Quantum Information Processing (2023) 22:279
https://doi.org/10.1007/s11128-023-04025-x

A universal quantum algorithm for weighted maximum cut and Ising problems

Natacha Kuete Meli[1] · Florian Mannel[1] · Jan Lellmann[1]

Received: 15 March 2023 / Accepted: 12 June 2023 / Published online: 11 July 2023

Abstract
We propose a hybrid quantum-classical algorithm to compute approximate solutions of binary combinatorial problems. We employ a shallow-depth quantum circuit to implement a unitary and Hermitian operator that block-encodes the weighted maximum cut or the Ising Hamiltonian. Measuring the expectation of this operator on a variational quantum state yields the variational energy of the quantum system. The system is enforced to evolve toward the ground state of the problem Hamiltonian by optimizing a set of angles using normalized gradient descent. Experimentally, our algorithm outperforms the state-of-the-art quantum approximate optimization algorithm on random fully connected graphs and challenges D-Wave quantum annealers by producing good approximate solutions. Source code and data files are publicly available (https://github.com/nkuetemeli/UQMaxCutAndIsing).

Keywords Quantum algorithms · Variational quantum circuits · Hybrid quantum circuits · Weighted maximum cut · Ising model

1 Introduction

Quantum computing has emerged as a powerful computation paradigm taking advantage of principles of quantum mechanisms. It involves two computational models that fundamentally differ in their functioning: The adiabatic quantum model is best

✉ Natacha Kuete Meli
meli@mic.uni-luebeck.de

Florian Mannel
mannel@mic.uni-luebeck.de

Jan Lellmann
lellmann@mic.uni-luebeck.de

[1] Institute of Mathematics and Image Computing, University of Lübeck, Maria-Goeppert-Straße 3, 23562 Lübeck, Schleswig-Holstein, Germany

suited for optimization problems, typically cast in quadratic unconstrained binary optimization form. Commercial devices such as D- Wave annealers [1] can already solve combinatorial problems in various fields such as computer vision [2, 3] and database engineering [4, 5]. The universal model of quantum computing, also referred to as the gate-based or circuit model, is more flexible and can potentially implement any classical operation, as shown by Bennett [6]. For selected problems such as Shor's factoring [7] and Grover's search [8] algorithms, strong theoretical convergence properties and drastic speed-up of the universal quantum computing model over classical counterparts could be proven. However, for problem sizes of practical interest, these algorithms still require more resources and run-time than existing universal devices can provide. In the era of noisy intermediate-scale devices, it is a challenging task to find real-world applications of universal quantum computing. Hybrid quantum-classical algorithms are therefore considered to be a promising way to obtain practical quantum supremacy.

The Ising model [9] describes a quantum mechanical system with $n \in \mathbb{N}$ particles or spins that can be in two possible states, i.e., the spin s_i, $i = 1, \ldots, n$ can be in the state ±1 represented by $s_i = (-1)^{q_i}$, $q_i \in \{0, 1\}$. Each spin s_i can interact with some external energy of strength $\mathcal{C}_{ii}$ or with an adjacent spin s_j by a mutual interaction energy $\mathcal{C}_{ij}$. The complete system can be modeled by a general un-directed n vertices graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$ with $\mathcal{S} = \{s_1, \ldots, s_n\}$, $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ and a cost function $\mathcal{C} : \mathcal{E} \to \mathbb{R}$ with $\mathcal{C}(s_i, s_j) := \mathcal{C}_{ij}$ on $\mathcal{E}$. For a quantum system in the state $|q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle$, $q_i \in \{0, 1\}$, the Ising model describes the total energy of the system as being the expectation $\langle \mathbf{C} \rangle := \langle q|\mathbf{C}|q\rangle$ of the $(2^n \times 2^n)$ Hamiltonian

$$\mathbf{C} = \sum_{i=1}^{n} \mathcal{C}_{ii}\mathbf{Z}_i + \sum_{1 \le i < j \le n} \mathcal{C}_{ij}\mathbf{Z}_i\mathbf{Z}_j, \tag{1}$$

where $\mathbf{Z}_k$ denotes the Pauli-$\mathbf{Z}$ operator acting on the kth particle of the system.

The problem is to

$$\begin{aligned} &\underset{|q\rangle}{\text{minimize}} \quad \langle \mathbf{C} \rangle \\ &\text{subject to} \quad |q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle,\ q_i \in \{0, 1\}\,, \end{aligned} \tag{2}$$

i.e., to search for the state $|q^\star\rangle$ of the system with the minimal energy according to the Ising model in Eq. (1). By setting $\mathcal{C}_{ii} = 0$ for all i, the problem reduces to the so-called weighted maximum cut (MaxCut) problem [10].

The observable $\mathbf{C}$ in Eq. (1) is a diagonal matrix and it is straightforward to verify that the expectation $\langle \mathbf{C} \rangle$ is lower-bounded by the smallest eigenvalue $\mathcal{C}_{min}$ of $\mathbf{C}$. Indeed, measuring $\mathbf{C}$ on a quantum system prepared in the superposition state $|\psi\rangle = \sum_{q=0}^{2^n-1} \alpha_q |q\rangle$ yields

$$\langle \mathbf{C} \rangle = \langle \psi | \mathbf{C} | \psi \rangle = \sum_{q=0}^{2^n-1} \left| \alpha_q \right|^2 \langle q | \mathbf{C} | q \rangle \tag{3}$$

$$\geq \sum_{q=0}^{2^n-1} \left| \alpha_q \right|^2 \langle q^\star | \mathbf{C} | q^\star \rangle = \mathcal{C}_{min}. \tag{4}$$

In other words, under all norm-1 vectors, $\langle \mathbf{C} \rangle$ reaches the minimum exactly for $|\psi\rangle$ being an eigenvector of $\mathbf{C}$ to the lowest eigenvalue, a *ground state* $|q^\star\rangle$ of $\mathbf{C}$. Computing $\langle \mathbf{C} \rangle$ for all eigenvectors of $\mathbf{C}$—which amounts to computing the diagonal elements of $\mathbf{C}$—is practically hard, as it amounts to brute-forcing the problem. Instead, in hybrid quantum-classical approaches, one aims for an efficient way to search for parameters $\Theta \in \Gamma$ that solve the variational problem

$$\underset{\Theta \in \Gamma}{\text{minimize}} \quad \mathcal{L}(\Theta), \quad \mathcal{L}(\Theta) := \langle \psi(\Theta) | \mathbf{C} | \psi(\Theta) \rangle, \tag{5}$$

where Γ is a suitable parameterization of the search space.

The Ising spin model is a powerful tool describing ferromagnetism in statistical mechanics [9] as well as many practically relevant NP-problems [11]. MaxCut is no less important. It finds practical applications in varying fields such as computer vision [12], data clustering [13], and communications network design [14]. Thus, determining solutions of the Ising spin model or the MaxCut problem, even approximately, is of great practical interest.

Adiabatic quantum computation (AQC) relies on the adiabatic theorem [15, 16] and solves the Ising problem by performing an adiabatic evolution of the quantum system from the known and easily-prepared ground state of an initial Hamiltonian to that of the problem Hamiltonian. D- Wave [1] quantum annealers are intermediate-scale devices implementing AQC. It is an established fact that the run-time of AQC algorithms scales inversely with the energy gap between the ground state and the first excited state of the system Hamiltonian [15–17]. This introduces the necessity of carefully designing the system Hamiltonian or to use spectral gap amplification techniques [18]. Another workaround is the conception of (hybrid) gate-based algorithms that use moderate quantum resources with outer-loop classical optimization [19].

Quantum approximate optimization algorithms (QAOA), firstly introduced by Fahri et al. [19] and widely discussed in the literature [20–23], mimic the AQC model, but use a low-depth variational circuit capable to run on near-term devices and benefiting from the maturity of classical optimization. QAOA is considered as one of the major candidates for dealing with realistic real-world applications at competitive performance using the universal model. However, in practice, optimizing QAOA parameters appears to be extremely difficult due to a non-convex objective [20, 22, 23]. Also, QAOA is threefold iterative and hence computationally expensive: First, the ansatz itself involves repetitive layers of gates; second, the classical optimization routine used around QAOA often needs to be iterative as the objective function is non-convex; third, evaluating this objective function requires a large number of quantum circuit measurements.

This work aims to facilitate combinatorial optimization on universal quantum computers. We first eliminate the repetitive layers of QAOA and propose an alternative, more effective encoding of the problem on universal quantum hardware. We present a new, easy to implement and low-depth variational circuit that block-encodes the MaxCut or the Ising Hamiltonian in an Hermitian and unitary operator. Measuring this circuit reveals direct information about the loss function Eq. (5). We then derive an optimization routine based on gradient descent for the proposed variational circuit in order to drive the quantum system toward the ground state of the problem Hamiltonian. We experimentally validate the novel algorithm and find that it outperforms the state-of-the-art gate-based QAOA model for solving MaxCut; on the Ising model, it also challenges the specialized D- Wave annealers.

