Inequality constraints in variational quantum circuits with qudits

Alberto Bottarelli [1,2,*] Sebastian Schmitt [3] and Philipp Hauke [1,2]
[1] Pitaevskii BEC Center, CNR-INO and Dipartimento di Fisica, Università di Trento, I-38123 Trento, Italy
[2] INFN-TIFPA, Trento Institute for Fundamental Physics and Applications, Trento, Italy
[3] Honda Research Institute Europe GmbH, Carl-Legien-Str. 30, 63073 Offenbach, Germany

(Received 18 October 2024; accepted 25 June 2025; published 29 August 2025)

Quantum optimization is emerging as a prominent candidate for exploiting the capabilities of near-term quantum devices. Many application-relevant optimization tasks require the inclusion of inequality constraints, usually handled by enlarging the Hilbert space through the addition of slack variables. This approach, however, requires significant additional resources especially when considering multiple constraints. Here, we study an alternative direct implementation of these constraints within the quantum approximate optimization algorithm, achieved using qudit-sum gates, and compare it to the slack variable method generalized to qudits. We benchmark these approaches on three paradigmatic optimization problems. We find that the direct implementation of the inequality penalties vastly outperforms the slack variables method, especially when studying real-world inspired problems with many constraints. Within the direct penalty implementation, a linear energy penalty for unfeasible states outperforms other investigated functional forms, such as the canonical quadratic penalty. The proposed approach may thus be an enabling step for approaching realistic industry-scale and fundamental science problems with large numbers of inequality constraints.

DOI: 10.1103/3l96-41xf

I. INTRODUCTION

In recent years, quantum optimization [1,2] has emerged as a highly promising field within near-term quantum computation, as it may help to solve combinatorial optimization problems prevalent in both physics and industrial applications. Examples include Max-Cut, k-sat, and various portfolio management, scheduling, and assignment problems. These problems have in common that they are formulated as a quadratic unconstrained binary optimization (QUBO) problem [3,4], where the cost function $C(\boldsymbol{x})$ is defined on the boolean cube and the objective is to find a string $\boldsymbol{x} \in \{0,1\}^N$ that minimizes or maximizes $C(\boldsymbol{x})$. Numerous quantum algorithms have been developed to tackle these optimization tasks, leveraging different concepts such as adiabaticity, employed in quantum annealing [5,6], or the variational principle, at the basis of the Variational Quantum Eigensolver [7–10] or Quantum Approximate Optimization Algorithm (QAOA) [11,12]. These algorithms are formulated in terms of a cost Hamiltonian, which is the matrix representation of the QUBO cost function. For realistic applications, a central aspect of the problems studied with quantum optimization algorithms is given by the presence of (typically many) linear constraints. See Ref. [13] for a recent example from the electromobility domain. Equality constraints are commonly implemented in quantum optimization routines by adding penalty terms to the cost Hamiltonian, analogous to classical penalty functions in constrained optimization approaches. However, many constraints in realistic applications are given in the form of inequalities. In a formulation such as QUBO the standard method for inequality constraints requires the introduction of additional slack variables in the cost Hamiltonian. Compared to equality constraints, which do not require any auxiliary variables, this constitutes a significant overhead and limits the approachable problem sizes as well as the achievable solution performance. Therefore, the current approach to the inclusion of inequality constraints represents a major bottleneck for realistic quantum-optimization protocols, in particular for their implementations on current noisy intermediate-scale quantum (NISQ) hardware.

In this paper, we focus on incorporating Hamming-weight inequality constraints in QAOA, a variational quantum algorithm suitable for NISQ devices. We propose a quantum circuit that adds energy penalties to the Hamiltonian, which produces additional phase shifts to the unfeasible states. This circuit utilizes additional ancillary qudit and qudit sum gates without requiring additional slack variables. We discuss the necessary resource scaling in terms of qudit levels and gates, and compare it to the standard constraint-handling approach based on slack variables. We illustrate the direct energy penalty on three different problems: a randomly interacting Ising spin model, a constrained state sampling problem, and an electric vehicle (EV) charging problem [see Figs. 1(a) and 1(b)]. We benchmark its performance against the qudit slack-variable approach using exact numerical simulations of a QAOA protocol [Fig. 1(c)], demonstrating that the functional form of the energy penalty plays a crucial role in the efficiency of the quantum optimization protocol. In particular,

*Contact author: alberto.bottarelli@untitn.it

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

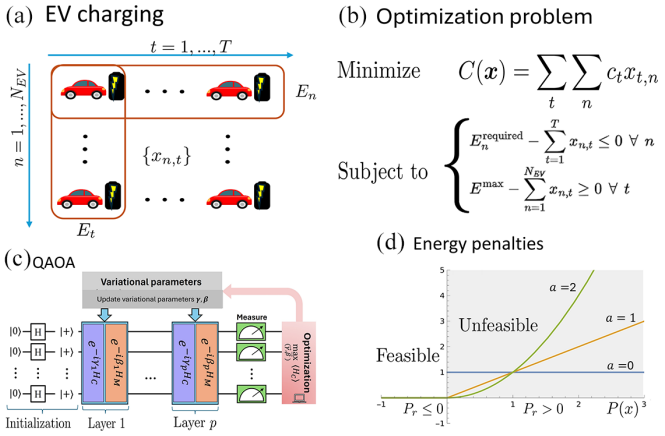

FIG. 1. Schematic representation of the aspects of an electric vehicle (EV) charging problem as a typical realistic optimization problem. (a) Sketch of the problem structure where $n = 1 \ldots N_{\mathrm{EV}}$ EVs need to be charged over a period of $t = 1 \ldots T$ time steps. Each EV n is represented by a set of binary variables $x_{n,t}$, indicating whether it is being charged at time t. The charging energy of a given EV at the end of the schedule is E_n (horizontal red square), while the total power provided at any time is E_t (vertical red square). (b) The optimization problem amounts to minimizing the total cost C, given as the sum of costs at each time step c_t, under several constraints: The maximal charging power at each time step cannot exceed a given value $E_{\max}$ and the charging energy of each EV needs to exceed a minimum value E_0 at the end of the schedule. (c) Schematic representation of the QAOA procedure. After initialization, the quantum register is subject to p layers consisting of the alternating application of a Hamiltonian representing the cost function, H_C, and a non-commuting "mixer" Hamiltonian, H_M. The corresponding rotation angles $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are variational parameters that are iteratively updated in interplay with a classical optimizer until the measured cost function converges to low values. (d) In this work, we propose to add a penalty term to the cost Hamiltonian H_C for each inequality constraint $P_r(\boldsymbol{x})$, $\boldsymbol{x} = (x_{1,1}, x_{1,2}, \ldots)^{\mathrm{T}}$. The penalty vanishes on feasible (constraint-obeying) configurations while it adds an additional cost to unfeasible (constraint-violating) configurations. Suitable choices are functions that increase as a power law with increasing constraint violation with power a. Importantly, the penalty can be implemented without the need for slack variables.

we find that a linear penalty function consistently outperforms constant and quadratic forms [Fig. 1(d)]. Moreover, the flexibility in choosing the functional form of the direct energy penalty allows for further optimization. Finally, we show that the proposed direct unitary method surpasses the qudit slack variable method, especially in scenarios involving multiple constraints.

This enhancement is facilitated by tapping into the rapid recent developments in qudit quantum information processing. Precise and universal control of quantum systems with $d > 2$ levels has been realized on various platforms ranging from trapped ions, over neutral atoms, to superconducting systems [14–21]. By compressing quantum information into higher-dimensional objects, the use of qudits is particularly appealing in the NISQ era, where even a constant or polynomial saving of resources can have a significant effect. The usefulness of qudits has been discussed, for example, in the context of combinatorial quantum optimization problems [22–25], where the natural description is in terms of d-ary integer variables $x_i \in \{0, \ldots, d-1\}$ rather than only binary variables. Our results illustrate another useful application by using ancilla qudits to implement inequality constraints as a direct energy penalty in the cost function, which may generate a significant performance advantage for realistic optimization problems.

The remainder of this paper is organized as follows. In Sec. II, we recap QAOA for qubit and qudit systems. In Sec. III, we discuss possible ways of imposing constraints on the solutions of the problem. In particular, we compare their resource scaling for the standard approach via slack variables and the method that is the main subject of this study, the inclusion of a nonlinear energy penalty in the cost function. Section IV presents a numerical study of the approaches described in previous sections, showing the beneficial performance of the direct approach to constraints through diagonal unitaries. We present our conclusions in Sec. V.

II. REVIEW OF THE QAOA

The QAOA [11,12,26] is a hybrid variational quantum algorithm [1,7] designed to solve combinatorial optimization problems. Its schematic is shown in Fig. 1(c). Given a cost function $C(\boldsymbol{x})$ of N classical variables $\boldsymbol{x} = (x_1, \ldots, x_N)$, the QAOA aims at finding the solution to

$$\min_{\boldsymbol{x} \in \{0,1\}^N} C(\boldsymbol{x}), \tag{1}$$

yielding the corresponding optimal configuration $\boldsymbol{x}_{\min}$. For now, we assume, as usual, the variables to be binary, $x_i \in \{0, 1\}$. Extending this scheme to multilevel qudits is straightforward and is described below. An important class of problems that is usually addressed with the QAOA consists of QUBO problems whose cost function is expressed as

$$C(\boldsymbol{x}) = 4\boldsymbol{x}^T J \boldsymbol{x} = 4 \sum_{i,j=1}^{N} J_{ij} x_i x_j \,, \tag{2}$$

where J is a symmetric and real $N \times N$ cost matrix, and where we included a factor of 4 for convenience. Such QUBO problems are important for industrial applications, as many paradigmatic problems can be mapped onto such a form [4,6], as well as for physics questions given their equivalence to Ising Hamiltonians.

