

Quantum approximate optimization algorithm for qudit systems

Yannick Deller,¹ Sebastian Schmitt^{1,2,*}, Maciej Lewenstein,^{3,4} Steve Lenk,⁵ Marika Federer,⁵ Fred Jendrzejewski,¹ Philipp Hauke^{1,6}, and Valentin Kasper³

¹*Universität Heidelberg, Kirchhoff-Institut für Physik, Im Neuenheimer Feld 227, 69120 Heidelberg, Germany*

²*Honda Research Institute Europe GmbH, Carl-Legien-Straße 30, 63073 Offenbach, Germany*

³*ICFO-Institut de Ciències Fotoniques, The Barcelona Institute of Science and Technology, Avenida Carl Friedrich Gauss 3, 08860 Barcelona, Spain*

⁴*ICREA, Passeig Lluís Companys 23, 08010 Barcelona, Spain*

⁵*Advanced System Technologies IOSB-AST, Fraunhofer IOSB, Am Vogelherd 90, 98693 Ilmenau, Germany*

⁶*INO-CNR BEC Center and Department of Physics, University of Trento, Via Sommarive 14, I-38123 Trento, Italy*



(Received 26 April 2022; accepted 16 May 2023; published 9 June 2023)

A frequent starting point of quantum computation platforms is the two-state quantum system, i.e., the qubit. However, in the context of integer optimization problems, relevant to scheduling optimization and operations research, it is often more resource-efficient to employ quantum systems with more than two basis states, so-called qudits. Here, we discuss the quantum approximate optimization algorithm (QAOA) for qudit systems. We illustrate how the QAOA can be used to formulate a variety of integer optimization problems such as graph coloring problems or electric vehicle charging optimization. In addition, we comment on the implementation of constraints and describe three methods to include these in a quantum circuit of a QAOA by penalty contributions to the cost Hamiltonian, conditional gates using ancilla qubits, and a dynamical decoupling strategy. Finally, as a showcase of qudit-based QAOA, we present numerical results for a charging optimization problem mapped onto a maximum- k -graph-coloring problem. Our work illustrates the flexibility of qudit systems to solve integer optimization problems.

DOI: [10.1103/PhysRevA.107.062410](https://doi.org/10.1103/PhysRevA.107.062410)

I. INTRODUCTION

Integer optimization problems [1,2] are at the heart of challenging real-world applications, such as scheduling optimization [3], operations research [4], and portfolio selection [5]. The practical importance of these problems makes the development of efficient solution algorithms a particularly active field of research. In recent years, quantum information processing technology has advanced substantially and a multitude of industry-relevant problems have been approached with quantum computing technology, for example, with quantum annealing [6,7]. Many problems have also been addressed by employing algorithms for gate-based universal quantum computing, such as job-shop scheduling [8], graph coloring [9–12], and flight-gate assignment [12,13]. A paradigmatic example of a hybrid classical-quantum algorithm is the quantum approximate optimization algorithm (QAOA), proposed in Refs. [14,15]. Furthermore, it was recognized that the QAOA (i) is a computational model itself [16], (ii) can lead to an optimal query complexity [17], and (iii) exhibits the possibility for quantum advantage [18]. Moreover, important research questions involve the role of quantum effects [19], the choice of the classical optimizer [20], and the performance of the QAOA for low- and high-depth circuits [21,22].

The typical starting point for the QAOA is the qubit, i.e., a quantum mechanical system with two basis states. Several qubits can then be used to represent integer numbers. However, such a binary representation of integers can lead to hardware overhead [12,23–25], and it may be more resource-friendly to work with quantum systems of a finite basis size with dimension $d > 2$, called qudits. Although the representation of qudits with arbitrary dimension in elementary qubits is computationally efficient, even small improvements in hardware requirements can be of great practical importance in the era of noisy intermediate-scale quantum (NISQ) devices [26]. In addition, there is an increased interest in employing qudit systems as quantum information platforms [27], and there has been great experimental progress in realizing quantum information processing with qudits such as photons [28], ions [29], superconducting circuits [30], nuclear magnetic resonance platforms [31], and Rydberg atoms [24,32].

In this paper, we discuss the QAOA for qudit systems and its possible realization in cold atomic systems with long-range interactions, e.g., in cold atomic mixtures [33] or quantum gases inside an optical cavity [34]. Specifically, we elaborate the representations of cost functions and constraints of integer optimization problems with qudits. Furthermore, we give examples of integer optimization problems such as graph coloring and electric vehicle (EV) charging problems, where the qudit formulation provides a convenient representation of integers. Finally, we numerically benchmark a simplified charging optimization problem for small instances.

*Corresponding author: sebastian.schmitt@honda-ri.de

The paper is organized as follows: In Sec. II, we discuss the QAOA for qudit systems and how to encode integer cost functions into Hamiltonians employing angular momentum operators and generalized Pauli operators. Furthermore, we discuss different ways to implement linear constraints in qudit systems generalizing the work of Refs. [15,35]. In Sec. III, we illustrate the implementation of concrete integer optimization problems, and in Sec. IV we numerically analyze the performance of the QAOA for a simplified EV charging optimization problem, which amounts to a graph k -coloring problem with additional coloring cost.

II. QAOA FOR QUDIT SYSTEMS

This section revisits the QAOA approach and discusses how to apply the QAOA to qudit systems. The approach is analogous to the case of qubits, only with enlarged local basis states and operators. We first discuss the Hilbert space for qudits and operators acting on this Hilbert space, namely, angular momentum operators and generalized Pauli operators. These two classes of operators can be implemented experimentally, for example, in atomic mixtures [33] or trapped-ion setups [29]. Next, we give a summary of the general structure of the QAOA [14]. Then, we give two different ways of encoding cost functions into Hamiltonians employing angular momentum operators and generalized Pauli operators. The two different encodings may prove advantageous for different experimental qudit implementations. This section mainly provides background information necessary for the following sections.

A. Hilbert space and operators

We consider the N -fold tensor product of a d -dimensional complex Hilbert space, i.e., $\mathcal{H} = \otimes_{i=1}^N \mathbb{C}^d$. The total dimension of the Hilbert space is $\dim \mathcal{H} = d^N$, and an orthonormal basis for the d -dimensional Hilbert space is denoted by $|z\rangle$ with $z \in \{0, \dots, d-1\}$. A state vector $|\psi\rangle \in \mathcal{H}$ can be written as

$$|\psi\rangle = \sum_{z_1=0}^{d-1} \cdots \sum_{z_N=0}^{d-1} \alpha_{z_1 \dots z_N} |z\rangle, \quad (1)$$

where $\alpha_{z_1 \dots z_N}$ is the complex amplitude and the states $|z\rangle = |z_1, \dots, z_N\rangle$ form an orthonormal basis, i.e., $\langle z|z'\rangle = \delta_{z,z'}$.

The generalized Pauli Z and X operators [27] for one qudit are defined via

$$Z = \sum_{z=0}^{d-1} e^{2\pi iz/d} |z\rangle\langle z|, \quad (2a)$$

$$X = \sum_{z=0}^{d-1} |(z+1) \bmod d\rangle\langle z|, \quad (2b)$$

where the eigenvalues of Z are the roots of unity

$$Z|z\rangle = e^{2\pi iz/d} |z\rangle. \quad (3)$$

The definition of the generalized Pauli operators on \mathcal{H} is given by

$$Z_j = \underbrace{1 \otimes \cdots \otimes 1}_{j-1} \otimes Z \otimes \underbrace{1 \cdots \otimes 1}_{N-j}, \quad (4a)$$

$$X_j = \underbrace{1 \otimes \cdots \otimes 1}_{j-1} \otimes X \otimes \underbrace{1 \cdots \otimes 1}_{N-j}, \quad (4b)$$

which only acts nontrivially on the j th qudit. By construction the basis states $|z\rangle$ are also eigenstates of products of generalized Pauli Z operators

$$\prod_{j=1}^N Z_j^{a_j} |z\rangle = e^{2\pi i \mathbf{a} \cdot \mathbf{z}/d} |z\rangle, \quad (5)$$

where $\mathbf{a} = (a_1, \dots, a_N)$ with $\mathbf{a} \in \mathbb{Z}^N$ summarizes the exponents in the previous expression.

In the following, we define angular momentum operators on the single qudit, which can be realized, for example, in cold atomic gases; see Appendix A for details. First, we define the vectors

$$|\ell, m\rangle \equiv |z\rangle \quad (6)$$

with $\ell = (d-1)/2$ and $m = z - (d-1)/2$. Using this basis, we define the angular momentum operators acting on the local Hilbert space

$$L_z |\ell, m\rangle = m |\ell, m\rangle, \quad (7a)$$

$$L_+ |\ell, m\rangle = \sqrt{(\ell-m)(\ell+m+1)} |\ell, m+1\rangle, \quad (7b)$$

$$L_- |\ell, m\rangle = \sqrt{(\ell+m)(\ell-m+1)} |\ell, m-1\rangle. \quad (7c)$$

Furthermore, the raising and the lowering operators, L_+ and L_- , allow us to define the x and y angular momentum operators,

$$L_x = \frac{1}{2}(L_+ + L_-), \quad (8a)$$

$$L_y = \frac{1}{2i}(L_+ - L_-), \quad (8b)$$

which generate rotations around the x and the y axis, respectively. Finally, we can relate the angular momentum operator to the generalized Pauli Z operator by

$$Z = e^{\frac{2\pi i}{d}[L_z + (d-1)/2]}, \quad (9)$$

where we used that $|z\rangle$ is an eigenvector of L_z .