This paper is structured as follows: in Sect. 2 we recall how D- Wave and QAOA solve combinatorial problems; in Sect. 3, we present the new variational quantum circuit and how its parameters are optimized; Sect. 4 provides experimental results and Sect. 5 concludes the work.

2 Related work

2.1 Adiabatic quantum computing

Adiabatic quantum computation (AQC) is an optimization paradigm which relies on the adiabatic theorem [15, 16] saying that the ground state of a quantum mechanical system is the solution of an optimization problem [24]. Theoretically, the evolution of a quantum system of $n \in \mathbb{N}$ particles at a time $t \in [0, T]$, typically re-scaled to $s = t/T \in [0, 1]$, can be described by a Hamiltonian $\mathbf{H}(s)$ on a Hilbert space $\mathcal{H} = \mathbb{C}^{\otimes 2^n}$, with the state of the system given by a unit vector $|\psi(s)\rangle \in \mathcal{H}$.

Industrial devices such as the D- Wave quantum annealer [1] have already been proposed to solve binary combinatorial problems based on the AQC optimization principle. Their main idea consists in *initializing* a quantum system with a Hamiltonian $\mathbf{B} = \sum_{k=1}^{n} \mathbf{X}_k$ whose ground state $|+\rangle^{\otimes n}$ is the perfect superposition state and easy to prepare. As above, $\mathbf{X}_k$ denotes the Pauli-$\mathbf{X}$ operator acting on the kth particle of the system. Then, another Hamiltonian $\mathbf{C}$, the *problem Hamiltonian,* is prepared as in Eq. (1). As the time s evolves from 0 to 1, the initial Hamiltonian $\mathbf{B}$ is transformed into the problem Hamiltonian $\mathbf{C}$, describing a time-dependent system Hamiltonian

$$\mathbf{H}(s) = \mathcal{B}(s) \cdot \mathbf{B} + \mathcal{C}(s) \cdot \mathbf{C}, \tag{6}$$

where $\mathcal{B}$ and $\mathcal{C}$ with $\lim_{s \to 1} \mathcal{B}(s) = 0$ and $\lim_{s \to 1} \mathcal{C}(s) = 1$ are annealing functions. The evolution of the system generated by $\mathbf{H}(s)$ over the time $s \in [0, 1]$ is governed by the Schrödinger equation; its solution defines a time-dependent unitary operator that transforms the ground state of $\mathbf{B}$ into the ground state of $\mathbf{C}$ with high probability if s varies sufficiently slowly [16]. The ground state of $\mathbf{C}$ is the solution of the optimization problem (2).

While being an efficient scheme for combinatorial optimization that has the potential to ultimately supercede classical computers, AQC has a caveat. It is known that the smaller the energy gap between the ground state and the first excited state of the adiabatic Hamiltonian $\mathbf{H}(s)$, the longer the required annealing time for guaranteeing the success of the optimization [15–17]. To overcome this, methods for universal quantum computers that take advantage of efficient classical optimization techniques have been proposed in the form of quantum approximate optimization algorithms [19].

2.2 Quantum approximate optimization algorithms

In quantum approximate optimization algorithms (QAOA, [19]), the problem (2) is *relaxed* to finding a state $|\psi\rangle = \sum_q \alpha_q |q\rangle$ such that $\langle \mathbf{C} \rangle$ is minimized. QAOA solves the MaxCut problem by *trotterizing* the evolution generated by Eq. (6). Its pseudo-code is recapitulated in Algorithm 1: First, the system is prepared in the perfect superposition state $|+\rangle^{\otimes n}$. Then, trotterization consists in repeatedly, say p times, applying to the state the unitaries $\mathbf{U}(\mathbf{C}, \gamma_k) := \exp(-i\gamma_k \mathbf{C})$ and $\mathbf{U}(\mathbf{B}, \beta_k) := \exp(-i\beta_k \mathbf{B})$ generated by the problem Hamiltonian $\mathbf{C}$ and the mixing Hamiltonian $\mathbf{B}$. The (small) time steps $\gamma_k, \beta_k \in \mathbb{R}$, $1 \leq k \leq p$, are optimization parameters in QAOA. The resulting state is called $|\gamma, \beta\rangle$:

$$|\gamma, \beta\rangle := \mathbf{U}(\mathbf{B}, \beta_p)\mathbf{U}(\mathbf{C}, \gamma_p) \cdots \mathbf{U}(\mathbf{B}, \beta_1)\mathbf{U}(\mathbf{C}, \gamma_1)|+\rangle^{\otimes n}. \tag{7}$$

Finally, the state $|\gamma, \beta\rangle$ is measured in the computational basis and is used to evaluate the cost function $\langle \mathbf{C} \rangle := \langle \gamma, \beta | \mathbf{C} | \gamma, \beta \rangle$ and if necessary its derivatives $\nabla_{(\gamma,\beta)} \langle \mathbf{C} \rangle$ and $\nabla^2_{(\gamma,\beta)} \langle \mathbf{C} \rangle$. A classical optimization routine is used to update the parameters $(\gamma, \beta) := (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)$.

Algorithm 1 Quantum Approximate Optimization Algorithms [19]

```
Require: $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$ and $p$
Ensure: $|\psi\rangle = |\gamma, \beta\rangle$
  Initialize the system in the state $|+\rangle = \sqrt{2^{-n}} \sum_{q=0}^{2^n-1} |q\rangle$
  Initialize parameter $(\gamma, \beta) \leftarrow (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)^{init}$
  while stopping criteria not met do
    Prepare $|\gamma, \beta\rangle := \mathbf{U}(\mathbf{B}, \beta_p)\mathbf{U}(\mathbf{C}, \gamma_p) \cdots \mathbf{U}(\mathbf{B}, \beta_1)\mathbf{U}(\mathbf{C}, \gamma_1)|+\rangle^{\otimes n}$
    Measure $|\gamma, \beta\rangle$ in the computational basis
    Compute $\langle \mathbf{C} \rangle := \langle \gamma, \beta | \mathbf{C} | \gamma, \beta \rangle$ and if necessary $\nabla_{(\gamma,\beta)} \langle \mathbf{C} \rangle$, $\nabla^2_{(\gamma,\beta)} \langle \mathbf{C} \rangle$
    Update $(\gamma, \beta) \leftarrow (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)^{new}$ using a classical optimizer
  end while
```

For $p \to \infty$, the results of [19] guarantee that there exist parameters (γ, β) for which measuring $|\gamma, \beta\rangle$ gives the desired ground state $|q^\star\rangle$ with high probability. However, the QAOA objective is difficult to optimize [20, 22, 23]. We believe that this is partially due to the fact that the QAOA ansatz encodes problem information in the argument of exp as phases of the qubits, which is partially lost at the measurement. Another issue of QAOA is that its repetitive layers are still too expensive for running on current and near-term devices. In this work, we propose a quantum circuit that encodes the problem more effectively and does not require the repetitive layers of

QAOA. Adhering to the promising concept of designing hybrid quantum algorithms, we embed the circuit in a classical optimization method to output the desired ground state $|q^\star\rangle$ with high probability.

3 Proposed universal quantum algorithm

Our method builds on the notion of *block-encoding* introduced in [25, 26] that allows to embed non-unitary matrices as the principal block of a unitary operator acting on the quantum system. Block-encoding is typically achieved by enlarging the Hilbert space of the quantum system.

Definition 1 (*Block-encoding* [25, 26]) Let $a, n \in \mathbb{N}$ and $m := a + n$. The $m \times m$ unitary $\mathbf{U}$ is said to be a block-encoding of the $n \times n$ matrix $\mathbf{C}$ if there is $\kappa \in (0, \infty)$ such that

$$\kappa\mathbf{C} = \left[\langle 0|^{\otimes a} \otimes \mathbf{I}_n\right] \mathbf{U} \left[|0\rangle^{\otimes a} \otimes \mathbf{I}_n\right]. \tag{8}$$

By adding a single qubit to the system, our goal is to implement the $(2^{1+n}) \times (2^{1+n})$ *unitary* operator

$$\mathbf{U} := \mathbf{U}(\mathbf{C}, K) := \begin{bmatrix} \sin(\hat{\mathbf{C}}) & \cos(\hat{\mathbf{C}}) \\ \cos(\hat{\mathbf{C}}) & -\sin(\hat{\mathbf{C}}) \end{bmatrix}, \quad \hat{\mathbf{C}} := \mathbf{C}/K, \tag{9}$$

for the Ising Hamiltonian $\mathbf{C}$ from Eq. (1) and a suitably chosen *constant* $K \in \mathbb{R}$. As $\mathbf{C}$ is a diagonal matrix, the sin and cos functions directly apply to the diagonal elements [27, Section 2.1.8]. This allows to encode information about the problem as probability amplitudes of the qubits. Note that because $\mathbf{C}$ is *Hermitian*, $\mathbf{U}$ is Hermitian as well and thus can serve as measurement observable. We use the constant K to re-scale all entries of $\mathbf{C}$ to $[-\pi/2, \pi/2]$, where sin is strictly increasing and invertible. Specially, Eq. (9) block-encodes a bijective transformation of $\mathbf{C}$. For reasons that will become clear in Sect. 3.1, this re-scaling also allows for an efficient implementation of $\mathbf{U}$. We stress that K should not be set to large since $\lim_{K\to\infty} \mathbf{U}(\mathbf{C}, K) = \mathbf{X}^{\otimes(1+n)}$, i.e., for large K the operator $\mathbf{U}$ behaves like a NOT-gate. In Sect. 3.2, we provide an optimization routine for the proposed circuit, we discuss its scalability and complexity in Sect. 3.3, and lastly we provide a suitable choice for K in Sect. 3.4.