In order to solve the problem using quantum computation, one promotes the classical variables to operators acting on a Hilbert space $\mathcal{H}_C = \mathbb{C}^{2N}$. This is usually defined as

$$x_i \to \frac{1 + \sigma_z^i}{2} \,, \tag{3}$$

where σ_z^i is the z-Pauli spin operator for qubit i. The cost function $C(\boldsymbol{x})$ is promoted to a Hamiltonian H_C by replacing each variable with the corresponding operator, as indicated in Eq. (3). Each state of the computational basis $\{|\boldsymbol{x}\rangle\}$ represents a classical bit string $\boldsymbol{x}$ and H_C is diagonal in this basis with eigenvalues $C(\boldsymbol{x})$:

$$H_C \, |\boldsymbol{x}\rangle = C(\boldsymbol{x}) \, |\boldsymbol{x}\rangle \,. \tag{4}$$

The solution of the optimization problem is thus encoded in the ground state of the Ising Hamiltonian

$$H_C = \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z\,, \tag{5}$$

where $h_i = 2\sum_{j=1}^{N} J_{ij}$.

The QAOA ansatz for finding the ground state of H_C consists in preparing a parametrized trial state

$$|\psi(\boldsymbol{\alpha},\boldsymbol{\beta})\rangle = U(\boldsymbol{\alpha},\boldsymbol{\beta})\,|\psi_0\rangle\,, \tag{6}$$

generated by alternating parametrized unitaries in the following way:

$$U(\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{l=1}^{p} e^{i\alpha_l H_C} e^{i\beta_l H_M}\,. \tag{7}$$

In the above, $|\psi_0\rangle$ is usually taken to be $\frac{1}{\sqrt{D}}\sum_{\boldsymbol{x}}|\boldsymbol{x}\rangle$, with $D = \dim\mathcal{H}_C = 2^N$ the Hilbert space dimension and $|\boldsymbol{x}\rangle$ all the possible states of the computational basis. A schematic representation of this circuit is shown in Fig. 1(c). Apart from a diagonal cost Hamiltonian H_C, the state is acted on with a nondiagonal mixing operator H_M, which has the role of inducing transitions between the different states. It is thus fundamental that the mixing operator and the cost Hamiltonian do not commute, i.e., $[H_C, H_M] \neq 0$. Due to the formulation of cost Hamiltonians in terms of Pauli σ_z operators, the most common mixer has the form $H_M = \sum_i \sigma_i^x$, which is also used in this work. The depth of the ansatz is defined by the integer hyperparameter p, also known as the number of layers of the algorithm. The expectation value of the cost Hamiltonian is calculated with the trial state of Eq. (6) as

$$E(\boldsymbol{\alpha},\boldsymbol{\beta}) = \langle\psi(\boldsymbol{\alpha},\boldsymbol{\beta})|H_C|\psi(\boldsymbol{\alpha},\boldsymbol{\beta})\rangle\,. \tag{8}$$

The set of $2p$ variational parameters $(\boldsymbol{\alpha},\boldsymbol{\beta})$ is used to minimize the expectation value by a classical optimization routine. Due to the variational principle, this value is ensured to be an upper bound to the ground-state energy of H_C.

It is rather straightforward to generalize the QAOA to solve problems formulated with d-ary integer variables $x_i \in \{0, ..., d-1\}$, which are represented by qudits. The local Hilbert space dimension of each operator presenting classical variables becomes $\mathbb{C}^d$ and the Pauli matrices are replaced with angular momentum operators $\{L_x, L_y, L_z\}$ with representation index $\ell = \frac{d-1}{2}$. Given a cost function $C(\boldsymbol{x})$ defined in terms of such d-ary variables, the quantum-mechanical cost Hamiltonian is obtained by replacing each variable as

$$x_i \to L_z^i + \tfrac{d-1}{2}. \tag{9}$$

For qudits, the available unitaries are given by the operators of the group SU(d), which has d^2-1 generators. To be able to generate all possible unitaries, it is necessary to modify the mixing operator. To the trivial generalization $H_M = \sum_{i=1}^{N} L_x^i$, we add a squeezing operator $\sum_i (L_z^i)^2$. It was shown [16,27] that the set $\{L_z^i, (L_z^i)^2, L_x^i\}$ allows one to generate all possible unitaries of SU(d) by repeated finite rotations of the qudit. Therefore, we use the mixer for qudits in the form

$$H_M = \sum_i \left(\beta L_x^i + \gamma\left(L_z^i\right)^2\right). \tag{10}$$

Once the cost Hamiltonian and the mixer are defined, the algorithm works in the same way as the qubit version. Below, we will use qudits only for implementing the slack variables, though many application-relevant cost functions can naturally benefit from formulation in terms of qudits [22–24,28–32].

III. CONSTRAINT HANDLING IN QAOA

For many physical and industrial problems, minimizing a cost function as in Eq. (1) does not necessarily suffice to give the desired solution. Often, the problem is subject to a set of additional constraints that need to be fulfilled. A common approach to handling constraints in optimization is by using penalty functions [33–35]. In this methodology the constrained problem is transformed to an unconstrained optimization problem by augmenting the cost function with penalty terms. These terms shift unfeasible solutions to high cost values, but leave the cost of feasible solutions unchanged, thereby ensuring that minimal cost solutions are feasible. In classical optimization, equality and inequality constraints can directly be transformed into penalty terms in the cost function. However, inequality terms require nonpolynomial penalty functions [33,35].

For quantum optimization approaches the problem is put in the form of a QUBO problem [4–6]. In that case only equality constraints can be added directly as quadratic penalty functions to the cost Hamiltonian [6,36] since they preserve the QUBO form. The standard approach for inequality constraints first transforms them into equality constraints by adding appropriately chosen slack variables [6,37], and then adds these extended equality constraints as quadratic penalty terms to the Hamiltonian. Significant overhead is incurred by this procedure as the slack variable extends the required Hilbert space substantially, and by the quadratic penalty terms additional interactions between all variables (slack and system) are introduced. Avoiding these overheads and the encompassing inefficiencies of the standard approach for handling inequality constraints in quantum optimization is the subject of this work.

We focus on optimization problems of the form

$$\boldsymbol{x}_{\min} = \underset{\boldsymbol{x}\in\{0,1\}^N}{\operatorname{argmin}}\, C(\boldsymbol{x}), \tag{11}$$

subject to R inequality constraints

$$P_r(\boldsymbol{x}) \leqslant 0, \qquad r = \{1, ..., R\}, \tag{12}$$

where r labels all the constraints included in the problem and $P_r(\boldsymbol{x})$ is a classical function of the search variables characterizing the constraint and which is given as part of the problem formulation.

We compare our proposed approach to the standard approach based on slack variables. In contrast to purely qubit-based methods, we will employ qudit slack variables, which avoid the overhead of encoding larger integers into multiple binary variables and thus can be seen as a best-case scenario for slack variable approaches (Sec. III A). As our main contribution, we propose a direct implementation of energy penalties, which acts only on the unfeasible subspace (Sec. III B) and allows one to keep the structure of the

variational algorithm unchanged [schematics in Fig. 1(c)]. In our numerical benchmarks in the next section, we compare this proposed approach to the qudit-based slack-variable implementation. For completeness, other possible approaches for inequality-constraint handling that have been discussed in the literature are mentioned in Sec. III E.

A. Slack variables as qudit operators

The standard approach to introduce inequality constraints in quantum optimization is via the use of slack variables [6,37]. Here, we first give a short review of slack variables for classical optimization problems and then discuss how to use qudits in order to encode them in quantum optimization algorithms.

We assume the function $P_r(\boldsymbol{x})$ to have N_r possible values, i.e., $P_r(\boldsymbol{x}) \in \{p_1, ..., p_{N_r}\}\ \forall \boldsymbol{x}$. Of these values, the first N_r^{feas} are negative or zero (i.e., they denote feasible solutions) and $N_r - N_r^{\text{feas}}$ are greater than zero (i.e., they denote unfeasible solutions).

Equality constraints can be directly incorporated as quadratic penalty terms to the cost function [36]. If the constraints are linear, the QUBO form is preserved. For inequality constraints this is not directly possible. The standard is to transform inequality constraints into equality constraints with the introduction of an appropriate slack variable s_r such that

$$P_r(\boldsymbol{x}) \leqslant 0, \tag{13}$$

$$\Leftrightarrow P_r(\boldsymbol{x}) + s_r = 0. \tag{14}$$

The slack variables are only allowed to take non-negative values ($s_r \geqslant 0$) and their range is determined by the values the constraint function attains for feasible configurations: only when Eq. (13) is satisfied can a value for s_r be found such that Eq. (14) is satisfied. Technically, $s_r \in \{-P_r(\boldsymbol{x}) : \forall \text{ feasible } \boldsymbol{x}\}$. This means that each slack variable s_r can assume N_r^{feas} values.

The transformed equality constraints of Eq. (14) can now be added as quadratic penalty terms to the cost function leading to the final penalized cost function considering all R different constraints:

$$C_{\text{slack}}(\boldsymbol{x}) = C(\boldsymbol{x}) + \sum_{r=1}^{R} \lambda_r [P_r(\boldsymbol{x}) + s_r]^2. \tag{15}$$

One slack variable s_r needs to be introduced for each inequality constraint, and each term gets a penalty factor $\lambda_r > 0$. In a standard QUBO formulation each slack variable needs to be encoded with multiple binary variables, which requires at least $\sum_r \log_2 N_r^{\text{feas}}$ auxiliary binary variables for a given constraint, i.e., slack variable s_r [35,37]. Typically, N_r^{feas} is larger than 2, rendering the associated spatial resource overhead rather costly, especially for current NISQ devices.

This resource overhead can be reduced by the use of qudits to represent the slack variables. If the qudit dimension d matches the number of feasible values N_r^{feas}, the computational basis needs to be extended by one qudit state for each constraint, i.e., slack variable,

$$|\boldsymbol{x}\rangle \Rightarrow |\boldsymbol{x}, \{s_1, \ldots, s_R\}\rangle \equiv |\boldsymbol{x}, \boldsymbol{s}\rangle. \tag{16}$$

In the quantum formulation, the constraint function as well as the slack variable are represented by operators with the appropriate eigenvalues,

$$(\hat{P}_r + \hat{S}_r)\,|\boldsymbol{x}, \boldsymbol{s}\rangle = [P_r(\boldsymbol{x}) + s_r]\,|\boldsymbol{x}, \boldsymbol{s}\rangle. \tag{17}$$

Feasible configurations of the search variables $\boldsymbol{x}$ can be identified by finding the appropriate basis state where the qudit slack variable has the correct value to produce a zero eigenvalue in the above equation, i.e., $P_r(\boldsymbol{x}) + s_r = 0$.

