

**Error mitigation for quantum approximate optimization**

Anita Weidinger<sup>1,\*</sup>, Glen Bigan Mbeng<sup>1,2</sup> and Wolfgang Lechner<sup>1,2</sup>  
<sup>1</sup>*Institute for Theoretical Physics, University of Innsbruck, A-6020 Innsbruck, Austria*  
<sup>2</sup>*Parity Quantum Computing GmbH, A-6020 Innsbruck, Austria*

 (Received 5 April 2023; accepted 7 August 2023; published 8 September 2023)

Solving optimization problems on near term quantum devices requires developing error mitigation techniques to cope with hardware decoherence and dephasing processes. We propose a mitigation technique based on parity encoding. This method uses a redundant encoding of logical variables to solve optimization problems on fully programmable planar quantum chips. We discuss how this redundancy can be exploited to mitigate errors in quantum optimization algorithms. In the specific context of the quantum approximate optimization algorithm, we show that errors can be significantly mitigated by appropriately modifying the objective cost function.

DOI: [10.1103/PhysRevA.108.032408](https://doi.org/10.1103/PhysRevA.108.032408)

**I. INTRODUCTION**

In recent years, immense effort has been made to leverage quantum computers to solve industry-relevant optimization problems [1–5]. The main obstacles to observing a quantum advantage are the limitations of the current noisy intermediate-scale quantum (NISQ) devices, which offer only a few hundred qubits and are prone to errors and noise [6,7]. Until technology advances, specific algorithms have been designed to utilize the maximum out of the imperfect hardware [8]. The proposed algorithms generally follow a hybrid quantum-classical approach, simultaneously exploiting quantum and classical computational power. The most prominent hybrid algorithms in current research are the variational quantum algorithms (VQAs), where parametrized quantum circuits are combined with a classical optimizer [9,10]. VQAs are suitable for various applications like chemistry [11,12] or machine learning [13,14]. In particular, the quantum approximate optimization algorithm (QAOA) is a promising VQA that aims at finding approximate solutions to combinatorial optimization problems [2,15]. Hybrid algorithms have the advantage of providing shallow circuit depths, which makes them less vulnerable to noise. Gates are potential error sources, however, in an ideal circuit, more circuit layers would improve the quality of the solution. In practice, there is a break-even point, where adding a layer negatively affects the algorithms' performance. The first experimental setups showed how severely limited the circuit depth, and therefore the algorithm's performance, is in current devices [5,16,17]. Hence, it is essential to lessen the effect of noise to exploit the capabilities of VQAs fully. Various quantum error mitigation (QEM) techniques have been developed to tackle this issue. In general, QEM aims to reduce the impact of noise via

classical postprocessing and multiple circuit runs, requiring no qubit overhead [18,19]. These resource sparing techniques are suitable for NISQ devices, in contrast to quantum error correction (QEC) codes, which instead require significant qubit overheads [20,21]. Some examples of QEM schemes are extrapolation and quasiprobability methods [22–24], quantum subspace expansion [25,26], individual error reduction [27], symmetry verification [28–31], and combinations like error extrapolation with symmetry verification [30,32] and quasiprobability method [32].

In this work we propose an error mitigation technique for QAOA that is based on the Lechner-Hauke-Zoller (LHZ) or parity architecture [33]. Instead of using the energy as cost function of the variational algorithm, our method introduces logical qubits and the decoded *logical energy* is used. This *logical energy* is the result of the evaluation of spanning trees in the parity variables, where each physical qubit contributes to multiple logical qubits. It is this redundancy that introduces the error mitigation features of the method.

The parity architecture was initially designed to tackle the issue of limited connectivity in quantum annealing hardware. However, when combined with digital hardware, the parity architecture provides the benefit of full parallelization of quantum gates [34] and universal quantum computing [35]. In addition, Ref. [36] shows that the parity architecture can be viewed as a classical low-density parity check code, and adding classical postprocessing procedures like belief propagation make it more robust against noise. Furthermore, Ref. [37] establishes a stabilizer-based formalism for the parity architecture.

The transformation from logical to physical qubits introduces redundant information which can be exploited in classical postprocessing to correct errors. Our method uses this redundancy not just in postprocessing but also during computation. Consequently, the noise-reducing benefit is propagated throughout the algorithm and not just instilled in the end. This is done by redesigning the information that is handed from the quantum circuit to the classical optimizer, represented by the cost function. With this adaptation, the computation is performed on the physical quantum hardware, while the classical optimizer always stays in the logical

\*anita.weidinger@uibk.ac.at

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

subspace. Our results demonstrate that this modification introduces quantitative error mitigation for QAOA in the presence of noise. The next sections are organized as follows: In Sec. II we outline the theoretical background for this paper, which includes a description of QAOA, parity QAOA, and a detailed explanation of our method: decoded parity QAOA. In Sec. III we discuss the numerical simulations' results where we compare the discussed QAOA approaches. Finally, we discuss the open questions and the prospects of this topic.

## II. METHODS

### A. Quantum approximate optimization algorithm with rerouting

The quantum approximate optimization algorithm (QAOA) [15] aims to find approximate solutions to combinatorial optimization problems, cast in the form of energy minimization of a general  $N$ -spin problem Hamiltonian:

$$H_p = \sum_i J_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z + \sum_{i<j<k} J_{ijk} \sigma_i^z \sigma_j^z \sigma_k^z + \dots, \quad (1)$$

where  $\sigma_j^{x,y,z}$  denote the Pauli spin operators and  $\{J_i, J_{ij}, J_{ijk}, \dots\}$  are long-range, multispin interactions.

The simplest  $p$ -level QAOA [15,38] starts from an initial state  $|\Psi_0\rangle = |+\rangle^{\otimes N}$  and alternates a phase separation gate  $U_p(\gamma) = e^{-i\gamma H_p}$  and a mixing gate  $U_x(\beta) = \prod_{j=1}^N e^{-i\beta \sigma_j^x}$  for  $p$  rounds in a quantum circuit. Running the circuit on suitable quantum hardware generates the variational state

$$|\Psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = U_x(\beta_p) U_p(\gamma_p) \dots U_x(\beta_1) U_p(\gamma_1) |\Psi_0\rangle, \quad (2)$$

which depends on the  $2p$  parameters  $\boldsymbol{\gamma} = \{\gamma_1, \dots, \gamma_p\}$  and  $\boldsymbol{\beta} = \{\beta_1, \dots, \beta_p\}$ . Then, by repeated measurements of the state in the computational basis, we can estimate the QAOA objective function

$$\mathcal{C}(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \Psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | H_p | \Psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle \quad (3)$$

for any parameter choice  $(\boldsymbol{\beta}, \boldsymbol{\gamma})$ . This objective function coincides with the expectation value of the original classical spin-glass energy on the measured  $N$ -bit strings. We use a classical computer to implement a feedback loop optimization algorithm to find the optimal parameters  $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$  that minimize  $\mathcal{C}(\boldsymbol{\beta}, \boldsymbol{\gamma})$ . Finally, we run the quantum circuit with parameters  $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$  and generate bit strings providing approximate solutions of the classical optimization problem. Due to the nature of the variational ansatz, the quality of the approximate solution generated by ideal noiseless circuits increases monotonically with the depth  $p$ . In particular, as discussed in Refs. [15,39], the adiabatic theorem [40] ensures that for  $p \rightarrow \infty$  the algorithm converges to an exact solution of the classical problem.

Although the mixing gate  $U_x(\beta)$  is straightforward to implement with single-qubit operations, hardware with local connectivity graphs require an additional compilation step [41,42] to decompose the phase gate  $U_p(\gamma)$  into a sequence of available local gates. In most QAOA implementations, this compilation step relies on a rerouting strategy [5,43–48], which performs additional layers of SWAP gates such that the interacting spins correspond to an edge in the hardware graph

at least once, enabling  $U_p(\gamma)$ 's implementation. However, the rerouting introduces a SWAP-gate overhead that should be minimized [43] to reduce the effect of decoherence and dephasing processes on the compiled quantum circuit.

### B. Parity quantum approximate optimization algorithm

In this section we describe *parity QAOA* [33,37,49–51], which uses an alternative compilation strategy, suitable for state-of-the-art quantum devices with planar chips [52–55]. To simplify the presentation, we follow Ref. [33] and specifically consider quadratic unconstrained binary optimization (QUBO) [56] problem Hamiltonians:

$$H_p = \sum_{i=1}^N \sum_{j<i} J_{ij} \sigma_i^z \sigma_j^z. \quad (4)$$

However, we refer to Refs. [37,49] for methods to tackle the more general problem Hamiltonian of Eq. (1) with parity QAOA.