3.1 Implementation

For a system prepared in the basis state $|q\rangle$, the cost of the cut q according to the Ising model in Eq. (1) is given by

$$\langle\mathbf{C}\rangle = \sum_{i=1}^{n} (-1)^{q_i} \mathcal{C}_{ii} + \sum_{1\le i<j\le n} (-1)^{q_i+q_j} \mathcal{C}_{ij}. \tag{10}$$

Crucially, for a $(1+n)$-qubit system prepared in the state $|\hat{\psi}, \psi\rangle := |\hat{\psi}\rangle_c \otimes |\psi\rangle$, where $|\hat{\psi}\rangle_c = \alpha|0\rangle_c + \beta|1\rangle_c$ is a 1-qubit register that we call the *cost qubit* and $|\psi\rangle$ is the n-qubit *working register,* if the cost qubit $|\hat{\psi}\rangle_c$ is kept in the state $|0\rangle_c$, it holds for $\mathbf{U}$ defined in Eq. (9) that

$$\langle \mathbf{U} \rangle = \langle \psi | \sin\left[\frac{\mathbf{C}}{K}\right] |\psi\rangle. \tag{11}$$

Thus, if K is chosen such that the entries of the scaled diagonal matrix $\frac{\mathbf{C}}{K}$ fit inside the monotone region of the sine, then minimizing $\langle \mathbf{U} \rangle$ with the cost qubit set to zero is equivalent to minimizing $\langle \mathbf{C} \rangle$.

Observe that $\mathbf{U}$ is a block matrix of diagonal matrices and applying it to the basis state $|0, q\rangle$ gives the same result as applying to the cost qubit $|0\rangle_c$ the controlled (2×2)-operator

$$\mathbf{U}_{2\times 2} := \begin{bmatrix} \sin\langle \hat{\mathbf{C}} \rangle & \cos\langle \hat{\mathbf{C}} \rangle \\ \cos\langle \hat{\mathbf{C}} \rangle & -\sin\langle \hat{\mathbf{C}} \rangle \end{bmatrix}. \tag{12}$$

As derived in “Appendix A,” this operator performs a reflection in the Bloch sphere of the cost qubit about the axis $\vec{n} = (1/2)\left[\cos\langle \hat{\mathbf{C}} \rangle \; 0 \; \sin\langle \hat{\mathbf{C}} \rangle\right]^\top$. Up to an irrelevant phase factor, $\mathbf{U}$ can be written as

$$\begin{aligned} \mathbf{U}(\mathbf{C}, K) &\equiv R_y\left[2\arccos(\sin\langle \hat{\mathbf{C}} \rangle)\right] && \cdot \mathbf{Z} \otimes \mathbf{I}^{\otimes n} \\ &\equiv R_y\left[\pi - 2\langle \hat{\mathbf{C}} \rangle\right] && \cdot \mathbf{Z} \otimes \mathbf{I}^{\otimes n} \\ &\equiv R_y(-2\langle \hat{\mathbf{C}} \rangle) \cdot R_y(\pi) && \cdot \mathbf{Z} \otimes \mathbf{I}^{\otimes n}. \end{aligned} \tag{13}$$

The second equation holds because $\arccos(\sin x) = \pi/2 - x$ for $x \in [-\pi/2, \pi/2]$. The last equation is obtained by applying the identity $R_y(\theta_1 + \theta_2) = R_y(\theta_2) \cdot R_y(\theta_1)$ for rotations in two dimensions, where $\theta_1, \theta_2 \in \mathbb{R}$. By recursively applying this same identity to $R_y(-2\langle \hat{\mathbf{C}} \rangle)$, we find

$$\begin{aligned} \mathbf{U}(\mathbf{C}, K) \equiv & \prod_{i=1}^{n} \mathbf{X}^{q_i} \cdot R_y(-2\hat{\mathcal{C}}_{ii}) \cdot \mathbf{X}^{q_i} \cdot \\ & \prod_{1 \le i < j \le n} \mathbf{X}^{q_i + q_j} \cdot R_y(-2\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X}^{q_i + q_j} \cdot \mathbf{X} \otimes \mathbf{I}^{\otimes n}. \end{aligned} \tag{14}$$

As a result, the weighted sum of Pauli-$\mathbf{Z}$ operators naturally translates into a product of unitary transformations, which is very compatible with the gate-based model of quantum computing. As the basis state $|q\rangle$ is chosen arbitrarily, $\langle \mathbf{U} \rangle$ outputs *sine* transformed costs as derived in Eq. (11) for arbitrary states. For a system prepared in the basis state $|\hat{\psi}, \psi\rangle = |0, q\rangle$, we can even recover the exact cost by $\langle \mathbf{C} \rangle = K \arcsin\langle \mathbf{U} \rangle$. The operator $\mathbf{U}$ can be efficiently implemented using the circuit given in Fig. 1:

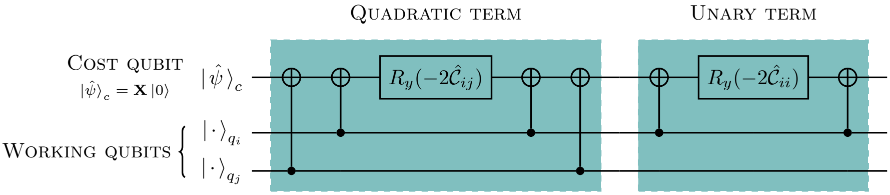

Fig. 1 Implementation of the $(2^{1+n}) \times (2^{1+n})$ operator $\mathbf{U}$. The cost qubit is initialized in the state $|\hat{\psi}\rangle_c = \mathbf{X}|0\rangle$ and is the target of all operations. Note that this does not contradict the idea of keeping it in the $|0\rangle$-state, as the operation $\mathbf{X} = R_y(\pi) \cdot \mathbf{Z}$ implements the last part of $\mathbf{U}$ [Eq. (13)]. **(Left)** For each coupling edge of weight $\mathcal{C}_{ij}$ between nodes q_i and q_j, rotate the cost qubit by $R_y(-2\hat{\mathcal{C}}_{ij})$ if their corresponding working qubits are in the same state, else by $R_y(2\hat{\mathcal{C}}_{ij}) = \mathbf{X} \cdot R_y(-2\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X}$. **(Right)** For each unary edge of weight $\mathcal{C}_{ii}$ involving node q_i, rotate the cost qubit by $R_y(-2\hat{\mathcal{C}}_{ii})$ if its corresponding working qubit is in the $|0\rangle$-state, else by $R_y(2\hat{\mathcal{C}}_{ii}) = \mathbf{X} \cdot R_y(-2\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X}$

- We initialize the cost qubit in the state $|\hat{\psi}\rangle_c = R_y(\pi) \cdot \mathbf{Z}|0\rangle = \mathbf{X}|0\rangle$.
- For each weight $\mathcal{C}_{ij}$ between two nodes q_i and q_j, we rotate the cost qubit by $R_y(-2\hat{\mathcal{C}}_{ij})$ if the corresponding working qubits are in the same state, and by $R_y(2\hat{\mathcal{C}}_{ij}) = \mathbf{X} \cdot R_y(-2\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X}$ if they are not.
- For each unary weight $\mathcal{C}_{ii}$ involving node q_i, we rotate the cost qubit by $R_y(-2\hat{\mathcal{C}}_{ii})$ if the corresponding working qubit is in the $|0\rangle$-state, and by $R_y(2\hat{\mathcal{C}}_{ii}) = \mathbf{X} \cdot R_y(-2\hat{\mathcal{C}}_{ii}) \cdot \mathbf{X}$ if it is not.