Interestingly, the number of additional dimensions due to the auxiliary qudits (slack variables) is proportional to N_r^{feas}, the number of feasible configurations of the corresponding constraint. Therefore, the more feasible configurations that exist, the larger the dimension of the auxiliary qudit Hilbert space. The dimension of the full Hilbert space including the auxiliary slack qudits increases exponentially with the number of constraints, i.e., $\dim(\mathcal{H}_{\text{slack}}) = 2^N \prod_{r=1}^{R} N_r^{\text{feas}}$ and $\mathcal{H}_{\text{slack}} = \mathcal{H}_C \otimes \prod_{r=1}^{R} \mathbb{C}^{N_r^{\text{feas}}}$. However, given a feasible configuration $|\boldsymbol{x}\rangle$, only one state of the extended Hilbert space $|\boldsymbol{x}, \boldsymbol{s}\rangle$ will represent a feasible total state [the one with eigenvalue $s_r = -P_r(\boldsymbol{x})$], rendering the majority of added quantum states unfeasible. Thus, it can be anticipated that for only lightly constrained problems (i.e., with large N_r^{feas}), the effective optimization problem including the slack variables is more difficult due to the large number of added unfeasible solutions. Specifically, the ratio of feasible solutions with respect to the Hilbert space dimension will decrease as $\frac{1}{N_r^{\text{feas}}}$ for each slack variable included.

The final form for the constrained Hamiltonian is then

$$H_{\text{slack}} = H_C + \sum_{r=1}^{R} \lambda_r (\hat{P}_r + \hat{S}_r)^2. \tag{18}$$

The quadratic form of the penalty terms guarantees that it can be made to vanish for all feasible configurations.

Adding qudit slack variables to the system requires modifications to the form of the QAOA mixer as described at the end of Sec. II. Explictly, we use the following form:

$$H_M = \beta \left(\sum_{i=1}^{N} \sigma_x^i + \sum_{r=1}^{R} L_x^{s_r} \right) + \gamma \sum_{r=1}^{R} \left(L_z^s\right)^2, \tag{19}$$

where β and γ are the variational parameters related respectively to the x and z components of the mixer. The second term is necessary due to the higher dimensionality of the qudits representing the slack variables: as explained around Eq. (10) of Sec. II, the additional $(L_z^s)^2$ allow for universality, i.e., reaching all qudit states. Apart from this change in the mixing operator, the QAOA is performed as described in Sec. II, just with the cost Hamiltonian given by Eq. (18).

B. Direct implementation of penalty terms for inequality constraints

We propose another possibility for including inequality constraints by using a penalized Hamiltonian of the form

$$H_{\text{penal}} = H_C + \sum_{r=1}^{R} \lambda_r \hat{G}_r, \tag{20}$$

where R is the total number of constraints and $\lambda_r > 0$ are penalty factors for each constraint. We introduced the operators $\hat{G}_r$ whose eigenvalues depend on the corresponding constraint function given in Eq. (12), namely,

$$\hat{G}_r \,|\boldsymbol{x}\rangle = g(P_r(\boldsymbol{x}))\,|\boldsymbol{x}\rangle. \tag{21}$$

(An explicit construction of $\hat{G}$ is given below.) The function $g(\cdot)$ is a penalty function as it is used in classical optimization. Its purpose is to increase the energy of unfeasible solutions while leaving the energy of feasible solutions unchanged. Therefore, it vanishes for feasible solutions $\boldsymbol{x}$, i.e., $g[P_r(\boldsymbol{x})] = 0$ when $P_r(\boldsymbol{x}) \leqslant 0$, and positive for unfeasible solutions, i.e., $g[P_r(\boldsymbol{x})] > 0$ when $P_r(\boldsymbol{x}) > 0$. In principle, arbitrary functional forms of g are possible. However, to facilitate a gradient towards the feasible subspace, we moreover desire $g(\cdot)$ to be nondecreasing for positive arguments. These requirements can be achieved by the typical choice [34]:

$$g(y) = y^a\,\Theta(y), \tag{22}$$

where $\Theta(y)$ is the Heaviside step function and $a \geqslant 0$ is an exponent that can be chosen freely. These types of penalty functions are illustrated in Fig. 1(d).

Including these penalties, the unitary operator to generate the trial wave function [Eq. (7)] is updated to

$$U(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{l=1}^{p} e^{i\alpha_l H_{\text{penal}}} e^{i\beta_l H_M} \quad \text{with} \tag{23}$$

$$e^{i\alpha_l H_{\text{penal}}} = e^{i\alpha_l (H_C + \sum_{r=1}^{R} \lambda_r \hat{G}_r)}. \tag{24}$$

The rotation generated by the penalizing Hamiltonian puts phases $\sim\alpha_l\lambda_r$ on the unfeasible states, thus permitting the QAOA procedure to select them out.

Since the terms proportional to the constraints are diagonal in the computational basis, it is possible to find constructions to implement $e^{i\alpha_l \sum_{r=1}^{R} \lambda_r \hat{G}_r}$ using either Fourier analysis or ancilla registries [38,39]. These methods are defined using only qubit systems and their efficiency is bound by the problem type and instance, and by the properties of the platform chosen for implementation.

In this work we propose an efficient implementation of diagonal unitaries of the form of Eqs. (23) and (24). In Sec. III C, we describe a construction that is valid for constraint functions $P_r(\boldsymbol{x})$, which only depend on the Hamming weight, i.e., the magnetization m_r of a subset $\mathcal{I}_r$ of problem qubits,

$$P_r(\boldsymbol{x}) = P_r(m_r(\boldsymbol{x})) \tag{25}$$

with

$$m_r(\boldsymbol{x}) = \sum_{x_i \in \mathcal{I}_r} x_i, \tag{26}$$

where $\mathcal{I}_r$ denotes a subset of the N qubits (generalization to summations of x_i and $\bar{x}_i$ is straightforward).

In contrast to the slack-variable approach, the ancilla qudit does not enter the cost function, but serves only to imprint appropriate phases according to Eq. (24) onto the trial wave function. One key advantage of this method is that there is no increase in the dimension of the total Hilbert space within which the solutions are searched. In particular, the fraction of feasible solutions in the Hilbert space is constant, regardless of the number of constraints. This is in contrast to the slack-variable approach, where each constraint increases the Hilbert space dimension by a factor of N_r^{feas}. Even more striking, the number of infeasible solutions is increased by a factor of $(N_r^{\text{feas}} - 1)$, since only one configuration of the slack variable represents a feasible solution. This is the main bottleneck and main issue of the slack-variable–based approach.

As shown in the next section, the bottleneck of the proposed procedure is the dimensionality of the ancilla qudit. For a constraint on the magnetization of N qubits considered here, it needs to be at least equal to $N + 1$. However, since the ancilla qudit can be used for many constraints simultaneously, we can expect the proposed method to be advantageous in particular when many constraints are present simultaneously, with each involving only a restricted number of problem qubits and small to intermediate sets of possible values P_r.

C. Penalties for Hamming weight constraints

This section shows how to implement unitaries of a Hamiltonian of the form (20) with constraints of the form of Eq. (25) using only one qudit ancilla. There exist ways to exponentiate arbitrary boolean functions in quantum computers. In general, these require computing the Fourier transform of the function that needs to be exponentiated. In principle, this allows one to implement diagonal unitaries of the form of Eq. (24), albeit with unfavorable, typically exponential scaling in the number of gates (see, e.g., [38,39]) or with a problem-dependent reduced gate count.

Here, we provide an alternative way to implement such a unitary using only one ancilla qudit and without the need to compute the Fourier transform. We assume a constraint of the form of Eq. (25), which only depends on the total Hamming weight (or magnetization),

$$P_r(\boldsymbol{x}) \equiv P_r(m(\boldsymbol{x})) \text{ with } m(\boldsymbol{x}) = \sum_{i=1}^{N} x_i \tag{27}$$

of an N-qubit basis state

$$|\boldsymbol{x}\rangle = |x_1, x_2, \ldots, x_N\rangle \quad (x_i \in \{0, 1\}). \tag{28}$$

For ease of notation, we assume only one constraint involving all N qubits, i.e., we drop the label r for the constraint function. The formulation of the general case with multiple constraints and where each constraint applies to a subset of qubits is straightforward.

The penalty function $P(m)$ indicates which values of m characterize feasible $[P(m) \leqslant 0]$ and unfeasible $[P(m) > 0]$ solutions. We denote the set of unfeasible values for m as $\mathcal{I} = \{m : P(m) > 0\}$ and the number of unfeasible values by $N_{\mathcal{I}} = |\mathcal{I}|$. For an N qubit state, the Hamming weight can take the $N + 1$ integer values $m \in \{0, 1, \ldots, N\}$, which determines the dimension of the ancilla qudit to be $N + 1$,

$$|y\rangle_a \text{ with } y \in \{0, 1, \ldots, N\}. \tag{29}$$

The circuit implementing the proposed approach is depicted in Fig. 2. We start from a product state of the problem qubits, which can be in an arbitrary state, and the ancilla qudit,

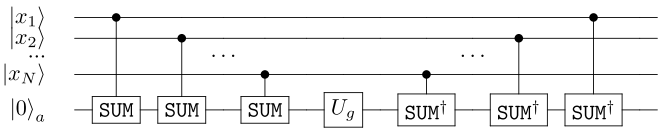

FIG. 2. Circuit showing how an arbitrary phase $g(m)$ as a function of the magnetization $m(\boldsymbol{x}) = \sum_i x_i$ can be applied on the qubit register using one ancilla qudit $|y\rangle_a$.

which is initialized in $|0\rangle_a$, i.e.,

$$|\psi\rangle\,|0\rangle_a = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle\,|0\rangle_a. \tag{30}$$

Then, we apply a sum gate that adds the boolean value corresponding to one of the qubits onto the target ancilla. Repeating this operation for each qubit, we end up in a state where the ancilla qudit has the value $m = m(\boldsymbol{x}) = \sum_i x_i$,

$$\prod_i \mathtt{SUM}_{i\to a}\,|\psi\rangle\,|0\rangle_a = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle\,|m\rangle_a\,. \tag{31}$$

After that, we apply a simple phase shift U_g to the ancilla qudit, where a phase is only added to the unfeasible states and the phase itself is determined by the penalty function $g(\cdot)$ as in Eq. (22), such that $U_g\,|m\rangle_a = e^{ig[P(m)]}\,|m\rangle_a$. Such an operation is easily realized in quantum devices as it only requires oone to be able to selectively apply a different phase to each level of a single qudit. The form of $g[P(m)]$ as a function of the qudit level index m is arbitrary in this case.