B. Structure of QAOA

The goal of the QAOA is to find the state \mathbf{z}_0 which minimizes a given cost function $C(\mathbf{z})$. In the following, we recount the building blocks and structure of the QAOA [14,15]. The starting point of the QAOA is the initial state $|\psi_0\rangle$, which we assume can be prepared efficiently and with high fidelity experimentally. We will frequently use the equal superposition state

$$|\psi_0\rangle = \frac{1}{d^{N/2}} \sum_{\mathbf{z}} |\mathbf{z}\rangle, \quad (10)$$

which is an eigenstate to the generalized Pauli X operators of Eq. (4b). Other initial states are possible, for example, an eigenstate to angular momentum operator L_x of Eq. (8a)

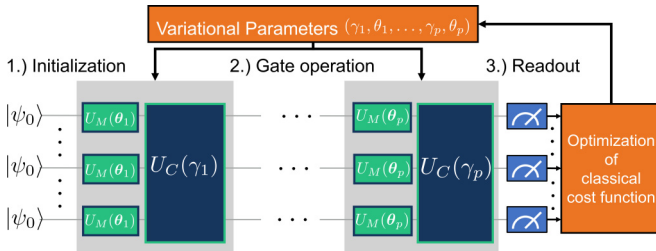


FIG. 1. QAOA for qudits. General structure of the QAOA, consisting of preparation of the initial state $|\psi_0\rangle$, which is the equal superposition of all basis states; application of the alternating QAOA circuit [consisting of phase separation gate $U_C(\boldsymbol{\gamma})$ and mixing gate $U_M(\boldsymbol{\beta})$]; and measurement together with subsequent classical optimization of the variational parameters with respect to the expectation value of the cost Hamiltonian.

or even an eigenstate to the L_z operator, e.g., $|0, \dots, 0\rangle$. We tested several choices of initial states in our numerical experiments and did not find any qualitative differences between these choices. It should therefore in principle not matter which initial state is used, and the choice should be guided by which states are most easily prepared in the experimental setup.

The quantum circuit of the QAOA starts from $|\psi_0\rangle$ with subsequent layers of gates (gray boxes in Fig. 1). Each layer is composed of two parametrized quantum gates, the so-called phase separation gate $U_C(\boldsymbol{\gamma})$ and the mixing gate $U_M(\boldsymbol{\beta})$, which are applied alternately. The generator of the phase separation gate is the cost Hamiltonian H_C encoding the classical cost function

$$U_C(\boldsymbol{\gamma}) = e^{-i\boldsymbol{\gamma}H_C} \quad (11)$$

with

$$H_C |\mathbf{z}\rangle = C(\mathbf{z}) |\mathbf{z}\rangle, \quad (12)$$

where $|\mathbf{z}\rangle$ denotes the computational basis states.

The mixing gate is defined as

$$U_M(\boldsymbol{\beta}) = e^{-i\boldsymbol{\beta}H_M} \quad (13)$$

via a mixing Hamiltonian H_M . Several forms of mixing gates are discussed in the literature [15]. Regarding the limitations of the current NISQ hardware, both the phase separation and the mixing gate should have an efficient decomposition into the native gate set of the experimental platform, which implements the QAOA.

The QAOA circuit of depth $p \geq 1$ is defined as

$$U(\boldsymbol{\gamma}, \boldsymbol{\beta}) = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C}, \quad (14)$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^p$ are free variational parameters to be determined during the execution of the algorithm and p denotes the number of layers. The trial state

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(\boldsymbol{\gamma}, \boldsymbol{\beta}) |\psi_0\rangle \quad (15)$$

is the quantum state approximating a possible solution to the optimization problem. The classical cost function, which is optimized with the QAOA, is

$$E_{\boldsymbol{\gamma}, \boldsymbol{\beta}} = \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | H_C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle, \quad (16)$$

which is the expectation value of the cost Hamiltonian.

Typically, the variational trial state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ is a superposition of the computational basis states, and the expectation value cannot be obtained in a single experimental run. The expectation value equation (16) is estimated by sampling from the trial wave function; see, e.g., Ref. [36]. In each sample, a specific configuration $\mathbf{z} = (z_1, \dots, z_N)$ is obtained from the quantum mechanical trial state with the probability $\mathcal{P}(\mathbf{z}) = |\langle \mathbf{z} | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle|^2$. The expectation value of the cost Hamiltonian is then obtained via

$$E_{\boldsymbol{\gamma}, \boldsymbol{\beta}} \approx \sum_{\text{samples } \mathbf{z}} \mathcal{P}(\mathbf{z}) C(\mathbf{z}), \quad (17)$$

where $\mathcal{P}(\mathbf{z})$ is estimated by sampling from the final QAOA state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$. In order to obtain a solution to the original optimization problem, one uses a classical optimization method to find the parameters $\boldsymbol{\gamma}^*$ and $\boldsymbol{\beta}^*$ that fulfill

$$\{\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*\} = \underset{\boldsymbol{\gamma}, \boldsymbol{\beta}}{\operatorname{argmin}} E_{\boldsymbol{\gamma}, \boldsymbol{\beta}}. \quad (18)$$

After the parameters of the QAOA circuit have been optimized, measuring the output state reveals potential solutions to the optimization problem. In the ideal case, when the QAOA optimization finds an optimal solution, the trial state is a single minimal energy state or a superposition of minimal energy eigenstates of the cost Hamiltonian H_C . In particular, when the cost Hamiltonian is invariant with respect to a symmetry transformation and the mixing operator does not break the symmetry, the final state may be a superposition of lowest-energy eigenstates. However, frequently a minimal energy state cannot be reached, either because the variational trial state cannot faithfully represent the ground state(s) or because the optimization procedure might not find the global optimum. Therefore the final QAOA state may have contributions from various computational basis states, which are low-energy states and have energies close to the optimal state.

Candidate solutions for the optimization problem are the computational basis states with substantial probabilities. Frequently, the final state needs to be prepared several times in order to sample from the trial state. In an experiment, the selection of the candidate solutions needs to consider the measurement error of the state sampling. In contrast, in numerical studies, the probabilities can be evaluated precisely, and we employ a fixed number of candidate solutions and select the ones with the lowest cost. For analyzing the theoretical performance of the QAOA, we neglect any effect of finite sampling.

C. Cost function Hamiltonian

This section discusses two possibilities of mapping certain classical cost functions C to cost Hamiltonians H_C . We first discuss a mapping employing generalized Pauli Z operators and then a second mapping using angular momentum operators L_z .

Mapping using generalized Pauli Z operators. The definition of the cost Hamiltonian in Eq. (12) implies the following diagonal representation:

$$H_C = \sum_{\mathbf{z}} C(\mathbf{z}) |\mathbf{z}\rangle \langle \mathbf{z}|. \quad (19)$$

In order to rewrite the cost Hamiltonian H_C as a polynomial of generalized Pauli Z operators, we use the discrete Fourier transform of the cost function, \widehat{C} . The Fourier transform and its inverse are given by

$$C(\mathbf{z}) = \frac{1}{d^N} \sum_{\mathbf{a}} \widehat{C}(\mathbf{a}) e^{2\pi i \mathbf{a} \cdot \mathbf{z} / d}, \quad (20a)$$

$$\widehat{C}(\mathbf{a}) = \sum_{\mathbf{z}} C(\mathbf{z}) e^{-2\pi i \mathbf{a} \cdot \mathbf{z} / d}, \quad (20b)$$

with $\mathbf{a} \in \mathbb{Z}^N$ and $0 \leq a_j \leq d - 1$. Using the Fourier transform \widehat{C} , we can rewrite the cost Hamiltonian as

$$H_C = \frac{1}{d^N} \sum_{\mathbf{z}} \sum_{\mathbf{a}} \widehat{C}(\mathbf{a}) e^{2\pi i \mathbf{a} \cdot \mathbf{z} / d} |\mathbf{z}\rangle \langle \mathbf{z}|. \quad (21)$$

Employing Eq. (5), the Hamiltonian becomes

$$H_C = \frac{1}{d^N} \sum_{\mathbf{a}} \widehat{C}(\mathbf{a}) \prod_{j=1}^N Z_j^{a_j}, \quad (22)$$

which is a polynomial in the generalized Pauli Z operators [35]. This encoding is especially useful when the Fourier transform of the cost function has few nonzero Fourier coefficients.

Mapping using L_z operators. Here we focus on polynomial functions $C(\mathbf{z})$ in the variables z_m . We obtain the cost Hamiltonian by substituting $z_m \rightarrow (\ell + L_{z,m})$ into the cost function leading to

$$H_C = C(\ell + L_{z,1}, \dots, \ell + L_{z,n}). \quad (23)$$

For examples and representations of cost functions we refer the reader to Sec. III and Appendix B.

D. Mixing Hamiltonian

The mixing operator has to be able to traverse the allowed state space of the optimization problem; see Ref. [15]. For the local qudit Hilbert space with d levels, $d^2 - 1$ local operators are in principle necessary to form an operator basis. However, as shown previously [33,37] a reduced set of three operators is sufficient to generate any state by (possibly many) repeated finite rotations. In this paper, the cost Hamiltonians include linear and higher-order terms in L_z , which allows us to consider a mixing Hamiltonian based only on the angular momentum operator in the x direction

$$H_M = \sum_{i=1}^N L_{x,i}, \quad (24)$$

which fulfills the abovementioned criteria for unconstrained integer optimization problems. As detailed in Appendix A, this mixing operator can be experimentally implemented in atomic qudit systems. Another viable choice would be to use the generalized Pauli X operators of Eq. (4b) as the basis for the mixing Hamiltonian. In principle this should not make a qualitative difference, which we explicitly confirmed by testing both choices in our numerical experiments.

E. Constraints

In many important optimization problems, the variables of the cost function must satisfy constraints, which can be given by equalities or inequalities, i.e.,

$$g_m(\mathbf{z}) \leq 0 \quad \text{or} \quad g_m(\mathbf{z}) = 0 \quad (25)$$

with $m = 1, \dots, M$. Hence it is an important question how to incorporate constraints in the QAOA. One common way to enforce constraints in QAOA circuits is by adding appropriate penalty terms to the cost function. Alternatively, one can engineer the mixing operator such that the evolution of the quantum state only takes place in the space of feasible solutions [15]. In this section, we explicitly implement strategies to enforce constraints in the QAOA. The results developed here are applicable to both qudit and qubit systems.

A standard route to implement constraints in classical optimization is by adding penalty terms to the cost function

$$\tilde{C}(\mathbf{z}) = C(\mathbf{z}) + \sum_m \lambda_m P_m[g_m(\mathbf{z})], \quad (26)$$

where λ_m are the penalty factors, P_m are the penalty functions, and $g_m(\mathbf{z})$ are the constraints given by Eq. (25). Adding penalty terms is typical for black-box optimization [38–40]. Possible penalty functions [41] are

$$P_{\text{eq}}[g(\mathbf{z})] = |g(\mathbf{z})|^a \quad (27)$$

for equality constraints $g(\mathbf{z}) = 0$ and

$$P_{\text{ineq}}[g(\mathbf{z})] = \max[0, g(\mathbf{z})]^a \quad (28)$$

for inequality constraints $g(\mathbf{z}) \leq 0$, with typical values for the exponents being $a = 1$ and $a = 2$. One way to implement the modified cost function $\tilde{C}(\mathbf{z})$ in the QAOA is to use the cost Hamiltonian H_C generated by $C(\mathbf{z})$ for the quantum circuit but employ $\tilde{C}(\mathbf{z})$ in the classical optimization loop with the cost function

$$\tilde{E}_{\gamma, \beta} = \sum_{\mathbf{z}} |\langle \mathbf{z} | \gamma, \beta \rangle|^2 \left\{ C(\mathbf{z}) + \sum_m \lambda_m P_m[g_m(\mathbf{z})] \right\}. \quad (29)$$

In addition, one has to tune the penalty parameters during the optimization process. Not including the constraints in the cost Hamiltonian simplifies the experimental realization of the phase separation gate and relaxes the requirements on the hardware. However, depending on the optimization problem and, in particular, on how the constraints confine the feasible search space, the trial states may produce infeasible solutions for randomly chosen γ and β .