Whenever the unary costs satisfy $\mathcal{C}_{ii} = 0$ for all i, we refer to Eq. (14) as the Universal Quantum Maximum Cut (UQMaxCut) model, else as the Universal Quantum Ising (UQIsing) model.

3.2 Optimization

3.2.1 Workflow

The overall workflow of our algorithm is presented in Fig. 2. The complete circuit consists of 3 registers: a 1-qubit register containing an ancilla qubit, another 1-qubit register for the cost qubit, and an n-qubit register for the working qubits encoding the variables of the problem.

First, the working qubits are rotated by a set of angles $\Theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$, constructing the ansatz

$$|\psi(\Theta)\rangle := R_y(\theta_1) \otimes \cdots \otimes R_y(\theta_n)|\psi\rangle, \tag{15}$$

from a system previously prepared in the state $|\psi\rangle$. Here, the qubit q_i, representing the ith node of the graph, is rotated by θ_i around the y-axis. Next, a Hadamard sandwich involving a controlled version of the $\mathbf{U}$-operator is applied to the ancilla qubit. Finally, according to the principle of implicit measurement of quantum computing [27, Section 4.4] stating that *all qubits that are not measured at the end of a quantum circuit can*

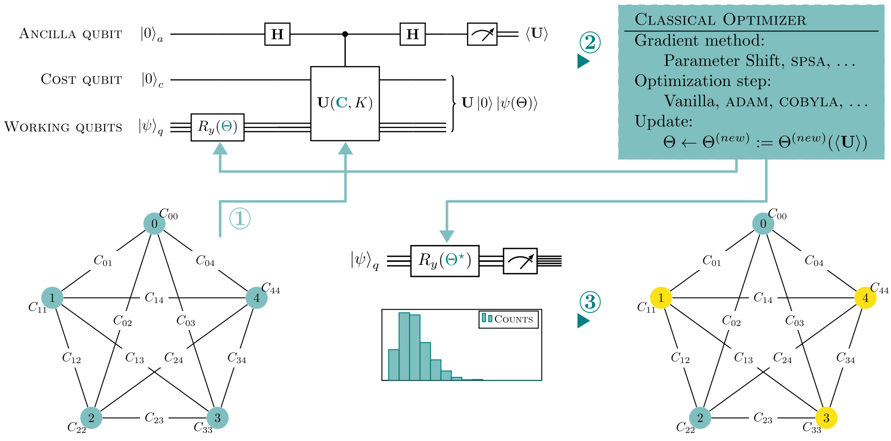
Ancilla qubit $|0\rangle_a$
H
H
$\langle \mathbf{U} \rangle$
Cost qubit $|0\rangle_c$
$\mathbf{U}(\mathbf{C}, K)$
$\mathbf{U}\,|0\rangle\,|\psi(\Theta)\rangle$
Working qubits $|\psi\rangle_q$
$R_y(\Theta)$
②
Classical Optimizer
Gradient method:
Parameter Shift, SPSA, …
Optimization step:
Vanilla, ADAM, COBYLA, …
Update:
$\Theta \leftarrow \Theta^{(new)} := \Theta^{(new)}(\langle \mathbf{U} \rangle)$
①
$|\psi\rangle_q$
$R_y(\Theta^\star)$
Counts
③
0 1 2 3 4
C_{00} C_{11} C_{22} C_{33} C_{44}
C_{01} C_{04} C_{14} C_{02} C_{03} C_{12} C_{13} C_{24} C_{34} C_{23}

Fig. 2 Complete workflow of our proposed algorithm for the Universal Quantum MaxCut and Ising Model (UQMaxCut and UQIsing). First ①, the cost (node and edge weight) information for the given graph is used to implement the operator $\mathbf{U} := \mathbf{U}(\mathbf{C}, K)$ that block-encodes the problem Hamiltonians. This operator is applied to a trial variational quantum state $|\psi(\Theta)\rangle$ tensored with the cost qubit. Second ②, using the principle of implicit measurement, the expectation value $\langle \mathbf{U} \rangle$, which equals the cost $\mathcal{L}(\Theta)$, is approximated by measuring several copies of the circuit. This computed expectation and eventually its gradient are iteratively used in a classical optimization routine that drives the parameterized state $|\psi(\Theta)\rangle$ toward the state $|\psi(\Theta^\star)\rangle$ that potentially gives the global minimal cost value. Finally ③, the optimal state $|\psi(\Theta^\star)\rangle$ is measured in the computational basis and the most frequently measured state corresponds to the desired optimal cut of the input graph

be assumed to be measured, only the ancilla qubit is measured, leaving the cost and working qubits in the state $\mathbf{U}|0, \psi(\theta)\rangle = |0\rangle_c \otimes \sin(\hat{\mathbf{C}})|\psi(\theta)\rangle + |1\rangle_c \otimes \cos(\hat{\mathbf{C}})|\psi(\theta)\rangle$.

Our goal is to solve the variational problem

$$\underset{\Theta \in \mathbb{R}^n}{\text{minimize}} \quad \mathcal{L}(\Theta), \quad \mathcal{L}(\Theta) = \langle 0, \psi(\Theta)|\mathbf{U}|0, \psi(\Theta)\rangle, \tag{16}$$

i.e., to find a set of angles $\Theta^\star$ such that $|\psi(\Theta^\star)\rangle = |q^\star\rangle$. The cost $\mathcal{L}(\Theta)$ can be efficiently calculated by $\mathcal{L}(\Theta) = p(0) - p(1)$, where $p(0)$ and $p(1)$ are the probabilities of measuring the ancilla qubit in the $|0\rangle$ and $|1\rangle$ state, see "Appendix B" for the proof. This allows us to compute the expectation $\langle \mathbf{U} \rangle$ without having to measure $\langle 0, \psi(\Theta)|$ needed for the scalar product. For the optimization, several approaches [28, 29] have fortunately been developed for evaluating or approximating gradients and improving optimization on quantum computers.

3.2.2 Parameter shift rule

The parameter shift rule [28] is a simple but exact method for evaluating the analytical gradient of a function given in the form of an expectation as in Eq. (16) on quantum hardware. For computing the partial derivative with respect to the parameter θ_i, it uses two function evaluations with the parameter θ_i shifted by $\pm\pi/2$:

$$\frac{\partial}{\partial\theta_i}\mathcal{L}(\Theta) = \frac{1}{2}\Big[\mathcal{L}(\Theta_{i+}) - \mathcal{L}(\Theta_{i-})\Big], \tag{17}$$

where $\Theta_{i\pm} := (\theta_1, \dots, \theta_{i-1}, \theta_i \pm \pi/2, \theta_{i+1}, \dots, \theta_n)$. The parameter shift rule uses the same circuit as the function evaluation, but allows to compute the *exact* gradient. Its drawback is that it requires $2n$ function evaluations to compute the gradient of $\mathcal{L}$.

3.2.3 Update rule

The circuits (UQMaxCut, UQIsing) are optimized by normalized gradient descent with decreasing step size. Normalized gradient descent has recently been proposed by Suzuki et al. [30] as a powerful optimization strategy for variational quantum algorithms. Specifically, in [30] it is demonstrated experimentally that normalized gradient steps are more effective in escaping non-global minima than gradient steps. It is also known [31] that normalized gradient descent evades saddle points. At each iteration k, our update rule reads

$$\Theta^{(k)} \leftarrow \Theta^{(k-1)} - \left[\frac{\pi n}{2}\right]^{1/2} \exp\left[-\frac{4k^2}{k_{max}^2}\right] \cdot \frac{\nabla_\Theta \mathcal{L}(\theta^{(k-1)})}{\|\nabla_\Theta \mathcal{L}(\theta^{(k-1)})\|_2}. \tag{18}$$

The design of the update rule Eq. (18) is motivated by the following consideration: We know that we produce bit-strings by either flipping $|q_i\rangle$ or not, thus $\theta_i^\star = \ell\pi, \ell \in \mathbb{Z}$. At iteration $k = 0$, the update rule Eq. (18) allows each θ_i to get updated to $\theta_i^1 \leftarrow \theta_i^0 \pm \pi \cdot g_i/2$, where $g_i \in [-1, 1]$ is the normalized contribution of θ_i^0 in the loss $\mathcal{L}(\Theta)$. Subsequently, we let the step size decay exponentially to zero when approaching the maximum number of iterations k_{max}. Given the noisy nature of quantum measurement, it is helpful that this frees us from the difficult task of determining a stopping condition for the algorithm.

3.3 Scalability and computational complexity

For a given graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$, the circuit construction presented in Fig. 1 requires at most one NOT-gate, $|\mathcal{E}|$ single-qubit rotation gates and $4|\mathcal{E}| - 2|\mathcal{S}|$ CNOT gates. Note that since the edges can be treated in arbitrary order, two consecutive CNOT gates that have the same control qubit cancel each other out as their product is the identity, further reducing the number of required CNOT gates.

