Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm

Michael Streif,[1,2] Sheir Yarkoni,[1,3] Andrea Skolik,[1,3] Florian Neukart,[1,3] and Martin Leib[1]
[1]*Data:Lab, Volkswagen Group, Ungererstrasse 69, 80805 München, Germany*
[2]*University Erlangen-Nürnberg (FAU), Institute of Theoretical Physics, Staudtstrasse 7, 91058 Erlangen, Germany*
[3]*LIACS, Leiden University, Snellius Gebouw, Niels Bohrweg 1, 2333 CA Leiden, Netherlands*

(Received 17 November 2020; revised 3 June 2021; accepted 15 June 2021; published 2 July 2021)

The binary paint shop problem (BPSP) is an APX-hard optimization problem of the automotive industry. In this work, we show how to use the quantum approximate optimization algorithm (QAOA) to find solutions of the BPSP. We demonstrate that QAOA with constant depth is able to beat all known heuristics for the binary paint shop problem on average in the infinite size limit $n \to \infty$. We complete our studies by running first experiments of small-sized instances on a trapped-ion quantum computer through Amazon Braket.

DOI: 10.1103/PhysRevA.104.012403

I. INTRODUCTION

Achievements in fabrication and control of two-level systems led to the first quantum computers with tens of qubits [1–4] and recently culminated in the demonstration of a quantum computer solving a computational task that is intractable for classical computers, also known as *quantum supremacy* [5]. This milestone raises expectations that quantum computing some day will accelerate research, speed up simulations in chemistry, and improve optimization processes in many branches of industry. Quantum algorithms with a proven scaling advantage over classical algorithms, such as Grover's [6] or Shor's [7] algorithm, require fault-tolerant quantum computers. However, the devices that will become available in the next 5–10 years will only have a limited number of qubits and will not feature error-correction. It is unclear if such noisy intermediate-scale quantum (NISQ) devices can be useful in solving real-world problems faster than classical computers or if larger error-corrected devices will be needed. To answer this question, it is especially important to develop quantum algorithms that are tailor-made for the characteristics of quantum processing units (QPU). A promising class of NISQ algorithms is the class of variational quantum algorithms, which are parametrized *Ansätze* optimized with classical learning loops. There exist various ideas on how to tailor the *Ansatz* for different tasks, such as the variational auantum eigensolver (VQE) for chemistry applications [8], or quantum neural networks (QNN) for machine learning [9,10]. The QAOA is a variational algorithm designed to solve classical optimization problems [11] and was applied to problems such as Max-Cut [11] or Max-3-Lin-2 [12]. Furthermore, there exist first insights on QAOAs performance [13–16], first experimental realizations on different quantum processors [17–19], and several proposals on how to further improve QAOA [20–26]. For example, in [26] it was shown that for some problem classes with certain topological characteristics, it is possible to find good parameters for QAOA with classical methods efficiently. Moreover, there exist results showing that it is classically hard to sample from the QAOA output [27] and that QAOA possesses a Grover-type speed-up [28]. However, performance bounds are only known for very short circuits [11] or classically easy instances [21]. Establishing scaling comparisons, beyond low depth circuits, between QAOA and classical methods for application relevant optimization problems is the next important step to find useful NISQ applications of QAOA.

In this present contribution, we take a step in this direction using the example of a combinatorial optimization problem from the automotive industry, i.e., the *binary paint shop problem* (BPSP). We show that the problem can be encoded into a spin glass in a constraint-free way and requires only a linear number of qubits for increasing problem size. As such, it is a perfect fit for QAOA on NISQ devices. We show that the problem has a fixed degree and coupling strength, which allows us to use the method developed in [26], which bypasses the NP-hard training procedure [29]. We present numerical results and run first small-scale experiments on a trapped ion quantum computer. We are numerically able to prove a minimal depth for QAOA to beat classical heuristics in the small to medium system-size limit (up to 100 cars) as well as for the infinite-size limit. With this we show that a constant time quantum algorithm can beat a polynomial time classical algorithm.

This paper is structured as follows. In Sec. II we review the quantum approximate optimization algorithm (QAOA). In Sec. III we review the binary paint shop problem, classical greedy algorithms to solve it, and we discuss the mapping of the problem onto an Ising Hamiltonian. In Sec. IV we show results of QAOA applied to the BPSP, and in Sec. IV we conclude.

II. REVIEW OF QAOA

In this section, we review the quantum approximate optimization algorithm (QAOA) [11], a variational quantum

algorithm designed to solve combinatorial optimization problems.

To solve a combinatorial optimization problem with QAOA, the first step is to reformulate it as a spin-glass problem. For our purposes, the spin glass can be represented as a problem graph $G = (V, E)$ with nodes $v \in V$ representing spins s_i and edges $e \in E$ representing the terms of the sum of the energy of the spin glass $E_P = \sum_{(i,j)\in E} J_{i,j} s_i s_j$ that needs to be minimized. We note that finding the optimal solution of a spin glass is NP-hard [30], thus there exist mappings with at most polynomial overhead from all NP problems to such a spin system, some of them shown in [31]. We search for low-energy configurations of the spin glass with a variational *Ansatz*. In QAOA, this is done by minimizing the expectation value of the problem Hamiltonian H_P with respect to a parametrized *Ansatz* state $|\Psi(\{\beta_l, \gamma_l\})\rangle$,

$$\min_{\{\beta_l,\gamma_l\}} \langle\Psi(\{\beta_l, \gamma_l\})|H_P|\Psi(\{\beta_l, \gamma_l\})\rangle. \tag{1}$$

Therefore, we translate the spin system to its quantum version,

$$\mathrm{H_P} = \frac{1}{2} \sum_{(i,j)\in E} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)}, \tag{2}$$

where each classical spin variable i is replaced by a qubit i with the Pauli-Z operator $\sigma_z^{(i)}$. The *Ansatz* state in QAOA is inspired by quantum annealing techniques and is generated by the repeated application of the mixing and problem unitary, $U_M(\beta_l) = e^{-i\beta_l H_M}$ and $U_P(\gamma_l) = e^{-i\gamma_l H_P}$, on the superposition state of all computational basis states, $|+\rangle^{\otimes n} = \bigotimes_i^n 1/\sqrt{2}(|0\rangle_i + |1\rangle_i)$. The generators of these unitaries are given by the mixing Hamiltonian, $H_M = \sum_i \sigma_x^{(i)}$, and the problem Hamiltonian, see Eq. (2). The full *Ansatz* state,

$$|\Psi(\{\beta_l, \gamma_l\})\rangle_{\mathrm{QAOA}} = \prod_l^p U_X(\beta_l) U_P(\gamma_l) |+\rangle^{\otimes n}, \tag{3}$$

includes p repetition of those unitaries, where each repetition is called a QAOA level. To find the optimal variational parameters $\{\beta_l^*, \gamma_l^*\}_{l=1}^p$, a quantum computer is used to estimate the expectation value of the problem Hamiltonian, while an outer learning loop on a classical computer updates the parameters to minimize the expectation value. Recent work has shown how to speed up the classical learning loop [20] or alternatively using entirely classical methods to find optimal parameters for certain problem classes [26]. Having found the optimal parameters, one then samples from the final state $|\Psi(\{\beta_l^*, \gamma_l^*\})\rangle$ in the computational basis, which yields solutions to the optimization problem.