In a situation where the cost Hamiltonian and the mixer are invariant with respect to a symmetry, the trial state is also symmetric. However, if one constraint violates the symmetry (not implemented in the Hamiltonian), the trial state will not be able to reflect this violation of the symmetry. This scenario may render the QAOA less effective because more candidate solutions must be sampled.

An alternative is to implement the constraint in the circuits. Here, we present three different ways to implement constraints in the quantum circuit: (i) including penalty terms in the cost Hamiltonian, (ii) using conditional gates, and (iii) employing dynamical decoupling.

1. Penalty terms in the cost Hamiltonian

One can include constraints in the QAOA by using $H_{\tilde{C}}$ instead of H_C ; see Refs. [6,7] for details. The penalized cost Hamiltonian is given by

$$H_{\tilde{C}} = H_C + \sum_m \lambda_m P_m(G_m), \tag{30}$$

where we introduced the constraint operator G_m via

$$G_m |z\rangle = g_m(z) |z\rangle. \tag{31}$$

The cost Hamiltonian including the constraints also has a diagonal representation

$$H_{\tilde{C}} = \sum_z \tilde{C}(z) |z\rangle\langle z|, \tag{32}$$

where $\tilde{C}(z)$ is given by Eq. (26). Using the results of Sec. II C, one can write the penalized Hamiltonian $H_{\tilde{C}}$ with the help of generalized Pauli Z or angular momentum operators, where the angular momentum encoding works when $\tilde{C}(z)$ is a polynomial. Again, constructing the Hamiltonian $H_{\tilde{C}}$ with generalized Pauli operators involves a discrete Fourier transform [see Eqs. (20a) and (20b)], which results in polynomials of generalized Pauli Z operators. Here, the max function in the inequality constraint may introduce higher powers of generalized Z operators.

Using the cost Hamiltonian $H_{\tilde{C}}$ in the unitary evolution of Eq. (14) leads to variational states $|\gamma, \beta\rangle$ that satisfy the constraints g_m for appropriate choice of the penalty parameters λ_m . A major disadvantage of including penalties in the cost Hamiltonian is the necessity for tuning the penalty factors λ_m . The penalty factors directly affect the cost function landscape and thus significantly influence search performance.

2. Constraints via conditional gates

Here we enforce equality and inequality constraints via conditional gates [35]. Specifically, we introduce the unitary operator U_g that acts on the N computational qudits and on one ancilla qubit $|y\rangle = \alpha |0\rangle + \beta |1\rangle$ via

$$U_g |z\rangle|y\rangle = \begin{cases} |z\rangle|y\rangle & \text{for } g(z) = 0 \text{ or } g(z) \leq 0 \\ |z\rangle X |y\rangle & \text{otherwise.} \end{cases} \tag{33}$$

If the quantum state $|z\rangle$ does fulfill the constraint, the ancilla qubit does not change, whereas we apply X on the ancilla if the constraint is not fulfilled. The operator U_g belongs to the class of conditional operators:

$$U_g(Q) = \sum_{z:g(z)\leq 0} |z\rangle\langle z| \otimes I + \sum_{z:g(z)>0} |z\rangle\langle z| \otimes Q, \tag{34a}$$

$$U_g(Q) = \sum_{z:g(z)=0} |z\rangle\langle z| \otimes I + \sum_{z:g(z)\neq 0} |z\rangle\langle z| \otimes Q, \tag{34b}$$

where the first equation is for inequalities and the second equation is for equalities. Furthermore, I is the identity operator and Q is an arbitrary unitary operator. The conditional operator $U_g(Q)$ does only apply the operator Q on the ancilla qubit if the quantum state $|z\rangle$ does not fulfill the inequality $g(z) \leq 0$ or equality $g(z) = 0$.

We represent $U_g(Q)$ as a matrix exponential in order to discuss the implementation with an appropriately chosen

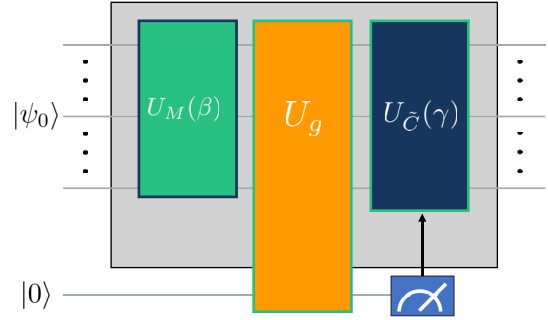


FIG. 2. QAOA with conditional gates for constraints: In order to enforce constraints, one performs a conditional gate U_g and measures the ancilla qubit $|0\rangle$. If the measurement returns the value 0, the constraint is fulfilled, and we do not change the cost function. If the measurement yields the value 1, then the constraint is violated, and we forward the measurement result to the cost function, use the Hamiltonian $H_{\tilde{C}}$, and penalize the violation of the constraint.

Hamiltonian. Direct calculation [35] shows

$$U_g(Q) = e^{-iH_g \otimes H_Q}, \tag{35}$$

with

$$H_g = \sum_z [1 - \delta_{\max[g(z),0],0}] |z\rangle\langle z|, \tag{36a}$$

$$H_g = \sum_z [1 - \delta_{g(z),0}] |z\rangle\langle z| \tag{36b}$$

and

$$H_Q = (\pi/2)X. \tag{37}$$

Particularly, we can implement H_g again using the Pauli operator encoding. In order to enforce one constraint g after one layer of the QAOA circuit, we initialize the ancilla qubit in $|0\rangle$; apply the conditional gate equation (33) on the quantum state $|\psi\rangle$, which we assume fulfills all constraints, leading to

$$U_g e^{-i\beta H_M} |\psi\rangle |0\rangle; \tag{38}$$

and measure the ancilla qubit $|y\rangle$. If the measurement on the ancilla qubit returns the value zero, the constraint is fulfilled, and we apply the unconstrained cost function Hamiltonian H_C . On the other hand, if the measurement of the ancilla qubit yields the result 1, the inequality is violated, and we use $H_{\tilde{C}}$ and the mixing Hamiltonian in the next step. Then we apply the conditional unitary operator U_g again and iterate this procedure. The quantum circuit illustrating this approach is schematically shown in Fig. 2.

Using conditional gates is more involved as it requires an additional qubit for tracking the constraint. Finally, conditional gates as described above effectively implement the max function in the cost Hamiltonian for inequality constraints.

3. Equality constraints via dynamical decoupling

Another way to implement equality constraints is via dynamical decoupling [42,43], a quantum control technique suppressing coupling to an environment. The technique dates back to nuclear magnetic resonance experiments [44–46]; for a review, see Ref. [47]. Dynamical decoupling techniques can also be employed to suppress transitions to undesired

subspaces; see, for example, Refs. [48,49] in the context of quantum simulations of lattice gauge theories. Here we discuss how to use dynamical decoupling to enforce constraints.

To apply dynamical decoupling to the QAOA, we choose an initial state $|\psi_0\rangle$ that already fulfills all equality constraints

$$G_m |\psi_0\rangle = 0 \quad (39)$$

for all m . The unitary operator $U(\boldsymbol{\gamma}, \boldsymbol{\beta})$ of the QAOA may lead to a trial state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$ where the constraints are not fulfilled, i.e.,

$$G_m |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle \neq 0. \quad (40)$$

The goal is to construct a unitary mixing operator that does not evolve the initial state out of the feasible subspace given by the equality constraints. Therefore the mixing term must commute with all constraints. In order to obtain such a unitary mixing term, we employ a dynamical decoupling strategy.

For simplicity, we assume that G_m has only integer eigenvalues and denote the largest eigenvalue by Λ_m . We start from the identity

$$e^{-i\theta G_m} |\psi_0\rangle = |\psi_0\rangle \quad \forall \theta \in \mathbb{R}, \quad (41)$$

which follows from Eq. (39). We define the symmetrization operation of any operator \mathcal{O} by

$$\bar{\mathcal{O}} = \prod_m \int_0^{\frac{2\pi}{\Lambda_m}} \frac{d\theta_m}{(2\pi/\Lambda_m)} e^{-i\theta_m G_m} \mathcal{O} e^{i\theta_m G_m}. \quad (42)$$

Specifically, the symmetrization implies

$$e^{-i\phi G_m} \bar{\mathcal{O}} e^{i\phi G_m} = \bar{\mathcal{O}} \quad (43)$$

for all m , which follows from using the integer spectrum of G_m and shifting the integration variables. The above equation is equivalent to $[e^{-i\phi G_m}, \bar{\mathcal{O}}] = 0$ for all m . Equality constraints with a rational spectrum can always be reformulated as constraints with an integer spectrum by multiplying with the least common multiple, while constraints with an irrational spectrum can be approximated with a rational spectrum. Employing Eq. (42), we symmetrize the mixing Hamiltonian according to

$$\bar{H}_M = \prod_m \int_0^{\frac{2\pi}{\Lambda_m}} \frac{d\theta_m}{(2\pi/\Lambda_m)} e^{-i\theta_m G_m} H_M e^{i\theta_m G_m}, \quad (44)$$

which leads to $[e^{-i\phi G_m}, \bar{H}_M] = 0$ for all m and its infinitesimal version $[G_m, \bar{H}_M] = 0$ for all m . Using the symmetrized mixing Hamiltonian, we engineer a new unitary mixing operator

$$U_M(\boldsymbol{\beta}) = e^{i\boldsymbol{\beta}\bar{H}_M} \quad (45)$$

that does not commute with the cost Hamiltonian [15] but guarantees that the final state

$$G_m U(\boldsymbol{\gamma}, \boldsymbol{\beta}) |\psi_0\rangle = 0 \quad (46)$$

also fulfills the constraints given $G_m |\psi_0\rangle = 0$.