The controlled-$\mathbf{U}(\mathbf{C}, K)$ gate appearing in Fig. 2 and conditioned by the ancilla qubit $|\cdot\rangle_a$ can be fully decomposed into single qubit rotations and CNOT gates without using any Toffoli gates. To see this, note that it can be expressed as

$$\begin{aligned}\text{controlled-}\mathbf{U}(\mathbf{C}, K) &= \mathbf{I} \otimes \mathbf{U}(\mathbf{C}, K)^a \\ &\equiv \mathbf{I} \otimes \prod_{i=1}^{n} \mathbf{X}^{q_i} \cdot [R_y(-2\hat{\mathcal{C}}_{ii})]^a \cdot \mathbf{X}^{q_i}\cdot\end{aligned} \tag{19}$$

Table 1 Resources and complexity comparison of a depth p conventional QAOA and our UQIsing model on a graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$

	QAOA	UQIsing [Ours]
# qubits	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}\rvert + 2$
# CNOT gates	$p \cdot (2\,\lvert\mathcal{E}\rvert - 2\lvert\mathcal{S}\rvert)$	$1 + 6\lvert\mathcal{E}\rvert - 2\lvert\mathcal{S}\rvert$
# single qubits rotations	$p \cdot (\lvert\mathcal{E}\rvert + \lvert\mathcal{S}\rvert)$	$2\,\lvert\mathcal{E}\rvert$
# Hadamard gates	$\lvert\mathcal{S}\rvert$	2
Qubit connectivity	Graph-dependent: $\lvert\mathcal{E}\rvert - \lvert\mathcal{S}\rvert$	One-to-all: $\lvert\mathcal{S}\rvert + 1$

Recall that the set $\mathcal{E}$ entails all the edges of the graph, i.e., the quadratic and unary ones. The set $\mathcal{S}$ is the set of vertices. Our construction requires less quantum and classical resources than QAOA for $p \geq 3$

$$\prod_{1\leq i<j\leq n} \mathbf{X}^{q_i+q_j} \cdot [R_y(-2\hat{\mathcal{C}}_{ij})]^a \cdot \mathbf{X}^{q_i+q_j} \cdot \mathbf{X} \otimes \mathbf{I}^{\otimes n}. \tag{20}$$

In particular, in order to control the complete $\mathbf{U}(\mathbf{C}, K)$ gate, it suffices to control only the rotation gate R_y.

Finally, the controlled rotation itself can be decomposed into two CNOT gates and two single qubit rotations as

$$[R_y(\theta)]^a = \mathbf{X}^a \cdot R_y(-\theta/2) \cdot \mathbf{X}^a \cdot R_y(\theta/2). \tag{21}$$

Table 1 recapitulates the main differences between UQIsing and a conventional QAOA of depth p. It shows that for $p \geq 3$, QAOA requires more quantum resources than our UQIsing model. Further, physically mapping the QAOA ansatz onto the quantum hardware has to take into account a graph-dependent qubit connectivity, while our method, independently of the input graph, requires that one qubit (the cost qubit) is connected to all other qubits (ancilla and working qubits).

3.4 Impact of the K-rescaling

When applying the method, it is important to appropriately choose the constant $K \in \mathbb{R}$ for re-scaling the costs $\hat{\mathbf{C}} = \mathbf{C}/K$. The goal is to fix K such that $\text{diag}(\hat{\mathbf{C}}) \in [-\pi/2, \pi/2]^{2^n}$, as this guarantees that $\sin(\text{diag}(\hat{\mathbf{C}}))$ preserves the order of the original costs in $\text{diag}(\mathbf{C})$.

Although it is tempting to choose $K \gg \mathcal{C}_{max} := \max_k \mathbf{C}_{kk}$, where $\mathbf{C}_{kk}$ denotes the kth diagonal element, recall that in Eq. (9) we have $\lim_{K\to\infty} \mathbf{U}(\mathbf{C}, K) = \mathbf{X}^{\otimes(n+1)}$, i.e., the larger K, the more shots are required to accurately measure the entries of $\sin(\hat{\mathbf{C}})$. Fortunately, in many problems an upper bound to the maximal cost $\mathcal{C}_{max}$ can be computed from the original weights without knowing $\mathbf{C}$. For example, choosing

$$K = \frac{2}{\pi} \cdot \mathcal{C}_{max} := \frac{2}{\pi} \cdot \left[\sum_{i=1}^{n} |\mathcal{C}_{ii}| + \sum_{1\leq i<j\leq n} |\mathcal{C}_{ij}| \right] \tag{22}$$

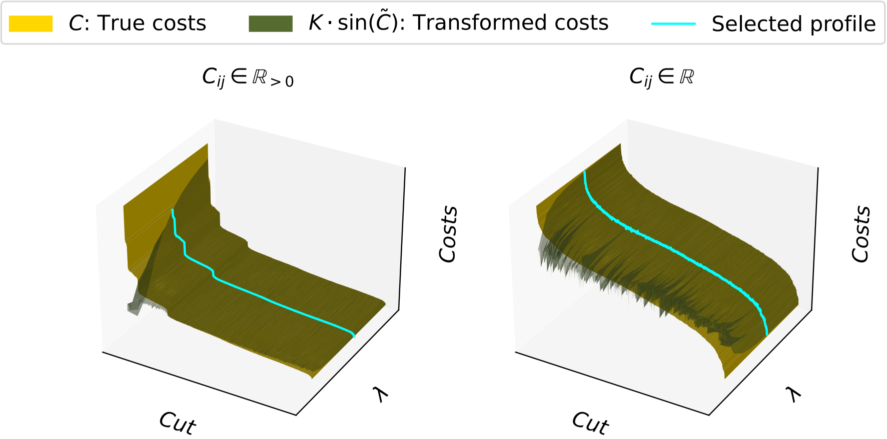

Fig. 3 Transforming the true costs $\mathrm{diag}(\mathbf{C})$ into $\mathrm{diag}(\sin(\mathbf{C}/K))$, where $K = \lambda \cdot \mathcal{C}_{max}$, c.f. Eq. (22). The true costs are sorted by their values and the x-axis represents the sorted cuts in the decimal basis. The y-axis showcases the influence of $0.1 \leq \lambda \leq 1$. The z-axis reports the costs, re-scaled to $[0, 1]$. The transformed costs, sorted by the sorted arguments of the true costs, show whether the order is preserved under the influence of λ. Example: If $\mathbf{C} = [2, 5, 7, 3]$ and, for some value of K, $\hat{\mathbf{C}} = [1, 3, 4, 2]$, then $\mathrm{sort}(\mathbf{C}) = [2, 3, 5, 7]$, $\mathrm{argsort}(\mathbf{C}) = [0, 3, 1, 2]$ and $\hat{\mathbf{C}}_{\mathrm{sorted}} = [1, 2, 3, 4]$. For each value of λ, the displayed costs are the averages over 10 instances of a 10-nodes fully-connected graph. Left: Graph instances with only positive edge weights. Right: Graph instances with both positive and negative edge weights. As expected, the more λ and thus K grows, the better the order of the true cost agrees with the order of the transformed costs. Also, for fixed λ the difference between the two costs is the largest when the true cost is large in absolute value. The cyan-marked line is the profile that we selected for the experiments; it indicates the value $\lambda = 2/\pi$, which is the minimum λ that guarantees an error-free transformation, cf. Eq. (22)

guarantees an error-free transformation of the initial problem (5) into the equivalent problem (16). As presented for 10-node graph examples in Fig. 3 , the choice of smaller values for K entails that the approximation of $\arccos(\sin x)$ by $\pi/2 - x$, used in Eq. (13), is inaccurate, with more pronounced error in the largest absolute entries of $\mathrm{diag}(\hat{\mathbf{C}})$. Notably, the solution of the transformed problem will only be a solution of the initial problem as long as $\arg\min_k \hat{\mathbf{C}}_{kk} = \arg\min_k \mathbf{C}_{kk}$.

4 Experimental results

In order to validate the practical usefulness of UQMaxCut and UQIsing, we benchmark against two state-of-the-art approaches for solving binary combinatorial optimization with quantum computing: QAOA for the gate-based model and D- Wave solvers for the adiabatic model. Random graphs in the experiments are generated using the PYTHON language package NETWORKX [32]. The unary and quadratic edge weights are all randomly and uniformly chosen in the range [1, 10] and the graphs are all fully connected. The gate-based circuits in the experiments (UQMaxCut, UQIsing, QAOA) are implemented in PYTHON and simulated in a noise-free framework using the QISKIT library and the IBM-QASM simulator [33]. For the adiabatic model, D- Wave solvers that run on the actual quantum hardware are used. D- Wave quantum annealers are made available through the leap quantum cloud service [34], and the D- Wave quantum algorithms can be implemented in Python using the Ocean software [35]. We perform 1024 measurement shots for all gate-based algorithms. On D- Wave [1], the experiment is run with the default annealing time of $20\mu s$ and 50 sample reads on the advantage topology.