Parity QAOA relies on the parity (or LHZ) transformation [33,49] to first encode the optimization problem into the local fields of a two-dimensional (2D) local problem Hamiltonian. In particular, for all-to-all connected QUBO problems, the parity mapping replaces the original  $N$  ‘‘logical’’ qubits  $\sigma_i^z$  with  $K = N(N-1)/2$  physical parity qubits  $\tilde{\sigma}_v^z$  each representing the relative configuration of two logical qubits  $\sigma_i^z \sigma_j^z \rightarrow \tilde{\sigma}_{i+(N-1)j}^z$  [see Fig. 1(a)]. In terms of the new physical qubits, the problem Hamiltonian of Eq. (4) takes the simple local field form

$$H_p \rightarrow \tilde{H}_p = \sum_{v=1}^K \tilde{J}_v \tilde{\sigma}_v^z, \quad (5)$$

with  $\tilde{J}_{i+(N-1)j} = J_{ij}$ . However, the parity transformation also enlarges the configuration space by introducing spurious (or invalid) physical states which do not represent any logical state. As suggested in Ref. [33], we can address this issue by arranging the qubits on a planar grid and considering a set of  $L = (N-1)(N-2)/2$  independent local four-body (and three-body) plaquette constraints terms  $H_{\uparrow\uparrow,l} = \tilde{\sigma}_{(l,1)}^z \tilde{\sigma}_{(l,2)}^z \tilde{\sigma}_{(l,3)}^z \tilde{\sigma}_{(l,4)}^z$  (and  $H_{\uparrow\downarrow,l} = \tilde{\sigma}_{(l,1)}^z \tilde{\sigma}_{(l,2)}^z \tilde{\sigma}_{(l,3)}^z$ ). This construction effectively recasts the problem of minimizing the nonlocal  $N$ -spin Hamiltonian  $H_p$  into the problem of minimizing the  $K$ -spin local Hamiltonian  $\tilde{H}_p$ , with the constraints  $\langle \psi | H_{\uparrow\uparrow,l} | \psi \rangle = 1$  for  $l = 1, \dots, L$ . Figure 1(b) illustrates the resulting parity architecture for a system of  $K = 15$  parity qubits (or  $N = 6$  logical qubits). It depicts an example of a valid physical configuration that fulfills all constraints (on the left), and the corresponding configuration of the logical qubits (on the right).

The simplest  $p$ -level parity QAOA approximates the solution of the constraint  $K$ -spin optimization problem by implementing the following variational state [34]:

$$|\tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega})\rangle = \tilde{U}_x(\beta_p) \tilde{U}_c(\Omega_p) \tilde{U}_z(\gamma_p) \dots \tilde{U}_x(\beta_1) \tilde{U}_c(\Omega_1) \tilde{U}_z(\gamma_1) |\tilde{\Psi}_0\rangle, \quad (6)$$

where  $|\tilde{\Psi}_0\rangle = |+\rangle^{\otimes K}$  is the initial state,  $\tilde{U}_x(\gamma) = e^{-i\gamma \tilde{H}_p}$  and  $\tilde{U}_c(\Omega) = \prod_{l=1}^L e^{-i\Omega H_{\uparrow\uparrow,l}}$  are the phase separation gates, and

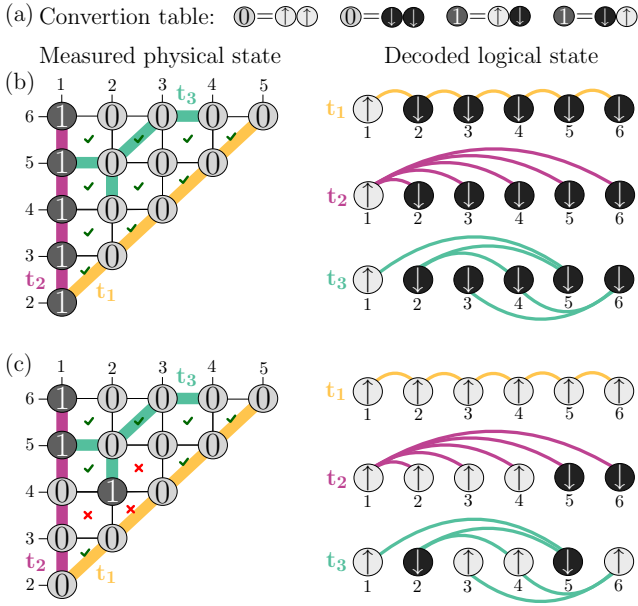


FIG. 1. Three decoding examples, based on the conversion table depicted in panel (a), for given physical configurations. Panel (b) shows a constraint fulfilling physical state (left) while the one in panel (c) violates some constraints (red crosses). The physical qubits used for decoding are marked with colored lines, labeled  $t_1$ ,  $t_2$ , and  $t_3$ . Those three decoding possibilities return three identical logical configurations (right) for the constraint fulfilling configuration in panel (b) and three different logical states for the constraint violating configuration in panel (c). The physical qubits used are marked as interactions in the logical graph. They are spanning trees in the logical graph, covering all logical qubits without forming a cycle.

$\tilde{U}_x(\beta) = \prod_{v=1}^K e^{-i\beta\sigma_v^x}$  is the mixing gate of the QAOA quantum circuit. Then, the optimal values of the  $3p$  parameters  $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$ ,  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)$ , and  $\boldsymbol{\Omega} = (\Omega_1, \dots, \Omega_p)$  are found using repeated measurements of the state in the computational basis to estimate and optimize the objective function

$$\tilde{C}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) = \langle \tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) | \tilde{H}_p + c \sum_l (1 - H_{\ddagger,l}) | \tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) \rangle, \quad (7)$$

where the penalty strength  $c$  is a positive constant introduced to penalize invalid states and should be larger than  $H_p$ 's lowest energy gap [57]. The objective function in Eq. (7) coincides with the weighted sum of the energy expectation values and the number of violated constraints of the measured  $K$ -bit strings. As in most QAOA implementations, the adiabatic theorem ensures that for  $p \rightarrow \infty$ , the parity QAOA variational state in Eq. (6) can represent the exact solution of the constrained  $K$ -spin optimization problem [33,34].

The main advantage of parity mapping lies within the structure of the required gates. On the one hand, the gates  $\tilde{U}_x(\beta)$  and  $\tilde{U}_p(\gamma)$  involve only single-qubit operations. On the other hand, the multiqubit phase gate  $\tilde{U}_c(\Omega)$  can be conveniently implemented on nearest-neighbor planar chips by a constant-depth sequence of parallel controlled-NOT (CNOT) gates and single-qubit rotations [34], see also Appendix A,

or via optimized fast four-qubit gate operations [51]. The numerical benchmarks in Ref. [58] confirmed that these parity QAOA implementations require significantly fewer multi-qubit gates than QAOA with rerouting when running large problem instances on planar chips. The use of less error-prone gates favors parity QAOA when running the two algorithms on NISQ hardware. However, the lower number of qubits  $N < K$  instead favors the rerouting strategy for QAOA. Although the trade-off between the number of qubits and the number of multiqubit gates still needs to be systematically analyzed, parity QAOA is a general alternative to the rerouting strategy for QAOA [50,59]. In the next section, we introduce a new decoded parity QAOA. The protocol is based on the evaluation of the energy of the decoded logical qubits instead of the actual energy of the physical qubits.

### C. Decoded parity quantum approximate optimization algorithm

In this section, we describe a decoding strategy to mitigate the errors arising in parity QAOA.