Implementing the continuous integral of the dynamical decoupling strategy of Eq. (42) in a circuit is challenging. One strategy is to sample or discretize the integral into a finite sum and use Floquet engineering to determine an appropriate discretization; see, e.g., Ref. [47]. Because of this sampling at each layer, the dynamical decoupling strategy is only possible for low circuit depth. However, in order to reduce the number

of decoupling layers, one can selectively introduce the dynamical decoupling, especially in the last layer.

III. APPLICATIONS

This section discusses optimization problems involving integer variables, which can be addressed with the QAOA based on qudits. Primarily, we illustrate the encodings of Sec. II C, which leads to feasible implementations in current qudit systems. Specifically, we treat a graph coloring problem and the optimization of an electric vehicle charging plan. Further integer optimization problems, i.e., a knapsack problem, multiway number partitioning, job-shop scheduling, and their respective qudit encodings, can be found in Appendix B.

A. Graph coloring

Let $G = (V, E)$ be a graph with N vertices and M edges. A proper vertex k -coloring of G is given, if one can assign one of k colors to each vertex such that adjacent vertices have different colors. If one can find such a proper vertex k -coloring, the graph G is k -colorable [15,23]. We denote the assignment of colors to the vertices by $\mathbf{z} = (z_1, \dots, z_N)$ with $z_i \in \{0, \dots, k-1\}$. The coloring task can be expressed as finding the minimum of an objective function which counts the number of edges between nodes with the same color, i.e.,

$$C(\mathbf{z}) = \sum_{(n,m) \in E} \delta_{z_n, z_m}, \quad (47)$$

where n and m denote vertices of the graph, z_n and z_m denote the color of the vertices, and E is the set of edges of the graph.

Minimizing $C(\mathbf{z})$ leads to the largest induced subgraph that can be properly k -colored. Moreover, the cost function can be encoded with Pauli Z operators [50] into the cost Hamiltonian

$$\begin{aligned} H_C &= \frac{1}{k^{N-1}} \sum_{(n,m) \in E} \sum_{a,b=0}^{k-1} \delta((a+b) \bmod k, 0) Z_n^a Z_m^b \\ &= \frac{1}{k^{N-1}} \sum_{(n,m) \in E} \left(1 + \sum_{a=0}^{k-1} Z_n^a Z_m^{k-a} \right), \end{aligned} \quad (48)$$

where we used the Fourier transform of the Kronecker delta, $\widehat{\delta}(a, b) = k\delta[(a+b) \bmod k, 0]$. For $k=3$ this becomes

$$H_C = \frac{1}{3^{N-1}} \sum_{(n,m) \in E} (1 + Z_m^3 + Z_n Z_m^2 + Z_n^2 Z_m). \quad (49)$$

This expression can be reformulated in terms of angular momentum operators $L_{z,n}$ resulting in a polynomial in powers of $L_{z,n}$ and $L_{z,m}$ with $L_{z,m}^{d-1} L_{z,n}^{d-1}$ as the largest power.

B. Charging optimization

Many problems in the energy management domain require optimizing a schedule for the distribution of electrical energy among technical devices. A representative problem is the charging schedule of electric vehicles (EVs). Designing these schedules typically leads to integer or mixed-integer programming problems; see, for example, Refs. [3,51–53]. We consider the following EV charging problem: An operator of charging stations needs to charge N EVs during the working

hours of a business complex. The operator can purchase and sell energy for real-time electricity market prices and charge or discharge each EV. The goal is to minimize the electricity cost for the operator while meeting operation constraints. Possible constraints are as follows: (i) Each EV has a desired target state of charge (SOC) of the battery, which needs to be reached at the end of the time period. (ii) Each battery has a minimal and maximal SOC. (iii) There is an upper and lower limit for the total cumulative charging power of all vehicles at all times. A simple variant of the EV charging problem without constraints and preemptive charging was discussed in Ref. [54] using the QAOA for qubits.

The charging plan for the EVs must be optimized for the total time T , where we divide T into equidistant time steps of duration Δt . In each time step the EV n can either be charged, not charged, or discharged, which is represented by the ternary variables $L_{n,t} = 1$, $L_{n,t} = 0$, or $L_{n,t} = -1$, respectively. The total electricity cost function is

$$C(\mathbf{L}) = \sum_{n=1}^N \sum_{t=1}^T \Delta t P_n^0 \left(\frac{e_t^c + e_t^d}{2} L_{n,t} + \frac{e_t^c - e_t^d}{2} L_{n,t}^2 \right), \quad (50)$$

where P_n^0 is the charging and discharging power of vehicle n and the prices for buying and selling energy are e_t^c and e_t^d . Specifically, the charging costs for car n are $\Delta t P_n^0 e_t^c$, whereas the discharging costs are $-\Delta t P_n^0 e_t^d$.

The SOC of each battery at time t is

$$E_{n,t} = E_n^{\text{init}} + \sum_{k=1}^t \Delta t P_n^0 (L_{n,k} - \delta_n L_{n,k}^2), \quad (51)$$

where E_n^{init} denotes the initial SOC of EV n and $\delta_n \geq 0$ encodes conversion losses in the EV since the SOC increases (decreases) by $(\pm 1 - \delta_n) \Delta t P_n^0$ during charging (discharging) with power P_n^0 . The constraints on the SOC of each battery are

$$E_{n,T} \geq E_n^{\text{target}} \quad \forall n, \quad (52a)$$

$$E_n^{\text{min}} \leq E_{n,t} \quad \forall n, t, \quad (52b)$$

$$E_{n,t} \leq E_n^{\text{max}} \quad \forall n, t, \quad (52c)$$

where E_n^{target} is the required minimal final SOC at time $t = T$ and E_n^{min} and E_n^{max} specify the generally allowed SOC for vehicle n . The limits on the maximum charging and discharging power are

$$P^{\text{min}} \leq \sum_n P_n^0 L_{n,t} \quad \forall t, \quad (53a)$$

$$P^{\text{max}} \geq \sum_n P_n^0 L_{n,t} \quad \forall t, \quad (53b)$$

with $P^{\text{min}} < 0$ being the largest possible discharging power and $P^{\text{max}} > 0$ being the maximum charging power. This amounts to $N(1 + 2T) + 2T$ constraints in total.

C. Combination of charging and graph coloring

Here we consider the EV charging optimization problem of the previous section with additional constraints on the charging time slots. Therefore we consider a graph where each vertex represents an EV and each edge indicates overlapping charging time slots. Furthermore, the vertex color represents

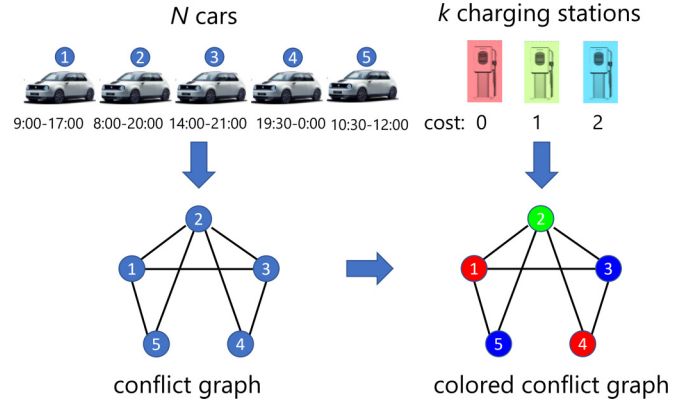


FIG. 3. Schematic representation of the simplified EV charging problem. N cars need to be assigned to k charging stations, where each car needs to be placed at a charging station for a given time period. Here, we consider $N = 5$ cars and $k = 3$ charging stations. No two cars with overlapping charging periods can be assigned to the same charging station, which can be formulated as a conflict graph where cars with overlapping time slots are connected by an edge. If we denote each charging station by a different color, the charging station assignment can be formulated as a coloring problem of the conflict graph. Furthermore, we assume that each charging station incurs different costs, which are dimensionless numbers.

the number of the charging station, whereby each charging station has different costs. Finally, the constraint that two EVs cannot be charged at the same station simultaneously is modeled by the condition that two connected vertices must not have the same color. This charging problem is schematically illustrated in Fig. 3.

A cost function, which combines the different charging station costs and penalizes charging two vehicles at the same station simultaneously, is

$$C(\mathbf{z}) = \sum_{n=1}^N \sum_{i=0}^{k-1} c_i z_n^i + \lambda \sum_{(u,v) \in E} \delta_{z_u, z_v}, \quad (54)$$

where z_n is the color of node n , the parameters c_i encode the cost for each color, and $\lambda > 0$ is the penalty factor. Note that we always use dimensionless cost functions and thus the c_i and the penalty factors λ are also dimensionless. When all colors have equal costs, all coefficients c_i for $i > 0$ are zero, and this problem reduces to the pure maximum- k -coloring problem of Sec. III A. For the case with three colors, $k = 3$, and associated costs c_{-1} , c_0 , and c_1 , the cost Hamiltonian is

$$H_C = \sum_{n=1}^N \left(c_0 + \frac{c_1 - c_{-1}}{2} L_{z,n} + \frac{c_1 + c_{-1} - 2c_0}{2} L_{z,n}^2 \right) + \lambda \sum_{(n,m) \in E} \left[1 - L_{z,n}^2 - L_{z,m}^2 + \frac{1}{2} L_{z,n} L_{z,m} + \frac{3}{2} L_{z,n}^2 L_{z,m}^2 \right], \quad (55)$$

where the vertices are denoted by n and m . Specifically, the coloring constraint induces two-site interactions with up to quadratic terms in L_z on each vertex.

IV. NUMERICAL RESULTS

In the following, we discuss numerical results of the QAOA for the cost Hamiltonian equation (55) for three colors $k = 3$. We consider different graphs ranging from $N = 4$ to $N = 9$ and include the graph coloring constraint term with a penalty factor $\lambda = 20$ directly in the quantum circuit, as detailed in Sec. II E. Furthermore, we consider two cases for the costs: (i) $(c_{-1}, c_0, c_1) = (0, 0, 0)$, where the charging problem reduces to maximum- k -graph-coloring, and (ii) $(c_{-1}, c_0, c_1) = (0, 1, 2)$. A recent work [50] benchmarked QAOA on pure maximum- k -graph-coloring with qutrits ($k = 3$) on random 3-colorable constant-degree graphs up to a size of $N = 300$, which was possible for $p = 1$ (a circuit with one layer). In contrast, we employ several layers ranging from $p = 1$ to $p = 8$, introduce an additional cost contribution for each vertex color, and focus on individual instances of highly connected graphs. Furthermore, we compare the performance of two classical optimization algorithms for the cost function encoded by Eq. (55).