This operation takes the state to

$$\sum_{\boldsymbol{x}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle\,U_g\,|m\rangle_a = \sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\notin\mathcal{I}}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle\,|m\rangle_a \tag{32}$$

$$+ \sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\in\mathcal{I}}} c_{\boldsymbol{x}} e^{ig[P(m)]}\,|\boldsymbol{x}\rangle\,|m\rangle_a. \tag{33}$$

Here, we made the conditional phase, which is only applied to the unfeasible states, apparent by splitting the sums explicitly. After undoing the sum gates, the final state is

$$\left(\sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\notin\mathcal{I}}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle + \sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\in\mathcal{I}}} c_{\boldsymbol{x}} e^{ig[P(m)]}\,|\boldsymbol{x}\rangle\right)|0\rangle_a. \tag{34}$$

This realizes the desired application of a unitary operator parametrized by a penalty function $g(\cdot)$, which introduces a phase shift only for those basis states in the qubit quantum state $|\psi\rangle$ that are infeasible [as indicated by $P(m)$].

For the case in which we want to constrain the magnetization to be below a given value m_0, the constraint function is just linear, $P(m) = m - m_0 \leqslant 0$. With the penalty function $g(\cdot)$ of Eq. (22) this leads the conditional phases to be

$$g[P(m)] = \Theta(m - m_0)\,(m - m_0)^a. \tag{35}$$

The corresponding final state is

$$\left(\sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\notin\mathcal{I}}} c_{\boldsymbol{x}}\,|\boldsymbol{x}\rangle + \sum_{\substack{\boldsymbol{x}\\ m(\boldsymbol{x})\in\mathcal{I}}} c_{\boldsymbol{x}} e^{i[m(\boldsymbol{x})-m_o]^a}\,|\boldsymbol{x}\rangle\right)|0\rangle_a, \tag{36}$$

which adds a phase to those states with $m \in \mathcal{I} = \{m_0 + 1, m_0 + 2, \ldots, N\}$.

The entire procedure for arbitrary $g[P(m)]$ requires only $2N$ qudit controlled sum gates and a single qudit phase shift unitary U_g. A requirement for it to work is the availability of qudits with at least $N + 1$ levels, where N is the number of qubits involved in the constraint.

This procedure becomes particularly favorable when many constraints need to be implemented. Each constraint is implemented sequentially by running the above circuit acting on the involved subset of qubits with the corresponding constraint functions. A single ancilla qudit is sufficient as it can be reused for all constraints. The use of multiple ancillas permits one to parallelize the implementation of different constraints.

The above procedure can also be generalized to penalties that depend on functions of the form

$$m^*(\boldsymbol{x}) = \sum_{i\in N} x_i + \sum_{i\in\bar{N}} \bar{x}_i, \tag{37}$$

where N and $\bar{N}$ are two distinct subsets of the system qubits and $\bar{x}_i$ is the inverse value of x_i. One needs only to include a π rotation (σ_x) for each qubit belonging to the set $\bar{N}$ before (after) the sum gate (sum† gate).

D. Summary of scalings

In this subsection, we summarize the resource requirements and scaling behavior of the slack-variable method and the direct qudit-based penalty implementation.

Using slack variables, each constraint $P_r(x)$ introduces an extra qudit with local dimension determined by the number of feasible values N_r^{feas}. For N register qubits, the total Hilbert space is thus enlarged to

$$\dim(\mathcal{H}_{\text{slack}}) = 2^N \prod_{r=1}^{R} N_r^{\text{feas}}. \tag{38}$$

The implementation requires R slack qudits with dimension N_r^{feas} or, alternatively, $\sum_r \log_2 N_r^{\text{feas}}$ qubits if a qubit encoding of the slack variables is used. For a constraint on the total magnetization of N qubits, we have $N^{\text{feas}} \in \{0, ..., N\}$ (representing the range from 0 to N qubits being in state $|1\rangle$). In the benchmarks below, we assume an implementation of the slack variables with qudits. This represents the most favorable situation for the slack-variable approach as the typical overhead when encoding integers in binary qubit variables is avoided (see, e.g., Ref. [40]).

Moreover, the implementation of the slack constraint in the cost Hamiltonian as given in Eq. (18) requires additional gates in the variational circuit. Assuming the constraint functions are linear in the register qubit operators σ_i^z, these are R single-qudit terms $\sim\hat{S}_r$, at most NR single-qudit terms $\sim\hat{P}_r$, and up to NR qubit-qudit interaction terms $\sim\hat{P}_r\hat{S}_r$ (depending on how many qubits contribute to a given constraint r). In case the slack variables are implemented via qubits, the terms involving $\hat{S}_r$ need to be further decomposed, potentially leading to higher-order interactions.

In contrast, the direct implementation of the penalty function for magnetization constraints as proposed in the previous section requires $O(N)$ qudit-sum gates and one diagonal single-qudit rotation. The qudit ancilla needs to be of dimension $N + 1$, where N is the number of qubits involved in the constraint. While the slack variables, being part of the cost

Hamiltonian, need to be present simultaneously, the direct implementation gives the design freedom to either implement all penalty functions in parallel (using one ancilla each) or sequentially by reusing a single ancilla. Moreover, unlike the slack-variable approach, the dimension of the search Hilbert space is not changed [excluding the ancilla(s), which does not enter the optimization procedure], thus avoiding the additional multiplicative factor from auxiliary qudit states.

E. Alternative approaches for inequality constraints from the literature

For completeness, we also mention other options that have been proposed in the literature. For example, one can start from a constrained state and perform the algorithm using only constraint-preserving evolution operators [41–43]. The drawback of this method is that usually such constraint-preserving operators are difficult to construct and need to be designed for each specific problem. Postselection of feasible solutions [44] can also be used, but depending on the problem this may be very inefficient as finding feasible solutions at all can be difficult, especially for problems where several disconnected domains of feasible solutions exist. Inequalities can also be enforced by midcircuit projection onto the feasible subspace through measurements of the constraint operators [45]. However, these projection operators are problem specific and can be challenging to implement, while the necessary measurements contribute to both algorithmic and qubit overhead.

Another recent approach utilizes the augmented Lagrangian formulation of constrained optimization problems [37], which also introduces energy penalties into the optimization cost function. However, this approach is only effective for problems where the constraints are active for the optimal solution, i.e., when the optimal solutions lie right at the boundary between feasible and unfeasible solutions such that the constraints become effective equality constraints.

IV. NUMERICAL BENCHMARKS

In this section, we show numerical results for the QAOA performed in different scenarios where inequality constraints play a role. First, we analyze the ability of the QAOA to obtain feasible solutions for a generic random spin model [46]. Then we show how to construct an initial constrained state for warm-starting the QAOA procedure. Finally, we test the performance on the industry-relevant problem of EV charging [47], which is subject to multiple constraints. For all of these problems, we compare the performance of the constraint-handling methods described in Secs. III A and III B.

We simulate the quantum part of the QAOA procedure using exact state vector simulations. To update the parameters defining the trial wave functions, we use the Powell classical optimizer from the scipy library [48] with default options, which means the algorithm runs until the predefined convergence criteria are met. For each problem instance, we perform $N_{\text{runs}} = 50$ executions of the QAOA as described in Sec. II, each time with initial parameters randomly drawn from a uniform distribution in the interval $[0, 2\pi]$. We will refer to a single execution of the QAOA for a given setup and initial set of parameters as a run. During the iterations of the optimization loop, the cost function is computed as the expectation

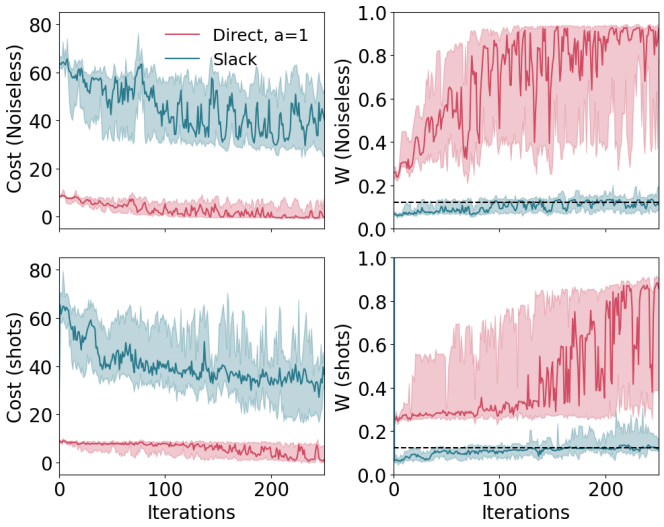

FIG. 3. Convergence of the optimized cost $\langle H_c \rangle$ (left) and feasible weight W (right) over optimization steps for a three-layer QAOA with $m_0 = -1.5$ and $N = 9$ qubits. Top row: state-vector evaluation of the cost function. Bottom row: noisy cost estimation using a finite number of $N_{\text{shots}} = 128$ samples in each iteration. Red curves correspond to our constraint implementation with $a = 1$; blue curves use slack variables. Results are obtained from $N_{\text{runs}} = 20$ optimizations for one specific cost Hamiltonian. Thick lines indicate the median, and shaded regions denote the 25th–75th percentiles. Dashed black line indicates the baseline probability for feasible states, i.e., the fraction of feasible over total number of states.

value of the cost Hamiltonian over the parametrized state. For each run, we extract the classical solutions of the optimization problem by sampling $N_S = 64$ solutions (measurement shots) from the final optimized quantum state $|\psi(\alpha^*, \beta^*)\rangle$.