Although parity QAOA's target state fulfills all parity constraints, running the algorithm on noisy quantum devices or with small  $p$  generates errors associated with constraint violations in the readouts. We can partially correct the leakage errors by implementing various decoding strategies, which recover logical states but do not guarantee the recovery of the target state. More specifically, a decoding strategy is a rule that assigns values to the  $N$  logical qubits  $\boldsymbol{q} = (q_1, \dots, q_N)$  given a readout of the  $K$  physical qubits  $\tilde{\boldsymbol{q}} = (\tilde{q}_1, \dots, \tilde{q}_K)$ . Here, following Refs. [33,60], we consider a simple strategy based on spanning trees (subgraphs where all pairs of nodes are connected by exactly one path) defined on the logical qubits. We use subsets of physical qubits, describing spanning trees  $t_i$  on the original logical qubits, to fully determine a configuration of the logical qubits  $\boldsymbol{q}^{(t)}$  (up to a global spin flip). Then, if  $\tilde{\boldsymbol{q}}$  satisfies all  $L$  parity constraints, all the  $N^{N-2}$  spanning trees return the same encoded logical qubit configuration [see Fig. 1(b)]. On the other hand, when  $\tilde{\boldsymbol{q}}$  does not satisfy all parity constraints, different spanning trees  $t, t'$  may result in a different logical configuration  $\boldsymbol{q}^{(t)} \neq \boldsymbol{q}^{(t')}$  [see Fig. 1(c)]. If the leakage error is sufficiently small, a majority vote over different spanning tree decoding strategies can correct the leakage errors that occurred during computation [33]. Reference [60] used this method to mitigate errors in adiabatic quantum optimization by adding a single decoding step at the end of the adiabatic protocol. We employ similar a decoding strategy to enhance parity QAOA's performance.

Let  $\tilde{\rho}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) = |\tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega})\rangle\langle\tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega})|$  be the density matrix representing the circuit's output state [61] and let  $P_{\boldsymbol{q}^{(t)}} = |\boldsymbol{q}^{(t)}\rangle\langle\boldsymbol{q}^{(t)}|$  be the projector on the decoded logical state  $|\boldsymbol{q}^{(t)}\rangle$ . For each spanning tree  $t$ , we define the linear decoding map

$$\begin{aligned} D_t[\tilde{\rho}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega})] &= \sum_{\tilde{\boldsymbol{q}} \in \{0,1\}^K} \langle \tilde{\boldsymbol{q}} | \tilde{\rho}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) | \tilde{\boldsymbol{q}} \rangle P_{\boldsymbol{q}^{(t)}} \\ &= \sum_{\tilde{\boldsymbol{q}} \in \{0,1\}^K} |\langle \tilde{\boldsymbol{q}} | \tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) \rangle|^2 P_{\boldsymbol{q}^{(t)}}, \end{aligned} \quad (8)$$

where each possible measurement outcome  $\tilde{q}$ , which appears with probability  $|\langle \tilde{q} | \tilde{\Psi}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) \rangle|^2$ , is translated to the corresponding logical configuration  $\mathbf{q}^{(t)}$ . Then, to apply the spanning tree decoding strategy to parity QAOA, we consider a set of  $M$  spanning trees  $T = \{t_1, t_2, \dots, t_M\}$  and we implement the same parity QAOA variational state of Eq. (6). However to find the optimal variational parameters  $(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\Omega}^*)$ , we instead minimize the expectation value of the decoded states' average energy:

$$\mathcal{C}^{(T)}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) = \frac{1}{M} \sum_{m=1}^M \text{Tr}(H_P D_{t_m}[\tilde{\rho}(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega})]), \quad (9)$$

which we estimate by repeatedly measuring the parity QAOA circuit and decoding the outcome. Finally, to generate approximate solutions, we prepare the optimal variational state  $|\tilde{\Psi}(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\Omega}^*)\rangle$ , we measure and decode the qubits  $\tilde{q} \rightarrow \{\mathbf{q}^{(t_1)}, \mathbf{q}^{(t_2)}, \dots, \mathbf{q}^{(t_M)}\}$ , and we return the best decoded logical state

$$\mathbf{q}^{(T)} = \arg \min_{t \in T} \langle \mathbf{q}^{(t)} | H_P | \mathbf{q}^{(t)} \rangle. \quad (10)$$

Meaning that the decoded state  $\mathbf{q}^{(t)}$  that returns the lowest energy  $\langle \mathbf{q}^{(t)} | H_P | \mathbf{q}^{(t)} \rangle$  for all  $t \in T$  contributes to the approximate solution. The output distribution is described by the density matrix  $D_T[\tilde{\rho}(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\Omega}^*)]$ . The objective function (9) and the decoding map  $D_T$  [Eq. (10)] play an important role in decoded parity QAOA. However, their definition is not unique [62,63]. In Appendix D, we study the effect of choosing a different decoding map  $D_T$ .

### III. RESULTS AND SIMULATIONS

In the following sections, we present numerical benchmarks of the decoded parity QAOA (Sec. II C) and compare it with the rerouting method (Sec. II A).

We benchmark QAOA on random QUBO problem instances, described by the Hamiltonian in Eq. (4) with uniformly distributed couplings  $J_{ij} \in \{\pm 0.1, \pm 0.2, \dots, \pm 1.0\}$ . In particular, we consider problems with  $N = 3, 4, 5, 6, 7$  logical qubits for the rerouting method, which correspond to  $K = 3, 6, 10, 15, 21$  physical qubits for the parity architecture.

To evaluate the algorithm's performance, we first find the target ground state  $\mathbf{q}^{(\text{gs})}$  configuration by brute force. Then, for QAOA, we generate 100 random start parameters  $\boldsymbol{\beta}$  and  $\boldsymbol{\gamma}$  for the rerouting strategy and  $\boldsymbol{\beta}$ ,  $\boldsymbol{\gamma}$ , and  $\boldsymbol{\Omega}$  for the parity strategy, followed by parameter updates using the Metropolis method. The run with the lowest energy  $\mathcal{C}(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)$  or  $\tilde{\mathcal{C}}(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\Omega}^*)$  is used for later calculations. We use the qiskit library [64] to simulate the quantum circuits and estimate the probability of outputting the ground state

$$P_{\text{gs}} = \langle \mathbf{q}^{(\text{gs})} | \rho_{\text{out}} | \mathbf{q}^{(\text{gs})} \rangle, \quad (11)$$

where the density matrix  $\rho_{\text{out}}$  describes the algorithm's output distribution. Specifically, we have  $\rho_{\text{out}} = |\Psi(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)\rangle\langle\Psi(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)|$  for QAOA with rerouting and  $\rho_{\text{out}} = D_T[\tilde{\rho}(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*, \boldsymbol{\Omega}^*)]$  for the decoded parity QAOA.

Since the parity strategy requires more qubits than rerouting, we cannot use  $P_{\text{gs}}$  to directly compare the algorithms. To enable a fairer comparison, we run the algorithms on multiple

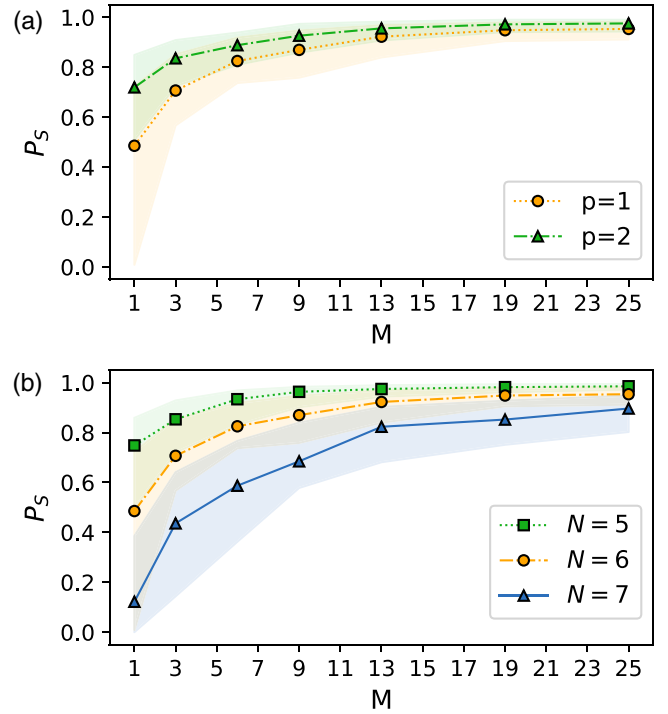


FIG. 2. Success probability  $P_S$  for parity QAOA in dependence of the total number of spanning trees  $M$  used for decoding for (a)  $N = 6$ ,  $p = 1, 2$  and (b) different system sizes  $N$  with  $p = 1$ . The (dashed) lines and markers represent the median, the shaded area the 25th and 75th percentile of 100 random instances.

copies in parallel and take the best outcome. Assuming the same fixed budget of  $K$  physical qubits for both algorithms, the resulting success probability is

$$P_S = 1 - (1 - P_{\text{gs}})^r, \quad (12)$$

where the number of copies is  $r = 1$  for parity strategy or  $r = K/N = (N - 1)/2$  for the rerouting strategy. For further details on the numerical simulations, we refer to Appendix B.

