# Optimal Flight-Gate Assignment on a Digital Quantum Computer

Yahui Chai,[1,*] Lena Funcke,[2,3] Tobias Hartung,[4] Karl Jansen,[1] Stefan Kühn,[1,5] Paolo Stornati,[6] and Tobias Stollenwerk[7]

[1] *Deutsches Elektronen-Synchrotron (DESY), Platanenallee 6, Zeuthen 15738, Germany*

[2] *Transdisciplinary Research Area "Building Blocks of Matter and Fundamental Interactions" (TRA Matter) and Helmholtz Institute for Radiation and Nuclear Physics (HISKP), University of Bonn, Nußallee 14–16, Bonn 53115, Germany*

[3] *Center for Theoretical Physics, Co-Design Center for Quantum Advantage, and National Science Foundation (NSF) AI Institute for Artificial Intelligence and Fundamental Interactions, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*

[4] *Northeastern University—London, Devon House, St Katharine Docks, London E1W 1LP, United Kingdom*

[5] *Computation-Based Science and Technology Research Center, The Cyprus Institute, 20 Kavafi Street, Nicosia 2121, Cyprus*

[6] *ICFO—Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, Avinguda Carl Friedrich Gauss 3, Castelldefels (Barcelona) 08860, Spain*

[7] *Institute for Quantum Computing Analytics (PGI-12), Forschungszentrum Jülich, Wilhelm-Johnen-Straße, Jülich 52428, Germany*

(Received 27 March 2023; revised 4 October 2023; accepted 8 November 2023; published 13 December 2023)

We investigate the performance of the variational quantum eigensolver (VQE) for the problem of optimal flight-gate assignment. This is a combinatorial-optimization problem that aims at finding an optimal assignment of flights to the gates of an airport, in order to minimize the passenger travel time. To study the problem, we adopt a qubit-efficient binary encoding with a cyclic mapping, which is suitable for a digital quantum computer. Using this encoding in conjunction with the conditional value at risk (CVaR) as an aggregation function, we systematically explore the performance of the approach by classically simulating the CVaR VQE. Our results indicate that the method allows for finding a good solution with high probability and that it significantly outperforms the naive VQE approach. We examine the role of entanglement for the performance and find that ansätze with entangling gates allow for better results than pure product states. Studying the problem for various sizes, our numerical data show that the scaling of the number of cost-function calls for obtaining a good solution is not exponential for the regimes that we investigate in this work.

DOI: 10.1103/PhysRevApplied.20.064025

## I. INTRODUCTION

In recent years, variational quantum algorithms (VQAs) [1–3] have become increasingly relevant due to substantial progress in quantum hardware development. Such algorithms typically do not require deep quantum circuits that could only be faithfully executed on fully error-corrected quantum computers. Instead, they are amenable to noisy intermediate-scale quantum (NISQ) devices (for various proof-of-principle demonstrations, see, e.g., Refs. [1,4–10]). While such algorithms are typically heuristics without proven performance guarantees, there are indications that VQAs can outperform classical algorithms for certain computationally hard problems. Besides various applications for quantum simulations, the solutions of combinatorial-optimization problems are further candidates for widespread applications that can be tackled with VQAs [11].

In order to assess the potential of VQA approaches for real-world applications, it can be useful to investigate applications beyond purely academic problems and to focus on certain industrial use cases, as they typically exhibit additional complexity. One such example is the flight-gate assignment (FGA) problem [12,13]. The FGA problem is a quadratic assignment problem [14] with additional constraints, as typical for real-world applications. Previous works have mainly investigated the solution of the FGA problem [13] and related problems [15–17] with quantum annealers. Here, the constraints are incorporated into an unconstrained cost function by penalty terms.

*yahui.chai@desy.de

 

These approaches have a number of disadvantages, one of which is the typically exponentially small subspace of valid solutions in the entire Hilbert space [17]. One method for mitigating this issue is to constrain the algorithm to only search the feasible subspace. This idea was originally proposed for quantum annealing [18] and was later adapted to variational algorithms [10,19]. The applicability of the latter approach for FGA has been investigated by deriving suitable algorithmic primitives for constraint invariance [20]. In Ref. [10], a proof-of-principle VQE has been implemented for the FGA problem using an encoding incorporating some of the constraints on IBM's quantum hardware, thus demonstrating the suitability of the problem for digital quantum devices.

In this paper, we systematically assess the performance of VQE for the FGA problem, by numerically studying its performance using the conditional value at risk (CVaR) [8] as an aggregation function. We adopt an encoding that avoids a dominant subspace of invalid solutions, which is similar to the one of Ref. [10], with the addition of a cyclic mapping. Our study demonstrates that utilizing this encoding, the CVaR VQE performs significantly better than the naive encoding used in previous works. From classically simulating the CVaR VQE for various problem sizes up to 18 qubits, our results indicate that the number of cost-function calls to obtain a reasonably large contribution of the optimal solution in the final state does not scale exponentially with the problem size. Furthermore, we examine the role of entangling gates in the ansatz. Our results demonstrate that ansätze creating entanglement between qubits show a significantly better performance than circuits preparing only product states.

The paper is organized as follows. In Sec. II, we first introduce the FGA problem, before discussing the one-hot encoding and the binary encoding of the problem. Subsequently, we discuss the CVaR-VQE method and the types of ansätze that we use in our simulations in Sec. III. Section IV shows our numerical results for classically simulating the CVaR VQE for various problem sizes and a comparison between entangling ansätze and ansätze that only produce product states. Finally, we conclude in Sec. IV.

## II. THE FLIGHT-GATE ASSIGNMENT PROBLEM AND ITS ENCODING INTO QUANTUM STATES

In this section, we first introduce the FGA problem and then proceed with discussing two ways of encoding the problem into quantum states: the one-hot encoding, which does not incorporate any of the constraints, and a binary encoding that integrates some of the constraints.

### A. The flight-gate assignment problem

The FGA problem aims at minimizing the total transit time of passengers in an airport by finding an optimal gate assignment of the flights. Although there are multiple scenarios for optimizing the gate assignment of flights at an airport, we choose the one where we seek to minimize the total transfer time of passengers at the airport [12]. In this scenario, we have three kinds of passengers in an airport: arriving passengers, departing passengers, and transfer passengers. The arriving passengers land at the airport with an inbound flight and need to walk from the arrival gate to the baggage claim before leaving the airport. Departing passengers enter the airport through the security checkpoint and leave with an outbound flight. Transfer passengers arrive at the airport with an inbound flight, have to walk to the gate of their connecting flight, and leave with an outbound flight. To model the problem mathematically given a set of flights $F$ and a set of gates $G$, we consider a set of binary decision variables $x_{i\alpha}$ that represent whether or not a flight $i$ is assigned to a gate $\alpha$:

$$x_{i\alpha} = \begin{cases} 1, & \text{if flight } i \in F \text{ is assigned to gate } \alpha \in G, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Throughout the paper, we refer to gates with Greek indices, to flights with Latin indices, and $x = (x_{i\alpha}) \in \{0,1\}^{|F|\times|G|}$ is a binary vector collecting all of the $|F| \times |G|$ decision variables. The total passenger travel time can then be expressed as a function of $x$ and is given by

$$T(x) = T^{\text{arr}}(x) + T^{\text{dep}}(x) + T^{\text{trans}}(x), \tag{2}$$

where the three parts arise from the contributions of the different types of passengers. The time $T^{\text{arr-dep}}$ represents the total transit time of arriving or departing passengers and is given by the partial sums

$$T^{\text{arr-dep}}(x) = \sum_{i\alpha} n_i^{\text{arr-dep}} t_\alpha^{\text{arr-dep}} x_{i\alpha}, \tag{3}$$

where $n_i^{\text{arr-dep}}$ is the number of passengers arriving or departing with flight $i$ and $t_\alpha^{\text{arr-dep}}$ is the time that it takes to walk from or to gate $\alpha$. The total time $T^{\text{trans}}$ of the transfer passengers is given by the sum of the times $t_{\alpha\beta}$ that it takes to go from gate $\alpha$ to gate $\beta$ for each of the $n_{ij}^{\text{trans}}$ passengers who transfer from flight $i$ to flight $j$ (or vice versa), given that flight $i$ is assigned to gate $\alpha$ and flight $j$ is assigned to gate $\beta$:

$$T^{\text{trans}}(x) = \sum_{i,j,\alpha,\beta} n_{ij}^{\text{trans}} t_{\alpha\beta} x_{i\alpha} x_{j\beta}. \tag{4}$$

Note that $T^{\text{trans}}(x)$ contains a term that is quadratic in the decision variables. Thus, minimizing the total time in Eq. (2) is an instance of a quadratic assignment problem, a category that is, in general, NP hard [21].

In addition, there are two constraints in the FGA problem. First, each flight can only be assigned to one gate, so

there can only be a single nonzero decision variable among those belonging to the same flight. This constraint can be enforced by imposing

$$\forall i \in F \quad \sum_{\alpha} x_{i\alpha} = 1. \tag{5}$$

Second, there can be at most a single flight at a gate at the same time, because flights departing at the same time from the airport cannot be assigned to the same gate. This can be expressed as

$$\forall \alpha \in G \text{ and } \forall (i,j) \in P \quad x_{i\alpha} \times x_{j\alpha} = 0, \tag{6}$$

where $P$ is the set of forbidden flight pairs,

$$P = \{(i,j) \in F \times F : t_i^{\text{in}} < t_j^{\text{in}} < t_i^{\text{out}} + t^{\text{buf}}\}. \tag{7}$$

In the above expression, $t_i^{\text{in-out}}$ is the time of arrival or departure of flight $i$ and $t^{\text{buf}}$ is a buffer time between two flights at the same gate. In the following, we refer to an assignment of the decision variables fulfilling the two above constraints as a feasible assignment.