III. THE BINARY PAINT SHOP PROBLEM

Assume an automotive paint shop and a random but fixed sequence of n cars. The task is to paint the cars in the order given by the sequence. Each individual car needs to be painted with two colors, once per color, in a to-be-determined color order. In other words, each car appears twice at random, uncorrelated positions in the sequence, and we are free to choose the color to paint the car first. A specific choice of first colors for every car is called a *coloring*. The objective of the optimization problem is to find a coloring that minimizes the number of color changes between adjacent cars in the sequence. This combinatorial optimization problem is called the *binary paint shop problem* (BPSP) [32–34]. In Fig. 1, we show a binary paint shop instance together with a suboptimal solution and the optimal solution. A formal definition of the binary paint shop problem is given in Definition III.I.

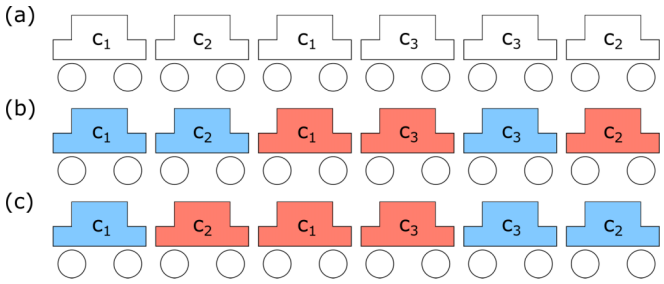

FIG. 1. (a) A binary paint shop instance with $n = 3$ cars $\{c_1, c_2, c_3\}$. (b) A valid but suboptimal coloring with $\Delta_C = 3$ color changes. (c) An optimal coloring that only requires $\Delta_C = 2$ color changes to paint the sequence.

Definition III.I (Binary paint shop problem). Let Ω be the set of n cars $\{c_1, \ldots, c_n\}$. An instance of the *binary paint shop problem* is given by a sequence $(w_1, \ldots, w_{2n})$ with $w_i \in \Omega$, where each car c_i appears exactly twice. We are given two colors $\mathcal{C} = \{1, 2\}$. A coloring is a sequence $f = f_1, \ldots, f_{2n}$ with $f_i \in \mathcal{C}$ and $f_i \neq f_j$ if $w_i = w_j$ for $i \neq j$. The objective is to minimize the number of color changes, $\Delta_C = \sum_i |f_i - f_{i+1}|$.

The binary paint shop problem belongs to the class of NP-hard optimization problems, thus there is no polynomial-time algorithm that finds the optimal solution for all problem instances. For many optimization problems in practice, rather than spending exponential time to find the optimal solution, fast approximate algorithms are used. However, the binary paint shop problem is proven to be APX-hard [33], i.e., it is as difficult to approximate as every problem in APX. Additionally, if the unique games conjecture (UGC) [35] holds, it is even not in APX, and thus there would not be a constant factor approximation algorithm for any α [34]. A constant factor approximation algorithm would be a polynomial-time algorithm that returns an approximate solution with at most αOPT color changes, where OPT is the optimal number of color changes. This is a key difference from previous problem classes to which QAOA has been applied, such as Max-Cut, where constant factor approximation algorithms are known [36].

Several greedy algorithms exist for the binary paint shop problem, which provide solutions with color changes linear in the number of cars n on average [32,33]. The greedy algorithm introduced in [32] starts at the first car w_1 of the sequence with one of both colors, goes through the sequence, and changes colors when necessary, i.e., only if the same car would be painted twice with the same color; see Fig. 2 for a pseudo code of this algorithm. For $n \to \infty$ cars, this greedy algorithm finds a solution with an average number of color changes $\mathbb{E}_G(\Delta_C) = n/2$ [37]. In Appendix A, we review two other greedy algorithms, the red first algorithm and the recursive greedy algorithm, yielding $\mathbb{E}_{RF}(\Delta_C) = 2n/3$ and $\mathbb{E}_{RG}(\Delta_C) = 2n/5$ color changes on average, respectively [37]. Numerical

Algorithm 1 Greedy algorithm

Input: a sequence $(w_1 \dots w_{2n})$, two colors $\mathcal{C} = \{1, 2\}$
Output: a coloring f

function GREEDY(w)
  first color of each car c_i: $\mathrm{FC}(c_i) = \mathrm{None}\ \forall i$
  Choose one of both colors: $c = 1, 2$

  for $k \leftarrow 1$ to $length(w)$ **do**
    $car \leftarrow w_k$
    if $\mathrm{FC}(car) \neq \mathrm{None}$ **then**
      $f_k = c$
      $\mathrm{FC}(car) = c$
    else
      $c = (c + 1) \mod 2$
      $f_k = c$
    end if
  end for
  return coloring sequence f
end function

FIG. 2. Pseudo code of the greedy algorithm to solve a binary paint shop instance.

Algorithm 2 Mapping of the binary paint shop problem onto a spin glass

Input: a sequence representing a BPSP instance, cf. Def. III.1
Output: an Ising Hamiltonian H_P

function MAPPING(w)
  $H_\mathrm{P} = 0$
  associate car c_i with qubit i
  $\#c_i = 0\ \forall i$
  for $k \leftarrow 1$ to $length(w)$ **do**
    $car_1, car_2 \leftarrow w_k, w_{k+1}$
    $H_\mathrm{P} \leftarrow H_\mathrm{P} + (-1)^{\#car_1 + \#car_2 + 1} \sigma_z^{(car_1)} \sigma_z^{(car_2)}$
    $\#car_1 \leftarrow \#car_1 + 1$
  end for
  return $\mathrm{H_C}$
end function

FIG. 3. Pseudo code for mapping a binary paint shop instance with n cars to an Ising Hamiltonian with n qubits.

results, however, suggest that the average number of optimal color changes is sublinear in the number of cars n [38]. Moreover, for some instances, the greedy algorithms only find solutions far from the optimal solution [37]. For example, for the instance shown in Fig. 1(a), the greedy algorithm finds the solution given in (b) rather than the optimal solution shown in (c).

More general versions of the binary paint shop problem can be found in practice. Typically the color set is augmented to contain more than two colors, and identical cars appear more than twice per word. These conditions correspond to the real-world application of painting thousands of car bodies per day, with numerous colors. Even restricting the color set to two colors has real-world relevance: before painting the final color of the body, each car is first painted with an undercoat called a *filler* coat. The filler colors are restricted to white and black, depending on the final car body color, and optimizing the number of color switches within the filler queue yields production cost savings. Given that the generalized paint shop problem is NP-hard in both the number of cars and colors [33], and that the binary color set is already industrially relevant, we restrict the color set to two colors in this study.

A. Reformulating the BPSP as a spin glass

In this section, we explain how to map the binary paint shop problem to a problem Hamiltonian as in Eq. (2).

