scheme.
- COBYLA's time per energy evaluation is approximately twice that of a single evaluation in either the QAOA or Hadamard scheme (see Appendix). In contrast, Powell's method requires only a few percent more time than a single evaluation. The Nelder–Mead algorithm occupies an intermediate position.

For the global search of the energy minimum, we employed two methods from the `scipy.optimize` module of the SciPy library:

---

[4] Gradient-free methods correspond to arguments of the `scipy.optimize.minimize` function in the SciPy library.

Table 3
Performance of local gradient-free optimization methods for QAOA and Hadamard circuits on a 5×5 lattice. The true ground-state energy is -2.1. Simulations performed using sampling with 1000 shots per evaluation and the `cuStateVec` backend

Method	**COBYLA**		**Powell**		**Nelder–Mead**	
Schemes	QAOA	Hadamard	QAOA	Hadamard	QAOA	Hadamard
Energy	-2.0990	-2.0999	-2.0996	-2.0999	-1.9339	-1.9621
Relative error (%)	0.0476	0.0048	0.0190	0.0048	7.9095	6.5667
Found parameters γ_1	0.0027	0.0007	0.0056	1.9979	0.0984	0.1006
β_1	0.1159	0.0968	0.4721	1.0655	0.1085	0.1031
γ_2	0.4982	0.5010	0.5054	0.5026	0.4343	0.4215
β_2	1.0002	1.0000	1.0011	0.9996	0.9976	0.9995
γ_3	1.5006	1.4992	1.5018	1.4992	1.4996	1.4930
β_3	0.5105	0.5026	0.5079	1.5013	0.5263	0.5325
Total NEEs	78	100	95	105	246	251
%%time Wall time	19.9s	51m 23s	12.4 s	27m 3s	44.2s	1h 29m

- Dual Annealing (DA) [18]: a stochastic global optimization algorithm based on a modified form of simulated annealing;
- Simplicial Homology Global Optimization (SHGO) [19] with Sobol sampling: a global optimization method based on simplicial integral homology and combinatorial topology.

For local optimization both methods used the COBYLA and Powell algorithms, combined with a sampling approach employing 1000 shots per parameter set. The performance of the global optimization methods is summarized in Table 4. Comparing the algorithms, the following observations can be made:

- Both global algorithms achieved solutions within a few percent of the true ground-state energy, independent of lattice size.
- When using COBYLA, fewer energy evaluations were required compared to Powell (consistent with standalone local method comparisons).
- When paired with COBYLA, both global methods required fewer energy evaluations than when using Powell, consistent with standalone comparisons of the local optimizers.
- The combination of SHGO and Powell resulted in the highest computational cost in terms of total energy evaluations.
- The DA algorithm with COBYLA required 3–4 times fewer energy evaluations than SHGO with Powell.
- The number of iterations and energy evaluations was largely independent of lattice size across all algorithm combinations.

Table 4

Performance of global optimization methods for QAOA and Hadamard circuits on a 5 × 5 lattice. The true ground-state energy is -2.1. Simulations use sampling with 1000 shots per evaluation and the `cuStateVec` backend. (Here n_{fev}: total number of energy evaluations; n_{lfev}: number of evaluations during local optimization)

Method	**Dual Annealing (DA)**				**SHGO**			
Local optimizer	COBYLA		Powell		COBYLA		Powell	
Scheme	QAOA	Hadamard	QAOA	Hadamard	QAOA	Hadamard	QAOA	Hadamard
Energy	-2.0002	-2.0960	-2.1000	-2.0962	-2.09928	-2.09936	-2.1	-2.1
Relative error (%)	4.7524	0.1905	0.0000	0.1810	0.0333	0.0286	0.0000	0.0000
Optimized parameters γ_1	1.5554	0.5091	0.6325	1.4458	1.8313	0.5015	1.3915	1.8454
β_1	1.0032	1.0019	0.0023	0.9998	1.0000	1.0005	0.9985	0.9988
γ_2	1.3628	0.3831	0.8668	0.5488	0.5059	0.3342	0.4918	0.4991
β_2	1.9072	1.9823	0.9987	0.4257	0.9959	1.9950	0.9932	1.0007
γ_3	1.0862	1.1088	0.4979	0.0746	1.6644	1.1650	0.1174	1.6564
β_3	0.5046	0.5053	1.4992	1.0929	1.5030	0.5024	0.5051	1.4990
Evals of energy: n_{fev} (DA) $n_{\text{fev}}/n_{\text{lfev}}$(SHGO)	1342	1387	2439	3072	1105/487	1146/528	4991/4372	3962/3343
%%time Wall time	2 min 53 s	5 h 55 min	5 min 14 s	13 h 6 min	3 min 47 s	7 h 9 min	11 min 11 s	17 h 2 min
Memory	256 KiB	25.5 MiB	256 KiB	56.0 MiB	213.7 MiB	240.4 MiB	28.5 MiB	99.8 MiB