4.1 Benchmark metrics

We denote by $|q^\star\rangle$ the ground-truth global minimizer and by $|\psi^\star\rangle$ the ground-state proposal of each method (UQMaxCut/UQIsing, QAOA and D- Wave). In the experiments, we adopt the following two metrics to evaluate the performance of the methods:

- The approximation ratio

$$r(\psi^\star) := \frac{\langle\psi^\star|\mathbf{C}|\psi^\star\rangle - \mathcal{C}_{max}}{\mathcal{C}_{min} - \mathcal{C}_{max}} \tag{23}$$

  informs about the quality of the result, i.e., how confident the method is with its solution proposal and how far the cost of this proposal is from the minimal cost $\mathcal{C}_{min} := \min_k \mathbf{C}_{kk}$, cf. [36]. All the terms appearing in r are classically evaluated. It holds $0 \leq r(\psi^\star) \leq 1$ and $r(\psi^\star) = 1$ iff $\psi^\star = q^\star$.
- The approximation index

$$i(\psi^\star) := \mathbf{1}_{\langle q^\star|\mathbf{C}|q^\star\rangle = \langle q_{max}|\mathbf{C}|q_{max}\rangle} \tag{24}$$

  is a Boolean variable that indicates whether the most likely state $|q_{max}\rangle$ obtained with probability $|\alpha_{max}|^2$ is the desired state $|q^\star\rangle$. It holds $i(\psi^\star) = 1$ if $\langle q^\star|\mathbf{C}|q^\star\rangle = \langle q_{max}|\mathbf{C}|q_{max}\rangle$ and 0 otherwise. Note that this differs from the usual approach of D- Wave, where the sampled state with the minimal energy is regarded as the best solution proposal.

4.2 Benchmarking UQMaxCut against QAOA and D-Wave

4.2.1 Symmetric solutions and entanglement

Before discussing the results on the MaxCut problem, it is important to notice that for MaxCut, solutions always exist in symmetric pairs. Specifically, for a basis state solution $|q^\star\rangle = \bigotimes_{i=1}^n |q_i^\star\rangle$, $q_i^\star \in \{0, 1\}$, the basis state $|\bar{q}^\star\rangle := \bigotimes_{i=1}^n |1 - q_i^\star\rangle$ is also a solution. This feature can be enforced by introducing the entanglement circuit given in Fig. 4after the rotation layer in Fig. 2. The circuit has the matrix representation

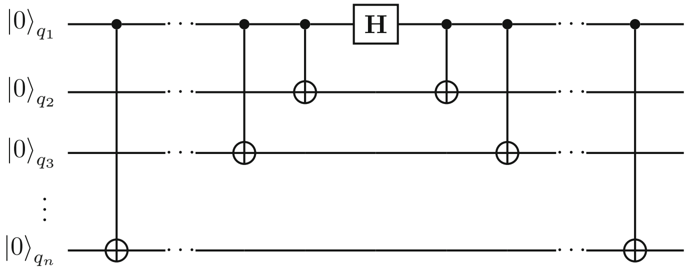

Fig. 4 Entanglement circuit to enforce symmetric solutions for MaxCut. Using this circuit, the optimization is performed only on the last $n-1$ variables and the qubit for the node q_1 is kept constant in the state $|0\rangle$ or $|1\rangle$

$$E = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & & & 1 \\ & \ddots & & & \iddots & \\ & & 1 & 1 & & \\ & & 1 & -1 & & \\ & \iddots & & & \ddots & \\ 1 & & & & & -1 \end{bmatrix}. \tag{25}$$

Also, this entanglement allows to optimize over $n-1$ angles instead of n. Without entanglement our method just outputs one of the two possible solutions. In the experiments, UQMaxCut is used without entanglement; if the algorithm outputs either one of the two solutions, we set the approximation index to 1.

4.2.2 Benchmark results

For the outer optimization algorithm of the UQMaxCut circuit, we use normalized gradient descent as described in Section 3.2. Other optimization algorithms such as vanilla gradient descent or adaptive moment estimation (ADAM [37]) could be used as well. However, it proved difficult in the experiments to find suitable step sizes for those methods, so we leave them for future research.

The QAOA layers depth in the experiments is set to $p = \lceil n/2 \rceil$ to allow for a fair comparison, as then both QAOA and UQMaxCut optimize over approximately n real variables. We also attempted to optimize QAOA using the same optimizer as for UQMaxCut, but the results were not competitive. Hence, we also show the results of QAOA when using a gradient-free optimizer; we used the COBYLA solver [38] available in the SCIPY library [39].

The results are presented in Fig. 5for fully connected graphs of $n = 3, 5$ and 10 nodes with strictly positive edge weights. For each n, the results are averaged over 20 graph instances and all algorithms are tested on the same instances. The angles for UQMaxCut and QAOA are all initialized to 0.

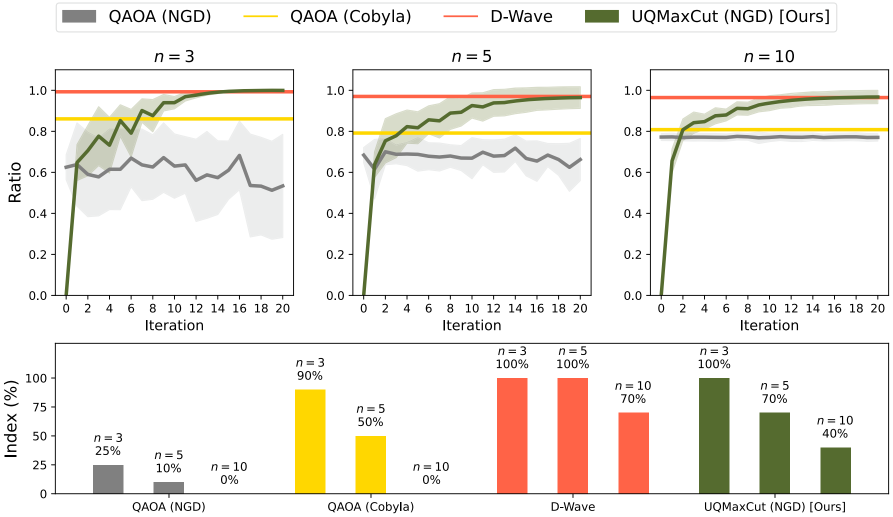

Fig. 5 Experimental comparison of the proposed Universal Quantum MaxCut (UQMaxCut) algorithm with QAOA and D- Wave. Results are shown for fully connected graphs of $n = 3, 5$ and 10 nodes with strictly positive edge weights. They are averaged over 20 random graph instances for each n. Top: The approximation ratio. The lines represent the averaged ratios over the 20 instances and the shaded areas indicate the standard deviations. Bottom: The approximation index, i.e., the percentage of instances where a global solution is found. All gate-based methods hardly return a global solution for larger n. Yet, in contrast to QAOA and Competitively with D- Wave, the proposed UQMaxCut consistently returns very good approximate solutions, i.e., points whose function values are very close to the global minimal function value

The approximation ratio for the three methods (QAOA, D- Wave, UQMaxCut) is not adversely affected by the number of variables n, but the approximation index drops sharply as the size of the problem increases. The gradient-free Cobyla-optimized version of QAOA performs much better than QAOA with gradient descent. The latter is on average as good as D- Wave regarding the approximation ratio. We conjecture that the gradient-based optimization of QAOA often gets trapped by saddle points of the QAOA loss function landscape. In contrast, UQMaxCut clearly outperforms the two QAOA variants and can compete with D- Wave in producing good approximate solutions. Furthermore, the approximation index demonstrates that it returns a global minimizer significantly more often than QAOA and less often than D- Wave whose architecture is specifically designed to solve such problems.

4.3 Benchmarking UQIsing against D-Wave

For the Ising model, we benchmark the proposed UQIsing algorithm against the D-Wave annealer, the adiabatic quantum computer specialized in solving this type of problems. The variational circuit for UQIsing is optimized in the same way as for UQMaxCut, see Sect. 3.2, with the exception that the initial angles are set to $\pi/2$ instead of 0. The results are depicted in Fig. 6.