We start by assigning a single qubit i to each car c_i in the sequence. The eigenstates of the σ_z-operator of each qubit indicate in which color each car is painted first. The second color choice for each car in the sequence is then unambiguously determined. To penalize color changes in the coloring, we use the coupling strengths J_{ij} between the qubits. We start at the first car w_1 in the sequence and step through the sequence adding couplings between the qubits representing the car w_k and its next neighbor w_{k+1} in the sequence. If both cars c_i and c_j, represented by w_k and w_{k+1}, respectively, appear both for the first or second time, a ferromagnetic coupling, $J_{ij} = -1$, is added. This ensures that consecutive cars favor being painted with the same color. If either car has already appeared in the sequence while the other has not, we instead add an antiferromagnetic coupling, $J_{ij} = 1$. We know that a solution with Δ_C color changes is separated by an energy $\Delta E = 1$ from a solution with $\Delta_C + 1$ color changes. We show the pseudo code of this mapping in Fig. 3. We note that the encoding of the problem does not include any constraints, thus all computational states embody valid solutions to the problem. Moreover, the encoding only requires n qubits for n cars, making it a better fit for NISQ devices than typical scheduling problems (like the traveling salesman problem) where the number of qubits required grows quadratically with the system size [31]. From the BPSP construction, we also know that a solution with Δ_C color changes is separated by an energy $\Delta E = 1$ from a solution with $\Delta_C + 1$ color changes.

1. Properties of the Ising Hamiltonian

In [26] it was shown how to calculate close-to-optimal QAOA parameters using a classical computer $\{\beta_i^{\mathrm{tree}}, \gamma_i^{\mathrm{tree}}\}$ for various levels p, and problem classes represented by graphs with a fixed degree and uniform coupling strengths, $J_{ij} =$ const. For the NISQ era, where typically p is small, this method circumvents optimizing the variational parameters $\{\beta_i, \gamma_i\}$ of the QAOA *Ansatz* state for each instance independently, and thus reduces the total QPU time used. In this section, we show that the binary paint shop instances represent graphs of fixed and average degree 4 and coupling strengths $J_{ij} = \pm 1$ (both for $N \to \infty$). As this method originally was proposed for the case $J_{ij} = J$ only, we prove in Appendix C that the method also works if $|J_{ij}| =$ const. In the following, we argue that the BPSP is a perfect fit for parameters calculated using this method.

a. Average degree. In the construction of the problem Hamiltonian, cf. Sec. III A, we add an interaction between two qubits if the corresponding cars are adjacent. As each car only appears twice, each car has at most four distinct neighbors.

It follows that the nodes in the graph G representing the spin system also have a maximal degree of 4. The degree is only smaller than 4 for the node representing the first or the last car in the sequence, or if the car is adjacent to the same car twice. In Fig. 7(a), we show that the average degree of the graph is converging to 4 from below and becomes effectively 4-regular when $n \to \infty$.

b. Coupling strengths J_{ij}. From the construction of the Ising Hamiltonian, we also know that the interaction values J_{ij} are integers and given by $\{-2, -1, 1\}$. In Appendix B, we show that the distribution of interaction values $P(J_{ij})$ converges to a distribution with $P(J_{ij} = -1) = \frac{2}{3}$ and $P(J_{ij} = 1) = \frac{1}{3}$, when $n \to \infty$. This means that the probability of a ferromagnetic coupling is twice as big as for an antiferromagnetic coupling, and that coupling strengths $|J_{ij}| = 2$ are suppressed in the infinite-size limit. We note, however, that the coupling strengths J_{ij} are not independent from each other.

IV. SOLVING THE BINARY PAINT SHOP PROBLEM WITH QAOA

In the following sections, we apply QAOA to the binary paint shop problem. For all simulations and experiments, we use the parameters $\{\beta_i^{\text{tree}}, \gamma_i^{\text{tree}}\}$ found with the method from Ref. [26], shown in Table II.

A. Numerical results

In this section, we numerically analyze the performance of QAOA on 100 randomly generated binary paint shop instances of sizes from $n = 5$ to 100 cars in increments of five cars [39].

For up to $n = 20$ cars, we calculate the QAOA output state, Eq. (3), for $p \in \{1, 2, 3, 4, 5\}$ levels of QAOA, and we determine the energy expectation value Eq. (1) together with the expected number of color changes Δ_C. For larger systems, the calculation of the QAOA output state is out of reach using a standard desktop computer.

However, since we are only interested in the energy expectation value and not in the output state, we use a proxy to calculate the expectation value for small values of p. First, we rewrite Eq. (1) as a sum over all expectation values of two-point correlation functions,

$$\langle H_{\text{P}} \rangle = \sum_{i,j} \frac{1}{2} \left\langle \sigma_z^{(i)} \sigma_z^{(j)} \right\rangle . \tag{4}$$

As pointed out in [11,26], the individual expectation values $\langle \sigma_z^{(i)} \sigma_z^{(j)} \rangle$ do not necessarily depend on the states of all qubits, but only on a subset, which can be used to reduce the computational cost to calculate them. Some of the gates defined in $U^{\text{QAOA}} = \prod_i^p U_{\text{X}}(\beta_i) U_{\text{P}}(\gamma_i)$ commute with the operator product $\sigma_z^{(i)} \sigma_z^{(j)}$, and since the gates are unitary we can completely skip them in the definition of the correlation function,

$$\begin{aligned} &\left\langle \sigma_z^{(i)} \sigma_z^{(j)} \right\rangle \\ &\quad = \langle +_{\text{supp}(\text{RCC}_{i,j})} | \, U_{\text{RCC}_{i,j}}^{\text{QAOA}\dagger} \sigma_z^{(i)} \sigma_z^{(j)} U_{\text{RCC}_{i,j}}^{\text{QAOA}} \, | +_{\text{supp}(\text{RCC}_{i,j})} \rangle , \end{aligned} \tag{5}$$

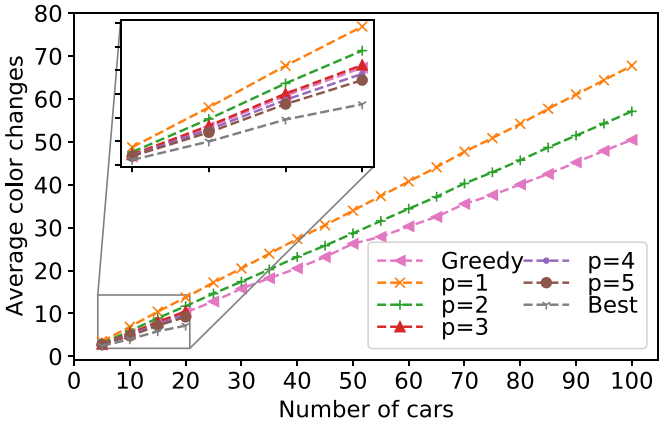

FIG. 4. Numerical results for the binary paint shop problem. The classical greedy algorithm is compared to QAOA with different levels p. Each data point is averaged over 100 randomly generated instances.