Regarding timing characteristics:

- For the DA algorithm, the average energy evaluation time closely matched that of standalone simulations for both the QAOA and Hadamard measurement schemes.
- SHGO with COBYLA exhibited 1.5–2× longer average evaluation time compared to standalone runs, while with Powell, the overhead was approximately 10%.

In terms of memory usage, SHGO required several hundred megabytes of additional memory, while DA used only several hundred kilobytes — independent of lattice size. Notably, SHGO provides a set of near-optimal parameter configurations, which represents a key advantage over DA, especially for identifying multiple low-energy states.

5. Summary

We investigate the performance of the Quantum Approximate Optimization Algorithm (QAOA) [6] through simulations of the Ising model in a longitudinal magnetic field. Using the `Cirq+qsim` framework, we construct a QAOA-generated variational wave function ansatz and employ optimization methods from `SciPy`'s library to estimate the ground-state energy.

The algorithm's advantage over other variational methods stems from its task-adaptive structure: the driver operator, defined as the time-evolution operator under the system Hamiltonian, enables efficient optimization while mitigating the barren plateau problem [4]. We employ the Ising model due to its importance in the study of spin systems and quantum field theories on space-time lattices.

We demonstrate a mapping between spin configurations on a spatial lattice and quantum register states, with each spin at a lattice site corresponding to a single qubit.

While the quantum register size scales linearly with the number of spins n, classical simulations require exponential resources (2^n complex amplitudes) to represent the full state vector and Hamiltonian matrix. To maintain high-fidelity ground-state approximations, the QAOA ansatz requires deeper circuits — i.e., an increasing number of layers — as n increases.

We analyze QAOA performance on both 2D and 3D lattice configurations and compare two measurement schemes: the standard QAOA scheme (see Figure 2) and the Hadamard test (see Figure 8), with energy calculations performed using either full state-vector simulations or sampling. The Hadamard test, introduced in Section 3.3, fully leverages the capabilities and demonstrates the advantage of a quantum computer by estimating the average energy through measurement of a single ancilla qubit. Modeling of state-vector simulations requires more memory but demonstrates faster execution than the Hadamard test (Sec. 3). Direct state-vector calculations for energy estimation in both schemes consume significantly more memory than sampling-based approaches. The sampling method exhibits nearly flat resource scaling: increasing the number of shots by an order of magnitude results in only a few percent increase in memory and computation time.

For parameter optimization, we compare gradient-based and gradient-free approaches. Gradient-based methods offer higher precision but are prone to convergence instability. In contrast, gradient-free techniques — such as COBYLA, Powell, and Nelder–Mead — exhibit greater robustness and lower computational overhead. The optimization landscape exhibits multiple local minima, which significantly hinders optimization — a well-known challenge in variational quantum algorithms. This obstacle provides another argument in favor of gradient-free methods, which can avoid local minima through adequate construction of the simplex in the parameter space.

The choice of measurement scheme and optimization algorithm for QAOA simulations ultimately depends on the research objectives — such as ansatz verification versus modeling real quantum hardware — as well as available computational resources and time constraints.

Table 5

Memory requirements for state vectors and unitary operators

Lattice Size	Qubits (n)	$\dim(\vert\psi\rangle)$	State Vector Memory	Unitary Operator Memory
3×3	9	512	4 KiB	4 MiB
4×4	16	65,536	512 KiB	64 GiB
5×5	25	33,554,432	256 MiB	16 PiB

The developed computational framework demonstrates straightforward scalability to arbitrary lattice dimensions. However, one should keep in mind the inevitable exponential growth in required computational resources with increasing system size. Our calculations were partially performed on JINR's HybriLIT quantum testbed [12]. Leveraging the cuStateVec library on NVIDIA A100 GPUs dramatically accelerated computations, reducing execution time (compared to an optimized PC used in this work) from three days to 14 minutes for energy surface plot (Fig. 4). Complete testbed implementation details are given in [20].