The encoding presented above requires $|G|$ decision variables $x_{i1} \cdots x_{i|G|}$ for each flight $i \in F$, which can be interpreted as a bit string. The constraint in Eq. (5) then implies that only a single entry in such a bit string can be nonzero. Hence, we call the encoding presented above the one-hot encoding. Since, for each flight, only $|G|$ assignments of the corresponding decision variables are compliant with the constraint in Eq. (5), the total number of feasible assignments is upper bounded by $|G|^{|F|}$.

### B. Hamiltonian formulation using the one-hot encoding

In order to treat the problem on a quantum computer, we have to formulate the problem as a (quantum) Hamiltonian. In order to minimize the objective function $T(x)$ subject to the constraints in Eqs. (5) and (6), we want to incorporate the constraints in the objective function. To this end, we translate them to positive semidefinite penalty terms the kernels of which corresponds to valid solutions fulfilling the constraints. These penalty terms can then simply be added to the objective function with a large positive constant in front, thus ensuring that the global minimum is the optimal solution fulfilling the constraints.

Equation (5) can be represented as a penalty term,

$$C^{\text{one}}(x) = \sum_i \left( \sum_{\alpha} x_{i\alpha} - 1 \right)^2, \tag{8}$$

while the second constraint in Eq. (6) can be formulated as

$$C^{\text{not}}(x) = \sum_{(i,j)\in P} \sum_{\alpha} x_{i\alpha} x_{j\alpha}. \tag{9}$$

Considering both the objective function and the penalty terms, the total cost function can be formulated as a quadratic unconstrained binary optimization (QUBO) problem:

$$\begin{aligned} Q(x) &= T(x) + \lambda^{\text{one}} C^{\text{one}}(x) + \lambda^{\text{not}} C^{\text{not}}(x) \\ &= c + \sum_{i\alpha} h_{i\alpha} \times x_{i\alpha} + \sum_{i\alpha j\beta} J_{i\alpha j\beta} \times x_{i\alpha} x_{j\beta}. \end{aligned} \tag{10}$$

In the above equation, $c$, $h_{i\alpha}$, and $J_{i\alpha j\beta}$ are the coefficients of the corresponding terms, which depend on $t_\alpha^{\text{arr-dep}}$, $t_{\alpha\beta}$, $n_i^{\text{arr-dep}}$, and $n_{ij}^{\text{trans}}$. The explicit formulas of these coefficients are shown in Eq. (A1) of Appendix A. The parameters $\lambda^{\text{one}}$ and $\lambda^{\text{not}}$ are constants that have to be chosen large enough to ensure that the solution of the above QUBO problem satisfies the constraints. For practical purposes, the values of these parameters might have to be set carefully to make the optimization procedure efficient [13].

In order to solve this problem using a quantum device, the QUBO problem has to be mapped to a Hamiltonian acting on qubits. This can be easily realized by replacing the binary decision variables $x_{i\alpha}$ in $Q(x)$ with the operators $(I - \hat{Z}_k)/2$, where $I$ is the identity and $\hat{Z}_k$ is the Pauli $Z$ matrix acting on the qubit that encodes the decision variable $x_{i\alpha}$. Substituting this transformation into the QUBO problem in Eq. (10), we obtain the (quantum) Hamiltonian

$$\hat{H} = c'\hat{I} + \sum_{p}^{N} h'_p \hat{Z}_p + \sum_{p<q}^{N} J'_{pq} \hat{Z}_p \hat{Z}_q, \tag{11}$$

where $N = |F| \times |G|$ and $c'$, $h'_p$, and $J'_{pq}$ are coefficients related to those of the original QUBO problem [for details, see Eq. (A3) in Appendix A]. The bit strings $x$ are now encoded by a computational basis state $|x\rangle$ and we call $|x\rangle$ a feasible state if $x$ represents a feasible assignment. The optimal solution of the FGA problem subject to the constraints corresponds to the ground state of the above Hamiltonian. By construction, the ground state will be a computational basis state, since $\hat{H}$ is diagonal in the $Z$ basis.

Note that in the encoding presented above, each decision variable is mapped to a single qubit. Hence, a total number of $|F| \times |G|$ qubits are required to address the problem on a quantum computer. However, only $|G|^{|F|}$ of the $2^{|G|\times|F|}$ basis states correspond to an assignment for which Eq. (8) is zero. Hence, the fraction of states in the Hilbert space fulfilling the first constraint, and correspondingly the number of feasible states, will decay exponentially with the problem size:

$$R_{\text{fea}}^{\text{one}} = \left( \frac{|G|}{2^{|G|}} \right)^{|F|}. \tag{12}$$

As a result, searching for the optimal solution will become increasingly challenging for increasing numbers of flights and gates.

### C. Hamiltonian formulation using a binary encoding

In order to avoid this exponential decay of the feasible subspace, we use a binary encoding for the FGA problem and derive the corresponding Hamiltonian, which is similar to the efficient embedding in Ref. [10]. In addition, we use a more efficient cyclic mapping as shown below.

As we have discussed in Sec. II A, there are $|G|$ assignments compliant with the first constraint in Eq. (5) for the decision variables corresponding to each flight. These assignments can be represented with $M = \lceil \log(|G|) \rceil$ (qu)bits using a binary encoding. Since $|G|$ is in general not a power of 2, we choose to map the elements in $G$ to the $2^M$ basis states $|\alpha'\rangle$ cyclically as

$$|\alpha'\rangle \leftrightarrow \text{gate } \alpha = \alpha' \bmod |G| \in G, \tag{13}$$

where $\alpha' = 0, \ldots, 2^M - 1$. In contrast, the previous work in Ref. [10] added a penalty term for the additional states $\{|\alpha'\rangle : |G| \leq \alpha' < 2^M\}$, in case $G$ was not a power of 2. However, this will lead to an exponential decay with $|F|$ for the fraction of feasible states, as these are given by $\left(G/2^M\right)^{|F|}$. The cyclic mapping used in this work can avoid this exponential decay of feasible states and will usually lead to many degenerate ground states (similar to the previously introduced concept of degeneracy engineering [22]), rendering it easier to find an optimal solution. All in all, for a total of $|F|$ flights, this encoding allows us to represent all possible assignments with $|F| \times M$ qubits, a lot less than that required for the one-hot encoding. Moreover, by construction, all solutions in this encoding automatically fulfill the constraint in Eq. (5).

In order to be able to solve the problem on a quantum computer using a VQA, we have to translate the Hamiltonian in Eq. (11) to this encoding. To this end, we define a set of projection operators $P_i(\alpha')$, $i = 0, \ldots, F-1$ given by

$$\begin{aligned}\hat{P}_i(\alpha') &= |\alpha'\rangle \langle\alpha'|_i = |z_0 \cdots z_{M-1}\rangle \langle z_0 \cdots z_{M-1}|_i \\ &= (|z_0\rangle \langle z_0| \otimes \cdots \otimes |z_{M-1}\rangle \langle z_{M-1}|)_i \,.\end{aligned} \tag{14}$$

In the above expression, $z_0 \cdots z_{M-1}$ is the bit string for the binary representation of $\alpha'$ and the index $i$ indicates the set of qubits related to flight $i$, on which the projection operators are acting. Applying $\hat{P}_i(\alpha')$ to one of the basis states encoding the solutions compliant with the first constraint for flight $i$ results in a 1, if and only if flight $i$ is assigned to gate $\alpha$, $\hat{P}_i(\alpha') |\beta'\rangle_i = \delta_{\alpha'\beta'}$. Using these projection operators, the Hamiltonian can be expressed as

$$\hat{H}(\hat{Z}) = \hat{H}^{\text{arr}} + \hat{H}^{\text{dep}} + \hat{H}^{\text{trans}} + \lambda^{\text{not}} \hat{H}^{\text{not}}, \tag{15}$$

where the individual terms are given by

$$\begin{aligned}\hat{H}^{\text{arr-dep}} &= \sum_i \sum_{\alpha'=0}^{2^M-1} n_i^{\text{arr-dep}} t_\alpha^{\text{arr-dep}} \hat{P}_i(\alpha'), \\ \hat{H}^{\text{trans}} &= \sum_{ij} \sum_{\alpha'\beta'=0}^{2^M-1} n_{ij}^{\text{trans}} t_{\alpha\beta}^{\text{trans}} \hat{P}_i(\alpha') \hat{P}_j(\beta'), \\ \hat{H}^{\text{not}} &= \sum_{(i,j)\in P} \sum_{\alpha'\beta'=0}^{2^M-1} \delta_{\alpha\beta} \hat{P}_i(\alpha') \hat{P}_j(\beta').\end{aligned} \tag{16}$$

In the above expression, $\alpha$ and $\beta$ refer to the gate indices after applying the mapping from Eq. (13). Note that we no longer have to impose the first constraint from Eq. (5) with a penalty term, as it is fulfilled by construction. Moreover, the Hamiltonian can be easily decomposed into Pauli operators using the relation

$$|z_k\rangle \langle z_k|_i = \left(\hat{I} + (-1)^{z_k} Z_{i\times M+k}\right)/2, \tag{17}$$

where we have chosen a linear ordering of the qubits.

The binary encoding with cyclic mapping still allows for unfeasible states, as the second constraint from Eq. (6) is not automatically fulfilled. Compared to the exponential decay observed for the one-hot encoding, the ratio of feasible solutions for the binary encoding is a lot larger and it decays only very slowly with the problem size, as shown in Fig. 1. In conjunction with its reduced qubit requirements, the binary encoding with cyclic mapping is significantly more amenable for NISQ devices, which provide only limited resources. The Hamiltonian corresponding to the binary encoding consists of $\mathcal{O}(|F|^2 \times |G|^2)$ Pauli $Z$ terms with order $2 \times \lceil \log(G) \rceil$ or less, meaning that each Pauli $Z$ term only acts nontrivially on at most $2 \times \lceil \log(G) \rceil$ qubits. Thus, the expectation value of the Hamiltonian can be evaluated efficiently on a quantum computer.