where $\text{RCC}_{i,j}$ is the set of gates not commuting with $\sigma_z^{(i)} \sigma_z^{(j)}$ called the reverse causal cone of the correlation function, $\text{supp}(\text{RCC}_{i,j})$ is the support of the reverse causal cone, i.e., the minimal set of qubits on which the reverse causal cone acts, and $|+_{\text{supp}(\text{RCC}_{i,j})}\rangle = \bigotimes_{l \in \text{supp}(\text{RCC}_{i,j})} 1/\sqrt{2}(|0\rangle_l + |1\rangle_l)$ is the superposition state of all computational basis states of the qubits in the reverse causal cone. The support of the reverse causal cone can be constructed in an iterative procedure [11,26]: for each layer in the QAOA circuit, we add all new neighbors in the problem graph of the support of the reverse causal cone of a QAOA circuit with one level less starting with the two qubits that define the correlation function. Therefore, the number of qubits affecting the expectation value depends on the number of QAOA levels p and the topology of the problem. The binary paint shop instances can be represented as graphs with a bounded degree of 4, thus the reverse causal cone includes up to $3^{p+1} - 1$ qubits. For $p = \{1, 2\}$ this results in system sizes that can be simulated using a standard desktop computer, independent of the actual size of the instance. After calculating the individual correlation functions independently, we find the QAOA expectation value by using Eq. (4).

In Fig. 4, we show the expected color changes from the QAOA output averaged over all instances together with the average result of the greedy algorithm, see Fig. 2, and exact solutions for up to $n = 20$ cars. Low-depth QAOA with $p = \{1, 2\}$ performs worse than the polynomial-time greedy algorithm, while for $p = 3$ levels the performance gap nearly vanishes. For $p = \{4, 5\}$, QAOA outperforms the greedy algorithm.

B. Beating the greedy algorithms for large instances

The greedy algorithms presented in Sec. III provide solutions with color changes growing linearly with the number of cars on average in polynomial time. Thus, they provide a good performance benchmark for QAOA. In Sec. IV A, numerical simulations revealed that QAOA with fixed level p is able to beat the greedy algorithm on average. In this section, we

strengthen this result with numerical insights in the infinite size limit, $n \to \infty$.

To understand the performance of the greedy algorithm, it is instructive to translate the action of the greedy algorithm presented in Fig. 2 into the spin glass picture. For the sake of clarity of presentation, we assume that all $2n$ couplings have magnitude $|J| = 1$, which is true in the infinite size limit. The greedy algorithm starts with assigning a random configuration to the spin representing the first car of the sequence. It then successively visits the spins representing the next car in the sequence. If it visits a car for the first time, it fixes the state of its spin such that the coupling between the car and its predecessor is fulfilled, i.e., the same state for ferromagnetic coupling and the opposite state for antiferromagnetic coupling.

The greedy algorithm is guaranteed to satisfy a coupling every time it approaches a spin representing a car that has not been visited yet. This happens $n-1$ times. The remaining $2n-n+1$ connections in the full graph are, however, not taken into account. On average, the energy of these unseen connections is equal to zero. In total, for $n \to \infty$, the greedy algorithm generates a solution with an average energy of $E_G/n = -1$, which results in solutions with color changes growing according to $\mathbb{E}_G(\Delta_C) = n/2$.

In comparison, in the limit of $n \to \infty$, the reverse causal cones of the two-point correlation functions after p levels of QAOA only include qubits in graphs that resemble trees of degree 4 and coupling strengths $J = \pm 1$ (see Appendix B). Thus, for systems of infinite size, the expectation value of each two-local operator is the same on average and given by the expectation value calculated on a tree.

To calculate the expectation value with a state-vector simulation, we would have to include $3^{p+1} - 1$ qubits, which is difficult to calculate classically even for shallow versions of QAOA. In [26], the authors developed a method that substantially increases the simulation capabilities using the small tree width of the involved tensor networks [40]. To calculate the two-point correlation functions on tree subgraph support, this method only scales polynomially in the number of qubits, but exponentially in the number of QAOA blocks. This allows us to calculate the expectation value up to $p = 7$ levels of QAOA, including 6560 qubits. With that we find average energies and color changes given in Table I. We note that the tensor network calculation also requires the optimization of the QAOA parameters. As the optimization might have found a local optimum rather than the global optimum, the values in Table I for QAOA represent lower bounds on the performance. If one could find a better set of parameters, the performance of the algorithm could even be improved.

We recognize that the performance of QAOA with $p = 3$ levels is close to the performance of the greedy algorithm, while for $p = 4$ there is a clear performance gap in favor of QAOA. While these arguments strictly hold for the limit of $n \to \infty$, we have shown that the results on smaller systems (see Fig. 4) agree with these results. When comparing the performance to two other heuristics, namely the red-first algorithm and the recursive greedy algorithm (see Appendix A), we see that QAOA outperforms the red-first algorithm on average with $p = 2$ levels, and for $p = 7$ QAOA also beats the recursive greedy algorithm on average. In Fig. 5, we show the data plotted against the number of QAOA blocks p together with a fit highlighting the improvement in performance when increasing p.

TABLE I. Average performance in terms color changes of QAOA with different levels p in comparison to greedy algorithms and random guessing in the limit $n \to \infty$. Ordered by the average performance.

Method	$\mathbb{E}(\Delta_C/n)$
Random guessing	1.000
$p = 1$	0.675
Red-first algorithm	0.666
$p = 2$	0.568
$p = 3$	0.503
Greedy algorithm, see Fig. 2	0.500
$p = 4$	0.462
$p = 5$	0.432
$p = 6$	0.411
Recursive greedy algorithm	0.400
$p = 7$	0.393

To determine the run time of the quantum algorithm on a state-of-the-art quantum processing unit, we recall that the binary paint shop problem can be represented as a graph of maximal degree 4 (see Sec. III A 1). On a fully connected quantum computer, the corresponding QAOA circuit with a fixed value of p thus only requires constant depth, making the QAOA for the BPSP a constant-time algorithm. This is in contrast with classical algorithms, which are polynomial-time algorithms. On quantum hardware with limited topology, e.g., for superconducting qubit devices with planar connectivity graphs, one could use methods such as the LHZ-encoding [41], which require a quadratic overhead in qubits, however one could still have constant computing time.