1. Appendix: Time and Memory Requirements for Energy Computation Methods

A state vector $|\psi\rangle$ and unitary matrix U for an n-qubit register have the following dimensions: $\dim(|\psi\rangle) = 2^n$, $\quad \dim(U) = 2^n \times 2^n = 2^{2n}$, where the state vector uses `complex64` and unitary operators `complex128` by default in the `Cirq` library. The required memory for storing $|\psi\rangle$ and U can be estimated using:

$$\mathrm{mem}(|\psi\rangle) = \frac{2^n \times 64}{8} \quad \text{bytes}, \qquad \mathrm{mem}(U) = \frac{2^{2n} \times 128}{8} \quad \text{bytes}.$$

As concrete examples, consider 2D lattices of different sizes and their corresponding qubit counts:

While the memory requirements for state vectors remain manageable even for larger systems, the storage needed for unitary operators grows exponentially. Despite these computational challenges, the `Cirq+qsim` framework enables efficient simulation of quantum circuits on personal computers through optimized sparse matrix representations, advanced state vector compression techniques, and parallel computation strategies.

Table 6 presents a comparative analysis of computational resources required for energy calculation by all the methods considered above on a 5×5 lattice using the same set of variational parameters (γ, β). We measured time and memory consumption using both the *psutil* utility and IPython's magic commands (%memit, %timeit).

The energy value is determined more accurately using the state-vector method. However, as shown in Table 7, comparable accuracy can be achieved by increasing the number of samples, with only a slight increase in time and memory requirements. As noted in Section 3, all methods employing the Hadamard test (see Table 7) require more time than those using just QAOA scheme (rows labeled "QAOA"). However, the advantage of the Hadamard-based methods lies in their superior memory efficiency, especially when combined with sampling. Based on the data in table 6, we can briefly characterize the studied methods as follows:

1. QAOA–State Vector: exact, low runtime, high memory usage;

Table 6
Performance metrics for 5 × 5 lattice simulations: wall time and memory usage. Energy values are averaged over 1000 shots for sampling methods (values near zero indicate negligible memory overhead)

Scheme	**Backend**	**Method**	**Energy**	**Wall time**	**Memory increment (%memit, MiB)**	**RSS (psutil, MiB)**
State Vector	CPU	QAOA	-1.6556	2.16 s	512.25	2125.49
		Hadamard	-1.6556	8 min 46 s	1536.41	331.5
	GPU (CUDA)	QAOA	-1.6556	550.2 ms	502.00	2221.46
		Hadamard	-1.6556	5 min 9 s	1536.03	426.46
	GPU (cuStateVec)	QAOA	-1.6556	558.2 ms	484.05	2231.09
		Hadamard	-1.6556	5 min 16 s	1535.87	437.59
Sampling (1000 shots)	CPU	QAOA	-1.6372	1.98 s	255.88	2125.72
		Hadamard	-1.6592	4 min 20 s	512.34	331.55
	GPU (CUDA)	QAOA	-1.6461	4.17 s	0.00	2222.59
		Hadamard	-1.6402	8 min 57 s	0.00	426.48
	GPU (cuStateVec)	QAOA	-1.6526	127.3 ms	0.00	2234.71
		Hadamard	-1.6545	15.23 s	0.12	440.70

Table 7
Time and memory consumption as a function of the number of measurement shots. Simulations for a 5 × 5 2D lattice using the `cuStateVec` backend.

Metric	**QAOA**			**Hadamard**		
Number of shots	1000	5000	10000	1000	5000	10000
Energy value	-1.6526	-1.6518	-1.6508	-1.6545	-1.6511	-1.6553
Wall time (%timeit)	127.3 ms	149.2 ms	175.6 ms	15.23 s	16.23 s	17.46 s
RSS (psutil, MiB)	2234.71	2235.84	2237.59	440.70	441.57	443.20

2. QAOA–Sampling: lower accuracy, lowest runtime, moderate memory;
3. Hadamard–State Vector: exact, highest runtime, low memory;
4. Hadamard–Sampling: lower accuracy, high runtime, lowest memory.