We also investigated using a finite number of shots for estimating the cost function expectation value during the optimization iterations. The results of the optimization are qualitatively the same as using the exact expectation values (see, e.g., Fig. 3 below). But since the simulation run times are substantially longer, we used the exact expectation values for our numerical experiments.

A. Metrics

We use various figures of merit to estimate the performance of the QAOA protocols under different constraint-handling techniques. As the first metric, we consider the approximation ratio, which is defined as

$$R = \min_{\text{samples } s} \frac{E_s - E_0}{|E_0|}, \tag{39}$$

where E_s with $s = 1, .., N_S$ is the energy of a state sampled from the final state and E_0 is the energy corresponding to the optimal (constrained) solution. This metric is relevant for studying the efficiency of the QAOA in solving industrial optimization problems, since in such situations the user cares about the best energy of a single configuration that could be achieved, and not the average over the final state.

A second relevant figure of merit is the success rate, defined as

$$r = \frac{\sum_{i=1}^{N_{\text{runs}}} X_i}{N_{\text{runs}}}, \tag{40}$$

where N_{runs} is the total number of runs and $X_i = 1$ if the algorithm sampled at least one of the (possibly degenerate) optimal states in the N_S samples of the ith run, and $X_i = 0$ otherwise.

A third figure of merit is the total weight of feasible solutions present in the final state,

$$W = \sum_{\boldsymbol{x}\in\text{feasible}} |c_{\boldsymbol{x}}(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)|^2, \tag{41}$$

where the $c_{\boldsymbol{x}}(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ are the amplitudes in the final QAOA state,

$$|\psi(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\rangle = \sum_{\boldsymbol{x}} c_{\boldsymbol{x}}(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\,|\boldsymbol{x}\rangle. \tag{42}$$

This metric serves as an indicator of the efficiency of the algorithm at sampling feasible solutions.

B. Random spin Hamiltonian

We start by studying the capabilities of the QAOA in finding the ground state for a model of randomly interacting spin $\frac{1}{2}$ in a random longitudinal field. We compare the qudit slack variables approach explained in Sec. III A to penalties discussed in Sec. III B. The cost Hamiltonian is

$$H_C = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z, \tag{43}$$

with independent Gaussian-distributed parameters with zero mean and unit variance, i.e., $J_{ij}, h_i \sim \mathcal{N}(0, 1)$.

For this Hamiltonian, we consider a single constraint affecting all spins of the form

$$P(\boldsymbol{\sigma}_z) = \sum_{i=1}^{N} \sigma_i^z - m_0 \equiv S^z_{\text{tot}} - m_0 \leqslant 0, \tag{44}$$

which filters out all the states with total spin $S^z_{\text{tot}} = \sum_{i=1}^{N} \sigma_z^i$ less than a target $m_0 \in \{-\frac{N}{2}, \frac{N}{2}+1, \ldots, \frac{N}{2}\}$. For the penalties described in Sec. III B, we choose the exponent a to be equal to 0, 1, or 2, resulting in the penalty Hamiltonians

$$H_{a=0} = \lambda\,\Theta\big(S^z_{\text{tot}} - m_0\big), \tag{45a}$$

$$H_{a=1} = \lambda\,\Theta\big(S^z_{\text{tot}} - m_0\big)\,\big(S^z_{\text{tot}} - m_0\big), \tag{45b}$$

$$H_{a=2} = \lambda\,\Theta\big(S^z_{\text{tot}} - m_0\big)\,\big(S^z_{\text{tot}} - m_0\big)^2. \tag{45c}$$

When using the penalty term for the slack variables as described in Sec. III A, the complete cost Hamiltonian takes the form

$$H_{\text{C,slack}} = H_C + \lambda\,\big(S^z_{\text{tot}} - m_0 + \hat{S}\big)^2, \tag{46}$$

where the slack variable operator $\hat{S}$ acts on a qudit with dimension $d = \frac{N}{2} + 1 + m_0$.

We start by considering a single instance of the Hamiltonian of Eq. (43) for a system of $N = 9$ qubits and compare the convergence of the optimizer for different setups. Figure 3 shows the convergence of the cost function (left column) and the weight of the feasible states (right column) over optimization iterations. We compare the results from using the exact state-vector evaluation of the cost function during the iterations (upper row) with estimating the cost from a finite number of samples employing $N_{\text{shots}} = 128$ shots in each iteration (lower row). Even with a low number of sampling shots, the optimizer converges in the same range of optimization iterations for both state-vector and finite sampling evaluation methods. We found the same behavior for all problems studied in this work, and therefore, limit ourselves to results from state-vector evaluations for the remaining simulations.

Figure 3 also compares the results from our proposed method of direct incorporation of penalties (red graphs) to the standard slack variable-based approach (petrol graphs). Both approaches converge within the shown iterations, but the slack-variable–based approach produces cost function values that are an order of magnitude larger than our proposed approach, indicating a failure to produce good low-energy solutions. Similarly, the weights of the feasible states for our proposed approach converge to above 90% whereas the slack-variable–based approach converges at around only 10%. These results provide a first indication that our proposed approach produces much more feasible solutions that also have lower cost values than the standard method.

We investigate this behavior further by considering a system of $N = 9$ qubits, for which we study the metrics defined in Sec. IV A for a $p = 1$ layer QAOA as a function of the value of the constrained magnetization m_0. We generate 20 random realizations of the cost Hamiltonian and run $N_{\text{runs}} = 50$ simulations for each Hamiltonian. All the results shown are obtained for a penalty factor of $\lambda = 4$, but higher values for λ show similar results.

We investigate the metrics given in Eqs. (39)–(41), obtained by running the different algorithms on the same problems, as the dimension of the feasible subspace is increased. In Figs. 4 and 5, we compare the results obtained with the proposed method to the approach using slack variables. We see that for all the metrics considered, the performance of the proposed approach is considerably better than that of the implementation with slack variables. For example, the success rate r for the penalties of Eqs. (45a)–(45c) is always significantly higher than for the slack-variable–based approach. Especially when m_0 approaches large values, the proposed methods converge to $r \gtrsim 0.5$ while for slack variables it converges toward low values around $r \approx 0.1$. Similarly, the approximation ratio R for slack variables has only small fluctuations around 0.3 for values of m_0 greater than -2.5, while for each instance of the approach based on direct penalization it quickly approaches 0 as m_0 increases and the system becomes less constrained.

Moreover, the performances of the flat, linear, and quadratic penalties can differ considerably. When considering r as figure of merit, the linear penalty shows the best behavior. Considering R for highly constrained problems ($m_0 \lesssim -1.5$) the best result is obtained with a flat penalty ($a = 0$). However, this effect is a direct result of the penalty terms in the Hamiltonian. Namely, for these problems the sampled low-energy states will also include unfeasible states, where the actual penalty term contributes to the energy, and

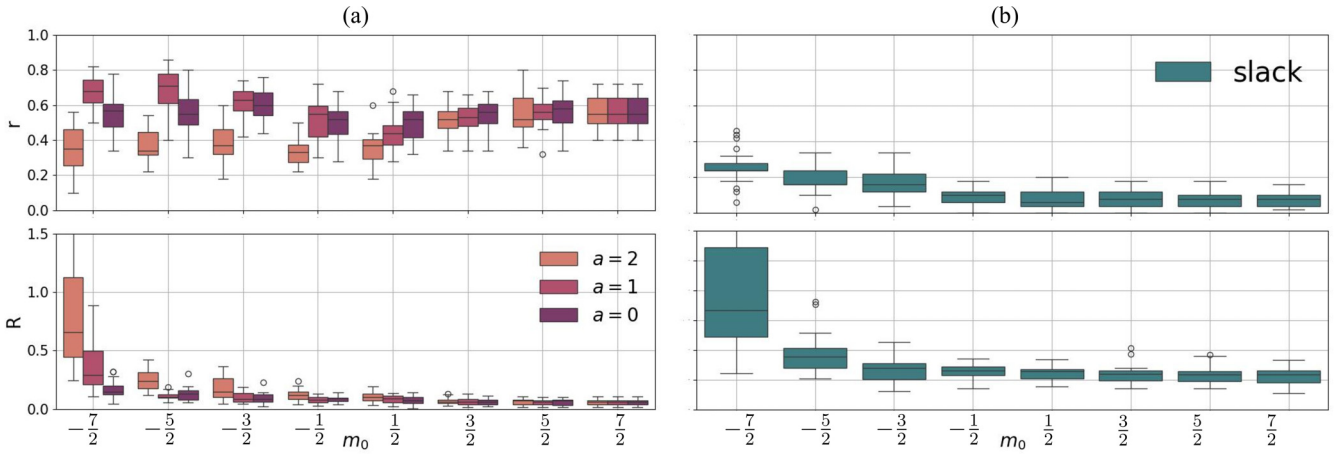

FIG. 4. Success rate r (top row) and approximation ratio R (bottom row) for constraint QAOA with a single layer ($p = 1$) and $N = 9$ qubits. The box plots show statistics for averages over 20 different instances of the random Hamiltonian of Eq. (43) and 50 runs per individual Hamiltonian. The results are shown as a function of the maximally allowed total spin, m_0. (a) Results from the proposed approach directly implementing the penalty functions of Eqs. (45a)–(45c). The linear form ($a = 1$) consistently performs best when considering the success rate r, while the flat penalty shows the best approximation ratio R. The results for different a become more similar as the value of m_0 increases, since the system becomes increasingly less constrained, reducing the impact of the different forms of penalties. (b) Results from runs using slack variables for handling the inequality constraints. The results clearly show the superiority of the direct penalty approaches of (a) over using slack variables as those achieve much higher success rates and lower approximation ratios.

consequently linear and quadratic terms will give larger energies in general.