Notable differences in performance between UQIsing and D- Wave are consistent with the MaxCut experiments. Specifically, its high approximation ratio, similar to that

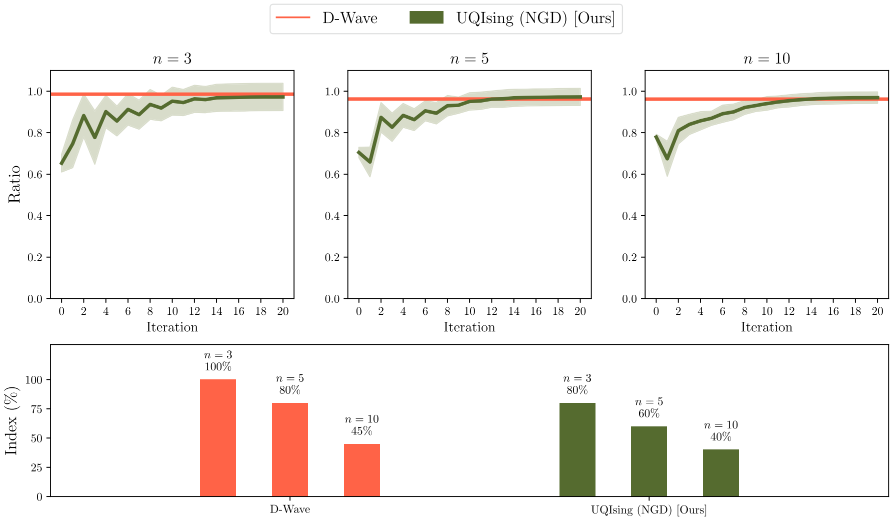

Fig. 6 Experimental comparison of the proposed Universal Quantum Ising Model (UQIsing) algorithm with D- Wave. Results are shown for fully connected graphs of $n = 3, 5$ and 10 nodes with strictly positive edge weights. They are averaged over 20 random graph instances for each n. Top: The approximation ratio. The lines represent the averaged ratios over the 20 instances and the shaded areas indicate the standard deviations. Bottom: The approximation index, i.e., the percentage of instances where a global solution is found. Our method can compete with D- Wave solvers in predicting approximate solutions and finding the global minimum for moderate n

of D- Wave, indicates that UQIsing always produces either globally optimal solutions or extremely good approximations thereof. On the other hand, the approximation index shows that D- Wave identifies a globally optimal solution more often than UQIsing.

5 Conclusion

We have presented a new low-depth quantum circuit to tackle two important combinatorial problems on universal quantum machines. The resulting Universal Quantum MaxCut (UQMaxCut) approach outperforms the state-of-the-art quantum approximate optimization algorithms (QAOA) by the lower depth, by the computed approximation ratios and by a higher probability of outputting optimal solutions. It also challenges the D- Wave-quantum annealers that are specifically designed to solve such combinatorial problems; on the MaxCut as well as on the Ising spin model, UQMaxCut, respectively, UQIsing can compete with D- Wave in producing globally optimal solutions.

We believe that the proposed approach enables the design of new methods for solving practically-sized problems on universal quantum machines. Inspired by the novel operator $\mathbf{U}$, future work should focus on designing fully universal algorithms without the classical outer optimization loop, replacing the latter with fully universal methods like for example Grover's search [8].

Acknowledgements The authors thank IBM and D-Wave for the freely available cloud devices and simulators which helped testing the algorithms.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The authors declare that the data supporting the findings of this study are random graphs that can be generated using the PYTHON language package NETWORKX [32].

Code Availability PYTHON source code for the algorithm and experiments is available at https://github.com/nkuetemeli/UQMaxCutAndIsing.

Declarations

Conflict of interest The authors declare no financial or non-financial competing interests.

Appendix A: Geometric interpretation of $\mathbf{U}_{2\times 2}$

We want to show that the operator

$$\mathbf{U}_{2\times 2} = \begin{bmatrix} \sin\langle\hat{\mathbf{C}}\rangle & \cos\langle\hat{\mathbf{C}}\rangle \\ \cos\langle\hat{\mathbf{C}}\rangle & -\sin\langle\hat{\mathbf{C}}\rangle \end{bmatrix} \tag{A1}$$

is up to a phase factor a reflection in the Bloch sphere about the axis $\vec{n} = (1/2)\left[\cos\langle\hat{\mathbf{C}}\rangle \; 0 \; \sin\langle\hat{\mathbf{C}}\rangle\right]^\top$. More precisely, we show that

$$\mathbf{U}_{2\times 2} \equiv R_y\left[2\arccos(\sin\langle\hat{\mathbf{C}}\rangle)\right] \cdot \mathbf{Z}. \tag{A2}$$

Proof For $\sin\langle\hat{\mathbf{C}}\rangle = \pm 1$, we have $\mathbf{U}_{2\times 2} = \pm\mathbf{Z}$, so Eq. (A2) holds. In the remainder of the proof, we can therefore assume $\sin\langle\hat{\mathbf{C}}\rangle \neq \pm 1$. The matrix $\mathbf{U}_{2\times 2}$ has the spectral decomposition $\mathbf{U}_{2\times 2} = \mathbf{X}^\top \boldsymbol{\Lambda} \mathbf{X}$, where $\boldsymbol{\Lambda} = \mathrm{diag}([-1 \;\; 1])$ is the diagonal matrix of eigenvalues of $\mathbf{U}_{2\times 2}$ and $\mathbf{X} = \left[\,|\psi\_\rangle \mid |\psi_+\rangle\,\right]$ is the matrix containing its eigenvectors

$$|\psi_+\rangle = \frac{1}{\sqrt{2 - 2\sin\langle\hat{\mathbf{C}}\rangle}} \begin{bmatrix} -\cos\langle\hat{\mathbf{C}}\rangle \\ \sin\langle\hat{\mathbf{C}}\rangle - 1 \end{bmatrix} \tag{A3}$$

$$\text{and} \qquad |\psi\_\rangle = \frac{1}{\sqrt{2 + 2\sin\langle\hat{\mathbf{C}}\rangle}} \begin{bmatrix} -\cos\langle\hat{\mathbf{C}}\rangle \\ \sin\langle\hat{\mathbf{C}}\rangle + 1 \end{bmatrix}. \tag{A4}$$

Let $\mathbf{\Sigma} = \operatorname{diag}([\pi \ 2\pi])$, so $\mathbf{\Lambda} = \exp(i\mathbf{\Sigma})$. The generalization of the exponential map to normal matrices [27, Section 2.1.8] implies that $\mathbf{U}_{2\times 2} = \exp(i\mathbf{X}^\top \mathbf{\Sigma}\mathbf{X})$. Now we express $\mathbf{X}^\top \mathbf{\Sigma}\mathbf{X}$ in the basis formed by the Pauli matrices:

$$\mathbf{X}^\top \mathbf{\Sigma}\mathbf{X} = \frac{1}{2}\begin{bmatrix} 3\pi + \pi \sin\langle \hat{\mathbf{C}} \rangle & \pi \cos\langle \hat{\mathbf{C}} \rangle \\ \pi \cos\langle \hat{\mathbf{C}} \rangle & 3\pi - \pi \sin\langle \hat{\mathbf{C}} \rangle \end{bmatrix} \tag{A5}$$

$$= \frac{3\pi}{2} \cdot \mathbf{I} + \frac{\pi}{2} \cos\langle \hat{\mathbf{C}} \rangle \cdot \mathbf{X} + 0 \cdot \mathbf{Y} + \frac{\pi}{2} \sin\langle \hat{\mathbf{C}} \rangle \cdot \mathbf{Z}. \tag{A6}$$

We can thus rewrite $\mathbf{U}_{2\times 2}$ as $\mathbf{U}_{2\times 2} = \exp(i\alpha)\exp(i\vec{v} \cdot \vec{\mathbf{\Sigma}})$, where $\alpha = 3\pi/2$ and $\vec{v} = (1/2)\left[\pi \cos\langle \hat{\mathbf{C}} \rangle \ 0 \ \pi \sin\langle \hat{\mathbf{C}} \rangle\right]^\top$. The notation $\vec{v} \cdot \vec{\mathbf{\Sigma}} := v_x \cdot \mathbf{X} + v_y \cdot \mathbf{Y} + v_z \cdot \mathbf{Z}$ denotes a linear combination of Pauli matrices. With a slight change of variables $\theta = -2\|\vec{v}\|_2 = -\pi$ and $\vec{n} = \vec{v}/\|\vec{v}\|_2 = \left[\cos\langle \hat{\mathbf{C}} \rangle \ 0 \ \sin\langle \hat{\mathbf{C}} \rangle\right]^\top$, this is equivalent to $\mathbf{U}_{2\times 2} = \exp(i\alpha)\exp(-i(\theta/2)\vec{n} \cdot \vec{\mathbf{\Sigma}})$.

The global phase $\exp(i\alpha)$ can be ignored in the implementation as it has no effect on the measurement. The exponential term $\exp(-i(\theta/2)\vec{n} \cdot \vec{\mathbf{\Sigma}})$ corresponds to a rotation of angle θ of the Bloch sphere about the axis $\vec{n}$. As $\vec{n}$ lies in the xz-plane and we have $\theta = -2\|\vec{v}\|_2 = -\pi$, it follows that $\mathbf{U}_{2\times 2}$ is a reflection about $\vec{n}$. Equivalently, Eq. (A2) holds.