C. Experimental results

In this section, we show the results from QAOA circuits of binary paint shop instances with $p = 1$ executed on a trapped-

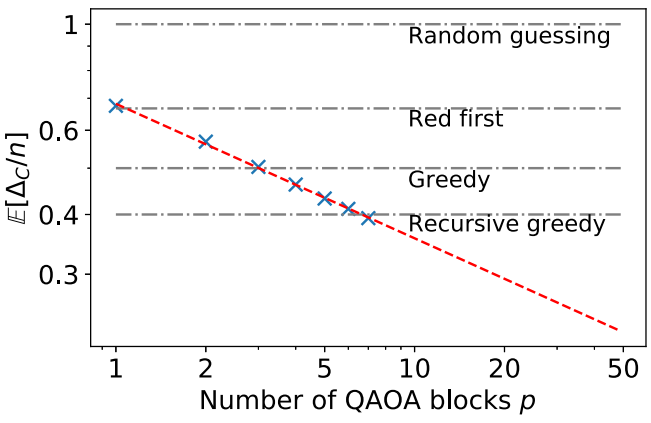

FIG. 5. The data from Table I shown in a log-log plot together with a fit to the function $f(p) = 10^b p^a$. The fit parameters are $a = (-0.279 \pm 0.005)$ and $b = (-0.168 \pm 0.003)$ with a coefficient of determination of $R^2 = 0.999$ [44].

ion quantum computer, the IonQ device [42], provided by Amazon Braket [43]. This device is composed of 11 fully connected qubits with average single- and two-qubit fidelities of 99.5% and 97.5%, respectively [42]. Like most available quantum hardware, trapped ion quantum computers only allow the application of gates from a restricted *native gate set* predetermined by the physical realization of the processor. To execute an arbitrary gate, compilation of the desired gate into available gates is required. For trapped ions, a generic native gate set consists of a parametrized two-qubit rotation, $R_{\mathrm{XX}}(\alpha) = \exp[-i\alpha\sigma_{\mathrm{x}}^{(i)}\sigma_{\mathrm{x}}^{(j)}/2]$ on qubits i and j, and a single qubit rotation, R,

$$R(\theta,\phi) = \begin{pmatrix} \cos(\theta/2) & -ie^{-i\phi}\sin(\theta/2) \\ -ie^{i\phi}\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \quad (6)$$

which includes $R_{\mathrm{X}}(\theta) = \exp[-i\theta/2\sigma_x^{(i)}] = R(\theta,0)$ and $R_{\mathrm{Y}}(\theta) = \exp[-i\theta/2\sigma_y^{(i)}] = R(\theta,\pi/2)$ [45]. These gates form a universal set of gates, i.e., all other gates can be synthesized with these gates.

The QAOA circuit, defined in Eq. (3), includes the parametrized two-qubit rotation $R_{\mathrm{ZZ}}(\gamma) = \exp[-i\gamma\sigma_{\mathrm{z}}^{(i)}\sigma_{\mathrm{z}}^{(j)}]$ on qubits i and j, parametrized single qubit $R_{\mathrm{X}}(\beta)$ rotations, and Hadamard gates. While the local $R_{\mathrm{X}}(\beta)$ is readily available on the hardware and can be executed without any overhead, the Hadamard gate and the two-qubit $R_{\mathrm{ZZ}}(\gamma)$ rotation require compilation, which will in turn increase the circuit depth.

To make the circuit as short as possible, we rotate the circuit by inserting Hadamard gates. For the sake of clarity, we introduce the unitary $U_{\mathrm{ZZ}} = U_{\mathrm{P}}$, which highlights that the problem unitary is a set of R_{ZZ} gates. Accordingly, U_{XX} is the unitary in which all R_{ZZ} in U_{ZZ} were replaced by R_{XX} gates. The same pattern applies for the definition of the mixing unitary U_{X}, i.e., U_{Y} denotes a unitary where all R_{X} gates are replaced by R_{Y} gates. The new circuit,

$$\begin{aligned} |\Psi\rangle_{\mathrm{QAOA}}^{p=1} &= U_{\mathrm{X}}(\beta)U_{\mathrm{ZZ}}(\gamma)\,|+\rangle \\ &= U_{\mathrm{X}}(\beta)\mathrm{H}^{\dagger}\mathrm{H}U_{\mathrm{ZZ}}(\gamma)\mathrm{H}\,|0\rangle \\ &= U_{\mathrm{X}}(\beta)\mathrm{H}U_{\mathrm{XX}}(\gamma)\,|0\rangle \\ &= U_{\mathrm{X}}(\beta-\pi)U_{\mathrm{Y}}(\pi/2)U_{\mathrm{XX}}(\gamma)\,|0\rangle\,, \end{aligned} \quad (7)$$

then only contains gates from the native gate set and thus needs no further compilation. For a higher p-value, the transformation is analogous and shown in Appendix D.

On IonQ devices, all gates are executed in sequence [46]. Thus, this representation of a QAOA circuit of a binary paint shop instance with n nodes and m edges can be carried out with circuit depth $d = m + 2n$, and it requires $2n$ single-qubit gates and m two-qubit gates. As the binary paint shop instances are bounded degree graphs with maximal degree 4, cf. Sec. III A 1, the circuit depth d scales linearly with the system size n, $O(d) \sim n$.

We execute the QAOA circuit with $p = 1$ for $N = 20$ randomly drawn binary paint shop instances from $n = 2$ to 11 cars [39]. For each instance, we take $M = 10^5$ samples and calculate the average number of color changes, $\langle\Delta_C^{\mathrm{QPU}}\rangle$. For comparison, we take data from an ideal (noiseless) simulation and random guessing. To compare the experimental output with the ideal simulation and random guessing, we calculate

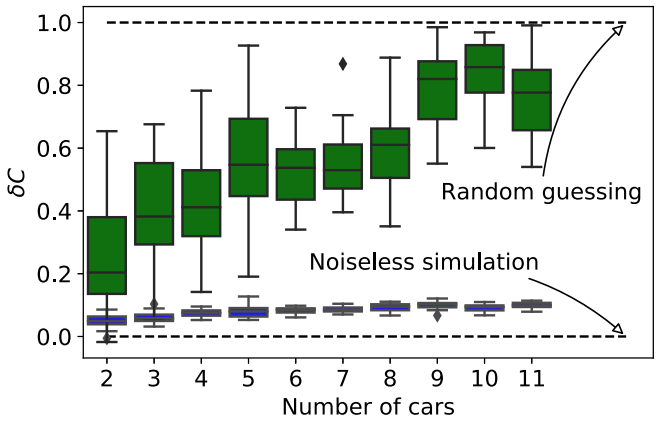

FIG. 6. Performance of the QAOA experiments using an IonQ QPU (green) in comparison to a simulation with noise (blue), cf. Appendix E, with $\delta C = \{0, 1\}$ corresponding to random guessing and an ideal (noiseless) simulation, respectively, as a function of the number of cars. Results are presented at each system size for $N = 20$ randomly drawn instances, averaged over $M = 10^5$ measurements. In this box plot, the black line shows to the median, the green boxes the interquartile range (IQR), the whiskers 1.5 times the IQR, and the diamonds are outliers.

the deviation in performance as

$$\delta C = \frac{\langle\Delta_C^{\mathrm{QPU}}\rangle - \langle\Delta_C^{\mathrm{sim}}\rangle}{\langle\Delta_C^{\mathrm{random}}\rangle - \langle\Delta_C^{\mathrm{sim}}\rangle}, \quad (8)$$

where $\langle\Delta_C^{\mathrm{sim}}\rangle$, $\langle\Delta_C^{\mathrm{QPU}}\rangle$, and $\langle\Delta_C^{\mathrm{random}}\rangle$ denote the expected instancewise color change obtained from the simulation, the QPU, and random guessing, respectively. A value of $\delta C = 0$ implies that the QPU found results as good as the ideal simulation did, while $\delta C = 1$ means that the QPU output mimics random guessing.