#### A. Noiseless simulations

First, we investigate how the performance of parity QAOA depends on the number  $M$  of spanning trees used. Then we fix the number  $M$  of spanning trees and simulate different system sizes  $N$ . We compare the latter results to the rerouting strategy.

##### 1. Decoded parity quantum approximate optimization algorithm

We randomly create a set of spanning trees  $T = \{t_1, t_2, \dots, t_M\}$ , for different  $M$ . More details can be found in Appendix B. Figure 2(a) shows the dependence of the success probability  $P_S$  on the number  $M$  of trees for  $N = 6$  and  $p = 1, 2$ . The results show the median of 100 different problem instances which use the same trees  $T$  for decoding. The shaded area ranges from the 25th to the 75th percentile. Increasing the number  $M$  of decoding trees improves  $P_S$ . In addition, the error bars decrease with increasing  $M$ . For example, a single tree,  $M = 1$ , reaches a median  $P_S$  of about 0.5 for  $p = 1$ , but the error bars range from 0 to 0.7. This means that half of the simulated instances



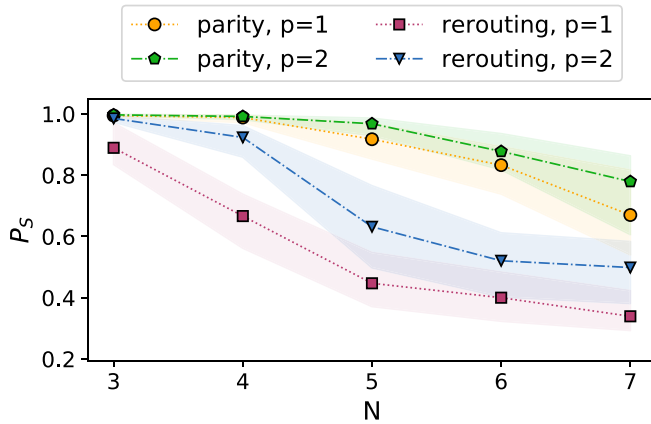


FIG. 3. Success probability  $P_S$  in dependence of the system size  $N$ ,  $m = N$ , and  $p = 1, 2$ . The dashed lines represent the median, the shaded area the 25th and 75th percentile of 100 random instances.

return a  $P_S$  within this range. The outcome is unpredictable, highly depending on which tree was used for decoding. It is, therefore, advantageous to use more trees, but the improvement will saturate with increasing  $M$ . Further details about this are provided in Appendix F and Fig. 7.

Results for different system sizes  $N$  with  $p = 1$ , are shown in Fig. 2(b). Here, one can observe a similar behavior as in Fig. 2(a), noting that with increasing system size  $N$  the problem becomes harder to solve.

## 2. Logical lines

In the previous section we studied the performance of parity QAOA using different amounts  $M$  of random spanning trees. Now we fix the decoding to a special set of spanning trees in the parity architecture: the *logical lines* [35,37], denoted by  $T^{(l)} = \{t_1^{(l)}, t_2^{(l)}, \dots, t_N^{(l)}\}$ . Logical line  $i$ , corresponding to tree  $t_i^{(l)}$ , includes all parity qubits containing the logical qubit  $i$ , e.g.,  $t_0^{(l)}$  covers the parity qubits ( $0x$ ) for  $x = 1, \dots, N$ . An example is  $t_2$  in Fig. 1. A (all-to-all connected QUBO) problem with  $N$  logical qubits has  $N$  logical lines, hence  $N = M$ .

Figure 3 shows the success probability  $P_S$  in dependence of the number  $N$  of logical qubits for the different QAOA approaches. The success probability  $P_S$  for QAOA with the rerouting strategy seems to decrease slower with increasing  $N$ . This is due to the fact that the number of repetitions  $r$  (to have the same number of qubits) increases with  $N$ , in fact  $r = (N - 1)/2$ .  $P_S$  for parity QAOA drops faster and it seems that the two curves will coincide for higher  $N$ . Here, we recall that  $M = N$ , but the total number of possible spanning trees scales exponentially with  $N$ . To keep an advantage it is necessary to increase  $M$  more than linearly with  $N$  for parity QAOA. How  $P_S$  for different sizes  $N$  scale with  $M$  is shown in Fig. 2(b).

## B. Noisy simulation

In this section we investigate the performance with noisy gates, i.e., we introduce a depolarizing error on all circuit

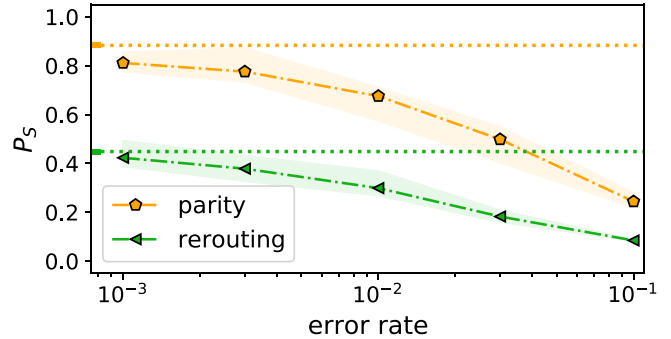


FIG. 4. Success probability  $P_S$  for different 2-qubit gate error rates for  $p = 2$ ,  $N = 6$ . The 1-qubit gate error rate is fixed to 0.001. Parity QAOA is decoded by  $M = N = 6$  spanning trees and uses 104 CNOT gates, the circuit for QAOA with rerouting includes 94 CNOT gates. The horizontal dashed lines represent the results for an ideal circuit. The Median of 20 random instances is shown, the shaded area is the 25th and 75th percentile.

gates. The 1-qubit gate error rate will be fixed to 0.001, while the 2-qubit error will range from 0.001 to 0.1.

With noise applied to the 2-qubit gates it is important to determine how many CNOT gates are needed in each circuit. The outlined description in the corresponding Secs. II A and II B is extended in Appendix A. There, it is shown that, for a problem with  $N = 6$  qubits, the rerouting layout has the advantage of fewer CNOT gates in the circuit (94 vs 104 for  $p = 2$ ).

Parity QAOA is decoded by the set of trees  $T^{(l)} = \{t_1^{(l)}, t_2^{(l)}, \dots, t_N^{(l)}\}$ , referred to as logical lines, as in Sec. III A 2. The obtained simulated and calculated results are shown in Fig. 4. As seen in the previous results, parity QAOA achieves a higher success probability  $P_S$  than QAOA with rerouting and is able to keep this advantage for all error rates ( $< 0.1$ ), despite having the disadvantage of a higher CNOT gate count. With an error rate of 0.03 the parity QAOA performs as well as the rerouting with no noise, executed 2.5 times.

## IV. CONCLUSIONS

In this work we have shown how to apply a decoding scheme into the parity QAOA optimization routine, which exploits the redundant information introduced by the parity transformation. This approach leads to a better success probability  $P_S$  compared with the standard rerouting method even when the number of used qubits is considered the same for both methods. For the studied problems, the QAOA with rerouting has the advantage of requiring fewer CNOT gates, but simulations with noise on small system sizes show that parity QAOA can keep the advantage even with high error rates ( $< 0.1$ ). The work in Ref. [58] indicates that this advantage will disappear for bigger systems and higher-order interaction terms, making the parity architecture more favorable. In addition, Ref. [65] shows how to reduce the gate count in parity QAOA further by replacing some CNOT gates by native ZZ gates. However, ongoing work will improve compiler quality for the rerouting circuit, hence, future work should investigate gate and constraint optimization in the parity architecture,

especially for complete graphs. Nevertheless, it is an open question how present the advantage in  $P_S$  of the introduced method will be for bigger systems sizes and how many spanning trees are the optimal choice. The results in this work suggest that the number of trees used for decoding has to be bigger than the number of logical qubits for  $N > 6$ . There exist an exponential number of spanning trees for a complete graph, meaning that there would be enough resources. However, the classical overhead should be kept in mind. A detailed discussion about the additional resources is in Appendix C. Although we only presented the decoding scheme for complete graphs, the extension to more general graphs with higher-order interactions is straightforward [49]. Those problems may have less qubit overhead, but then fewer spanning trees exist for decoding. Future work should investigate this trade-off. In addition, it might be advantageous to adapt the objective function if one increases the system size. In this work, we used the mean energy of all used spanning trees, but a different formulation could be more beneficial considering bigger problem instances.

A Python code implementation of the algorithms presented in this work and a tutorial on how to use it is available online [66].