Figure 5 shows the total weight of the feasible states as a function of the constraint target value m_0, for the three forms of the penalty given in Eqs. (45a)–(45c) (left panel) as well as for the slack-variable approach (right panel). The continuous lines show the total weight of the feasible states for a uniformly distributed quantum state as a baseline, which is simply given by the fraction of the number of feasible states over all states (including qudit–slack-variable states in the case of slack-variable approaches), i.e., Base $= \dim(\mathcal{H}_{\text{feasible}})/\dim(\mathcal{H}_{\text{tot}})$. For the proposed approach, the baseline increases monotonically from $1/2^N$ for $m_0 = -\frac{N}{2}$ toward 1 for the unconstrained problem at $m_0 = \frac{N}{2}$. In contrast, for the slack-variable approach it does not approach unity when increasing m_0; instead, it has a maximum around an intermediate value of $m_0 \approx 1.5$ and slightly decreases for larger m_0. This behavior is due to the changing dimension of the slack variable with m_0. On the one hand, the absolute number of feasible states for a given m_0 is the same as for the case without slack variables since only one out of all possible values for the slack variable represents a feasible solution. On the other hand, the dimension of the slack variable increases with increasing m_0 and thus in the case of slack variables the baseline acquires an additional factor $\frac{1}{d} = 1/(m_0 + 1 + \frac{N}{2})$.

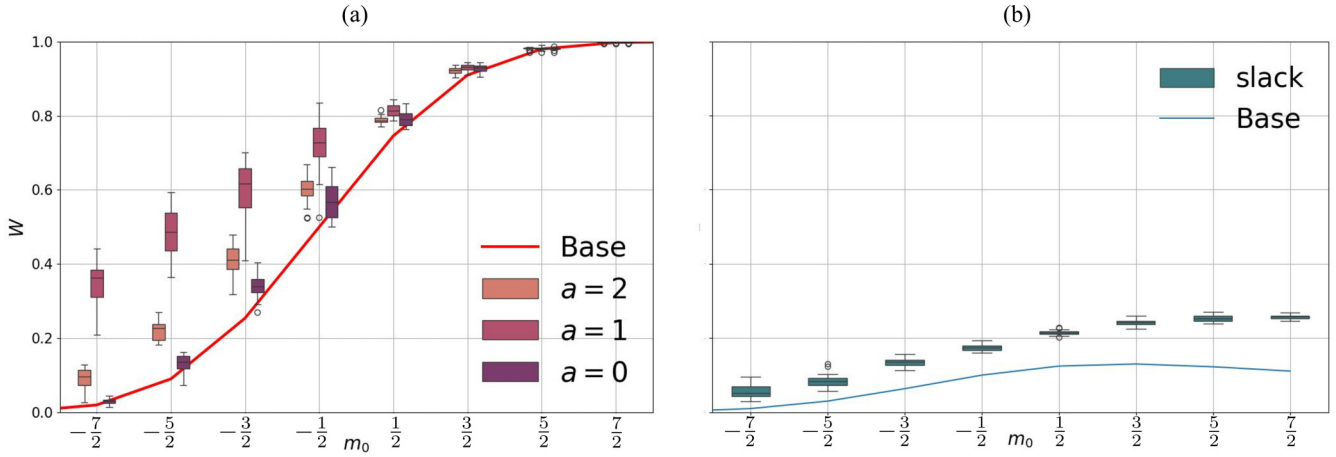

FIG. 5. Total weight W of all feasible states of a single-layer QAOA with constrained cost function defined in Eqs. (45a)–(45c) (left) and slack variables of Eq. (46) (right) for the same setup as Fig. 4. The solid lines show the baseline probability of finding a feasible state from an equal superposition of all basis states. The slack-variable approach has a consistently lower probability of sampling a feasible configuration from the final state, even if almost all qubit configurations are feasible for larger m_0 where it saturates around 25% while this probability reaches 100% for the proposed direct penalty method. This trend reflects what we see in Fig. 4 and serves as an indicator of the poor performance of slack variables for inequality-constrained optimization.

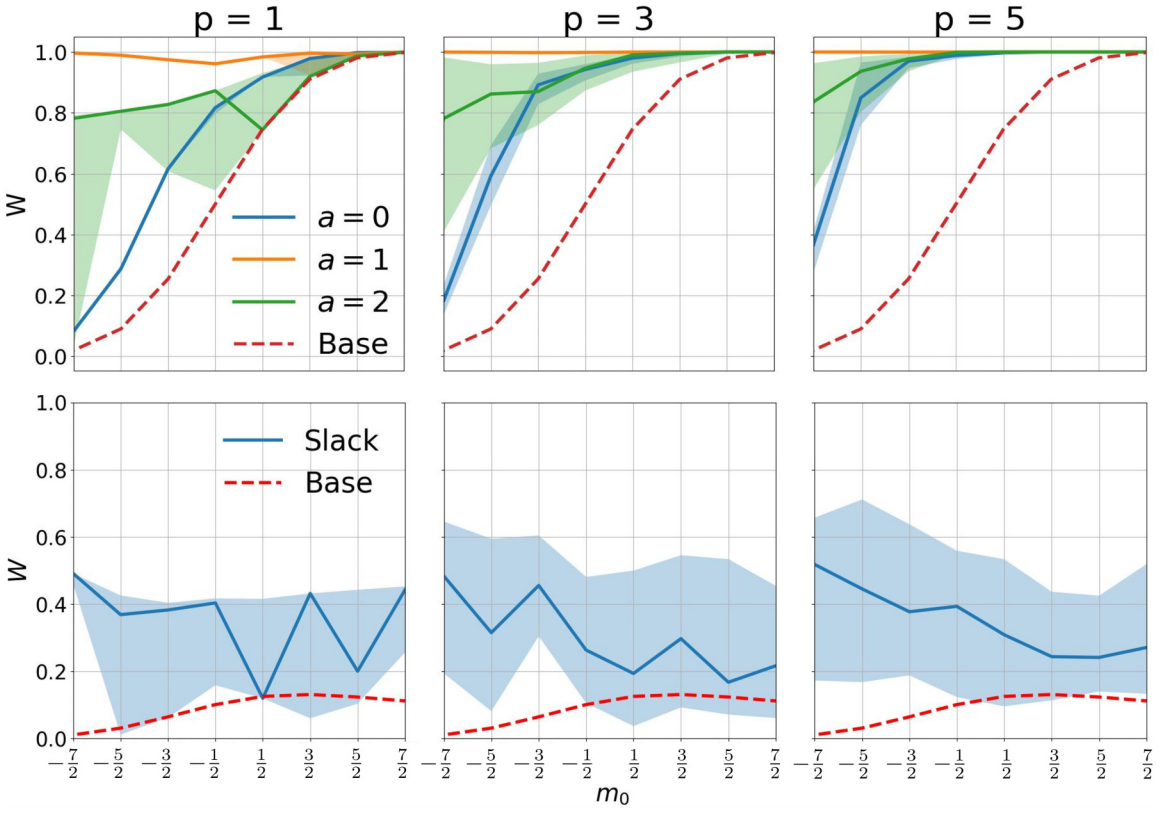

FIG. 6. Probability W of sampling a feasible state from the final state of a QAOA with constraint-only cost function, for the direct penalty approach (top row) and slack-variable–based approach (bottom), for $p = 1$, 3, and 5 QAOA layers (left to right panels). The solid lines show median values for the different penalty types and the shaded areas indicate the 20%–80% quantiles of the 50 runs performed. The dashed red lines marked as "Base" indicate the baseline probabilities from the equal superposition state of all solutions. The direct penalty functions outperform the slack-variable approach significantly. In particular, the penalty function with $a = 1$ already gives close to 100% probability for all constraint values for only the $p = 1$ layer.

The results of Fig. 5 show that all the tested methods increase the probability of sampling a feasible state over the baseline. However, similarly to the trend of Fig. 4, the proposed direct penalty methods considerably outperform the slack-variable approach. This effect is most pronounced for the linear penalty function and with increasing m_0, i.e., for less constrained problems.

C. Constrained state preparation and sampling of feasible states

In certain scenarios, the goal of the algorithm may be to sample from a subset of states that fulfill the constraints, without any preference on the states within that subset. Examples include topological models and lattice gauge theories (LGTs) [49], spin ice [50], and the sampling of polymer melts [51,52]. Another scenario is when trying to construct a superposition of many or all feasible states as is required for the initialization of approaches utilizing constraint-preserving mixing operators [41–43].

Motivated by this, we consider a QAOA where the cost function is given only by the penalties encoding the constraint of the form of Eq. (44). The QAOA cost Hamiltonian is then of the form of Eqs. (45a)–(45c) for the proposed approach and of the form of Eq. (46) for the slack-variable approach. We run the QAOA for up to $p = 5$ layers, and we do $N_{\text{runs}} = 50$ for each setup (penalty type, number of layers, and constraint value m_0).

Figure 6 shows the total weight of feasible configurations in the final quantum state for numbers of layers $p = 1$, 3, and 5. The linear penalty term considerably outperforms the slack-variable approach as well as the other penalty forms considered, already reaching close to the maximum of $W = 1$ after the first layer for almost all values of the minimum magnetization m_0. As in the previous example, all approaches improve as expected over the baseline, given by the weight of the feasible states on the initial state (dashed line). Again, for all chosen forms of the penalty function, the proposed direct penalty approach improves significantly more over the baseline than the slack-variable–based approach. As would be expected, the direct approach also achieves overall better performances with increasing number of layers p. Interestingly such a trend is not observed for the slack-variable–based approaches.

D. EV charging problem

To make contact with the QAOA for solving combinatorics problems of industrial relevance and test the different approaches with multiple constraints, we evaluate the performance of the proposed approach for an EV charging problem.

It is schematically represented in Figs. 1(a) and 1(b). The target of this problem is to find the charging schedule for a fleet of EVs, i.e., the charging power for each EV at each time step. This problem is naturally formulated in terms of qudit variables [23] as soon as more than two charging levels (charging or not charging) are considered. Importantly in our context, this problem naturally has many inequality constraints since each EV has minimum requirements for the total energy delivered to it while at each time step the total charging power must not exceed the fuse limits. It is thus a useful benchmark problem to study how the different approaches perform with the inclusion of multiple constraints.