## III. VARIATIONAL QUANTUM EIGENSOLVER USING THE CONDITIONAL VALUE AT RISK

The VQE is a hybrid quantum classical algorithm for finding an approximation to the ground state of a given Hamiltonian $\hat{H}$ by minimizing $\langle\psi(\boldsymbol{\theta})|\hat{H}|\psi(\boldsymbol{\theta})\rangle$. Here, $|\psi(\boldsymbol{\theta})\rangle$ is a normalized ansatz state, which is parametrized by real numbers $\boldsymbol{\theta}$. To find an optimal set of parameters, the VQE utilizes a feedback loop between a quantum device and a classical computer. The former is used to realize a variational ansatz $|\psi(\boldsymbol{\theta})\rangle$ in the form of a parametric quantum circuit and to measure the expectation value of the

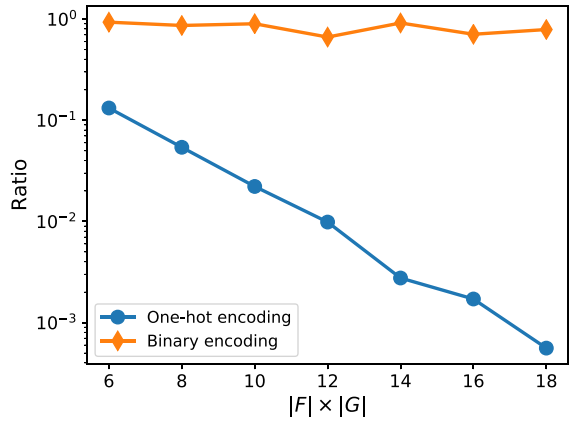


FIG. 1. The ratio of the feasible states in the Hilbert space, which are the states that fulfill both the first and the second constraint, as a function of the problem size $|F| \times |G|$ for the one-hot encoding (blue dots) and the binary encoding (orange diamonds). As a guide for the eye, the markers are connected with lines. The ratio of the feasible states for the one-hot encoding decays approximately exponentially with the problem size (see Eq. (12)), which is mainly caused by states violating the first constraint. For details on instance generation, see Sec. IV.

Hamiltonian. The classical computer is running a minimization algorithm suggesting a new set of parameters $\boldsymbol{\theta}'$ based on the measurement outcome of the quantum device. Running the feedback loop until convergence, the parametric circuit encodes an approximation of the ground state of the given Hamiltonian. Due to its modest quantum hardware requirements and its partial resilience to noise, the VQE is one of the most promising candidates for applications on NISQ devices. While the VQE was originally proposed for finding the ground state of a molecule [1], it can be readily applied to many other fields (see, e.g., Refs. [5,23–26]).

In particular, the VQE has been proposed to solve combinatorial-optimization problems [27–29]. Contrary to strongly correlated quantum many-body systems, for combinatorial-optimization problems, the problem Hamiltonian is diagonal and the possible solutions correspond to basis states. Since we are only interested in obtaining a good candidate for the solution of the combinatorial-optimization problem, the resulting state at the end of the VQE does not necessarily have to be dominated by the state encoding this solution. As long as it produces a state that has a reasonably large component of such a solution, the projective measurements at the end will reveal it, provided that enough measurements are taken. Due to this property, in Ref. [8] it has been argued that the CVaR is better suited as a cost function for combinatorial-optimization problems than the expectation value of the Hamiltonian. The CVaR for a random variable $X$ with the cumulative density function $F_X$ is defined as the conditional expectation over the left $\xi$ tail of the distribution,

$$\mathrm{CVaR}_\xi(X) = \mathbb{E}\left[X | X \leq F_X^{-1}(\xi)\right], \tag{18}$$

where $\xi \in (0, 1]$. This can be applied to VQE by considering only a subset of the samples obtained during the measurement process. Suppose that we perform $K$ measurements resulting in the bit strings $\{z_1, z_2, \ldots, z_K\}$ and the corresponding energy values $\{E_1, E_2, \ldots, E_K\}$. Assuming that the energy values are sorted in ascending order, the CVaR can be calculated as

$$\mathrm{CVaR}_\xi = \frac{1}{\lceil \xi K \rceil} \sum_{i=1}^{\lceil \xi K \rceil} E_i. \tag{19}$$

Note that for $\xi = 1$, the $\mathrm{CVaR}_\xi$ is simply the usual estimate for the expectation value with $K$ samples. In the opposite limit, $\xi \to 0$, the $\mathrm{CVaR}_\xi$ corresponds to selecting the measurement that produces the lowest energy. Moreover, the definition in Eq. (19) shows that the $\mathrm{CVaR}_\xi$ does essentially not reward increasing the fidelity of the VQE solution with the ground state beyond $\xi$, as we only consider the subset of the $\lceil \xi K \rceil$ measurements with the lowest energy.

In the following, we use VQE with the $\mathrm{CVaR}_\xi$ as a cost function to address the FGA problem. In particular, we explore the performance for various choices of $\xi$ as a function of the problem size.

## IV. SIMULATION RESULTS

In order to explore the performance of the VQE using the CVaR for the FGA problem, we perform classical simulations using the QISKIT [30] framework, assuming a perfect quantum device without shot noise, which means that we evaluate the cost function exactly. For our experiments, we use the EfficentSU2 ansatz from QISKIT consisting of parametric $R_Y(\theta) = \exp(-i\theta Y/2)$ rotation gates and linear entangling layers of controlled-NOT (CNOT) gates (for an illustration, see Fig. 2). The classical minimization is performed with constrained optimization by linear approximation (COBYLA) [31], the maximum number of function evaluations for which is set to 50 times the number of qubits, in order to avoid excessively long optimization times.

In the following, we examine three aspects. First, we investigate the performance of the VQE using the CVaR for various values of $\xi$ as a function of the problem size. Second, we explore the effect of entanglement on the optimization by using an ansatz that generates product states only and we compare the results to those obtained with the EfficentSU2 ansatz. Finally, we explore the scaling of the method with the problem size.

### A. Performance of the VQE using the CVaR

To investigate the performance of the VQE using the CVaR for the FGA problem, we use the CVaR VQE

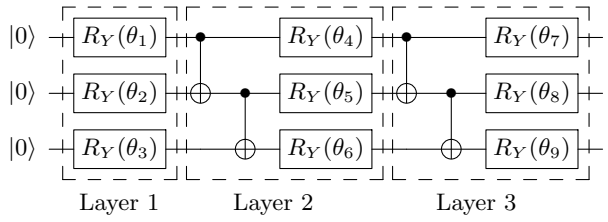


FIG. 2. The EfficientSU2 ansatz with linear CNOT entangling layers, shown for $l = 3$ layers and three qubits.

to explore various instances of the FGA problem for both the one-hot encoding and the binary encoding with up to 18 qubits. For the one-hot encoding, the number of qubits is equal to the problem size $|F| \times |G| \in \{6, 8, 10, 12, 14, 16, 18\}$. For the binary encoding, we can solve the FGA problem up to $|F| \times |G| = 34$, due its better resource efficiency. For each problem size, we randomly generate a set of nontrivial instances with multiple flights and gates and pick the ones that are difficult for the classical solver, meaning the ones that take the longest time to solve. We investigate 50 random instances of the FGA problem for each problem size and for each instance we run the VQE 5 times, using random choices for the initial parameters in the ansatz. Hence, in total, we explore 250 random instances for each problem size. Moreover, we study the dependence of the results on the choice of the parameter $\xi$ and the number of layers $l$ in the ansatz. To this end, we run simulations with $\xi \in \{0.01, 0.1, 0.25, 1\}$ and $l = 1, 2, 3$.

First, we monitor the fraction of instances that reach a certain fidelity threshold with the ground state as a function of the iteration number. The fidelity is defined as the probability of sampling a ground state from the quantum state $|\psi\rangle$ prepared by the VQE, $\sum_{z^*} |\langle z^*|\psi\rangle|^2$, where $\{|z^*\rangle\}$ is the set of the ground states that might be degenerate because of the cyclic mapping in Eq. (13). We choose low fidelity thresholds, 1% and 10%, to evaluate the performance, because even with a fidelity of 1%, the probability of obtaining the optimal solution at least once using 1000 shots is $1 - (1 - 0.01)^{1000}$, which is more than 99.99%. To be able to combine data for different problem sizes, we follow Ref. [8] and consider the normalized number of iterations corresponding to the number of cost-function evaluations divided by the number of qubits. Figure 3 shows the results for the binary encoding and fidelity thresholds 1% and 10% as a function of the normalized iterations. Comparing the different columns of Fig. 3, corresponding to different numbers of layers in the ansatz, we observe that in general, adding more layers yields better results. In particular, going beyond a single rotation layer, in which case the ansatz is able to produce entangled states, the fraction of instances that reach the threshold at the end of the simulation increases noticeably. Moreover, we observe that the CVaR VQE is able to generate a significant fraction of instances above the fidelity threshold within just a few normalized iterations. The latter indicates that