#### ACKNOWLEDGMENTS

The authors thank E. Wybo, M. Fellner, K. Ender, and R. S. ter Hoeven, for useful discussions. Work was supported by the Austrian Science Fund (FWF) through a START grant under Project No. Y1067-N27 and the SFB BeyondC Project No. F7108-N38. This project was funded within the QuantERA II Program that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 101017733.

#### APPENDIX A: QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM ON PLANAR CHIPS

In this section we describe in more detail the embedding of the different QAOA strategies on digital quantum devices. This is important when we introduce noise and the number of error-prone 2-qubit gates (CNOT) matter.

Here, we consider the two approaches outlined in Secs. II A and II B: QAOA with rerouting and parity QAOA. The parity QAOA scheme described in Sec. II C introduces a modification for the optimizer and read-out and, therefore, uses the same embedding as the original version. For both methods we consider a planar chip with nearest-neighbor connectivity to be in line with modern state-of-the-art quantum devices [52–55].

The problem Hamiltonian for the rerouting strategy, Eq. (4), is described by two-body interactions. The corresponding phase separation unitary operator

$$U_p(\gamma) = e^{-i\gamma\hat{H}_p} = \prod_{i=1}^N \prod_{j<i} e^{-i\gamma J_{ij}\sigma_i^z\sigma_j^z}, \quad (\text{A1})$$

implements each interaction with two CNOT gates and a single  $R^z(\alpha)$  rotation, as shown in Fig. 5(a). The rotation angle  $\alpha$

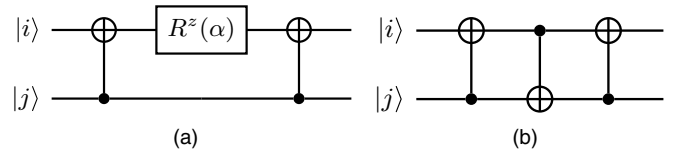


FIG. 5. Interactions between (neighboring) qubits  $i$  and  $j$  is realized with two CNOT gates and a single qubit z-Rotation, as shown in panel (a). A SWAP gate, shown in panel (b), is implemented via three CNOT gates.

is determined by the product of the variational parameter  $\gamma$  and the interaction  $J_{ij}$ :  $R_j^z(\alpha) = e^{-i\alpha\sigma_j^z}$  with  $\alpha = \gamma J_{ij}$ . To be able to realize all qubit interactions non-neighboring qubits need to be made local at least once via SWAP gates. A SWAP gate is shown in Fig. 5(b). It consists of three CNOT gates, which makes them resource intensive. As the problem graphs we study in this paper are fully connected, every qubit needs to interact with every other qubit. To minimize the usage of SWAP gates we optimize the circuit with the `t|ket>` transpiler by Cambridge Quantum Computing (CQC) [43].

In the main text, Sec. III B, we investigate a problem with  $N = 6$  qubits, results shown in Fig. 4. For a circuit of this size the `t|ket>` transpiler returns 94 CNOT gates for  $p = 2$ , arranged on a  $2 \times 3$  qubit lattice. (Note that the compiler did not return the most optimal circuit.)

In the parity architecture we transformed the qubit interactions  $J_{ij}$  to local fields  $\tilde{J}_v$ , see Eq. (5) in the main text. The local field unitary,  $\tilde{U}_z(\gamma) = \prod_{v=1}^K e^{-i\gamma J_v \tilde{\sigma}_v^z}$  consists of single qubit rotations, where the rotation angle is determined by  $\gamma$  and  $\tilde{J}_v$ . However, the constraint unitary,

$$\tilde{U}_c(\Omega) = \prod_{l=1}^L e^{-i\Omega H_{\leftrightarrow,l}} = \prod_{l=1}^L e^{-i\Omega \tilde{\sigma}_{(l,1)}^z \tilde{\sigma}_{(l,2)}^z \tilde{\sigma}_{(l,3)}^z [\tilde{\sigma}_{(l,4)}^z]}, \quad (\text{A2})$$

needs to be realized via CNOT gates. The unitary contains 3 or  $4\sigma^z$  terms, depending whether it is a three- or four-body constraint. Four-body (three-body) constraints consist of a consecutive sequence of three (two) CNOT gates, a  $R^z(\Omega)$  rotation and three (two) CNOT gates back on the same qubits. That are in total six (four) CNOT gates per four-body (three-body) constraint. This sequence is visualized in Fig. 6.

The problem studied in the main text, Sec. III B, results shown in Fig. 4, has  $K = 15$  qubits in the parity representation. In total it has six four-body and four three-body

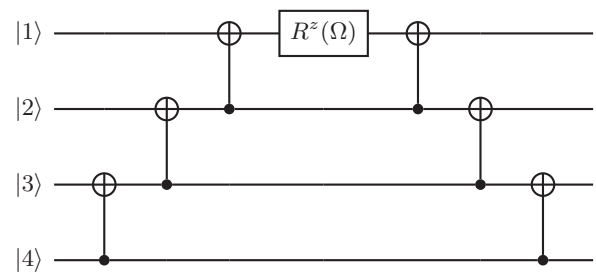


FIG. 6. Circuit to implement a four-body constraint. It consists of six CNOT gates and a single qubit  $R^z$  rotation. In case of a three-body constraint the last line, qubit four, is not required.

constraints. This sums up to 52 CNOT gates for each layer, resulting in 104 CNOT gates for  $p = 2$ .

In this example QAOA, with rerouting has the advantage of requiring fewer CNOT gates (94 vs 104 CNOT gates). However, Ref. [58] gives a more detailed comparison about the two mentioned models and used CNOT gates for different optimization problems. There, the result suggest a significant advantage of the parity encoding for bigger system sizes and higher-order interaction terms. In this work we study small system sizes and two-qubit interactions only. The described implementation of constraints in parity QAOA can be further enhanced, as shown in the recent work of Ref. [65]. This improvement was not considered in this work or in Ref. [58].

**APPENDIX B: SIMULATION DETAILS**

The QAOA simulation details are the following: To optimize the QAOA variational parameters  $\beta$  and  $\gamma$  and  $\beta, \gamma$ , and  $\Omega$  for rerouting QAOA and parity QAOA, respectively, we initialize 100 random start parameters (as proposed in Ref. [67]) followed by parameter updates. Updates are accepted with the Metropolis criterion at a constant temperature. In general, there will be 10.000 measurements for each step. We simulate 100 problem instances for the noiseless and 20 instances for the noisy simulations, whereas the best run [lowest energy  $\mathcal{C}(\beta^*, \gamma^*)$  or  $\tilde{\mathcal{C}}(\beta^*, \gamma^*, \Omega^*)$ ] out of the 100 random initialization is taken for latter statistics. For the simulations with noise the number of measurements during the optimization is set to 1000 and readjusted to 10 000 for executing the circuit with the best found parameters  $\beta^*$  and  $\gamma^*$  or  $\beta^*, \gamma^*$ , and  $\Omega^*$ . With the choice  $J_{ij} \in \{\pm 0.1, \pm 0.2, \dots, \pm 1.0\}$  the search space of parameter  $\gamma$  is restricted to  $[0, 10\pi]$ . The search space of the other parameters is limited to  $[0, \pi]$ .

The set of random spanning trees  $T = \{t_1, t_2, \dots, t_M\}$  for decoded parity QAOA are generated the following way: One starts with a random edge, containing two nodes. One of these nodes is randomly chosen to continue the tree. The next new edge is chosen randomly from the set of possible edges, excluding cycles. This procedure is repeated until all nodes are included.

**APPENDIX C: DECODED PARITY QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM**

This chapter gives a more detailed description of the decoding strategy for parity QAOA (Sec. II C). The adapted QAOA procedure is shown at the top of Fig. 7. The unitaries are applied on an initial state  $|\Psi_0\rangle = |+\rangle^K$  as described in Eq. (6). This is followed by a measure and decoding scheme, which returns a logical, and not a physical energy to the optimizer. The optimizer then feeds the adapted parameters back into the circuit.