We employ two classical optimization algorithms: The limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [55] and the covariance matrix adaptation evolutionary strategy (CMA-ES) [56] taken from Ref. [57]. For each setting (graph, cost function, and circuit depth), we execute 50 different QAOA optimization runs of the CMA-ES with randomly chosen initial values for $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. The population-based CMA-ES evaluates between 6 and 12 candidate solutions in each generation, depending on the search space dimension $2p$. For the L-BFGS optimizer, we use between 300 and 600 optimization runs to be comparable to the number of circuit evaluations with CMA-ES. Figure 4 shows the probability distribution of the computational basis states determined by the final QAOA state for $N = 6$, different circuit depth, and with or without coloring cost. For example, the pure graph coloring problem without coloring cost has 12 optimal states for the graph given in Fig. 4. The probability distribution of the computational basis states is shown in Figs. 4(a) and 4(c), and example graph colorings are depicted in Fig. 4(e). However, lifting the color symmetry by including costs for different colors leads to a single optimal solution, as can be seen from the asymmetric probability distribution in Figs. 4(b) and 4(d).

In Fig. 5, we show results for a representative example of the simplified charging problem on a graph with $N = 6$ nodes and three colors ($k = 3$) with and without coloring cost. Figures 5(a) and 5(b) show the optimality gap, i.e., the difference between the exact minimum and the minimum obtained from the QAOA cost function for different circuit depths $p \leq 8$. The exact minimum was obtained by exhaustive search of the whole state space, which was possible for the limited problem size considered here. Generally, the lowest value of the optimality gap decreases for both optimizers with increasing circuit depth, indicating that a deeper circuit can, in principle, achieve smaller values of the cost function [21, 58]. However, there are considerable variations in the optimality gap between the runs.

This behavior of the optimality gap is understandable as the cost function landscape of the QAOA is typically highly multimodal with many local minima and maxima, as can

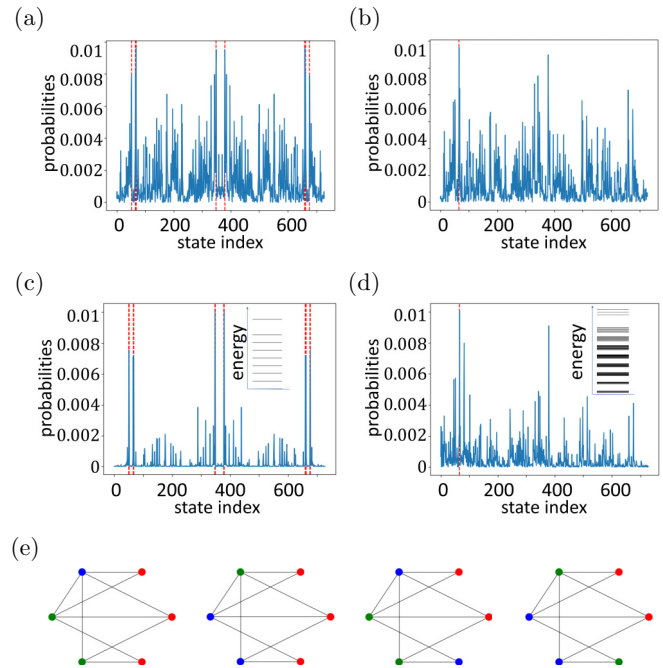


FIG. 4. Probability distributions of the optimized state and optimal solution graphs. Probability distribution of representative final QAOA states for a $N = 6$ graph without coloring cost [(a) and (c)] and with coloring cost $(c_{-1}, c_0, c_1) = (0, 1, 2)$ [(b) and (d)]. (a) and (b) depict results for shallow circuits with depth $p = 1$, while (c) and (d) show results for depth $p = 5$. The red dashed lines indicate the 12 (1) optimal solutions without (with) coloring cost. The insets in (c) and (d) show the energy spectra of the respective Hamiltonian in arbitrary units. (e) Candidate solutions for the simplified charging problem, where the colors red, green, and blue have cost c_{-1} , c_0 , and c_1 , respectively. Without coloring cost, all shown graphs are optimal solutions, and the additional eight optimal solutions can be generated by pairwise color exchange. With coloring cost $(c_{-1}, c_0, c_1) = (0, 1, 2)$, only the leftmost graph of (e) is optimal.

be seen in Figs. 5(c) and 5(d), where a part of the $p = 1$ two-dimensional search landscape is shown. The multimodal and, in particular, the ridgelike structure of the cost function landscape makes the optimization problem considerably harder for algorithms such as L-BFGS, which use gradient information, as it introduces saddle-point-like features known to cause problems in many settings including deep-learning applications [59–61]. Hence the L-BFGS optimization process may end up in a local minimum with a high probability, essentially determined by the location of the random initial starting point. In contrast, the CMA-ES is a population-based global optimizer capable of dealing with this cost function landscape and finds lower cost minima more reliably.

However, for larger circuits, particularly for $p \gtrsim 4$, the CMA-ES may also result in suboptimal local minima, a common behavior for evolutionary algorithms in a larger search space. In Fig. 6 we show the progress of the CMA-ES optimization runs as a function of the internal generation number. For the more straightforward problem with $p = 1$ shown in Fig. 6(a), the CMA-ES converges rather quickly in about 200 generations. In contrast, for the more challenging

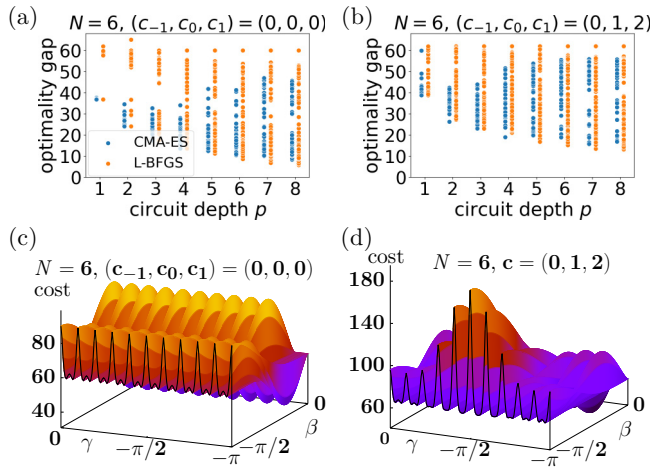


FIG. 5. Optimality gap and cost function landscape. Optimality gap of the final QAOA states for a graph with $N = 6$ (a) without coloring cost, i.e., $(c_{-1}, c_0, c_1) = (0, 0, 0)$, and (b) with coloring cost $(c_{-1}, c_0, c_1) = (0, 1, 2)$, as a function of the circuit depth p and for the two optimizers L-BFGS (orange) and CMA-ES (blue). The plots show the best result for each of the 50 (300) CMA-ES (L-BFGS) runs. (c) and (d) show the cost function landscape for circuit depth $p = 1$ in a certain parameter region of the search space for the same $N = 6$ graph without (c) and with (d) coloring cost.

optimization problem with $p = 5$, we can observe that some runs do not converge even after 1400 generations. In principle, this limitation can be removed by running the algorithms for more generations, which comes at the cost of more circuit evaluations.

It is instructive to investigate how the improvement of the optimality gap for deeper circuits (cf. Fig. 5) is reflected in the probability distribution. For shallow circuits, i.e., $p = 1$, the global minima are clearly visible, as indicated by the red dashed lines in Figs. 4(a) and 4(b). However, other states with higher energy do have sizable contributions in the probability distribution of the final state. Increasing the circuit depth leads to better separation of the optimal states compared with suboptimal states and a smaller value of the cost function; see Figs. 4(c) and 4(d).

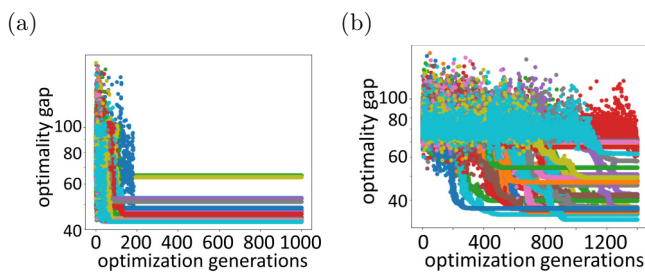


FIG. 6. Optimization progress. Optimality gap values of 50 CMA-ES optimization runs as a function of the internal optimization generation number for an $N = 6$ graph without coloring cost and with a circuit depth of $p = 1$ (a) and $p = 5$ (b). Each color represents one individual optimization run with different initial values for the search parameters γ and β .

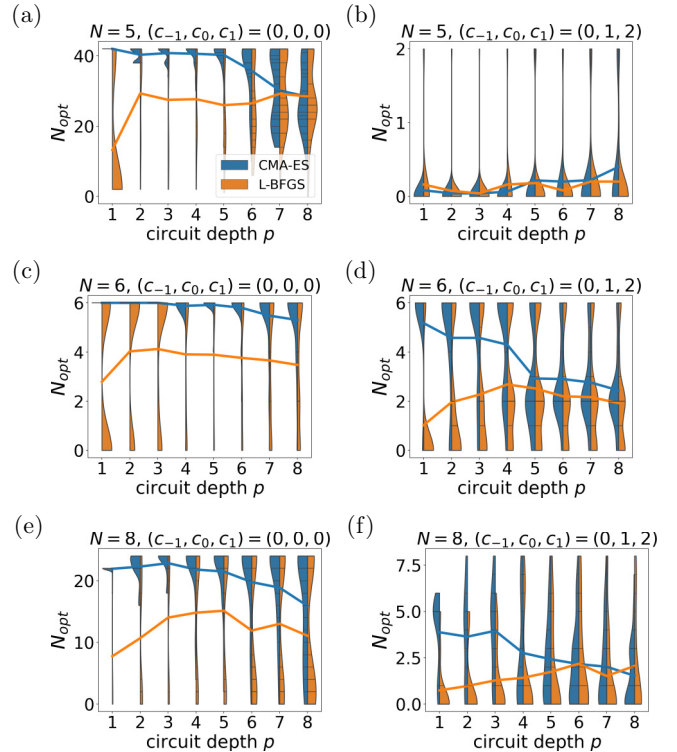


FIG. 7. Distribution of optimal solutions over several QAOA runs. Number of optimal solutions found in one QAOA optimization run for simplified charging problem instances on graphs with $N = 5$ [(a) and (b)], $N = 6$ [(c) and (d)], and $N = 8$ [(e) and (f)] without [(a), (c), and (e)] and with [(b), (d), and (f)] charging cost. The colored bars show the number of found optimal solutions in each run, where a larger width implies a larger number of found solutions. Lines show the mean number of found solutions aggregated over all 50 and 300–600 different optimization runs using CMA-ES and L-BFGS, respectively.