The cost function of the considered EV charging problem is defined as

$$C(\boldsymbol{x}) = \sum_{t=1}^{T} c_t \sum_{n=1}^{N_{\mathrm{EV}}} x_{n,t}, \tag{47}$$

where the variables $x_{n,t} \in \{x_{\min}, \ldots, x_{\max}\}$ characterize the amount of energy charged (or discharged if $x_{\min} < 0$) to EV $n \in \{1, \ldots, N_{\mathrm{EV}}\}$ at time step $t \in \{1, \ldots, T\}$. The (time-dependent) cost for a unit of electric energy is given by the coefficients c_t. This problem is naturally subject to many constraints. In particular, we consider the following two types:

$$E_n^{\mathrm{required}} - \sum_{t=1}^{T} x_{n,t} \leqslant 0 \quad \forall n, \tag{48}$$

$$\sum_{n=1}^{N_{\mathrm{EV}}} x_{n,t} - E^{\max} \leqslant 0 \quad \forall t. \tag{49}$$

The first one, Eq. (48), reflects the requirement of each EV to obtain a minimal amount of electricity E_n^{required} after the charging is finished. The second constraint, Eq. (49), ensures that the maximal charging energy never exceeds the fuse limits. These are $N_{\mathrm{EV}} + T$ linear constraints in total, where each couples only a specific subset of variables, but taken together they couple all variables with each other. This problem is illustrated in Fig. 1.

For demonstration purposes, we consider a rather simple problem instance where we only include two charging power levels, $x_{n,t} \in \{0, 1\}$ and two vehicles ($N_{\mathrm{EV}} = 2$) for four time steps ($T = 4$). The constraints are specified by $E^{\max} = 1$ and $E_0^{\mathrm{required}} = E_1^{\mathrm{required}} = 2$. With this setup, the classical combinations that satisfy the constraint are easy to define: each vehicle must be charged at least in two time steps, but the vehicles cannot be charged at the same time.

The quantum formulation is obtained by the usual replacement of the search variables with spin operators as shown in Eq. (3). The Hilbert space dimension for this simple example is $\dim(\mathcal{H}) = 2^{TN_{\mathrm{EV}}} = 2^8 = 256$, out of which only 12 basis states are feasible. Due to the symmetry with respect to the EVs, the solutions are always doubly degenerate. To incorporate the constraints of Eqs. (48) and (49) using slack variables, we need a total of six auxiliary variables: two qudits for the constraints of Eq. (48) with dimension $d = 3$ and four qubits ($d = 2$) for those of Eq. (49). The Hilbert space dimension for the constrained problem is then $\dim\mathcal{H}_{\mathrm{slack}} = 2^8 \times 3^2 \times 2^4 = 36864$, which is 2 orders of magnitude larger than the Hilbert space of the setup without slack variables.

We study the performance of the QAOA as a function of the number of layers and for different random realizations of the Hamiltonian of Eq. (47), where the prices are chosen in each instance from a uniform distribution $c_t \sim \mathcal{U}(0, 1)$. We run $N_{\mathrm{runs}} = 50$ runs for each problem instance. For the approach using direct penalties, we consider $p = 1, \ldots, 5$ layers, while we did only run simulations for up to $p = 3$ layers for the slack-variable approach due to its vastly larger Hilbert-space dimension and the consequently significantly longer run times. The results, summarized in Fig. 7, enhance the understanding of the previous benchmarks. The approach based on direct penalty Hamiltonians already achieves success rates on the order of 30% for $p = 1$, a value that increases up to 60%–90% for $p = 5$ layers. The approximation ratio also rapidly approaches zero with increasing p, especially for the best-performing case of $a = 1$.

In stark contrast, the results of the slack-variable approach essentially indicate a failure of the method. Although the success rate does increase slightly for a larger number of layers, the overall scale is only around 10^{-3}. Similarly, the approximation ratio is an order of magnitude worse than for the direct penalty approach and essentially indicates that the algorithms never come close to finding the true ground-state energy region.

The reason for this failure is found in the vast increase in Hilbert-space dimension needed to encode all the different constraints. For each slack variable, only one of all possible configurations represents a feasible state, which leads to the majority of added states being unfeasible. In the above example where there are typically only two feasible configurations for the charging variables $x_{n,t}$, this leads to a reduction of the fraction of feasible solutions from $2/2^8 \approx 4 \times 10^{-3}$ without slack variables to $2/(2^8 \times 3^2 \times 2^4) \approx 5 \times 10^{-5}$ with slack variables. This large increase in search space due to the slack variables makes the corresponding search problem for the QAOA much more difficult.

V. CONCLUSION AND OUTLOOK

We have presented an approach to efficiently include energy penalties for many inequality constraints in the QAOA routine. In contrast to standard slack-variable–based approaches, it does not increase the Hilbert space dimension of the search problem regardless of the number of constraints to be included. It only requires one additional ancilla qudit, which does not enter the cost function. We have benchmarked the approach on three different problems, and compared the results to the standard approach utilizing slack variables. The proposed approach, which includes constraints directly without relying on additional qudits in the cost Hamiltonian, vastly outperforms the slack-variable–based method. Moreover, we find that a linear energy penalty outperforms constant and quadratic penalty terms.

While including a single constraint through slack variables can lead to decent results, the performance drops drastically when including multiple constraints. However, the presence of a multitude of constraints is the typical case for a large class of combinatorial problems suited for the QAOA (e.g., resource allocation problems [13]). Including many slack variables in the problem would be unfeasible for NISQ architectures.

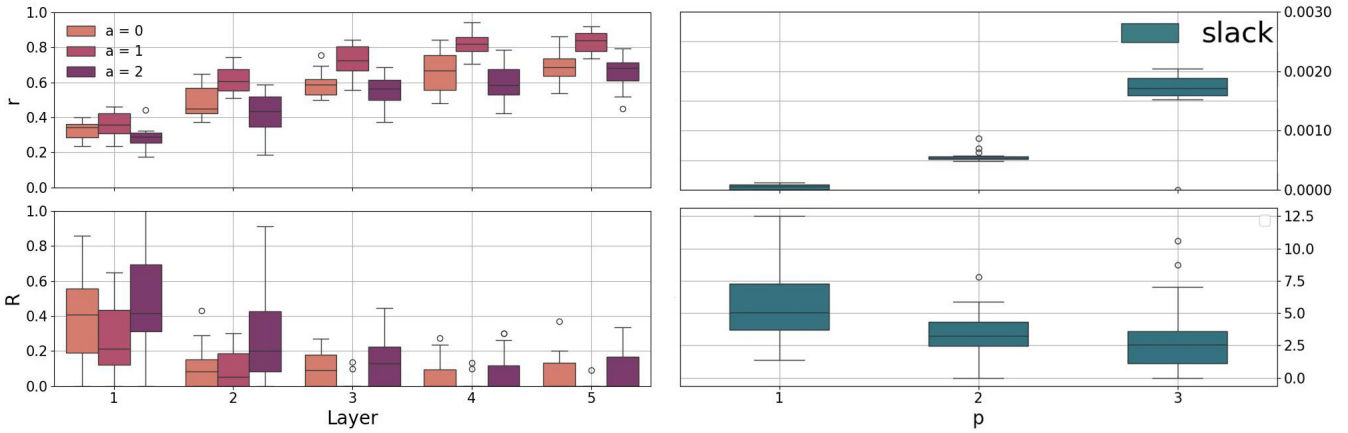

FIG. 7. Results of the EV charging problem of Eqs. (47)–(49) for $N_{\mathrm{EV}} = 2$, $T = 4$, $E^{\max} = 1$, and $E_n^{\mathrm{required}} = 2$, for penalties encoded directly (left) and with slack variables (right) as a function of the number of QAOA layers p. The approach based on direct penalty terms vastly outperforms the one based on slack variables (note the different scale of the y axes). In line with the previous findings, the case with linear penalty $a = 1$ performs best among the direct penalty approaches. In particular, for $p \geqslant 3$ the median as well as the 20th and the 80th quantiles of the distribution of the approximation ratio R all vanish, indicating that an optimal solution is always found for $a = 1$.

There, it leads to a drastic increase in required quantum resources that is hard to meet and in turn amplifies the noise in the circuit execution. In a fault-tolerant scheme where pure resources and noise are expected to not be the dominating limiting factors, it still considerably increases the complexity of the circuit, which is known to be a possible cause for the barren plateaus phenomenon [53]. Additionally, as shown in this work, the problem of finding low-energy states, i.e., the optimization problem to determine the parameters of the parametrized quantum circuit, is much more difficult to solve due to the vastly increased search space.

In the future, an interesting direction will be to combine the use of qudits for constraint handling and representation of the cost-function register, in order to maximize the use of available qudit levels in a given machine. Moreover, while qudit encodings of cost functions have been proposed, the potential performance advantages in variational quantum algorithms with respect to the usual qubit formulations need further thorough analysis, in particular also in view of the additional engineering overhead required. Our results may also stimulate cross-fertilization with other fields of quantum technologies. For example, including constraints in a quantum algorithm is of fundamental relevance also for the task of quantum simulation of lattice gauge theories [54,55], as these theories are characterized by an extensive set of physical constraints that needs to be preserved (e.g., Gauss' law in quantum electrodynamics) [56–58].

Further along the road, the proposed approach of including the energy penalties directly by using only a very small number of ancilla qudits, and thus avoiding the incorporation of slack variables into the energy functions, may be an enabling step for approaching realistic industry-scale and fundamental science problems with large numbers of inequality constraints.

ACKNOWLEDGMENTS

We acknowledge fruitful discussions with Gopal Chandra Santra, Linus Ekstrøm, and Mikel Garcia de Andoin. A.B. acknowledges funding from the Honda Research Institute Europe. S.S. and P.H. acknowledge funding by the European Union under Horizon Europe Programme, Grant Agreement 101080086–NeQST. This project has received funding from the Italian Ministry of University and Research (MUR) through the FARE grant for the project DAVNE (Grant No. R20PEX7Y3A), and was supported by the Provincia Autonoma di Trento, and Q@TN, the joint laboratory between University of Trento, FBK—Fondazione Bruno Kessler, INFN—National Institute for Nuclear Physics, and CNR—National Research Council. This project was funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.4–Call for tender No. 1031 of 17/06/2022 of Italian Ministry for University and Research funded by the European Union–NextGenerationEU (Project No. CN_00000013). This work received funds from Project DYNAMITE QUANTERA2_00056 funded by the Ministry of University and Research through the ERANET COFUND QuantERA II–2021 call and co-funded by the European Union (H2020, GA No. 101017733).