To obtain a logical energy of a physical configuration it is necessary to decode the physical configuration  $\tilde{q}$  into a logical one  $q$ . As mentioned in Sec. II B in the main text, configurations in the parity architecture which violate the constraints, have no corresponding logical configuration. Such an unphysical state is shown as the measured configuration in the

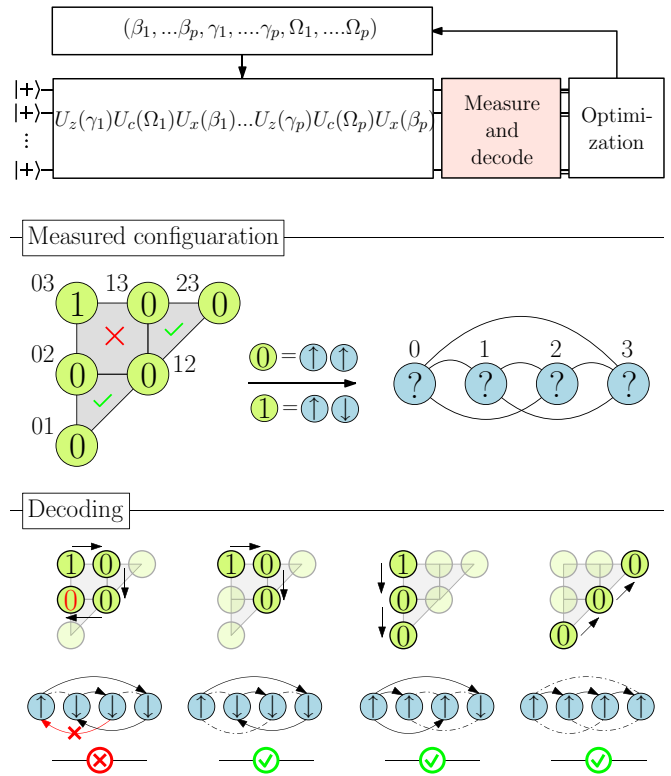


FIG. 7. QAOA on the parity architecture with decoding. On top, the circuit applies the unitaries as usual, with  $U_z(\gamma)$  and  $U_c(\Omega)$  as problem terms and  $U_x(\beta)$  as the driver term. With the measurement, physical configurations are decoded into logical ones, such that a logical energy serves as objective function to the classical optimizer. An example of a physical configuration (green qubits) is given in the middle. The configuration violates a constraint (red cross) and, therefore, has no corresponding logical state (blue qubits). Four decoding examples are shown in the bottom. In the first example the four qubits of the four-body constraint are used for decoding. With the last physical qubit one runs into a contradiction, not all parities of the logical qubits can be fulfilled. The examples shown take physical qubits for decoding that relate to a spanning tree in the logical one, marked as black arrows in the logical graph. With this subset of physical qubits it is possible to construct a valid logical state.

middle of Fig. 7 (green qubits). This configuration fulfils the three-body constraints (green check), but not the four-body constraint (red cross). Therefore, we cannot construct a logical configuration  $q$  with the usual decoding rules:  $0 \rightarrow \uparrow\uparrow$  or  $\downarrow\downarrow$  (parallel logical qubits),  $1 \rightarrow \uparrow\downarrow$  or  $\downarrow\uparrow$  (antiparallel logical qubits). To see this, consider the first decoding example on the bottom left of Fig. 7: The decoding starts at the top-left qubit 03. It has the value 1 and therefore the logical qubits 0 and 3 have to be antiparallel. We can set 0 to up and 3 to down. The next two physical qubits, 13 and 12, give the information that qubit 1 is parallel to qubit 3 and qubit 2 is parallel to 1. We can set the qubits 1 and 2 to down. The fourth and last physical qubit, 02, has the value 0, this means qubit 0 and 2 should be parallel, but they are already antiparallel. There is no valid logical configuration that fulfills all parity qubits. Before considering the last physical qubits (02) all the logical qubits were determined. This is the

second example in the figure. Here, three of the four previous qubits are taken for the decoding. The three physical qubits correspond to the interactions, marked as arrows, in the logical graph. Those interactions cover all the qubits without making a cycle (the previous example made a cycle). This is the definition of a spanning tree in a graph. The last two examples show different spanning trees and their corresponding decoded state. Note that they do not return the same logical state because the corresponding physical configuration violates a constraint.

#### Added resource costs

The error mitigation technique presented in this work has no measurement overhead but requires additional resources at the classical optimization level. We describe its additional computational costs in this section.

To mitigate the errors, after each measurement, we decode a physical configuration  $\tilde{q}$  with  $M$  spanning trees to obtain the  $M$  logical configurations,  $\tilde{q} \rightarrow \{q^{(t_1)}, q^{(t_2)}, \dots, q^{(t_M)}\}$ , which we then use to compute the cost function with Eq. (9). This procedure introduces a classical overhead proportional to the number  $M$  of spanning trees considered. In particular, considering all the  $N^{N-2}$  spanning trees of a complete graph with  $N$  nodes would lead to an exponential overhead. We can avoid introducing exponential overheads by considering a subset of  $M = N$  spanning trees. The results presented in Figs. 2 and 4 of the main text suggest that such a linear overhead scaling for  $M = N$  is sufficient to considerably enhance the quantum optimization algorithm's performance. Furthermore, Fig. 2(b) shows that, for  $N = 7$ , taking  $M > N$  could further increase the algorithm's success probability. However, to identify the optimal number of spanning trees, one should carefully consider the trade-off between performance and overhead because further increasing  $M$  would also lead to larger computational overheads.

For complete graphs, the parity embedding also introduces a quadratic qubit overhead over a direct implementation with rerouting. To account for this overhead, we simulated running the rerouted algorithm on  $K/N$  copies. However, various real-world optimization problems instead require implementing sparse graphs or hypergraphs (with  $k$ -body interactions with  $k > 2$ ), where the parity embedding is more efficient than a direct implementation in terms of CNOT gate count [58]. Moreover, the parity embedding enables a straightforward parallelization of the gates' execution, leading to shorter variational circuits [34].

#### APPENDIX D: COMPARING DIFFERENT STRATEGIES

In this section we use a different measure for the success probability  $P_S$  than outlined in Sec. II C.

In the main text, after measuring the optimal variational state  $|\tilde{\Psi}(\beta^*, \gamma^*, \Omega^*)\rangle$  we returned the best decoded state  $q^{(T)}$  defined in Eq. (10) in the main text. The argument is that, for the measurement in the very end, one can keep the decoded state with the lowest energy and discard the others. However, in this section, we want to keep all the decoded states and consider the probability to measure the ground state within them. In this case, this is Eq. (10) in the main text. Instead,

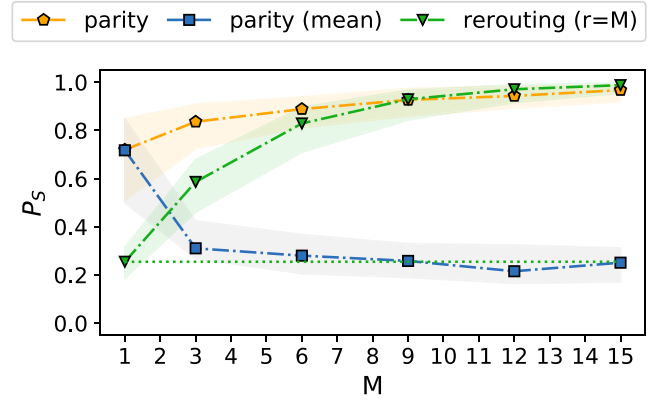


FIG. 8. Success probability  $P_S$  in dependence of the number of spanning trees  $M$  used,  $N = 6$ ,  $p = 2$ . When creating the variational state  $|\tilde{\Psi}(\beta^*, \gamma^*, \Omega^*)\rangle$  with the best found parameters, parity QAOA keeps the best decoded logical state per physical state while w-parity QAOA keeps all decoded logical states to determine the success probability  $P_S$ . For the rerouting strategy the results for  $P_S$  are adapted with Eq. (11) (main text) with  $r = M$ .

the output distribution is described by the quantum density matrix  $D_T[\tilde{\rho}(\beta^*, \gamma^*, \Omega^*)]$ , where  $D_T$  is the linear decoding map  $D_T[\tilde{\rho}] = \frac{1}{M}(D_{t_1}[\tilde{\rho}] + D_{t_2}[\tilde{\rho}] + \dots + D_{t_M}[\tilde{\rho}])$ . The definitions of the ground-state probability (11) and the success probability Eq. (12) do not change, both in the main text. We compare parity QAOA as described in the main text with the here defined “mean parity” QAOA method. In both cases we set  $r = 1$  in Eq. (12).

In addition, we also look at the rerouting method, setting the number of repetitions  $r = M$  of Eq. (12), compared with  $r = N/K$  in the main text.

In Fig. 8 we can see the results for the success probability  $P_S$  for parity, mean parity, and rerouting for  $N = 6$ . For  $M = 1$  both parity methods return the same  $P_S$ . For increasing  $M$  the probability for mean parity drops and converges to a value of  $P_S \approx 0.3$ . Similar  $P_S$  is achieved by the rerouting method, with a single copy (green dashed line).