Comparing the probability distribution of the case without color cost in Figs. 4(a) and 4(c) and with color cost in Figs. 4(b) and 4(d) shows a qualitative difference, which can be understood from the spectrum of the cost Hamiltonian. The spectrum of the pure graph coloring Hamiltonian [see the inset of Fig. 4(c)] has a large gap between the (degenerate) ground state manifold and the first excited states. As the QAOA circuit is an approximation to an adiabatic time evolution, a large energy gap between the ground state and excited state is beneficial for finding the ground state. In contrast, we do not observe a clear energy gap when considering coloring costs; see the inset of Fig. 4(d). Especially the appearance of multiple low-lying energy states makes the problem of separating those states in the quantum circuit harder. Consequently, nonoptimal states have higher amplitudes for the problem containing coloring costs compared with the pure graph coloring problem.

By preparing and sampling from the final QAOA state, one can then extract multiple candidates for the optimum of the cost function. We extract several candidate solutions and test for optimality for different graphs for the simplified charging problem with and without coloring cost. In addition, we perform several optimization runs and depict the average

number of optimal solutions (with and without coloring costs) in Fig. 7. The vertical axis of Fig. 7 always extends to the total number of optimal solutions (determined by a classical optimizer), and the error bars indicate the minimal and maximal numbers of solutions found with the QAOA after the classical optimization loop. In most cases, we can find β and γ such that the QAOA state allows for detecting all optimal solutions regardless of the circuit depth, the optimization algorithm, and the problem instance. Exceptions, where the QAOA could not find all optimal solutions, are shown in Fig. 7(f) for $N = 8$ and $p \leq 3$ with coloring costs and in Fig. 7(e) for $N = 8$ and $p = 1$ without coloring cost.

Inspecting the average number of found optimal solutions, we can observe that the variation between different runs is substantial. In particular, for the instance without coloring cost on a graph with $N = 5$ nodes shown in Fig. 7(a), the QAOA reliably finds almost all 42 optimal solutions when employing the CMA-ES for not-too-deep circuits, while introducing coloring cost on the same graphs [see Fig. 7(b)] leads to a situation where on average almost none of the two optimal solutions are found. We point out that finding all optimal solutions with the CMA-ES becomes more difficult with increasing circuit depth, which is a typical behavior for evolutionary algorithms when increasing parameter space. Both optimization algorithms show lower performance in finding all optimal solutions in one run on average for the problem with color cost than the pure graph coloring problem. This effect results from lifting the degeneracy by introducing the coloring cost, which leads to multiple low-lying states close in energy. The relation between the spectrum of the Hamiltonian and the performance of the QAOA is a field of current study [21].

V. CONCLUSION

In this paper, we discussed the QAOA for current or upcoming qudit experiments. First, we described how to map cost functions onto cost Hamiltonians utilizing generalized Pauli or angular momentum operators. Additionally, we illustrated different ways to incorporate equality or inequality constraints. Therefore we laid out a scheme to include constraints in the classical optimization loop. We also presented three alternative methods to incorporate constraints into the quantum circuit. The first method adds penalty functions for the constraints into the cost Hamiltonian. The second method realizes equality and inequality constraints using conditional gates and ancilla qubits, similar to an error-correcting code. Finally, the third approach implements constraints by exploiting dynamical decoupling, which suppresses computational basis states which violate equality constraints.

As an application of the QAOA with qudits, we discussed theoretical and industry-relevant optimization problems, for example, the graph k -coloring or an EV charging problem with global power constraints. Since these problems only involve bounded integer variables, they can be mapped on qudits. Motivated by current experiments with ultracold atoms or ions, we propose to use the x -angular momentum operator L_x as a mixing operator. Finally, we numerically studied a simplified EV charging optimization problem, which amounts to a maximum- k -graph-coloring problem with an

additional color cost term on the vertices. We compared solutions of the QAOA obtained with the gradient-based classical L-BFGS optimizer and the global evolutionary CMA-ES optimizer for our numerical studies. Our results showed that the global evolutionary optimizer was less sensitive to the initialization of the search and reliably produced better results than the gradient-based approach for the instances considered. This performance behavior can be understood by the highly multimodal cost function landscape. Starting from pure maximum- k -coloring and introducing a coloring cost function, the performance with both optimizers typically decreased. This behavior is a direct consequence of reducing the symmetry of the cost Hamiltonian.

We extracted solutions from the final state by selecting states with the largest amplitudes. In our examples, we could find multiple optimal solutions. In particular, the final state is also symmetric when the cost Hamiltonian is invariant with respect to a symmetry, and the mixing operator does not break this symmetry. The final state thus includes equal-weight superpositions of symmetry-related states. Finding several candidate solutions is a highly desirable feature for practical applications, as one has the chance to obtain a large subset of all possible solutions. However, these degeneracies may lead to detrimental performance since the amplitude of the optimal states may be distributed such that the sampling of the candidate solutions becomes inefficient. Notably, the signal-to-noise ratio between optimal and suboptimal states may be reduced. In order to improve the signal-to-noise ratio, one can single out optimal states by investigating ways to reduce the number of candidate solutions, e.g., via sparsity constraint, on the QAOA trial state. This approach is left for future study.

In this paper, we studied a selection of optimization problems. However, we expect that the insights generated here are relevant for general problem instances on larger graphs and different types of problems. Specifically, the cost function landscape will generally be multimodal due to the structural form of the mixer and phase separation operators. Therefore global black-box optimizers are expected to be very useful for the QAOA [20]. Another promising and highly relevant aspect of qudit-based implementations of QAOA is the possibility of resource-efficient implementation on hardware, which was shown in previous work [24]. The question of how this advantage over qubit-based implementations extends to the formulation of realistic problems including constraints in details is left for future research. In total, we have extended the QAOA toolbox for qudit systems and applied it to relevant theoretical and practical applications opening up an avenue for current and future qudit platforms to solve integer optimization problems.

ACKNOWLEDGMENTS

We acknowledge fruitful discussions with R. Blatt, A. Bottarelli, A. Garcia-Sala, D. Gonzalez-Cuadra, M. K. Oberthaler, M. Ringbauer, H. Türeci, T. V. Zache, and P. Zoller. S.L. acknowledges support from the project Quantum Hub Thüringen, 2021 FGI 0047, Free State of Thuringia, Thüringer Aufbaubank. F.J. acknowledges the DFG support through the project FOR 2724, the Emmy-Noether grant (Project No. 377616843). This work is supported by the DFG

Collaborative Research Centre “SFB 1225 (ISOQUANT),” by the Bundesministerium für Wirtschaft und Energie through the project “EnerQuant” (Project No. 03EI1025C), and by the Bundesministerium für Bildung und Forschung through the project “HFAK” (Project No. 13N15632). P.H. acknowledges support from Provincia Autonoma di Trento, the ERC starting grant StrEnQTh (Project No. 804305), the Google Research Scholar Award ProGauge, and Q@TN–Quantum Science and Technology in Trento. V.K. and M.L. acknowledge support from ERC AdG NOQIA; Agencia Estatal de Investigación (R&D Project No. CEX2019-000910-S, funded by MCIN/AEI/10.13039/501100011033, Plan National FIDEUA PID2019-106901GB-I00, FPI, QUANTERA MAQS PCI2019-111828-2, Proyectos de I+D+I “Retos Colaboración” QUSPIN RTC2019-007196-7); Fundació Cellex; Fundació Mir-Puig; Generalitat de Catalunya through the CERCA program, AGAUR Grant No. 2017 SGR 134, QuantumCAT U16-011424, cofunded by ERDF Operational Program of Catalonia 2014-2020; EU Horizon 2020 FET-OPEN OPTOLogic (Grant No. 899794); National Science Centre, Poland (Symfonia Grant No. 2016/20/W/ST4/00314); Marie Skłodowska-Curie grant STREDCH No. 101029393; “La Caixa” Junior Leaders fellowships (No. 100010434); and EU Horizon 2020 under Marie Skłodowska-Curie Grant Agreement No. 847648 (LCF/BQ/PI19/11690013, LCF/BQ/PI20/11760031, LCF/BQ/PR20/11770012, and LCF/BQ/PR21/11840013).

APPENDIX A: REALIZATION OF THE QUDIT-QAOA WITH ATOMIC SYSTEMS

This Appendix discusses the experimental capabilities of ultracold atoms to realize the angular momentum encoding of quadratic cost functions and the mixing Hamiltonian given in Eq. (24). Specifically, quadratic cost functions can be experimentally realized in three distinct atomic platforms: Cold atomic mixtures [33], cold quantum gases in a cavity, and Rydberg atoms [34,62]. In both systems, the qudit is realized as a long collective spin by cooling atoms with internal degrees of freedom to the ground state of optical lattice sites. In the mixture system, the effective interaction between different qudits is mediated by phononic excitations, theoretically proposed in Ref. [33]. In the cavity system, the long-range interaction between the atoms is mediated via a photonic mode, which was experimentally demonstrated with a high degree of control over the interaction and the connectivity in Ref. [34].

Both the mixture and the cavity system are described by the effective Hamiltonian

$$H_C = \sum_{\mathbf{x}, \mathbf{y}} U(\mathbf{x}, \mathbf{y}) L_z(\mathbf{x}) L_z(\mathbf{y}) + \sum_{\mathbf{x}} b(\mathbf{x}) L_z(\mathbf{x}), \quad (\text{A1})$$

where \mathbf{x} and \mathbf{y} denote the minima of the lattice potential, $U(\mathbf{x}, \mathbf{y})$ is the long-range potential between the qudits, and $b(\mathbf{x})$ is a locally controllable energy shift. The mixing Hamiltonian can be engineered by standard tools such as global microwave pulses [63] that lead to terms of the form

$$H_M = \Omega \sum_{\mathbf{z}} L_x(\mathbf{x}). \quad (\text{A2})$$

A major advantage of employing these two platforms with high connectivity is the natural implementation of quadratic cost functions.