In Fig. 6, we plot the distribution of δC over all N instances for increasing system size. As is clearly visible, for the smallest system size ($n = 2$) the results are close to an ideal simulation, while for the largest studied instances ($n = 11$), the output is almost random. Similar to previous QAOA experiments [17–19], the present results highlight the strong influence of noise on the performance of the quantum algorithm. Moreover, in the same figure, we show the quantity δC where we replaced the QPU results with those obtained from a simulation with noise. In this simulation, the noise was tuned such that we find agreement with the reported single- and two-qubit fidelities in [42]. For more details on the noise model used, we refer to Appendix E. As is visible, the simulation with noise cannot fully explain the experimental data. This could be due to several reasons:

(i) On IonQ all gates are carried out strictly successively [46], which leads to long idle times of the qubits. This is not taken into account by errors of single- and two-qubit gates.

(ii) The fidelities reported in [42] were obtained for isolated one- and two-qubit gates.

This outcome highlights that single- and two-qubit fidelities are not sufficient to fully characterize the hardware's performance.

V. CONCLUSION AND OUTLOOK

In this work, we applied QAOA to the binary paint shop problem, a computational problem from the automotive industry. We have shown numerical simulations together with experimental data obtained from a trapped ion quantum computer. Moreover, we were able to provide a comparison between the performance of QAOA and classical heuristics in the infinite-size limit for noiseless quantum computation.

The experimental results of this paper highlight the deterioration of the quantum algorithm's performance when increasing the problem size. To push forward to industry-relevant binary paint shop instances with hundreds of cars, either noise mitigation techniques or adaptations of QAOA must be developed to make this application on NISQ devices superior to random guessing. In this direction, the recursive adaption of QAOA introduced in [47] or the encoding of QAOA into a lattice gauge model [41] might provide improvements. Moreover, it would be interesting to investigate whether classical local algorithms [23] are able to outperform the results shown herein and to improve the classical performance bounds.

Furthermore, providing an answer on the question of whether QAOA is a constant factor approximation algorithm could open up new room for quantum advantage.

This work has impact far beyond the restricted use case of the binary paint shop. In the way the binary paint shop problem is formulated, we can already see that it is more general: every resequencing problem with a cost function that only depends on the relative orientation of variables that are adjacent in the sequence is similar to the binary paint shop problem. Even a generalization to next-to-adjacent orientation is straightforward. These kinds of resequencing problems are prevalent in many production facilities. For example, consider the optimization problem of cars being assembled by workers: cars with sunroofs are more difficult to assemble and should not be adjacent in a sequence so that the workers can keep up with the constant speed of the conveyor belt. Furthermore, since we use a generic quantum algorithm, QAOA, we are confident that the observed runtime and solution quality enhancements will prevail as long as we do not change the general problem characteristics: a regular problem graph with interaction strengths of magnitude 1. These strict requirements can also be relaxed considerably, as shown in [26]. We relegate the search for interesting use-cases within this group to future work.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 828826. We thank Martin Schütz for technical support with Amazon Braket, and Héctor Valverde for technical support with AWS.

APPENDIX A: CLASSICAL HEURISTICS

1. Red-first heuristic

The red-first heuristic is a greedy algorithm for the binary paint shop problem in which all first occurrences of each car have the same color. This heuristic has a proven average performance for $n \to \infty$ of

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{RF}}(\Delta_C) = \frac{2n}{3}. \tag{A1}$$

After mapping the binary paint shop problem to an Ising Hamiltonian, the action of the red-first heuristic on the sequence representing a BPSP instance is equivalent to setting all spins in the spin system to the same value. The average energy for $n \to \infty$ is given by

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{RF}}(E) = \frac{1}{2}\sum J_{ij}s_i s_j = \frac{1}{2}\sum J_{ij} = -\frac{n}{3}, \tag{A2}$$

where we used the results from Appendix B on the average degree and coupling strengths of the graph.

2. Recursive greedy heuristic

The recursive greedy heuristic starts by iteratively deleting both occurrences of the last car of the sequence until the sequence has length 2 and only one car. Subsequently, the sequence of length 2 is painted optimally. After that, the occurrences of the last deleted car are added back to the sequence. While keeping the already painted cars fixed, the new car is painted optimally. This is repeated until the whole sequence is painted.

From the spin-system perspective, this corresponds to the procedure of deleting all spins except a single spin and adding back spin by spin while setting each state to the best possible energetic configuration.

This heuristic has a proven average performance for $n \to \infty$ of

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{RG}}(\Delta_C) = \frac{2n}{5}. \tag{A3}$$

As single color change yields an increase of energy by 1, we use the results on the red-first heuristic to determine the average energy to

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{RG}}(E) = -\frac{3n}{5}. \tag{A4}$$

3. Random guessing

As a baseline for comparison, we show here the performance of random guessing. For $n \to \infty$,

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}_{\mathrm{random}}(E) &= \sum_{s_i,s_j=-1,1} \sum_{i,j} J_{ij}s_i s_j \\ &= \sum_{i,j}(1/2J_{ij} - 1/2J_{ij}) = 0,\end{aligned} \tag{A5}$$

and thus

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{random}}(\Delta_C) = n. \tag{A6}$$

APPENDIX B: CHARACTERISTICS OF THE SPIN SYSTEM

In this Appendix, we discuss the distribution of coupling strengths J_{ij} and the average degree of the Ising model resulting from the mapping given in Fig. 3.