The views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Commission, the European Union, or of the Ministry of University and Research. Neither the European Union nor the granting authority can be held responsible for them.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

[1] A. Abbas, A. Ambainis, B. Augustino *et al.*, Challenges and opportunities in quantum optimization, Nat. Rev. Phys. **6**, 718 (2024).

[2] B. C. B. Symons, D. Galvin, E. Sahin, V. Alexandrov, and S. Mensa, A practitioner's guide to quantum algorithms for optimisation problems, J. Phys. A Math. Theor. **56**, 453001 (2023).

[3] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, The unconstrained binary quadratic programming problem: A survey, J. Comb. Optim. **28**, 58 (2014).

[4] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 5 (2014).

[5] 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).

[6] S. Yarkoni, E. Raponi, T. Bäck, and S. Schmitt, Quantum annealing for industry applications: Introduction and review, Rep. Prog. Phys. **85**, 104001 (2022).

[7] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[8] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[9] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: A review of methods and best practices, Phys. Rep. **986**, 1 (2022).

[10] D. A. Fedorov, B. Peng, N. Govind, and Y. Alexeev, VQE method: A short survey and recent developments, Mater. Theory **6**, 2 (2022).

[11] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[12] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, A review on quantum approximate optimization algorithm and its variants, Phys. Rep. **1068**, 1 (2024).

[13] S. Limmer, J. Varga, and G. R. Raidl, Large neighborhood search for electric vehicle fleet scheduling, Energies **16**, 4576 (2023).

[14] M. S. Blok, V. V. Ramasesh, T. Schuster, K. O'Brien, J. M. Kreikebaum, D. Dahlen, A. Morvan, B. Yoshida, N. Y. Yao, and I. Siddiqi, Quantum information scrambling on a superconducting qutrit processor, Phys. Rev. X **11**, 021010 (2021).

[15] 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).

[16] V. Kasper, D. Gonzalez-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).

[17] Y. Chi, J. Huang, Z. Zhang, J. Mao, Z. Zhou, X. Chen, C. Zhai, J. Bao, T. Dai, H. Yuan *et al.*, A programmable qudit-based quantum processor, Nat. Commun. **13**, 1166 (2022).

[18] D. González-Cuadra, T. V. Zache, J. Carrasco, B. Kraus, and P. Zoller, Hardware efficient quantum simulation of non-Abelian gauge theories with qudits on Rydberg platforms, Phys. Rev. Lett. **129**, 160501 (2022).

[19] P. Hrmo, B. Wilhelm, L. Gerster, M. W. van Mourik, M. Huber, R. Blatt, P. Schindler, T. Monz, and M. Ringbauer, Native qudit entanglement in a trapped ion quantum processor, Nat. Commun. **14**, 2242 (2023).

[20] X. Gao, P. Appel, N. Friis, M. Ringbauer, and M. Huber, On the role of entanglement in qudit-based circuit compression, Quantum **7**, 1141 (2023).

[21] L. E. Fischer, A. Chiesa, F. Tacchino, D. J. Egger, S. Carretta, and I. Tavernelli, Universal qudit gate synthesis for transmons, PRX Quantum **4**, 030327 (2023).

[22] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Hybrid quantum-classical algorithms for approximate graph coloring, Quantum **6**, 678 (2022).

[23] Y. Deller, S. Schmitt, M. Lewenstein, S. Lenk, M. Federer, F. Jendrzejewski, P. Hauke, and V. Kasper, Quantum approximate optimization algorithm for qudit systems, Phys. Rev. A **107**, 062410 (2023).

[24] G. Bottrill, M. Pandey, and O. Di Matteo, Exploring the potential of qutrits for quantum optimization of graph coloring, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, WA, USA* (IEEE, New York, 2023), pp. 177–183.

[25] M. Karácsony, L. Oroszlány, and Z. Zimborás, Efficient qudit based scheme for photonic quantum computing, SciPost Phys. Core **7**, 032 (2024).

[26] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, arXiv:1602.07674.

[27] 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).

[28] N. L. Wach, M. S. Rudolph, F. Jendrzejewski, and S. Schmitt, Data re-uploading with a single qudit, Quantum Mach. Intell. **5**, 36 (2023).

[29] S. Roca-Jerat, J. Román-Roche, and D. Zueco, Qudit machine learning, Mach. Learn.: Sci. Technol. **5**, 015057 (2024).

[30] D. H. Useche, A. Giraldo-Carvajal, H. M. Zuluaga-Bucheli, J. A. Jaramillo-Villegas, and F. A. González, Quantum measurement classification with qudits, Quantum Inf. Process. **21**, 12 (2022).

[31] M. G. De Andoin, A. Bottarelli, S. Schmitt, I. Oregi, P. Hauke, and M. Sanz, Formulation of the electric vehicle charging and routing problem for a hybrid quantum-classical search space reduction heuristic, in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain* (IEEE, 2023), pp. 5318–5323.

[32] V. Vargas-Calderón, N. Parra-A, H. Vinck-Posada, and F. A. González, Many-qudit representation for the travelling salesman problem optimisation, J. Phys. Soc. Jpn. **90**, 114002 (2021).

[33] A. E. Smith and D. W. Coit, Constraint-handling techniques—penalty functions, in *Evolutionary Computation 2, Advanced Algorithms and Operators*, edited by T. Baeck, D. B. Fogel, and Z. Michalewicz (Institute of Physics, London, 2000), Chap. 7, pp. 41–48.

[34] C. A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A

survey of the state of the art, Comput. Methods Appl. Mech. Eng. **191**, 1245 (2002).
[35] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering (Springer, New York, 2006).
[36] K. Kuroiwa and Y. O. Nakagawa, Penalty methods for a variational quantum eigensolver, Phys. Rev. Res. **3**, 013197 (2021).
[37] H. N. Djidjev, Quantum annealing with inequality constraints: The set cover problem, Adv. Quantum Technol. **6**, 2300104 (2023).
[38] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, Efficient quantum circuits for diagonal unitaries without ancillas, New J. Phys. **16**, 033040 (2014).
[39] S. Hadfield, On the representation of Boolean and real functions as Hamiltonians for quantum computing, ACM Trans. Quantum Comput. **2**, 1 (2021).
[40] N. Chancellor, Domain wall encoding of discrete variables for quantum annealing and QAOA, Quantum Sci. Technol. **4**, 045004 (2019).
[41] F. G. Fuchs, K. O. Lye, H. Møll Nilsen, A. J. Stasik, and G. Sartor, Constraint preserving mixers for the quantum approximate optimization algorithm, Algorithms **15**, 202 (2022).
[42] 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).
[43] A. Bärtschi and S. Eidenbenz, Grover mixers for qaoa: Shifting complexity from mixer design to state preparation, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA* (IEEE, 2020), pp. 72–82.
[44] P. Díez-Valle, J. Luis-Hita, S. Hernández-Santana, F. Martínez-García, Á. Díaz-Fernández, E. Andrés, J. J. García-Ripoll, E. Sánchez-Martínez, and D. Porras, Multiobjective variational quantum optimization for constrained problems: An application to cash handling, Quantum Sci. Technol. **8**, 045009 (2023).
[45] D. Herman, R. Shaydulin, Y. Sun, S. Chakrabarti, S. Hu, P. Minssen, A. Rattew, R. Yalovetzky, and M. Pistoia, Constrained optimization via quantum zeno dynamics, Commun. Phys. **6**, 219 (2023).
[46] N. Sauerwein, F. Orsi, P. Uhrich, S. Bandyopadhyay, F. Mattiotti, T. Cantat-Moltrecht, G. Pupillo, P. Hauke, and J.-P. Brantut, Engineering random spin models with atoms in a high-finesse cavity, Nat. Phys. **19**, 1128 (2023).
[47] O. Sassi and A. Oulamara, Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches, Int. J. Prod. Res. **55**, 519 (2017).
[48] P. Virtanen *et al.* and SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in python, Nat. Methods **17**, 261 (2020).
[49] A. Y. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. **303**, 2 (2003).
[50] *Spin Ice*, edited by M. Udagawa and L. Jaubert (Springer, New York, 2021).
[51] C. Micheletti, P. Hauke, and P. Faccioli, Polymer physics by quantum computing, Phys. Rev. Lett. **127**, 080501 (2021).
[52] F. Slongo, P. Hauke, P. Faccioli, and C. Micheletti, Quantum-inspired encoding enhances stochastic sampling of soft matter systems, Sci. Adv. **9**, eadi0204 (2023).
[53] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, Barren plateaus in variational quantum computing, Nat. Rev. Phys. **7**, 174 (2025).
[54] M. C. Banuls, R. Blatt, J. Catani, A. Celi, J. I. Cirac, M. Dalmonte, L. Fallani, K. Jansen, M. Lewenstein, S. Montangero *et al.*, Simulating lattice gauge theories within quantum technologies, Eur. Phys. J. D **74**, 165 (2020).
[55] J. C. Halimeh, M. Aidelsburger, F. Grusdt, P. Hauke, and B. Yang, Cold-atom quantum simulators of gauge theories, Nat. Phys. **21**, 25 (2025).
[56] J. C. Halimeh, H. Lang, J. Mildenberger, Z. Jiang, and P. Hauke, Gauge-symmetry protection using single-body terms, PRX Quantum **2**, 040311 (2021).
[57] J. C. Halimeh and P. Hauke, Stabilizing gauge theories in quantum simulators: A brief review, arXiv:2204.13709.
[58] A. Rajput, A. Roggero, and N. Wiebe, Quantum error correction with gauge symmetries, npj Quantum Inf. **9**, 41 (2023).