However, quadratic Hamiltonians do not suffice to encode the cost functions for all problems we consider in this paper, e.g., the graph coloring of Sec. III A. Nevertheless, cost functions containing higher powers of angular momentum operators may be implemented by Trotterization or by employing resource Hamiltonians, as demonstrated in the context of variational quantum simulation [64]. Another possibility is to employ a universal quantum computer which is based on qudits. For example, trapped-ion platforms are able to implement generalized Pauli operators and can entangle qudits, and as such, they can implement the QAOA; see Ref. [29] for more details.

APPENDIX B: OTHER OPTIMIZATION PROBLEMS

This Appendix introduces several optimization problems whose cost functions are naturally expressed in terms of qudits, namely, the knapsack problem, multiway number partitioning, and job-shop scheduling.

1. Knapsack problem

The knapsack problem consists in assigning a set of items to a container [23]. There are N different items with c copies each. Furthermore, each item i has a weight w_i and a value v_i , and the goal is to maximize the total value in the container while not exceeding a given weight limit W . The cost function of the bounded knapsack problem [65] is

$$C(\mathbf{z}) = \sum_{i=1}^N v_i z_i, \quad (\text{B1})$$

which has to be maximized and is subject to the weight constraint

$$\sum_{i=1}^N w_i z_i \leq W, \quad (\text{B2})$$

where $z_i \in [0, c]$.

The bounded knapsack problem can be straightforwardly mapped to qudits by using the angular momentum encoding discussed in Sec. II C promoting the integer variables z_i to qudits with $d = c + 1$. Using the angular momentum operator $L_{z,i}$ for $\ell = c/2$, we obtain the cost function

$$H_C = - \sum_{i=1}^N v_i L_{z,i}, \quad (\text{B3})$$

where we included a minus sign in order to transform the problem into a minimization problem. The constraints (B2) are linear in the angular momentum operators and can be implemented by using the methods developed in Sec. II E.

2. Multiway number partitioning

The number-partitioning problem is the task of partitioning a list S of n positive integers, $S = (s_1, s_2, \dots, s_n)$, into k subsets S_1, S_2, \dots, S_k , such that the numbers are as equally distributed as possible. That is, the sum of the numbers in

different subsets $V_i = \sum_{l \in S_i} s_l$ for $1 \leq i \leq k$ is requested to be as similar as possible. For instance, if $S = (1, 1, 2, 3, 4, 5)$ and $k = 2$, the optimal partitions are $(1,1,2,4)$ and $(3,5)$, which in this case yields a completely balanced partition with $V_1 = V_2 = 8$. For the case of $k = 2$ the decision version of the number-partitioning problem is NP-complete [66], though there are various algorithms that solve the problem efficiently in many cases. A trapped-ion setup for two-way number partitioning has been proposed in Ref. [67].

The multiway number partitioning can be cast into a mathematical cost function as follows. The sum of the elements in the set S_i is

$$V_i(\mathbf{z}) = \sum_{l=1}^n s_l \delta_{i,z_l}, \quad (\text{B4})$$

where the value of the variable $z_l = 1, \dots, k$ indicates the subset S_i of which s_l is a member. We then choose the cost function

$$C(\mathbf{z}) = \sum_{a < b} [V_a(\mathbf{z}) - V_b(\mathbf{z})]^2, \quad (\text{B5})$$

which minimizes the differences between the sum of the partitions with the vector $\mathbf{z} = (z_1, \dots, z_n)$.

The implementation in k -level systems z_l automatically ensures that each s_l is a member of exactly one set S_i with $i = 1, \dots, k$. In general, the realization of δ_{i,z_l} in Eq. (B4) requires a polynomial of order k in the z_l . For example, for $k = 3$, $\delta_{1,z_l} = 3 - 5z_l/2 + z_l^2/2$. Importantly, these higher-order terms within $V_i(\mathbf{z})$ act locally, but the qudits are then coupled in a pairwise fashion via $C(\mathbf{z})$.

In the literature there exist various other approaches to mathematically formulate the multiway number-partitioning problem, which become equivalent in the case of $k = 2$; see Ref. [68]. Here, we have opted for a cost function that employs integer variables and leads to a direct construction using only two-qudit interactions.

3. Job-shop scheduling

The problem consists of the task of scheduling the execution of N jobs j on M machines. Each job is subdivided into K operations, where $j_{n,k}$ denotes the operation k of job n and each operation has a predefined processing duration $p_{n,k}$, where $n \in [1, N]$ and $k \in [1, K]$. The operations of one job must be executed in a predefined order $j_{n,1} \rightarrow j_{n,2} \rightarrow \dots \rightarrow j_{n,K}$ and must not overlap. Furthermore, each operation $j_{n,k}$ has to be executed on one specific machine $m_{n,k} \in [1, M]$, and operations executed on one machine must not overlap. A schematic representation of this problem is shown in Fig. 8.

An encoding based on qudits is formulated by discretizing the time into T equally spaced time intervals, $t = 1, \dots, T$. The problem is then formulated with the variables $t_{n,k} \in \{1, \dots, T\}$, which specify the time at which the execution of operation $j_{n,k}$ on machine $m_{n,k}$ starts.

There are two constraints to be respected. First, two operations of the same job must not overlap, i.e., the predecessor operation must finish before the successor can start:

$$t_{n,k} + p_{n,k} < t_{n,k+1}, \quad (\text{B6})$$

	$t=$	1	2	3	4	T
m_1			j_{11}		j_{1N}		j_{22}	...	j_{1K}				
m_2				j_{21}			j_{N2}	...					
\vdots								...					
m_M					j_{12}				j_{2K}		j_{NK}		

FIG. 8. Job-shop scheduling. Table representation of a job schedule. The horizontal line denotes the discretized time, whereas the vertical axis denotes the machine. Filling the box corresponds to using the machine with the job $j_{n,k}$.

which needs to be fulfilled for all $k \in [1, K - 1]$ and all jobs n . Second, two operations on one machine must not overlap, i.e., only one operation can run at any given time on one machine. This means that for any two operations $j_{n,k}$ and $j_{n',k'}$ to be executed on the same machine, the one operation must either be finished before the other operation or start after it, i.e.,

$$(t_{n,k} + p_{n,k} < t_{n',k'}) \text{ XOR } (t_{n,k} > t_{n',k'} + p_{n',k'}) \quad (\text{B7})$$

for all machines m and all $(n, k), (n', k') \in o_m$, where $o_m = \{(n, k) | m_{n,k} = m\}$ is the list of operations to be run on machine m . The latter condition can also be transformed into a quadratic constraint

$$(t_{n,k} + p_{n,k} - t_{n',k'})(t_{n,k} - t_{n',k'} - p_{n',k'}) > 0. \quad (\text{B8})$$

Depending on the application scenario, multiple different cost functions can be employed [69,70]. A typical cost function is given by the average job completion time,

$$C(\mathbf{t}) = \frac{1}{N} \sum_{n=1}^N (t_{n,K} + p_{n,K}), \quad (\text{B9})$$

which represents overall machine-usage efficiency and needs to be minimized.

Another cost function is the makespan, which is the finishing time of the last operation,

$$C(\mathbf{t}) = \max_n (t_{n,K} + p_{n,K}). \quad (\text{B10})$$

In order to avoid the nonlinear max function, a linear cost function can be formulated with an additional auxiliary variable,

$$C'(\mathbf{t}') = t_{N+1,K}, \quad (\text{B11})$$

which needs to fulfill N additional linear constraints

$$t_{n,K} + p_{n,K} < t_{N+1,K} \quad (\text{B12})$$

for all jobs $n = 1, \dots, N$.

The above formulation directly lends itself to addressing the job-shop scheduling problem with QAOA by replacing the classical variable $t_{n,k}$ with an angular momentum operator with total spin $\ell = (T - 1)/2$. All constraints and the cost functions can be expressed as operators by replacing the classical variables with these L_z operators. Thus we need kN qudits ($kN + 1$ qudits) for the average completion time (makespan) formulation, where each qudit has dimension $d = T$.