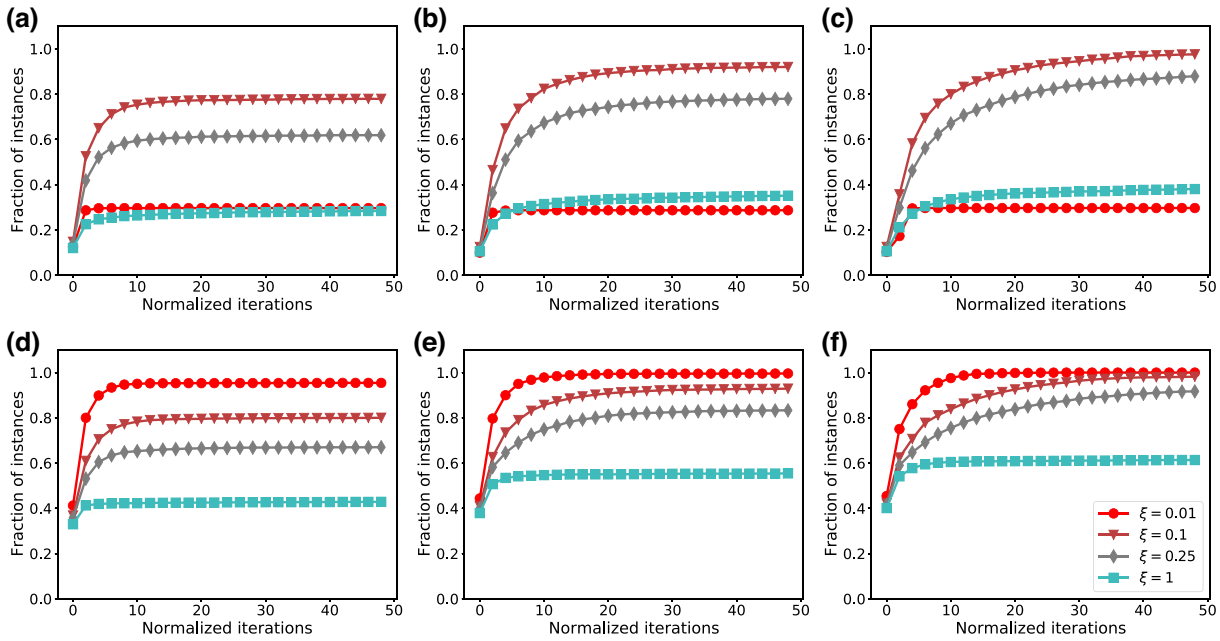


FIG. 3. The fraction of instances attaining a fidelity with the exact solution state of at least (a)–(c) 10% and (d)–(f) 1% as a function of the number of normalized iterations using the binary encoding. The different markers correspond to different choices of $\xi = 0.01$ (red dots), 0.1 (dark red triangles), 0.25 (gray diamonds), and 1 (cyan squares). The columns correspond to different numbers of layers: (a),(d) $l = 1$, (b),(e) $l = 2$, and (c),(f) $l = 3$.

even on NISQ hardware, where one might be restricted to a small number of iterations, we have a reasonable chance of finding a good solution.

Focusing on the results for a fidelity threshold of 1% in Figs. 3(d)–3(f), we see that decreasing $\xi$ improves the results. In particular, the conventional VQE using the expectation value of the Hamiltonian as a cost function shows the worst performance and reaches the 1% fidelity threshold for no more than 60% of all instances, even for three layers [see Fig. 3(f)]. Considering a larger fidelity threshold of 10%, shown in Figs. 3(a)–3(c), the observation is qualitatively similar, except for $\xi = 0.01$. The poor performance of $\xi = 0.01$ in that case can be explained by the nature of the CVaR cost function. As outlined in Sec. III, the CVaR cost function does not reward increasing the fidelity with the ground state beyond $\xi$. Thus, a choice of $\xi = 0.01$ does in general not allow for reliably reaching a fidelity with the ground state of 10%. Interestingly, the conventional VQE (corresponding to $\xi = 1$) only shows a slightly better performance than the CVaR VQE with $\xi = 0.01$. Increasing $\xi$ to the fidelity threshold, we again observe good performance and for three layers more than 95% of all instances reach a fidelity of at least 10% with the exact solution. Note that the CVaR VQE also has a better performance in optimizing the QUBO Hamiltonian using one-hot encoding and achieves a quite high success rate up to 18 variables (qubits) if we use $\xi = 0.1$ and three layers (for details, see Fig. 8 in Appendix B).

In our theoretical study, in which we evaluate the cost function exactly, we observe a higher success rate and faster convergence for a lower fidelity threshold and lower values of $\xi$. On quantum hardware, we have to consider that the measurement process involves taking a finite number of samples. Smaller values for $\xi$ imply discarding a larger fraction of samples and, thus, fewer statistics when estimating the cost function using sampling results. Hence, for simulations on quantum hardware, $\xi$ has to be chosen carefully. Reference [8] suggests a choice of $\xi$ in the range of $[0.1, 0.25]$ based on empirical results on quantum hardware. In addition, current NISQ devices suffer from a noticeable level of noise that might affect the results further. In this work, we focus on benchmarking the performance of the VQE on the FGA problem in an ideal setting and investigating the best choice of $\xi$ for NISQ hardware is beyond the scope of this paper. Hence, for simplicity, we focus on two scenarios, $\xi \in \{0.1, 1\}$ with a fidelity threshold of 0.1, as well as $\xi \in \{0.01, 1\}$ with a fidelity threshold of 0.01.

Using these scenarios, we compare the performance of both encodings utilizing the CVaR VQE in Fig. 4. For each problem size, we monitor the fraction of instances the maximal fidelity of which throughout the whole optimization process reaches the fidelity threshold. For a fidelity threshold of 10%, the binary encoding has a significantly better performance, especially in the case of $\xi = 1$, which corresponds to conventional VQE using the expectation value

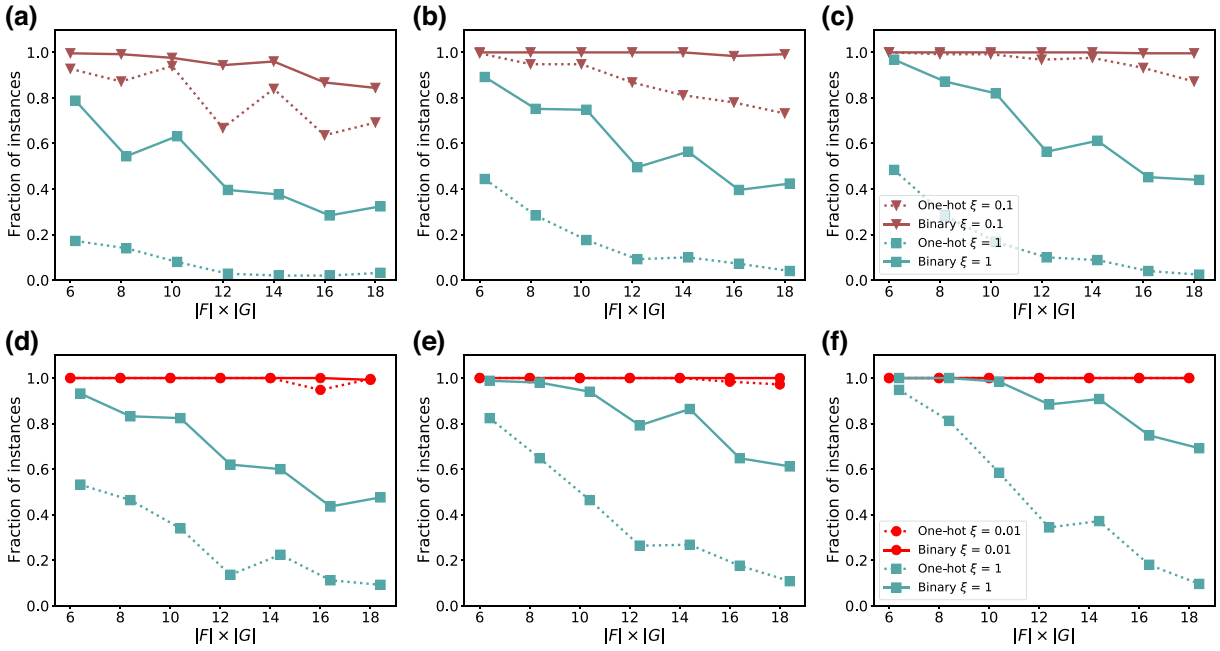


FIG. 4. The fraction of instances attaining a fidelity with the exact solution state of at least (a)–(c) 10% and (d)–(f) 1% at the end of the VQE for problems with up 18 variables, corresponding to up to 18 qubits for the one-hot encoding and up to nine qubits for the binary encoding. The different markers represent the different choices of $\xi = 0.01$ (red dots), 0.1 (brown triangles), and 1 (cyan squares). The columns correspond to different numbers of layers: (a),(d) $l = 1$, (b),(e) $l = 2$, and (c),(f) $l = 3$. The results for the binary encoding are connected with solid lines, while those for the one-hot encoding are connected with dashed lines.

as the cost function [see Figs. 4(a)–4(c)]. For our largest problem size, $|F| \times |G| = 18$, three layers of the ansatz, and $\xi = 1$, the one-hot encoding only reaches the fidelity threshold for a few percent of the instances, while for the binary encoding, around 50% of all instances still produce a fidelity of at least 10% [see Fig. 4(c)]. Decreasing the value of $\xi$ to 0.1, the drop in the fraction of instances reaching the fidelity threshold with increasing $|F| \times |G|$ is still more pronounced for the one-hot encoding. This reflects the hardness of the VQE using a hardware-efficient ansatz and a normal expectation value as the cost function: it is almost impossible to find the optimal solution of the FGA problem using conventional VQE if the number of qubits is larger than 18. Fortunately, we can use the CVaR as a cost function to overcome this problem, which shows a quite high success rate even for the largest problem size studied in this work. Moreover, while the one-hot encoding can reach the fidelity threshold of 1% with a similarly high success rate as the binary encoding if one chooses $\xi = 0.01$ [see Figs. 4(d)–4(f)], the average number of function evaluations to achieve the fidelity threshold of 1% is a lot less for the binary encoding compared to the one-hot encoding for the same problem size. As we will examine in detail in Sec. IV C, we observe that the average number of function evaluations for our largest problem size to reach the fidelity threshold of 1% with $\xi = 0.01$ is about $\mathcal{O}(10)$ for the binary encoding and $\mathcal{O}(10^2)$ for the one-hot encoding [see also Figs. 7(b) and 10(b)].