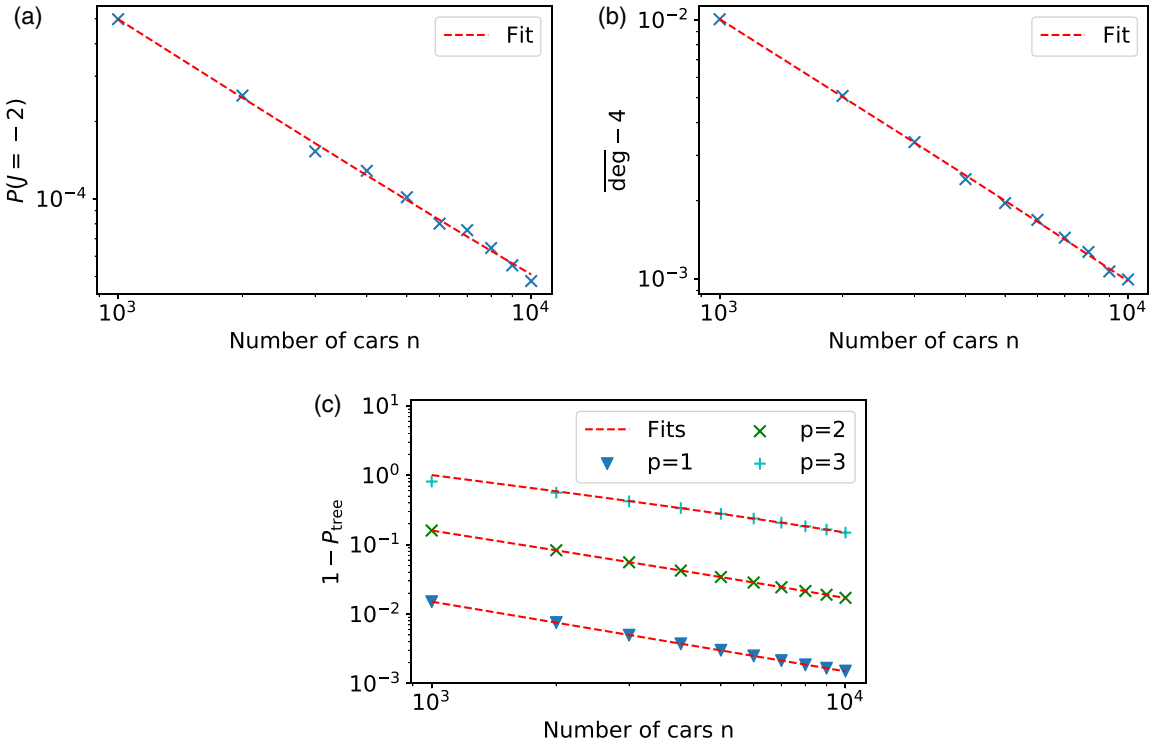

FIG. 7. This plot provides numerical insight into the properties of the graph representation of the binary paint shop problems for increasing system size n. For all plots, we fit the data to a function of the form $f(n) = 10^a n^b + c$. (a) The probability to draw an edge with coupling strength $J = \{-2\}$. Fit parameters are $(b = -1.028 \pm 0.048, c = 0.000 \pm 0.000)$. (b) The deviation of the average degree $\overline{\text{deg}}$ of random binary paint shop instances from degree 4. Fit parameters are $(b = -0.991 \pm 0.019, c = 0.000 \pm 0.000)$. (c) The probability to draw a QAOA subgraph that is not a tree for $p = 1, 2, 3$ levels of QAOA. Fit parameters are $(b = -1.006 \pm 0.006,\ c = 9\times10^{-6} \pm 3\times10^{-5})$ for $p = 1$, $(b = -0.980 \pm 0.005,\ c = -0.002 \pm 0.000)$ for $p = 2$, and $(b = -0.732 \pm 0.029,\ c = -0.046 \pm 0.010)$ for $p = 3$. For the fit of $p = 3$, only the last six data points were used. All data points were averaged over 1000 randomly drawn instances for each system size.

1. Coupling strengths

We show here that the probability of finding a ferromagnetic interaction ($J = -1$) in the spin glass representation of the binary paint shop is twice as big as the probability of finding an antiferromagnetic coupling ($J = 1$) when $n \to \infty$, while the probability of finding a coupling of $J = -2$ is converging to zero.

If we draw a random pair of cars (w_i, w_{i+1}) of the sequence, the probability that we find the same car c_i twice is given by $P_{\text{samecar}} = 1/(2n-1)$. A similar argument can be made for the probability that a single car c_i has the same neighbor c_j twice. As couplings of $J = -2$ are only generated when a car has the same neighbor twice, this already means that $P_{ij}(-2) = 0$ when $n \to \infty$. This is in agreement with the numerics shown in Fig. 7(a).

Thus, when looking at $n \to \infty$, we now exclude the cases of the previous paragraph and look at the probabilities $P_{ij}(+1)$ and $P_{ij}(-1)$. By construction, a ferromagnetic coupling is generated whenever two cars in the sequence are neighbors and both occur for the first time or both have already occurred. If we draw a random pair of the sequence, the probability that this pair generates a ferromagnetic coupling is given by

$$P(J=-1) = \frac{1}{2n-1}\sum_{\langle ij\rangle}[P_{ij}(00) + P_{ij}(11)], \tag{B1}$$

where $P_{ij}(00)$ is the probability that both cars at positions i and j in the sequence occur the first time, and $P_{ij}(11)$ is the probability that both cars appear for the second time. For position i in the sequence, the probability that the car already appeared before is given by $(i-1)/2n$. With this, we reformulate Eq. (B1) as

$$\begin{aligned} P(J=-1) &= \frac{1}{2n-1}\sum_{i=1}^{2n-1}\left[\left(1-\frac{i-1}{2n-1}\right)\left(1-\frac{(i+1)-1}{2n-1}\right)\right. \\ &\qquad \left. + \frac{i-1}{2n-1}\frac{(i+1)-1}{2n-1}\right] \\ &= \frac{1-6n+8n^2}{6n(2n-1)}. \end{aligned} \tag{B2}$$

For $n \to \infty$, we find $P(J=-1) = 2/3$, i.e., the probability of finding a ferromagnetic interaction strength is $2/3$. The probability of finding an antiferromagnetic coupling, $P(J=+1)$, can be calculated in a similar way by calculating the probability that one of two consecutive cars in the sequence was already seen while the other did not occur yet. This can be written as

$$P(J=+1) = \frac{1}{2n-1}\sum_{\langle ij\rangle}[P_{ij}(01) + P_{ij}(10)] \tag{B3}$$

$$= \frac{1}{2n-1}\sum_{i=1}^{2n-1}\left[\left(\frac{i-1}{2n-1}\right)\left(1-\frac{(i+1)-1}{2n-1}\right) + \frac{i-1}{2n-1}\left(1-\frac{(i+1)-1}{2n-1}\right)\right] = \frac{4n^2-1}{6n(2n-1)}, \tag{B4}$$

which for $n \to \infty$ gives $P_{ij}(J=+1) = 1/3$. Thus, the spin-system formulation of the binary paint shop problems has (for $n \to \infty$) integer ferromagnetic or antiferromagnetic couplings with probabilities $2/3$ and $1/3$, respectively.

2. Average degree

As the probability that a car is next to the same car twice in the sequence is vanishing when $n \to \infty$, each car representing a spin in the spin system has four connections in the infinite-size limit. This intuition is supported by Fig. 7(b), where we show the deviation from an average degree of 4 for 1000 randomly drawn binary paint shop instances while increasing the system size.

3. Number of tree subgraphs

The expectation value of QAOA is given by the sum of the expectation values of two-point correlation functions $\sigma_z^{(i)}\sigma_z^{(j)}$; see Eq. (4). As was discussed in Sec. IV, each expectation value can be calculated over the reverse causal cone of the corresponding edge (i, j) in the graph. In Fig. 7(c) we show the probability $1 - P_{\text{tree}}$, with P_{tree} the probability that a randomly drawn subgraph resembles a tree of degree 4 for $p = 1$, 2, and 3. The numerical experiment together with the fits suggest that the probability to draw subgraphs that are not trees is zero in the infinite-size limit ($n \to \infty$).