- [1] D.-S. Chen, R. G. Batson, and Y. Dang, *Applied Integer Programming: Modeling and Solution* (Wiley, New York, 2009).
- [2] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization* (Wiley, New York, 1999).
- [3] T. Ishihara and S. Limmer, Optimizing the hyperparameters of a mixed integer linear programming solver to speed up electric vehicle charging control, in *Applications of Evolutionary Computation 2020* (Springer, New York, 2020), pp. 37–53.
- [4] R. L. Rardin, *Optimization in Operations Research*, 2nd ed. (Pearson, Upper Saddle River, NJ, 2016).
- [5] G. Cornuéjols, J. Peña, and R. Tütüncü, *Optimization Methods in Finance*, 2nd ed. (Cambridge University Press, Cambridge, 2018).
- [6] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, *Rep. Prog. Phys.* **83**, 054401 (2020).
- [7] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, Quantum annealing for industry applications: Introduction and review, *Rep. Prog. Phys.* **85**, 104001 (2022).
- [8] D. Venturelli, D. J. J. Marchand, and G. Rojo, Quantum annealing implementation of job-shop scheduling, [arXiv:1506.08479](https://arxiv.org/abs/1506.08479).
- [9] D. Marx, Graph colouring problems and their applications in scheduling, *Period. Polytech. Electr. Eng.* **48**, 11 (2004), <https://pp.bme.hu/ee/article/view/926>.
- [10] O. Titiloye and A. Crispin, Quantum annealing of the graph coloring problem, *Discrete Optimi.* **8**, 376 (2011).
- [11] O. Titiloye and A. Crispin, Graph coloring with a distributed hybrid quantum annealing algorithm, in *Agent and Multi-Agent Systems: Technologies and Applications*, edited by J. O’Shea, N. T. Nguyen, K. Crockett, R. J. Howlett, and L. C. Jain (Springer, Berlin, 2011), pp. 553–562.
- [12] J. Chen, T. Stollenwerk, and N. Chancellor, Performance of domain-wall encoding for quantum annealing, *IEEE Trans. Quantum Eng.* **2**, 3102714 (2021).
- [13] T. Stollenwerk, S. Hadfield, and Z. Wang, Toward quantum gate-model heuristics for real-world planning problems, *IEEE Trans. Quantum Eng.* **1**, 3101816 (2020).
- [14] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [15] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, *Algorithms* **12**, 34 (2019).
- [16] S. Lloyd, Quantum approximate optimization is computationally universal, [arXiv:1812.11075](https://arxiv.org/abs/1812.11075).
- [17] Z. Jiang, E. G. Rieffel, and Z. Wang, Near-optimal quantum circuit for Grover’s unstructured search using a transverse field, *Phys. Rev. A* **95**, 062317 (2017).
- [18] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, [arXiv:1602.07674](https://arxiv.org/abs/1602.07674).
- [19] M. S. Alam, F. A. Wudarski, M. J. Reagor, J. Sud, S. Grabbe, Z. Wang, M. Hodson, P. A. Lott, E. G. Rieffel, and D. Venturelli, Practical Verification of Quantum Properties in Quantum Approximate Optimization Runs, *Phys. Rev. Appl.* **17**, 024026 (2022).
- [20] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, Classical optimizers for noisy intermediate-scale quantum devices, in *Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Piscataway, NJ, 2020), pp. 267–277.
- [21] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, *Phys. Rev. X* **10**, 021067 (2020).
- [22] J. R. McClean, M. P. Harrigan, M. Mohseni, N. C. Rubin, Z. Jiang, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Low-Depth Mechanisms for Quantum Optimization, *PRX Quantum* **2**, 030312 (2021).
- [23] A. Lucas, Ising formulations of many NP problems, *Front. Phys.* **2** (2014).
- [24] J. R. Weggemans, A. Urech, A. Rausch, R. Spreuw, R. Boucherie, F. Schreck, K. Schoutens, J. Minář, and F. Speelman, Solving correlation clustering with QAOA and a Rydberg qudit system: A full-stack approach, *Quantum* **6**, 687 (2022).
- [25] N. Chancellor, Domain wall encoding of discrete variables for quantum annealing and QAOA, *Quantum Sci. Technol.* **4**, 045004 (2019).
- [26] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [27] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais, Qudits and high-dimensional quantum computing, *Front. Phys.* **8** (2020).
- [28] H.-H. Lu, Z. Hu, M. S. Alshaykh, A. J. Moore, Y. Wang, P. Imany, A. M. Weiner, and S. Kais, Quantum phase estimation with time-frequency qudits in a single photon, *Adv. Quantum Technol.* **3**, 1900074 (2019).
- [29] M. Ringbauer, M. Meth, L. Postler, R. Stricker, R. Blatt, P. Schindler, and T. Monz, A universal qudit quantum processor with trapped ions, *Nat. Phys.* **18**, 1053 (2022).
- [30] A. Morvan, V. V. Ramasesh, M. S. Blok, J. M. Kreikebaum, K. O’Brien, L. Chen, B. K. Mitchell, R. K. Naik, D. I. Santiago, and I. Siddiqi, Qutrit Randomized Benchmarking, *Phys. Rev. Lett.* **126**, 210504 (2021).
- [31] S. Dogra, Arvind, and K. Dorai, Determining the parity of a permutation using an experimental NMR qutrit, *Phys. Lett. A* **378**, 3452 (2014).
- [32] S. R. Cohen and J. D. Thompson, Quantum computing with circular Rydberg atoms, *PRX Quantum* **2**, 030322 (2021).
- [33] V. Kasper, D. González-Cuadra, A. Hegde, A. Xia, A. Dauphin, F. Huber, E. Tiemann, M. Lewenstein, F. Jendrzejewski, and P. Hauke, Universal quantum computation and quantum error correction with ultracold atomic mixtures, *Quantum Sci. Technol.* **7**, 015008 (2022).
- [34] A. Periwal, E. S. Cooper, P. Kunkel, J. F. Wienand, E. J. Davis, and M. Schleier-Smith, Programmable interactions and emergent geometry in an array of atom clouds, *Nature (London)* **600**, 630 (2021).
- [35] S. Hadfield, On the representation of Boolean and real functions as Hamiltonians for quantum computing, *ACM Trans. Quantum Comput.* **2**, 18 (2021).
- [36] M. Willsch, D. Willsch, F. Jin, H. D. Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, *Quantum Inf. Process.* **19**, 197 (2020).
- [37] P. Giorda, P. Zanardi, and S. Lloyd, Universal quantum control in irreducible state-space sectors: Application to bosonic and spin-boson systems, *Phys. Rev. A* **68**, 062320 (2003).

- [38] E. Mezura-Montes and C. A. Coello Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm Evol. Comput.* **1**, 173 (2011).
- [39] A. E. Smith and D. W. Coit, Constraint-handling techniques—penalty functions, in *Evolutionary Computation 2, Advanced Algorithms and Operators*, edited by T. Bäck, D. B. Fogel, and Z. Michalewicz (Institute of Physics, London, 1997), Chap. 7, pp. 41–48.
- [40] Z. Michalewicz, A survey of constraint handling techniques in evolutionary computation methods, in *Proceedings of the 4th Annual Conference on Evolutionary Programming* (MIT Press, Cambridge, MA, 1995), pp. 135–155.
- [41] Other penalty functions are possible, but they have to fulfill the minimal requirement (a) to produce the same constant value (typically zero) for all solutions which fulfill the constraint, (b) to be larger than the constant value for infeasible solutions, and (c) to increase monotonically with the degree of constraint violation.
- [42] L. Viola and S. Lloyd, Dynamical suppression of decoherence in two-state quantum systems, *Phys. Rev. A* **58**, 2733 (1998).
- [43] L. Viola, E. Knill, and S. Lloyd, Dynamical Decoupling of Open Quantum Systems, *Phys. Rev. Lett.* **82**, 2417 (1999).
- [44] E. L. Hahn, Spin Echoes, *Phys. Rev.* **80**, 580 (1950).
- [45] H. Y. Carr and E. M. Purcell, Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments, *Phys. Rev.* **94**, 630 (1954).
- [46] S. Meiboom and D. Gill, Modified spin-echo method for measuring nuclear relaxation times, *Rev. Sci. Instrum.* **29**, 688 (1958).
- [47] D. A. Lidar, Review of decoherence free subspaces, noiseless subsystems, and dynamical decoupling, in *Quantum Information and Computation for Chemistry* (Wiley, New York, 2012), pp. 295–354.
- [48] J. C. Halimeh, H. Lang, J. Mildenerger, Z. Jiang, and P. Hauke, Gauge-symmetry protection using single-body terms, *PRX Quantum* **2**, 040311 (2021).
- [49] V. Kasper, T. V. Zache, F. Jendrzewski, M. Lewenstein, and E. Zohar, Non-Abelian gauge invariance from dynamical decoupling, *Phys. Rev. D* **107**, 014506 (2023).
- [50] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Hybrid quantum-classical algorithms for approximate graph coloring, *Quantum* **6**, 678 (2022).
- [51] R. Vujanic, P. Esfahani, P. Goulart, S. Mariethoz, and M. Morari, A decomposition method for large scale MILPs, with performance guarantees and a power system application, *Automatica* **67**, 144 (2016).
- [52] T. Mao, X. Zhang, and B. Zhou, Intelligent energy management algorithms for EV-charging scheduling with consideration of multiple EV charging modes, *Energies* **12**, 265 (2019).
- [53] J. Han, J. Park, and K. Lee, Optimal scheduling for electric vehicle charging under variable maximum charging power, *Energies* **10**, 933 (2017).
- [54] C. Dalyac, L. Henriët, E. Jeandel, W. Lechner, S. Perdrix, M. Porcheron, and M. Veshchezerova, Qualifying quantum approaches for hard industrial optimization problems. a case study in the field of smart-charging of electric vehicles, *EPJ Quantum Technol.* **8**, 12 (2021).
- [55] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey *et al.*, SciPy 1.0: Fundamental algorithms for scientific Computing in Python, *Nat. Methods* **17**, 261 (2020).
- [56] N. Hansen, The CMA evolution strategy: A comparing review, in *Towards a New Evolutionary Computation*, edited by J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Studies in Fuzziness and Soft Computing Vol. 192 (Springer, New York, 2006), pp. 75–102.
- [57] M. Shibata, <https://github.com/CyberAgentAILab/cmaes>, 2022.
- [58] C. Moussa, H. Calandra, and V. Dunjko, To quantum or not to quantum: Towards algorithm selection in near-term quantum optimization, *Quantum Sci. Technol.* **5**, 044009 (2020).
- [59] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, in *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14* (MIT Press, Cambridge, MA, 2014), Vol. 2, pp. 2933–2941.
- [60] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, The loss surfaces of multilayer networks, in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, edited by G. Lebanon and S. V. N. Vishwanathan, Proceedings of Machine Learning Research Vol. 38 (PMLR, San Diego, CA, 2015), pp. 192–204.
- [61] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization methods for large-scale machine learning, *SIAM Rev.* **60**, 223 (2018).
- [62] G. Anikeeva, O. Marković, V. Borish, J. A. Hines, S. V. Rajagopal, E. S. Cooper, A. Periwal, A. Safavi-Naeini, E. J. Davis, and M. Schleier-Smith, Number partitioning with grover's algorithm in central spin systems, *PRX Quantum* **2**, 020319 (2021).
- [63] H. Strobel, W. Muessel, D. Linnemann, T. Zibold, D. B. Hume, L. Pezzè, A. Smerzi, and M. K. Oberthaler, Fisher information and entanglement of non-Gaussian spin states, *Science* **345**, 424 (2014).
- [64] C. Kokail, C. Maier, R. van Bijnen, T. Brydges, M. K. Joshi, P. Jurcevic, C. A. Muschik, P. Silvi, R. Blatt, C. F. Roos, and P. Zoller, Self-verifying variational quantum simulation of lattice models, *Nature (London)* **569**, 355 (2019).
- [65] D. Pisinger, A minimal algorithm for the bounded knapsack problem, in *Integer Programming and Combinatorial Optimization*, edited by E. Balas and J. Clausen (Springer, Berlin, 1995), pp. 95–109.
- [66] S. Mertens, The easiest hard problem: Number partitioning, [arXiv:cond-mat/0310317](https://arxiv.org/abs/cond-mat/0310317).
- [67] P. Hauke, L. Bonnes, M. Heyl, and W. Lechner, Probing entanglement in adiabatic quantum optimization with trapped ions, *Front. Phys.* **3** (2015).
- [68] R. E. Korf, Objective functions for multi-way number partitioning, in *Third Annual Symposium on Combinatorial Search* (Association for the Advancement of Artificial Intelligence, Washington, DC, 2010), pp. 71–72.
- [69] A. Jain and S. Meeran, Deterministic job-shop scheduling: Past, present and future, *Eur. J. Oper. Res.* **113**, 390 (1999).
- [70] W.-Y. Ku and J. C. Beck, Mixed integer programming models for job shop scheduling: A computational analysis, *Comput. Oper. Res.* **73**, 165 (2016).