According to the results, the rerouting approach needs to be repeated at least  $M = 9$  times to reach the same or more  $P_S$  than parity QAOA.

#### APPENDIX E: PARITY QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM+

There are several adaptations to the simple  $p$ -level QAOA method described in the main text in Sec. II A [38,42,68–75]. For instance, the quantum alternation operator ansatz [38] (QAOA+) provides a more expressive variational state by modifying the definitions of  $U_x(\beta)$  and  $U_p(\gamma)$ . In Ref. [50] the authors propose an approach for parity QAOA which is based on QAOA+. Here, one starts with constraint fulfilling states and the driver Hamiltonian  $\tilde{H}_x^{(\text{mod})}$  is adapted such that it only allows transitions between constraint fulfilling states (in an ideal circuit). No parity constraints required. This gives the advantage of increased performance, but with the drawback of nonparallelizable gates. With a hybrid approach, where some parity constraints are enforced implicit with the driver Hamiltonian and the others are applied explicit with parity



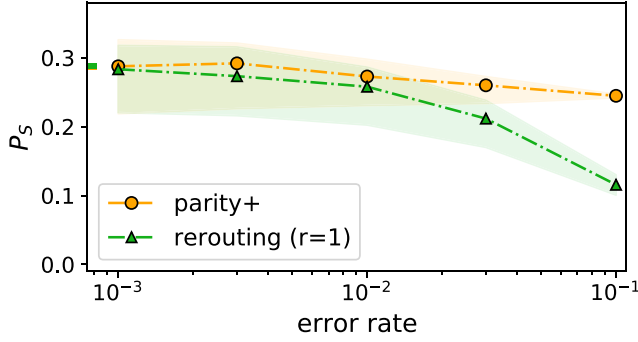


FIG. 9. Noise on parity QAOA+ and QAOA with rerouting,  $N = 5$ ,  $p = 1$ . The 1-qubit gate error rate is fixed to 0.001.

constraints, one obtains a trade-off between performance and parallelizability of the gates. Nevertheless, here we want to focus on the first. We use a driver unitary  $\tilde{U}_x^{(\text{mod})}(\beta) = e^{-\beta \tilde{H}_x^{(\text{mod})}}$  such that the states during the parity QAOA procedure stay in the constraint fulfilling subspace. As in the main text, we study QUBO problems with all-to-all connectivity, here with  $N = 5$ . In Sec. III A 2 we defined a set of spanning trees that correspond to the logical lines in the parity architecture, namely,  $T^{(l)} = \{t_1^{(l)}, t_2^{(l)}, \dots, t_N^{(l)}\}$  with  $t_i^{(l)}$ , including all parity qubits containing the logical qubit  $i$ . Now one can define a driver term with

$$X_i = \prod_{k \in t_i^{(l)}} \sigma_x^{(k)}. \quad (\text{E1})$$

The new driver Hamiltonian is defined as the sum over the driver terms  $\tilde{H}_x^{\text{mod}} = \sum_{i=1}^N X_i$ . With this Hamiltonian, parity QAOA+ and QAOA with rerouting are mathematically equivalent. Here, we consider single copies in both cases, hence  $r = 1$  in Eq. (12) of the main text. Simulations on an ideal circuit yield the same output. This can be seen in Fig. 9, where the results for the success probability  $P_S$  for  $N = 5$  and  $p = 1$  are shown. The solid bars on the right represent the results with no noise. As soon as a depolarizing 2-qubit gate error is applied, the simulation results between the different approaches differ. Here, parity + needs more CNOT gates than the rerouting layout. Nevertheless, applying the decoding to parity + shows better noise stability.

#### APPENDIX F: NUMBER OF PARAMETERS AND SPANNING TREES

Figure 10 (left) shows the success probability in dependence of the circuit layers  $p$ . Nevertheless, one needs to keep in mind that the different QAOA approaches have different numbers of parameters associated with each depth. The variational state created by the rerouting method, Eq. (2) of the main text, depends on  $2p$  parameters  $\beta$  and  $\gamma$ . In contrast, the parity circuit creates the variational state, Eq. (6), with  $3p$  parameters  $\beta$ ,  $\gamma$ , and  $\Omega$ . This means that the rerouting strategy has less parameters to optimize, e.g., one can perform QAOA with depth  $p = 3$  and one has as many parameters to optimize as with parity QAOA with depth  $p = 2$ . This is

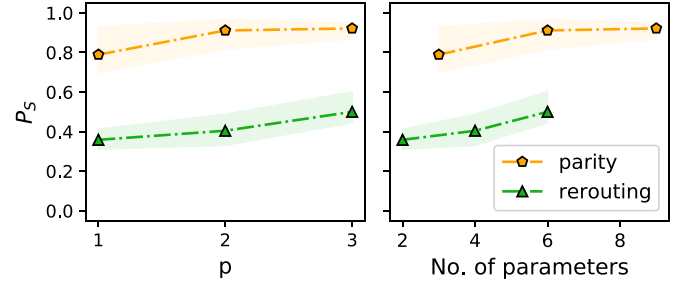


FIG. 10. Success probability  $P_S$  in dependence of the circuit depth  $p$  (left) and in dependence of the total number of parameters (right). The rerouting method needs to optimize two parameters per cycle ( $\beta$  and  $\gamma$ ), the parity method 3 ( $\beta$ ,  $\gamma$ , and  $\Omega$ ). The number of parameter updates is 100, 500, 1100 for  $p = 1, 2, 3$ , respectively. The number of measurements is set to 1000 during optimization and to 10.000 for the final measurement.

shown in Fig. 10 (right). We can see that the rerouting method with  $p = 3$  (and 1100 parameter updates) still cannot reach the same success probability as the parity method with  $p = 2$  (500 parameter updates).

In the main text, Sec. III A 1, we looked at the performance of parity QAOA in dependence of the number of spanning trees  $M$  used for decoding. Here we want to look at the 100 instances individually and, in addition, compare it to the rerouting method, which is independent of  $M$ . Figure 11 shows the success probabilities of the two different methods for different  $M$ . In Fig. 11(a) we see the results for  $M = 1$ . The results for parity QAOA ( $P_{\text{parity}}$ ) are distributed over the whole interval  $[0,1]$ , meaning that the outcome is unpredictable. On the other hand, QAOA with rerouting returns stable

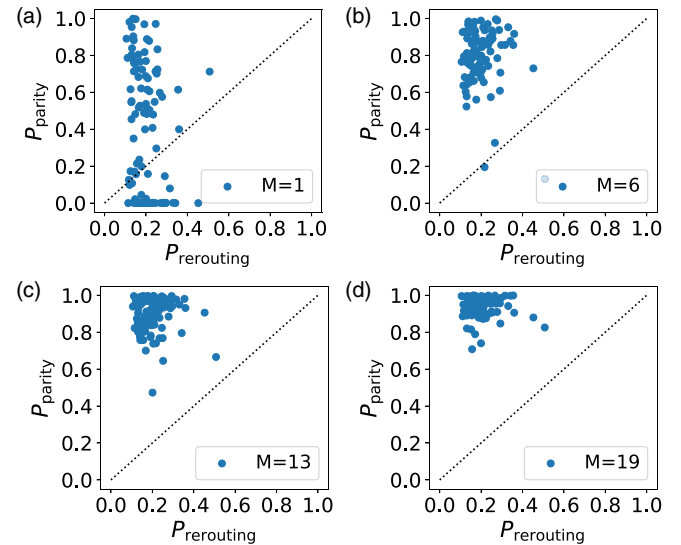


FIG. 11. Success probabilities of the rerouting ( $P_{\text{rerouting}}$ ) and parity strategy ( $P_{\text{parity}}$ ) for different amount  $M$  of random spanning trees,  $N = 6$ ,  $p = 1$ . One tree for decoding,  $M = 1$  returns unpredictable success rates for parity QAOA, as shown in panel (a), while QAOA with rerouting gives a stable performance. With increasing  $M$ ,  $M = 6$  in panel (b),  $M = 13$  in panel (c), and  $M = 19$  in panel (d), this problem for parity QAOA will disappear.

results. This disadvantage for parity QAOA will disappear with increasing  $M$ , meaning that it will be necessary to include

enough spanning trees in the decoding. Figures 11(b)–11(d) shows the outcome of  $M = 6, 13, 19$ , respectively.