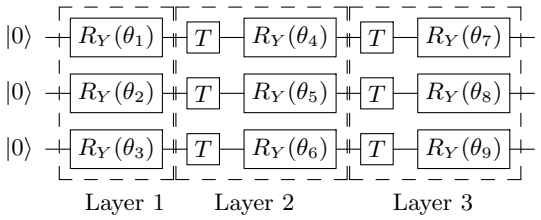


FIG. 5. The quantum circuit without entangling gates, which prepares a product state, illustrated for three qubits.

### B. Effect of entanglement on the performance

The simulation results for both the binary encoding and the one-hot encoding improve when using a larger number of layers in the ansatz, as Figs. 3 and 4 (and also Fig. 8 in Appendix B) reveal. However, it is not clear if the improvement of the performance is due to an increased number of entangling layers or merely because of the presence of more parameters in the ansatz. In order to investigate the role of entanglement in the VQE, we perform simulations using a quantum circuit without entangling gates by replacing the CNOT layers with single-qubit $T$

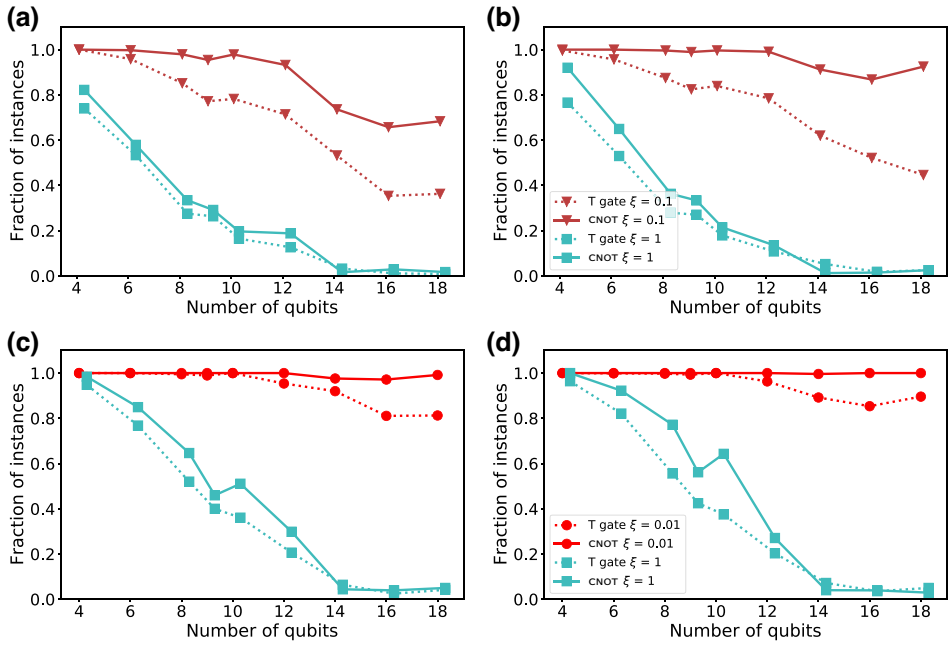


FIG. 6. A comparison of the performance of quantum circuits with and without entanglement, for solving the FGA problem using binary encoding. The circuit with entanglement is the EfficientSU2 circuit with CNOT layers (solid lines); in the circuit without entanglement, the CNOT layers are replaced with single-qubit $T$ gates (dotted lines). Similar to Fig. 4, the fraction of instances are shown that attain a fidelity with the exact solution state of at least (a),(b) 10% and (c),(d) 1% at the end of the VQE. The columns correspond to different numbers of layers: (a),(c) $l = 2$ and (b),(d) $l = 3$. The different markers represent different choices of $\xi = 0.01$ (red dots), 0.1 (brown triangles), and 1 (cyan squares).

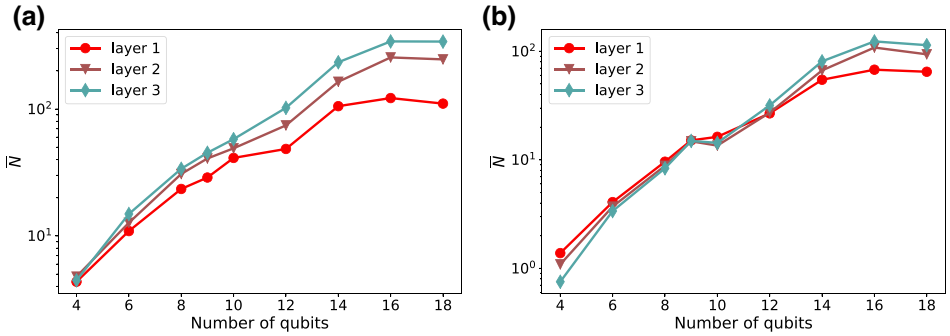


FIG. 7. The scaling of the average number of function evaluations $\overline{N}$ with the number of qubits for binary encoding. The simulation results are obtained using $l = 1$ (red dots), 2 (brown triangles), and 3 (cyan triangles) for the EfficientSU2 circuit. (a) Results corresponding to a fidelity threshold of 10% with $\xi = 0.1$. (b) Results corresponding to a fidelity threshold of 1% with $\xi = 0.01$. The $y$ axes are in logarithmic scale.

gates following Ref. [28]. The corresponding circuit is shown in Fig. 5.

Figure 6 shows the results for the performance of the CVaR VQE as a function of the problem size with and without entanglement, using the binary encoding for the FGA problem. For two and three layers in the ansatz, the VQE with entanglement shows a clear advantage for both $\xi = 0.1$ and $\xi = 0.01$, especially for larger problem sizes. In contrast, the advantage of the entangling circuit is not obvious for the VQE using the conventional expectation value, corresponding to $\xi = 1$. For the one-hot encoding, we observe a similar behavior; however, the difference between the two circuits is a lot smaller than that for the binary encoding (see Fig. 9 in Appendix B). Ideally, this comparison should be extended beyond using only product states, thus also including correlations between qubits that can be efficiently generated by a classical computer, as we aim to address in future work.

### C. Scaling of the number of cost-function evaluations

The results shown in the previous sections indicate that CVaR VQE in conjunction with the binary encoding is suitable to effectively solve the FGA problem. In the following, we examine the scaling of the number of cost-function calls with the problem size during the classical minimization, in order to benchmark the efficiency. To this end, we study the average number of cost-function evaluations of the successful instances that achieve the chosen fidelity threshold for the final state, which we refer to as $\overline{N}$ in the following. We adopt a best-case scenario and count the number of cost-function evaluations until the quantum state generated by the VQE achieves the desired fidelity threshold for the first time [32].

In Fig. 7, we show the scaling of $\overline{N}$ for the fidelity thresholds of 1% and 10%. For both cases, we observe a rather similar behavior. After an initially approximately exponential increase of the number of cost-function evaluations, the number eventually begins to saturate around a problem size of 14 qubits. While for a threshold of 10% with $\xi = 0.1$, the value of the final plateau in the number of cost-function calls shows a slight dependence on the number of layers, this dependence seems to be weaker for a threshold of 1% with $\xi = 0.01$. Moreover, a comparison between Figs. 7(a) and 7(b) shows that lowering the fidelity threshold and the value of $\xi$ by one order of magnitude only slightly reduces the number of cost-function calls required until convergence. Regarding the random instances that we have used here, some instances have degenerate ground states because of the cyclic mapping in Eq. (13). In case of a degeneracy, it can be easier to find a ground state; however, this might influence the scaling of the number of cost-function calls to reach a certain fidelity. In particular, we observe a slight "staircase shape" of the data in Fig. 7. In order to examine the effect of degeneracies carefully, we also explore the same scaling using a set of random instances with $|G|$ equal to a power of 2, such that there are no degenerate ground states due to the cyclic mapping in Eq. (13). For this case, we also do not observe an exponential increase of the number of cost-function calls up to the problem size of 18 qubits that we study (for details, see Appendix C). While a larger number of qubits is needed to obtain a solid conclusion for the scaling of $\overline{N}$, which is beyond the scope of this work, our results are promising, as they indicate that the FGA problem can be efficiently solved on digital quantum computers.

## V. CONCLUSIONS AND OUTLOOK

In this work, we have systematically explored the performance of the CVaR VQE for the FGA problem using a resource-efficient binary encoding in conjunction with a cyclic mapping, which is suitable for digital quantum computers. Compared to the one-hot encoding used in Refs. [13,20], the binary encoding with cyclic mapping requires a noticeably smaller number of qubits and allows for addressing the problem in a resource-efficient manner

on NISQ devices. In particular, the fraction of solutions in the Hilbert space that are compatible with the constraints is substantially larger than for the case of the one-hot encoding.

We have numerically benchmarked the performance of the CVaR VQE for the binary encoding and compared it to the previously used one-hot encoding and conventional VQE. We find that using the CVaR [8] as an aggregation function greatly improves the performance for both encodings compared to conventional VQE. In particular, using a fidelity threshold of at least 10% with the exact solution, the CVaR VQE is able to reach this threshold for more than 80% of all instances, provided that a large enough number of layers is chosen in the ansatz, compared to less than 40% of all instances using conventional VQE. Moreover, the binary encoding performs significantly better than the one-hot encoding. In particular, the CVaR-VQE approach to the binary encoding does not show a noticeable performance decrease with the problem size for the range of parameters we study, in contrast to the one-hot encoding.