Comparing computational performance between CPU and GPU implementations (using CUDA and `cuStateVec`), the `cuStateVec` library demonstrates superior efficiency across all metrics. For energy calculations, `cuStateVec` outperforms both CPU and native CUDA implementations in terms of computational speed (wall time), memory efficiency (peak memory usage), and numerical accuracy (energy expectation values). The native CUDA implementation shows comparable performance to `cuStateVec` for state-vector simulations but suffers significant performance degradation with sampling-based methods (1000 shots), where it becomes less efficient than both the CPU baseline and the `cuStateVec` implementation.

Author Contributions: Conceptualization, Yuri G. Palii; Methodology, Yuri G. Palii and Alla A. Bogolubskaya; Software, Yuri G. Palii, Alla A. Bogolubskaya and Denis A. Yanovich; Validation, Yuri G. Palii and Alla A. Bogolubskaya; Formal analysis, Yuri G. Palii; Investigation, Yuri G. Palii, Alla A. Bogolubskaya, Denis A. Yanovich; Writing — original draft preparation, Yuri G. Palii; Writing — review and editing, Yuri G. Palii and Alla A. Bogolubskaya; Visualization, Alla A. Bogolubskaya and Denis A. Yanovich; Supervision, Yuri G. Palii; Project administration, Yuri G. Palii and Alla A. Bogolubskaya. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Codes are available by demand from authors.

Acknowledgments: The authors thank Dmitry V. Belyakov, lead programmer at MLIT JINR, and Maxim I. Zuev, research scientist at MLIT JINR, for configuring the `Cirq` library software environment and providing computational technical support.

Conflicts of Interest: The authors declare no conflict of interest.

Declaration on Generative AI: The authors have not employed any Generative AI tools.

References

1. Pedersen, J. W., Lamm, H., Lawrence, S. & Yeter-Aydeniz, K. Quantum Simulation of Finite Temperature Schwinger Model via Quantum Imaginary Time Evolution. *Phys. Rev. D* **108,** 114506. doi:10.1103/PhysRevD.108.114506. arXiv: 2304.01144 `[hep-lat]` (2023).
2. Jordan, S. P., Lee, K. S. M. & Preskill, J. Quantum Algorithms for Quantum Field Theories. *Science* **336,** 1130–1133. doi:10.1126/science.1217069. arXiv: 1111.3633 `[hep-th]` (2012).
3. Abhijith, J. *et al.* Quantum Algorithm Implementations for Beginners. *Quantum* **6,** 881. doi:10.22331/q-2022-12-22-881. arXiv: 1804.03719 `[cs.ET]` (2022).
4. Bharti, K. *et al.* Noisy intermediate-scale quantum (NISQ) algorithms. *Rev. Mod. Phys.* **94,** 015004. doi:10.1103/RevModPhys.94.015004. arXiv: 2101.08448 `[quant-ph]` (2022).
5. Hidary, J. D. *Quantum Computing: An Applied Approach* 2nd. doi:10.1007/978-3-030-23927-6 (Springer Nature Switzerland AG, Cham, Switzerland, 2021).
6. Farhi, E., Goldstone, J. & Gutmann, S. *A Quantum Approximate Optimization Algorithm* 2014. arXiv: 1411.4028 `[quant-ph]`.
7. Lotshaw, P. C. *et al.* Simulations of Frustrated Ising Hamiltonians using Quantum Approximate Optimization. *Phil. Trans. R. Soc. Lond. A* **381,** 20210414. doi:10.1098/rsta.2021.0414. arXiv: 2206.05343 `[quant-ph]` (2022).
8. Ozaeta, A., van Dam, W. & McMahon, P. L. Expectation Values from the Single-Layer Quantum Approximate Optimization Algorithm on Ising Problems. *Quantum Sci. Technol.* **7,** 045036. doi:10.1088/2058-9565/ac88d4. arXiv: 2012.0342 `[quant-ph]` (2022).