- 
- [1] T. Albash and D. A. Lidar, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [2] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, *Phys. Rev. X* **10**, 021067 (2020).
- [3] N. Moll *et al.*, *Quantum Sci. Technol.* **3**, 030503 (2018).
- [4] G. G. Guerreschi and A. Y. Matsuura, *Sci. Rep.* **9**, 6903 (2019).
- [5] M. P. Harrigan *et al.*, *Nat. Phys.* **17**, 332 (2021).
- [6] J. Preskill, *Quantum* **2**, 79 (2018).
- [7] D. Stilck França and R. García-Patrón, *Nat. Phys.* **17**, 1221 (2021).
- [8] K. Bharti *et al.*, *Rev. Mod. Phys.* **94**, 015004 (2022).
- [9] M. Cerezo *et al.*, *Nat. Rev. Phys.* **3**, 625 (2021).
- [10] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New J. Phys.* **18**, 023023 (2016).
- [11] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature (London)* **549**, 242 (2017).
- [12] F. Arute *et al.*, *Science* **369**, 1084 (2020).
- [13] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature (London)* **549**, 195 (2017).
- [14] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, *Phys. Rev. A* **98**, 032309 (2018).
- [15] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [16] G. Pagano *et al.*, *Proc. Natl. Acad. Sci. USA* **117**, 25396 (2020).
- [17] N. Lacroix *et al.*, *PRX Quantum* **1**, 020304 (2020).
- [18] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [19] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O’Brien, Quantum error mitigation, [arXiv:2210.00921](https://arxiv.org/abs/2210.00921).
- [20] D. Gottesman, *Quantum Information Science and Its Contributions to Mathematics, Proceedings of Symposia in Applied Mathematics* (American Mathematical Society, Providence, Rhode Island, 2010).
- [21] R. Raussendorf and J. Harrington, *Phys. Rev. Lett.* **98**, 190504 (2007).
- [22] Y. Li and S. C. Benjamin, *Phys. Rev. X* **7**, 021050 (2017).
- [23] K. Temme, S. Bravyi, and J. M. Gambetta, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [24] S. Endo, S. C. Benjamin, and Y. Li, *Phys. Rev. X* **8**, 031027 (2018).
- [25] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, *Phys. Rev. A* **95**, 042308 (2017).
- [26] J. R. McClean, Z. Jiang, N. C. Rubin, R. Babbush, and H. Neven, *Nat. Commun.* **11**, 636 (2020).
- [27] M. Otten and S. K. Gray, *npj Quantum Inf.* **5**, 11 (2019).
- [28] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O’Brien, *Phys. Rev. A* **98**, 062339 (2018).
- [29] P. Czarnik, A. Arrasmith, L. Cincio, and P. J. Coles, Qubit-efficient exponential suppression of errors, [arXiv:2102.06056](https://arxiv.org/abs/2102.06056).
- [30] S. McArdle, X. Yuan, and S. Benjamin, *Phys. Rev. Lett.* **122**, 180501 (2019).
- [31] L. Botelho, A. Glos, A. Kundu, J. A. Mischak, O. Salehi, and Z. Zimborás, *Phys. Rev. A* **105**, 022441 (2022).
- [32] Z. Cai, *npj Quantum Inf.* **7**, 80 (2021).
- [33] W. Lechner, P. Hauke, and P. Zoller, *Sci. Adv.* **1**, e1500838 (2015).
- [34] W. Lechner, *IEEE Trans. Quantum Eng.* **1**, 3102206 (2020).
- [35] M. Fellner, A. Messinger, K. Ender, and W. Lechner, *Phys. Rev. Lett.* **129**, 180503 (2022).
- [36] F. Pastawski and J. Preskill, *Phys. Rev. A* **93**, 052325 (2016).
- [37] A. Rocchetto, S. C. Benjamin, and Y. Li, *Sci. Adv.* **2**, e1601246 (2016).
- [38] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, *Algorithms* **12**, 34 (2019).
- [39] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes, [arXiv:1906.08948](https://arxiv.org/abs/1906.08948).
- [40] A. Messiah, *Quantum Mechanics* (Dover Publications, Mineola, New York 2014).
- [41] Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima, *Quantum Info. Comput.* **11**, 142 (2011).
- [42] B. Bakó, A. Glos, O. Salehi, and Z. Zimborás, Near-optimal circuit design for variational quantum optimization, [arXiv:2209.03386](https://arxiv.org/abs/2209.03386).
- [43] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, *Quantum Sci. Technol.* **6**, 014003 (2021).
- [44] M. Alam, A. Ash-Saki, and S. Ghosh, in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE, Piscataway, NJ, 2020), pp. 215–228.
- [45] L. Lao, H. van Someren, I. Ashraf, and C. G. Almudever, *IEEE Trans. Comput. Aid. D.* **41**, 359 (2022).
- [46] G. Li, Y. Ding, and Y. Xie, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’19* (Association for Computing Machinery, New York, 2019), pp. 1001–1014.
- [47] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS ’19* (Association for Computing Machinery, New York, 2019), pp. 1015–1029.
- [48] B. Tan and J. Cong, in *2020 IEEE/ACM International Conference on Computer Aided Design (ICCAD)* (IEEE, Piscataway, NJ, 2020), pp. 1–9.
- [49] K. Ender, R. ter Hoeven, B. E. Niehoff, M. Drieb-Schön, and W. Lechner, Parity quantum optimization: Compiler, *Quantum* **7**, 950 (2023).
- [50] K. Ender, A. Messinger, M. Fellner, C. Dłaska, and W. Lechner, *PRX Quantum* **3**, 030304 (2022).
- [51] C. Dłaska, K. Ender, G. B. Mbeng, A. Kruckenhauser, W. Lechner, and R. van Bijnen, *Phys. Rev. Lett.* **128**, 120503 (2022).
- [52] F. Arute *et al.*, *Nature (London)* **574**, 505 (2019).
- [53] L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, *Quantum* **4**, 327 (2020).
- [54] M. Saffman, T. G. Walker, and K. Mølmer, *Rev. Mod. Phys.* **82**, 2313 (2010).
- [55] I. Bloch, J. Dalibard, and W. Zwerger, *Rev. Mod. Phys.* **80**, 885 (2008).

- [56] A. Lucas, *Front. Phys.* **2**, 5 (2014).
- [57] M. Lanthaler and W. Lechner, *New J. Phys.* **23**, 083039 (2021).
- [58] M. Fellner, K. Ender, R. ter Hoeven, and W. Lechner, Parity quantum optimization: Benchmarks, *Quantum* **7**, 952 (2023).
- [59] M. Fellner, Master's thesis, Institute for Theoretical Physics, 2020, (unpublished).
- [60] T. Albash, W. Vinci, and D. A. Lidar, *Phys. Rev. A* **94**, 022327 (2016).
- [61] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge 2010).
- [62] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, *Quantum* **4**, 256 (2020).
- [63] I. Kolotouros and P. Wallden, *Phys. Rev. Res.* **4**, 023225 (2022).
- [64] M. S. Anis *et al.* Qiskit/qiskit: Qiskit 0.39.4 (2022).
- [65] J. Unger, A. Messinger, B. E. Niehoff, M. Fellner, and W. Lechner, Low-depth circuit implementation of parity constraints for quantum optimization, [arXiv:2211.11287](https://arxiv.org/abs/2211.11287).
- [66] <https://gitlab.com/AnitaW/qaqa-error-mitigation>.
- [67] R. Shaydulin, I. Safro, and J. Larson, in *2019 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, Piscataway, NJ, 2019), pp. 1–8.
- [68] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, *Phys. Rev. A* **101**, 012320 (2020).
- [69] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, *Sci. Rep.* **12**, 6781 (2022).
- [70] Y. Yu, C. Cao, C. Dewey, X.-B. Wang, N. Shannon, and R. Joynt, *Phys. Rev. Res.* **4**, 023249 (2022).
- [71] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, *Phys. Rev. Lett.* **125**, 260505 (2020).
- [72] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, *Quantum* **6**, 678 (2022).
- [73] D. J. Egger, J. Mareček, and S. Woerner, *Quantum* **5**, 479 (2021).
- [74] L. C. G. Gavia, C. Poole, M. Saffman, and H. K. Krovi, *Phys. Rev. A* **104**, 062428 (2021).
- [75] L. Zhu, H. L. Tang, G. S. Barron, F. A. Calderon-Vargas, N. J. Mayhall, E. Barnes, and S. E. Economou, *Phys. Rev. Res.* **4**, 033029 (2022).