Appendix B: Implicit measurement

Let $\mathbf{U}$ be a Hermitian and unitary operator, so $\mathbf{U}$ can be used both as quantum gate and measurement observable. According to the implicit measurement principle of quantum computation [27, Section 4.4], *all qubits that are not measured at the end of a quantum circuit can be assumed to be measured.* Specially, the measurement of the first qubit of the circuit in Fig. 7performs a measurement of the observable $\mathbf{U}$ for a system prepared in the state $|\psi_{in}\rangle$.

Proof As $\mathbf{U}$ is both Hermitian and unitary, its eigenvalues are both real and unitary, and therefore either +1 or −1. Then, the operators $\mathbf{P}_\pm = \frac{1}{2}(\mathbf{I} \pm \mathbf{U})$ are orthogonal projectors into the eigenspaces of $\mathbf{U}$ to the eigenvalues ±1. Evidently, $\mathbf{U}$ can be expressed as $\mathbf{U} = \mathbf{P}_+ - \mathbf{P}_-$. Before measurement, the system is in the state

$$\frac{1}{2}\Big[(|0\rangle + |1\rangle) \otimes \mathbf{I}|\psi_{in}\rangle + (|0\rangle - |1\rangle) \otimes \mathbf{U}|\psi_{in}\rangle\Big] \tag{B7}$$

$$= |0\rangle \otimes \mathbf{P}_+|\psi_{in}\rangle + |1\rangle \otimes \mathbf{P}_-|\psi_{in}\rangle. \tag{B8}$$

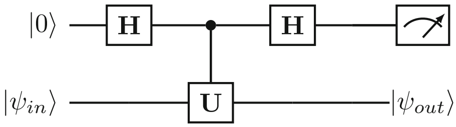

Fig. 7 Circuit illustrating the implicit measurement principle

A measurement of the first qubit in the standard basis, with the measurement operators $\mathbf{P}_0 = |0\rangle\langle 0| \otimes \mathbf{I}$ and $\mathbf{P}_1 = |1\rangle\langle 1| \otimes \mathbf{I}$, yields the probabilities

$$p(0) = \langle \psi_{in}|\mathbf{P}_+|\psi_{in}\rangle \qquad \text{and} \qquad p(1) = \langle \psi_{in}|\mathbf{P}_-|\psi_{in}\rangle, \tag{B9}$$

and leaves the system in the post-measurement state

$$|0\rangle \otimes \frac{\mathbf{P}_+|\psi_{in}\rangle}{\sqrt{\langle \psi_{in}|\mathbf{P}_+|\psi_{in}\rangle}} \tag{B10}$$

if the output 0 is observed, respectively,

$$|1\rangle \otimes \frac{\mathbf{P}_{-}|\psi_{in}\rangle}{\sqrt{\langle\psi_{in}|\mathbf{P}_{-}|\psi_{in}\rangle}} \tag{B11}$$

if the output 1 is observed.

Thus, the second register of the circuit is in post-measurement state of the observable $\mathbf{U}$ on a trial state $|\psi_{in}\rangle$. Its expectation is

$$\langle\mathbf{U}\rangle = 1 \cdot p(0) + (-1) \cdot p(1) = \langle\psi_{in}|(\mathbf{P}_{+} - \mathbf{P}_{-})|\psi_{in}\rangle = \langle\psi_{in}|\mathbf{U}|\psi_{in}\rangle, \tag{B12}$$

which shows that we can evaluate the expectation $\langle\mathbf{U}\rangle$ by measuring a single qubit. This is a crucial feature of our algorithm, as it makes it possible to accurately approximate $\langle\mathbf{U}\rangle$ without having to sample $\langle\psi_{in}|$ needed for the scalar product. Also, it allows to use a moderate number of measurements that is independent of the number of qubits in $|\psi_{in}\rangle$.

References

1. D-Wave Systems: QPU Solver Datasheet. https://docs.dwavesys.com/docs/latest/doc_qpu.html (2022)
2. Benkner, M.S., Lähner, Z., Golyanik, V., Wunderlich, C., Theobalt, C., Moeller, M.: Q-match: iterative shape matching via quantum annealing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7586–7596 (2021)
3. Kuete Meli, N., Mannel, F., Lellmann, J.: An iterative quantum approach for transformation estimation from point sets. In: Computer Vision and Pattern Recognition, pp. 529–537 (2022)
4. Groppe, S., Groppe, J.: Optimizing transaction schedules on universal quantum computers via code generation for Grover's search algorithm. In: Proceedings of the 25th International Database Engineering & Applications Symposium, pp. 149–156 (2021)
5. Uotila, V.J.E.: Synergy between quantum computers and databases. In: Proceedings of the VLDB 2022 PhD Workshop Co-located with the 48th International Conference on Very Large Databases (VLDB 2022) (2022)
6. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
7. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999)
8. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
9. McCoy, B.M., Wu, T.T.: The Two-Dimensional Ising Model. Courier Corporation, Cambridge (2014)
10. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM (JACM) **42**(6), 1115–1145 (1995)
11. Lucas, A.: Ising formulations of many np problems. Front. Phys. **2**, 5 (2014)
12. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
13. Ding, C.H., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 107–114 (2001)
14. Barahona, F., Grötschel, M., Jünger, M., Reinelt, G.: An application of combinatorial optimization to statistical physics and circuit layout design. Oper. Res. **36**(3), 493–513 (1988)
15. Albash, T., Lidar, D.A.: Adiabatic quantum computation. Rev. Mod. Phys. **90**(1), 25 (2018). https://doi.org/10.1103/revmodphys.90.015002
16. Jansen, S., Ruskai, M.-B., Seiler, R.: Bounds for the adiabatic approximation with applications to quantum computation. J. Math. Phys. **48**(10), 102111 (2007). https://doi.org/10.1063/1.2798382
17. Zener, C.: Non-adiabatic crossing of energy levels. Proc. R. Soc. Lond. Ser. A Contain. Pap. Math. Phys. Character **137**(833), 696–702 (1932)
18. Somma, R., Boixo, S.: Spectral gap amplification. SIAM J. Comput. **42**(2), 593–610 (2013)

19. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
20. Guerreschi, G.G., Matsuura, A.Y.: QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. Sci. Rep. **9**(1), 1–7 (2019)
21. Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E.G., Venturelli, D., Biswas, R.: From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. Algorithms **12**(2), 34 (2019)
22. Herrman, R., Ostrowski, J., Humble, T.S., Siopsis, G.: Lower bounds on circuit depth of the quantum approximate optimization algorithm. Quantum Inf. Process. **20**(2), 1–17 (2021)
23. Shaydulin, R., Alexeev, Y.: Evaluating quantum approximate optimization algorithm: a case study. In: 2019 Tenth International Green and Sustainable Computing Conference (IGSC), pp. 1–6 (2019)
24. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. Phys. Rev. E **58**, 5355–5363 (1998). https://doi.org/10.1103/PhysRevE.58.5355
25. Gilyén, A., Su, Y., Low, G.H., Wiebe, N.: Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pp. 193–204 (2019)
26. Camps, D., Beeumen, R.V.: Fable: Fast approximate quantum circuits for block-encodings. In: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 104–113 (2022). https://doi.org/10.1109/QCE53715.2022.00029
27. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, 10th edn. Cambridge University Press, Cambridge (2010)
28. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**(3), 032309 (2018)
29. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Tech. Digest **19**(4), 482–492 (1998)
30. Suzuki, Y., Yano, H., Raymond, R., Yamamoto, N.: Normalized gradient descent for variational quantum algorithms. In: 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 1–9 (2021)
31. Murray, R., Swenson, B., Kar, S.: Revisiting normalized gradient descent: fast evasion of saddle points. IEEE Trans. Autom. Control **64**(11), 4818–4824 (2019). https://doi.org/10.1109/TAC.2019.2914998
32. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using networkX. Technical report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States) (2008)
33. Cross, A.: The IBM Q experience and QISKit open-source quantum computing software. In: APS March Meeting Abstracts, vol. 2018, pp. 58–003 (2018)
34. D-Wave Systems: D-Wave Leap. https://www.dwavesys.com/take-leap (2021)
35. D-Wave Systems: D-Wave Ocean Software Documentation. https://docs.ocean.dwavesys.com/ (2021)
36. Willsch, M., Willsch, D., Jin, F., De Raedt, H., Michielsen, K.: Benchmarking the quantum approximate optimization algorithm. Quantum Inf. Process. **19**(7), 1–24 (2020)
37. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Powell, M.J.D.: A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Advances in Optimization and Numerical Analysis, pp. 51–67. Springer, Dordrecht (1994)
39. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods **17**(3), 261–272 (2020)