9. AI, G. Q. *Cirq: Quantum approximate optimization algorithm for the Ising model* https://quantumai.google/cirq/experiments/qaoa/qaoa_ising. Accessed: 2024-04-01. 2023.
10. Bengtsson, I. & Życzkowski, K. *Geometry of Quantum States. Introduction to Quantum Entanglement* Second. doi:10.1017/9781139030710 (Cambridge University Press, Cambridge, UK, 2017).
11. Palii, Y., Bogolubskaya, A. & Yanovich, D. Quantum Approximation Optimization Algorithm for the Ising Model in an External Magnetic Field. *Phys. Part. Nucl.* **55,** 600–602. doi:10.1134/S1063779624040185 (2024).
12. Belyakov, D. V., Bogolyubskaya, A. A., Zuev, M. I., Palii, Y. G., Podgajny, D. V., Streltsova, O. I. & Yanovich, D. A. *Polygon for quantum computing on the heterogeneous HybriLIT platform* Russian. in *Proc. of the International Conference "Information Technologies and Mathematical Methods" (ITTMM-2024)* (In Russian) (2024), 303.
13. Corporation, N. *NVIDIA cuStateVec: A High-Performance Library for Quantum Circuit Simulation* https://docs.nvidia.com/cuda/cuquantum/custatevec/index.html. Accessed: 2024-04-01. 2023.
14. Bayraktar, H. *et al. NVIDIA cuQuantum SDK, Accelerate quantum computing research* 2023. arXiv: 2308.01999 `[quant-ph]`.
15. Powell, M. J. D. A direct search optimization method that models the objective and constraint functions by linear interpolation. *Advances in Optimization and Numerical Analysis* (eds Gomez, S. & Hennart, J.-P.) 51–67. doi:10.1007/978-94-015-8330-5_4 (1994).
16. Powell, M. J. D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7,** 155–162. doi:10.1093/comjnl/7.2.155 (1964).
17. Nelder, J. A. & Mead, R. A simplex method for function minimization. *Comput. J.* **7,** 308–313. doi:10.1093/comjnl/7.4.308 (1965).
18. Xiang, Y., Sun, D. Y., Fan, W. & Gong, X. G. Generalized simulated annealing algorithm and its application to the Thomson model. *Phys. Lett. A* **233,** 216–220. doi:10.1016/S0375-9601(97)00484-1 (1997).
19. Endres, S. C., Sandrock, C. & Focke, W. W. A simplicial homology algorithm for Lipschitz optimization. *J. Glob. Optim.* **72,** 181–217. doi:10.1007/s10898-018-0645-y (2018).
20. Palii, Y., Belyakov, D., Bogolubskaya, A., Zuev, M. & Yanovich, D. *Simulation of the QAOA algorithm at the JINR quantum testbed* Russian. in *Proc. of the International Conference "Mathematical Problems in Quantum Information Technologies" (MPQIT 2024)* In press (Dubna, JINR, 2024).

Information about the authors

Palii, Yuri G.—Candidate of Physical and Mathematical Sciences, Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: palii@jinr.ru, phone: +7(496)2162235, ORCID: 0000-0001-9000-9794)

Bogolubskaya, Alla A.—Candidate of Physical and Mathematical Sciences, Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: abogol@jinr.ru, phone: +7(496)2164015, ORCID: 0000-0003-4356-8336, Scopus Author ID: 6508333497)

Yanovich, Denis A.—Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: yan@jinr.ru, phone: +7(496)2162552)

УДК 519.6, 530.145.67
PACS 03.67.Ac, 03.67.Lx,
DOI: 10.22363/2658-4670-2025-33-4-440-460 EDN: HYWEXV

Моделирование работы QAOA с использованием квантовых фреймворков Cirq и qsim

Ю. Г. Палий, А. А. Боголюбская, Д. А. Янович

Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. В работе рассмотрено решение задачи поиска состояния с минимальной энергией в модели Изинга с продольным магнитным полем для двух- и трёхмерных решёток различных размеров на квантовом компьютере с использованием квантового приближённого алгоритма оптимизации (QAOA). Базисные состояния квантового регистра соответствуют конфигурациям спинов на пространственной решётке, а гамильтониан модели реализуется с помощью последовательности квантовых вентилей. Среднее значение энергии эффективно измерено с помощью теста Адамара. Работа алгоритма QAOA моделируется для последовательно усложняющихся решёточных конфигураций с применением библиотек `Cirq` и `qsim`. Результаты оптимизации, проведённой градиентным и безградиентными методами, свидетельствуют о предпочтительности последних как с точки зрения моделирования работы, так и с точки зрения использования квантового компьютера. Приведены ключевые аргументы в пользу преимуществ квантовых вычислений для решения данной задачи.

Ключевые слова: квантовые вычисления, квантовый приближённый алгоритм оптимизации (QAOA), модель Изинга, симуляция квантовых вычислений, оптимизация, Cirq, qsim, cuStateVec