Comparing the performance of the CVaR VQE for the binary encoding using the entangling EffiecientSU2 ansatz to a simple ansatz without entangling gates (and, thus, producing only product states), we find that the presence of entanglement significantly improves the results if using CVaR as a cost function. In particular, for larger problem sizes, our results indicate that entanglement is beneficial for efficiently exploring the energy landscape. Using a fidelity threshold of 10% and three layers, with $\mathrm{CVaR}_{0.1}$ as a cost function, we observe that the entangling ansatz is able to reach this threshold for roughly 90% of all instances for the largest problem size that we study, compared to roughly 40% of all instances for the product-state ansatz.

Focusing on the number of cost-function calls to obtain a certain fidelity threshold with the exact ground states, our data suggest that this number does not scale exponentially with the problem size for the range of parameters that we study. Consequently, the FGA problem seems to be efficiently addressable with digital quantum computers.

The encoding that we have used in this work can be readily implemented on gate-based quantum devices and the number of layers and cost-function calls in our study seem within reach on existing quantum hardware. In the future, we plan to investigate the performance of the binary encoding and the CVaR VQE in a realistic scenario with noise and eventually on a quantum device.

## ACKNOWLEDGMENTS

S.K. acknowledges financial support from the Cyprus Research and Innovation Foundation under projects "Future-Proofing Scientific Applications for the Supercomputers of Tomorrow (FAST)", Contract No. COMPLEMENTARY/0916/0048, and "Quantum Computing for Lattice Gauge Theories (QC4LGT)", Contract No. EXCELLENCE/0421/0019. L.F. is partially supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage ($C^2QA$) under Contract No. DE-SC0012704, by the DOE QuantiSED Consortium under Subcontract No. 675352, by the National Science Foundation under Cooperative Agreement PHY-2019786 [33], and by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under Grant Contracts No. DE-SC0011090 and No. DE-SC0021006. P.S. acknowledges support from the European Research Council (ERC) Advanced Grant (AdG) "NOvel Quantum simulators—connectIng Areas" (NOQIA) project; the Ministerio de Ciencia y Innovation Agencia Estatal de Investigaciones (PGC2018-097027-B-I00/10.13039/501100011033, CEX2019-000910-S/10.13039/501100011033, Plan National FIDEUA PID2019-106901GB-I00, FPI, QUANTERA MAQS PCI2019-111828-2, QUANTERA DYNAMITE PCI2022-132919, and Proyectos de I+D+I "Retos Colaboración" QUSPIN RTC2019-007196-7); MICIIN, with funding from the European Union (EU) NextGenerationEU project (Grant No. PRTR-C17.I1) and by the Generalitat de Catalunya; the Fundació Cellex; Fundació Mir-Puig; the Generalitat de Catalunya [through the European Social Fund FEDER and CERCA program, the Agencia de Gestión de Ayudas Universitarias y de Investigación (AGAUR) Grant No. 2021 SGR 01452, QuantumCAT U16-011424, cofunded by the European Regional Development Fund (ERDF) Operational Program of Catalonia 2014–2020]; the Barcelona Supercomputing Center MareNostrum (Grant No. FI-2022-1-0042); the EU Horizon 2020 FET-OPEN "Optical Topologic Logic" (OPTOlogic) program (Grant No. 899794); the EU Horizon Europe program (Grant Agreement No. 101080086—NeQST); the National Science Centre, Poland (Symfonia Grant No. 2016/20/W/ST4/00314); the ICFO Internal "QuantumGaudi" project; and the EU Horizon 2020 research and innovation program under the Marie-Skłodowska-Curie Grant Agreements No. 101029393 (STREDCH) and No. 847648 ("La Caixa" Junior Leaders fellowships ID100010434: LCF/BQ/PI19/11690013, LCF/BQ/PI20/11760031, LCF/BQ/PR20/11770012, and LCF/BQ/PR21/11840013). The views and opinions expressed in this work are, however, those of the author(s) only and do not necessarily reflect those of the EU, European Climate, Infrastructure and Environment Executive Agency (CINEA), or any other granting authority. Neither the EU nor any granting authority can be held responsible for them. This work is supported with funds from the Ministry of Science, Research and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA). This work is funded by the EU Horizon Europe framework program (HORIZON) under the ERA Chair scheme with Grant Agreement No. 101087126.

## APPENDIX A: DETAILS OF THE HAMILTONIAN FORMULATION USING ONE-HOT ENCODING

In Sec. II B, the FGA problem is formulated using the one-hot encoding with the cost function in form of a QUBO problem, as shown in Eq. (10). The corresponding Ising-type Hamiltonian is given in Eq. (11). In this appendix, we provide the explicit formulas of the coefficients for both formulations of the problem.

Considering the QUBO problem in Eq. (10), the coefficients $c$, $h_{i\alpha}$, and $J_{i\alpha j\beta}$ depend on the number of passengers, $n_i^{\text{arr-dep}}$ and $n_{ij}^{\text{trans}}$, and the different times, $t_\alpha^{\text{arr-dep}}$ and $t_{\alpha\beta}$, and read

$$\begin{aligned} c &= |F| \times \lambda^{\text{one}}, \\ h_{i\alpha} &= n_i^{\text{arr}} t_\alpha^{\text{arr}} + n_i^{\text{dep}} t_\alpha^{\text{dep}} - 2\lambda^{\text{one}}, \\ J_{i\alpha j\beta} &= n_{ij}^{\text{trans}} t_{\alpha\beta} + \delta_{ij}\lambda^{\text{one}} + \delta_{\alpha\beta}\delta_{ij}^{P}\lambda^{\text{one}}, \end{aligned} \tag{A1}$$

where $\delta_{ij}^P$ is nonzero if and only if the flights $i$ and $j$ are a forbidden pair of flights,

$$\delta_{ij}^P = \begin{cases} 1, & \text{if } (i,j) \in P, \\ 0, & \text{otherwise.} \end{cases} \tag{A2}$$

Given a QUBO problem as in Eq. (10), we can easily obtain the corresponding Ising Hamiltonian by replacing the binary variables $x_{i\alpha}$ in the QUBO with the operators $(I - \hat{Z}_{i\alpha})/2$,

$$\begin{aligned} H &= c + \sum_{i\alpha} h_{i\alpha} \frac{(I - \hat{Z}_{i\alpha})}{2} + \sum_{i\alpha j\beta} J_{i\alpha j\beta} \frac{(I - \hat{Z}_{i\alpha})(I - \hat{Z}_{j\beta})}{4} \\ &= c + \frac{1}{2}\sum_{i\alpha} h_{i\alpha} + \frac{1}{4}\sum_{i\alpha j\beta} J_{i\alpha j\beta} + \frac{1}{4}\sum_{i\alpha} J_{i\alpha i\alpha} \\ &\quad + \sum_{i\alpha}\left[-\frac{1}{2}h_{i\alpha} - \frac{1}{4}\sum_{j\beta}(J_{i\alpha j\beta} + J_{j\beta i\alpha})\right]\hat{Z}_{i\alpha} \\ &\quad + \frac{1}{4}\sum_{i\alpha \neq j\beta} J_{i\alpha j\beta}\hat{Z}_{i\alpha}\hat{Z}_{j\beta}. \end{aligned} \tag{A3}$$

The index $i\alpha$ can be mapped to the qubit index $p$ by a linear mapping, $p = i \times |G| + \alpha$, which allows for expressing the Hamiltonian as

$$\hat{H} = c'\hat{I} + \sum_{p}^{N} h'_p \hat{Z}_p + \sum_{p<q}^{N} J'_{pq}\hat{Z}_p\hat{Z}_q, \tag{A4}$$

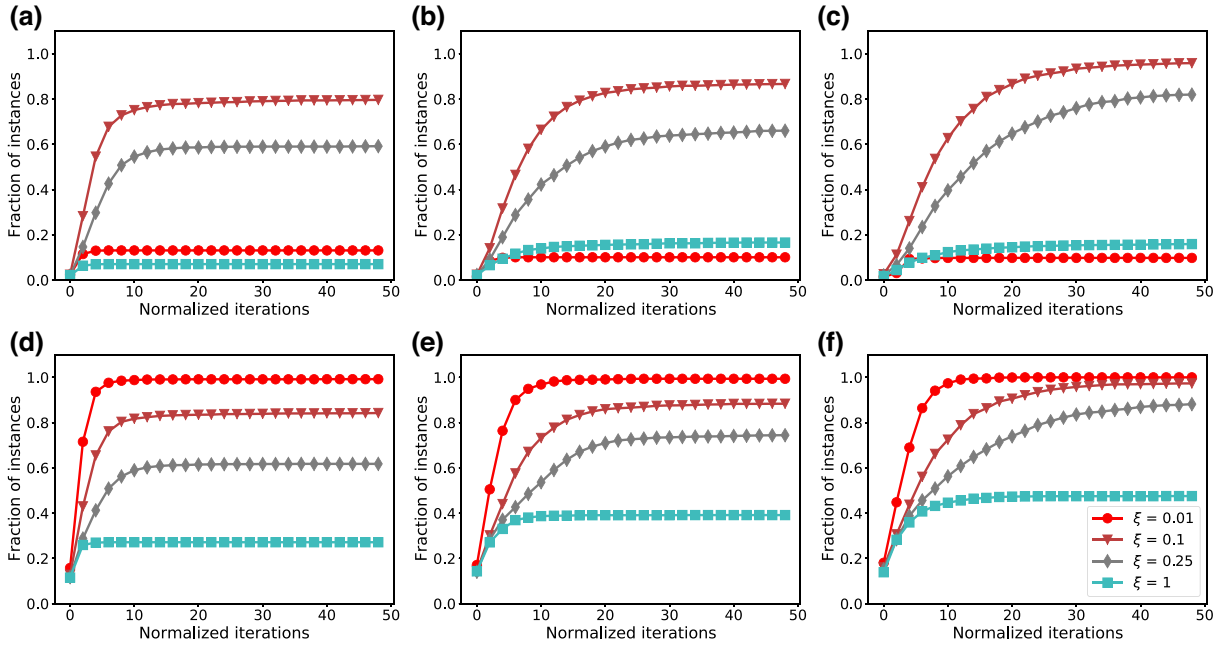


FIG. 8. The fraction of instances with the number of normalized iterations using one-hot encoding. There are four different CVaR coefficients, $\xi \in \{0.01, 0.1, 0.25, 1\}$. The columns correspond to different numbers of layers: (a),(d) $l = 1$, (b),(e) $l = 2$, and (c),(f) $l = 3$. As mentioned in Sec. IV A, the problem size of the FGA problem in one-hot encoding is from six variables to 18 variables and there are 250 random instances for each problem size, so there are 1750 instances in total in the one-hot encoding for each $\xi$ and each $l$. The plots show the fraction of instances that achieve a certain fidelity in the 1750 total instances.

with the coefficients

$$c' = c + \frac{1}{2}\sum_{i\alpha} h_{i\alpha} + \frac{1}{4}\sum_{i\alpha j\beta} J_{i\alpha j\beta} + \frac{1}{4}\sum_{i\alpha} J_{i\alpha i\alpha},$$
$$h'_p = -\frac{1}{2}h_{i\alpha} - \frac{1}{4}\sum_{j\beta}(J_{i\alpha j\beta} + J_{j\beta i\alpha}), \tag{A5}$$
$$J'_{pq} = \frac{1}{4}\left(J_{i\alpha j\beta} + J_{j\beta i\alpha}\right).$$

In the above expression, the indices $p$ and $q$ correspond to $p = i \times |G| + \alpha$ and $q = j \times |G| + \beta$.

## APPENDIX B: SIMULATION RESULTS OF ONE-HOT ENCODING

In this appendix, we provide extended simulation results for the FGA problem using the one-hot encoding, which results in the Ising Hamiltonian in Eq. (11). As shown in Fig. 8, the CVaR VQE helps to improve the performance and to have a larger number of instances that reach the desired fidelity threshold. In particular, similar to the binary encoding, we observe that a smaller $\xi$ leads to better results.

Figure 9 shows the performance of the one-hot encoding using ansatz circuits with and without entanglement. As the figure reveals, entanglement also plays a positive role in this case; however, the advantage of the entangling circuit is not as great as in the case of the binary encoding, as a comparison with Fig. 6 reveals.

Finally, we also explore the scaling of the number of function evaluations of the CVaR VQE to achieve a certain fidelity threshold in the one-hot encoding. The results for this case can be found in Fig. 10. For the one-hot encoding, our scaling results are unfortunately inconclusive, as the figure shows. After an initial exponential increase, similar to the binary encoding, it seems that the curve starts to flatten and goes toward a plateau. However, the system sizes that we can reach in our classical simulations are too small to solidify this conjecture.

## APPENDIX C: FURTHER EXPLORATION OF THE SCALING OF THE NUMBER OF COST-FUNCTION EVALUATIONS

In this appendix, we examine the scaling of the number of cost-function calls with instances that have four gates, $|G| = 4$, which can be exactly represented by $M = 2$ qubits and will not have degenerate ground states caused by the cyclic mapping in Eq. (13). Therefore, we generate a set of random instances with the number of qubits $|F| \times 2 \in \{4, 6, 8, 10, 12, 14, 16, 18\}$, with 50 random instances for each problem size, and study the CVaR VQE for five randomly chosen initial parameter sets for each instance. As shown in Figs. 11(a) and 11(b), the fraction of instances that achieve a certain fidelity threshold using the relevant

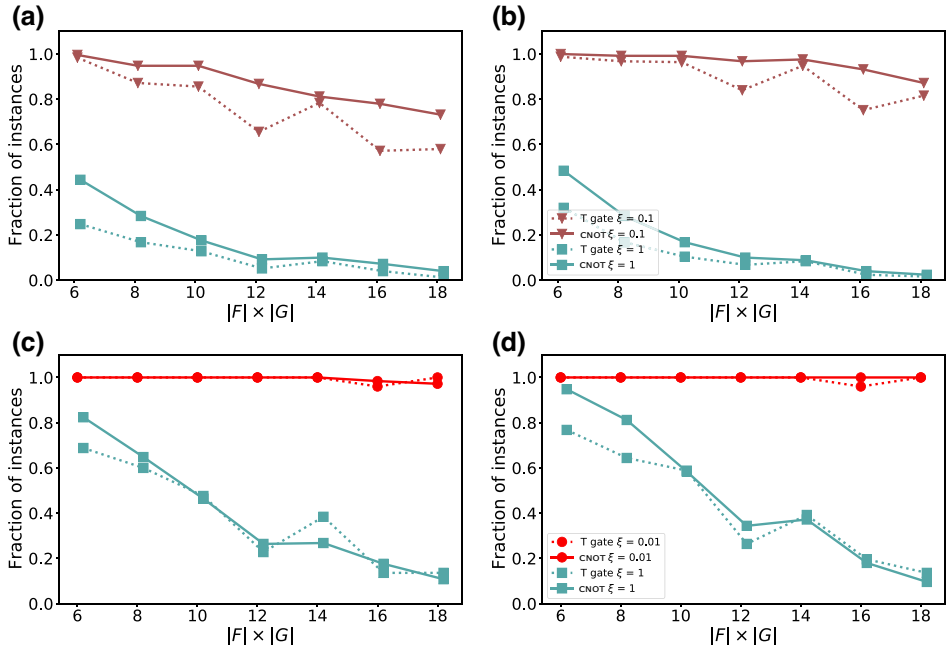


FIG. 9. A performance comparison of the quantum circuit with and without entanglement for one-hot encoding. The fraction of instances that attain a fidelity with the exact solution state of at least (a),(b) 10% and (c),(d) 1%. The columns correspond to different numbers of layers: (a),(c) $l = 2$ and (b),(d) $l = 3$. The dotted lines are related to the results obtained by the quantum circuit that only has the one-qubit $R_y$ and T gates (Fig. 5) and the solid lines are related to the circuit with entanglement generated by the CNOT gates (Fig. 2). The circuit with entanglement performs better in most cases, especially for the CVaR VQE with the coefficient $\xi = 0.1$.

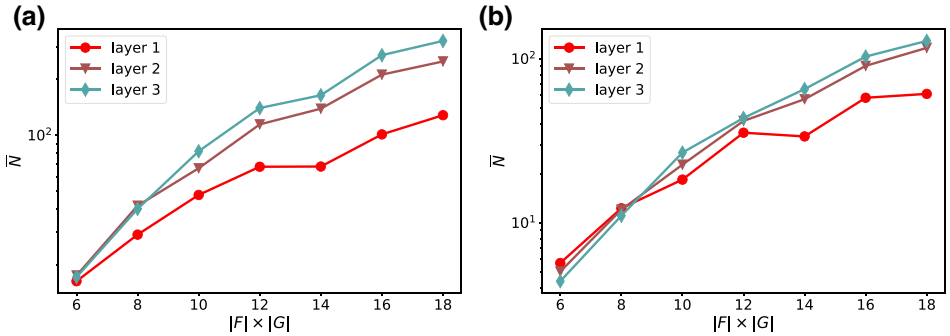


FIG. 10. The scaling of the number of the function evaluations ($\overline{N}$) to get the fidelity (a) 0.1 with $\xi = 0.1$ or (b) 0.01 with $\xi = 0.01$ for the FGA problem in the one-hot encoding. The results are obtained using the entangling quantum circuit with the number of layers $l \in \{1, 2, 3\}$. Just as with the binary encoding (see Sec. IV C), we count the average optimal number of function evaluations of the successful instances that have a fidelity larger than 0.1 or 0.01, respectively.

CVaR coefficient ($\xi = 0.1$ for fidelity 0.10, $\xi = 0.01$ for fidelity 0.01) is still very high and almost all of instances up to 18 qubits can achieve the fidelity threshold of 0.01 with $\text{CVaR}_{0.01}$.

Next, we examine the average number of function calls of the instances that achieve the fidelity threshold, which is what we have done in Sec. IV C but for different instances. In Figs. 11(c) and 11(d), similar to the scaling of the number of function calls shown above, the curve seems to bend and start to flatten, so we do not observe an exponential scaling up to 18 qubits. However, more qubits are required to draw any solid conclusions about the scaling of the number of function calls.

## APPENDIX D: FURTHER DISCUSSION ABOUT THE EFFECTS OF ENTANGLEMENT ON PERFORMANCE

In Sec. IV B, we show that the entangling quantum circuit has a better performance than the nonentangling

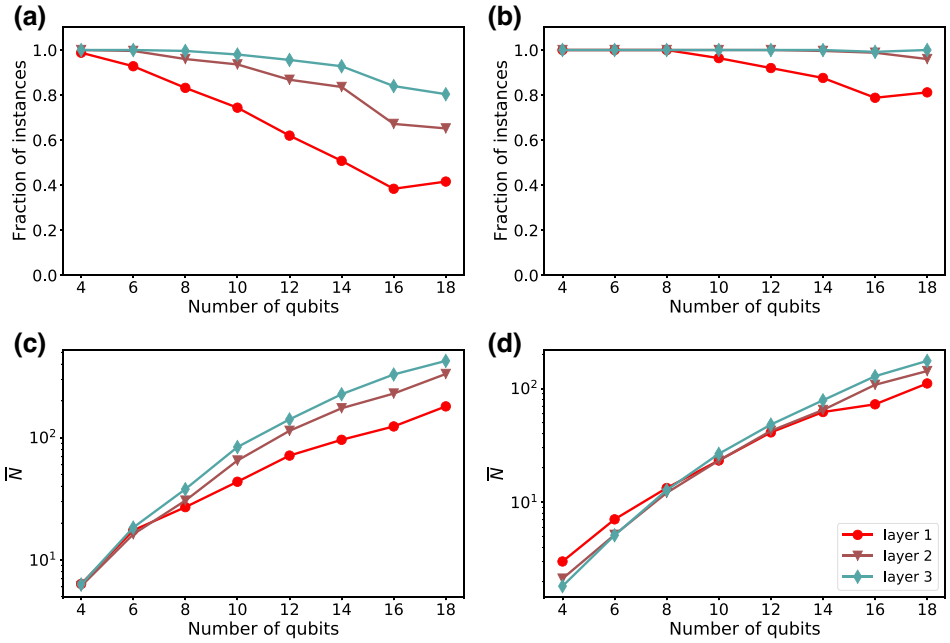


FIG. 11. The results of the FGA instances without the degenerate instances in the binary encoding using the CVaR VQE. (a),(b) The fraction of instances that achieve the fidelity threshold using the CVaR VQE with $l = 1$ (red dots), 2 (brown triangles), and 3 (cyan triangles). The results are obtained using a CVaR coefficient of (a) $\xi = 0.1$ for the fidelity threshold of 10% and (b) $\xi = 0.01$ for the fidelity threshold of 1%. (c),(d) The average number of cost-function calls ($\overline{N}$) to achieve the fidelity threshold with different quantum layers: (c) $\xi = 0.1$ for the fidelity threshold of 10% and (d) $\xi = 0.01$ for the fidelity threshold of 1%. The $y$ axes in (c) and (d) are in logarithmic scale.

circuit if using CVaR as a cost function. However, the nonentangling circuit can be efficiently simulated by a classical computer. Thus a question arises: can the poor performance of the nonentangling circuit be mitigated by the cheaper sampling? In other words, can we still get the optimal solution in a state that has lower fidelity by more sampling of the nonentangling circuit?

To clarify the situation where the advantage of entangling is more possible, we calculate the necessary shots of the failed instances with a nonentangling circuit in binary encoding, if we want to get the optimal solution with a probability $p = 99.9\%$. Focusing on the setup with CVaR coefficient $\xi = 0.01$, i.e., three layers in Fig. 6(d), almost all of the instances with the entangling ansatz can get a fidelity larger than 0.01. For the nonentangling ansatz, if we want to compensate its poor performance with more sampling, the necessary shots for the failed instances will be $S = \log(1-0.999)/\log(1-\bar{p}_{\rm ov})$ with $\bar{p}_{\rm ov} = \max(p_{\rm ov}, 1/2^N)$, where $p_{\rm ov}$ is the maximal fidelity over the entire iteration and $N$ is the qubit number. There are cases in which VQE converges to a local minimum and the probability of the optimal solution $p_{\rm ov}$ is almost zero, resulting in an extremely large shot number $S$. Thus we set $\bar{p}_{\rm ov} = \max(p_{\rm ov}, 1/2^N)$ to avoid the extremely large shots in this case. From the estimation in Fig. 12, the necessary sampling for the failed case increases rapidly with the qubits, to values even larger than the $2^N$ that is needed for brute-force search. In this case, the scaling of the shots $S$ with the qubit number seems exponential but due to the finite data set and the specific setup we used here, it is hard to get a solid and general conclusion about the function of the scaling. However, it is promising that the number of samples needed to compensate for the poor performance of the nonentangling ansatz increases rapidly with the qubit number, so the advantage of better trainability of the entangling ansatz will become more possible when the problem size is large.

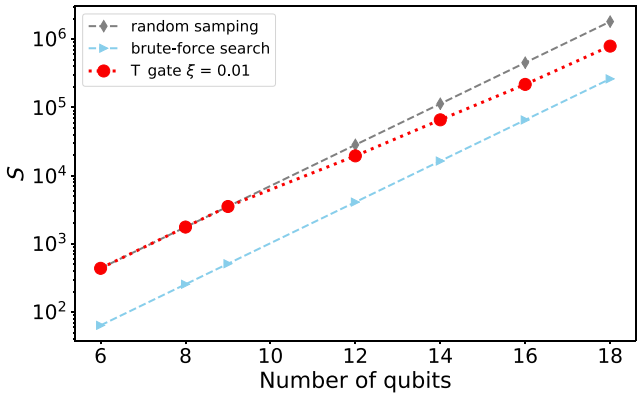


FIG. 12. The shots necessary to sample the optimal solution at least once with a probability of 99.9% for the failed instances with a nonentangling circuit in Fig. 6(d) (number of layers $l = 3$). The gray dashed line shows the shots needed for random sampling $S = \log(1-0.999)/\log[1-(1/2^N)]$ and the light blue dashed line shows $2^N$ related to the brute-force approach, where $N$ represents the qubit number.


[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 1 (2014).

[2] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

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

[4] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).

[5] 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 the lattice Schwinger model, Nature **569**, 355 (2019).

[6] T. Hartung and K. Jansen, Zeta-regularized vacuum expectation values, J. Math. Phys. **60**, 093504 (2019).

[7] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, A. Aspuru-Guzik, R. Blatt, and C. F. Roos, Quantum chemistry calculations on a trapped-ion quantum simulator, Phys. Rev. X **8**, 031022 (2018).

[8] P. Kl. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, Improving variational quantum optimization using CVaR, Quantum **4**, 256 (2020).

[9] Y. Atas, J. Zhang, R. Lewis, A. Jahanpour, J. F. Haase, and C. A. Muschik, SU(2) hadrons on a quantum computer, Nat. Commun. **12**, 6499 (2021).

[10] H. Mohammadbagherpoor, P. Dreher, M. Ibrahim, Y.-H. Oh, J. Hall, R. E. Stone, and M. Stojkovic, Exploring airline gate-scheduling optimization using quantum computers, ArXiv:2111.09472 (2021).

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

[12] S. Hyun Kim, E. Feron, J.-P. Clarke, A. Marzuoli, and D. Delahaye, Airport gate scheduling for passengers, aircraft, and operations, J. Air Transp. **25**, 109 (2017).

[13] T. Stollenwerk, E. Lobe, and M. Jung, in *Proceedings of the First International Workshop on Quantum Technology and Optimization Problems*, Theoretical Computer Science and General Issues No. 9 (Springer, Munich, Germany, 2019).

[14] G. Finke, R. E. Burkard, and F. Rendl, in *Surveys in Combinatorial Optimization*, edited by S. Martello, G. Laporte, M. Minoux, and C. Ribeiro, North-Holland Mathematics Studies, Vol. 132 (North-Holland, 1987), p. 61.

[15] D. Venturelli, D. J. J. Marchand, and G. Higuera Rojo, Job shop scheduling solver based on quantum annealing, ArXiv:1506.08479 (2016).

[16] T. Stollenwerk, B. O'Gorman, D. Venturelli, S. Mandrà, O. Rodionova, H. Ng, B. Sridhar, E. Gilbert Rieffel, and R.

Biswas, Quantum annealing applied to de-conflicting optimal trajectories for air traffic management, IEEE Trans. Intell. Transp. Syst. **21**, 285 (2020).

[17] T. Stollenwerk, V. Michaud, E. Lobe, M. Picard, A. Basermann, and T. Botter, Agile earth observation satellite scheduling with a quantum annealer, IEEE Trans. Aerosp. Electron. Syst. **57**, 3520 (2021).

[18] I. Hen and F. M. Spedalieri, Quantum annealing for constrained optimization, Phys. Rev. Appl. **5**, 034007 (2016).

[19] 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).

[20] T. Stollenwerk, S. Hadfield, and Z. Wang, Toward quantum gate-model heuristics for real-world planning problems, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., USA, 1979).

[22] E. R. Anschuetz, L. Funcke, P. T. Komiske, S. Kryhin, and J. Thaler, Degeneracy engineering for classical and quantum annealing: A case study of sparse linear regression in collider physics, Phys. Rev. D **106**, 056008 (2022).

[23] D. Paulson, L. Dellantonio, J. F. Haase, A. Celi, A. Kan, A. Jena, C. Kokail, R. van Bijnen, K. Jansen, P. Zoller, and C. A. Muschik, Simulating 2D effects in lattice gauge theories on a quantum computer, PRX Quantum **2**, 030334 (2021).

[24] A. Avkhadiev, P. E. Shanahan, and R. D. Young, Accelerating lattice quantum field theory calculations via interpolator optimization using noisy intermediate-scale quantum computing, Phys. Rev. Lett. **124**, 080501 (2020).

[25] G. Mazzola, S. V. Mathis, G. Mazzola, and I. Tavernelli, Gauge-invariant quantum circuits for $U(1)$ and Yang-Mills lattice gauge theories, Phys. Rev. Res. **3**, 043209 (2021).

[26] 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).

[27] D. Amaro, M. Rosenkranz, N. Fitzpatrick, K. Hirano, and M. Fiorentini, A case study of variational quantum algorithms for a job shop scheduling problem, EPJ Quantum Technol. **9**, 5 (2022).

[28] G. Nannicini, Performance of hybrid quantum-classical variational heuristics for combinatorial optimization, Phys. Rev. E **99**, 013304 (2019).

[29] S. Mugel, C. Kuchkovsky, E. Sánchez, S. Fernández-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orús, Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks, Phys. Rev. Res. **4**, 013006 (2022).

[30] Qiskit contributors, Qiskit: An open-source framework for quantum computing (2023), https://doi.org/10.5281/zenodo.2573505.

[31] M. J. D. Powell, in *Advances in Optimization and Numerical Analysis* (Springer, Springer Dordrecht, 1994), p. 51.

[32] Note that this is possible because we assume a perfect quantum computer without shot noise, i.e., we have direct access to the state vector and we can monitor the overlap with exact solutions throughout the optimization procedure.

[33] The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/