APPENDIX C: EXTENSION OF TREE-QAOA

In this Appendix, we prove that the optimal QAOA parameters on a tree-graph with coupling strength $J = +1$ are equivalent to the optimal QAOA parameters on a tree-graph with coupling strengths $J = \pm 1$. As a starting point, we assume that we have an Ising model defined on a tree,

$$H_{\text{tree}} = \sum_{(i,j)\in E} J_{ij}\sigma_i\sigma_j, \tag{C1}$$

with the edge set E defining the tree graph. To prove the assumption, we show that the optimal parameters of tree-QAOA stay the same when replacing the coupling strengths $J_{ij} = 1$ with any $J_{ij} = \pm 1$. We assume that there exists a transformation

$$U_{\text{T}} H_{\text{tree}}^{J=1} U_{\text{T}}^\dagger = H_{\text{tree}}^{J=\pm 1}, \tag{C2}$$

with $U_{\text{T}} = \bigotimes_j^k X_j$ defining a k-local spin flip operation on a subset of k qubits. Inserting this transformation into the expectation value of the QAOA circuit, Eq. (1), yields

$$\begin{aligned}
&\langle +| \cdots e^{i\gamma_p H_{\text{tree}}^{J\pm1}} U_{\text{X}}(\pm Z_i Z_j) U_{\text{X}}^\dagger e^{-i\gamma_p H_{\text{tree}}^{J\pm1}} \cdots |+\rangle \\
&= \langle +| \cdots e^{i\gamma_p U_{\text{T}} H_{\text{tree}}^{J=1} U_{\text{T}}^\dagger} U_{\text{X}}(\pm Z_i Z_j) U_{\text{X}}^\dagger e^{-i\gamma_p U_{\text{T}} H_{\text{tree}}^{J=1} U_{\text{T}}^\dagger} \cdots |+\rangle \\
&= \langle +| \cdots U_{\text{T}} e^{i\gamma_p H_{\text{tree}}^{J=1}} U_{\text{T}}^\dagger U_{\text{X}}(\pm Z_i Z_j) U_{\text{X}} U_{\text{T}} e^{-i\gamma_p H_{\text{tree}}^{J=1}} U_{\text{T}}^\dagger \cdots |+\rangle \\
&= \langle +| \cdots e^{i\gamma_p H_{\text{tree}}^{J=1}} U_{\text{T}}^\dagger U_{\text{X}}(\pm Z_i Z_j) U_{\text{X}}^\dagger U_{\text{T}} e^{-i\gamma_p H_{\text{tree}}^{J=1}} \cdots |+\rangle \\
&= \langle +| \cdots e^{i\gamma_p H_{\text{tree}}^{J=1}} U_{\text{X}} U_{\text{T}}^\dagger (\pm Z_i Z_j) U_{\text{T}} U_{\text{X}}^\dagger e^{-i\gamma_p H_{\text{tree}}^{J=1}} \cdots |+\rangle \\
&= \langle +| \cdots e^{i\gamma_p H_{\text{tree}}^{J=1}} U_{\text{X}}(Z_i Z_j) U_{\text{X}}^\dagger e^{-i\gamma_p H_{\text{tree}}^{J=1}} \cdots |+\rangle .
\end{aligned} \tag{C3}$$

This shows that, if a transformation Eq. (C2) exists, then the expectation value of the tree-QAOA with couplings strengths $J_{ij} = \pm 1$ is equivalent to the initial case with $J_{ij} = 1$ and the variational parameters are the same. On trees, where no frustration is present, it is always possible to find such a transformation.

APPENDIX D: CIRCUIT OPTIMIZATION FOR TRAPPED ION QUANTUM COMPUTERS

In Eq. (7), we showed how to transform the QAOA with a $p = 1$ circuit such that only native gates are used. In this Appendix, we show how this can be done for an arbitrary number of QAOA levels p. By inserting Hadamard gates, the circuit transforms to

$$\begin{aligned}
|\Psi\rangle_{\text{QAOA}}^p &= U_{\text{X}}(\beta_p) U_{\text{ZZ}}(\gamma_p) \cdots U_{\text{X}}(\beta_2) U_{\text{ZZ}}(\gamma_2) U_{\text{X}}(\beta_1) U_{\text{ZZ}}(\gamma_1) |+\rangle \\
&= U_{\text{X}}(\beta_p) \text{HH} U_{\text{ZZ}}(\gamma_p) \text{HH} \cdots \text{HH} U_{\text{X}}(\beta_2) \text{HH} U_{\text{ZZ}}(\gamma_2) \text{HH} U_{\text{X}}(\beta_1) \text{HH} U_{\text{ZZ}}(\gamma_1) \text{H} |0\rangle \\
&= U_{\text{X}}(\beta_p) \text{H} U_{\text{XX}}(\gamma_p) \cdots U_{\text{Z}}(\beta_2) U_{\text{XX}}(\gamma_2) U_{\text{Z}}(\beta_1) U_{\text{XX}}(\gamma_1) |0\rangle \\
&= U_{\text{X}}(\beta_p - \pi) U_{\text{Y}}(\pi/2) U_{\text{XX}}(\gamma_p) \cdots U_{\text{Z}}(\beta_2) U_{\text{XX}}(\gamma_2) U_{\text{Z}}(\beta_1) U_{\text{XX}}(\gamma_1) |0\rangle ,
\end{aligned} \tag{D1}$$

including only native gates.

APPENDIX E: NOISE MODEL

In this Appendix, we introduce the noise model used in the noisy simulation shown in Fig. 6. After each application of a gate on a set of qubits, we subsequently apply a local depolarizing channel,

$$\mathcal{E}(\rho) : \rho \to \sum_i K_i \rho K_i^\dagger, \tag{E1}$$

TABLE II. The QAOA parameters obtained with the method given in [26] and used for the numerical simulations and experiments in this paper.

p	γ_1	β_1	γ_2	β_2	γ_3	β_3	γ_4	β_4	γ_5	β_5	γ_6	β_6	γ_7	β_7
1	0.52358	−0.39269												
2	0.40784	−0.53411	0.73974	−0.28296										
3	0.35450	−0.58794	0.65138	−0.42318	0.75426	−0.22301								
4	0.31500	−0.60498	0.58754	−0.47780	0.67322	−0.36127	0.77120	−0.18753						
5	0.29092	−0.62254	0.54678	−0.50507	0.60334	−0.41672	0.68722	−0.32534	0.78446	−0.16280				
6	0.26872	−0.62933	0.51278	−0.52317	0.56359	−0.45282	0.61410	−0.38834	0.69565	−0.29814	0.78667	−0.14595		
7	0.25377	−0.63776	0.48903	−0.53260	0.53172	−0.47185	0.57572	−0.43247	0.62149	−0.36318	0.69778	−0.27774	0.78866	−0.13380

with the Kraus operators

$$K_1 = \sqrt{1-\eta}\mathbb{1}, \quad K_2 = \sqrt{\frac{\eta}{3}}\sigma_x, \quad K_3 = \sqrt{\frac{\eta}{3}}\sigma_y,$$
$$K_4 = \sqrt{\frac{\eta}{3}}\sigma_z, \tag{E2}$$

on all qubits that participated in the gate. Thus, for a single-qubit gate on qubit i, we apply a single local depolarizing channel on qubit i. For a two-qubit gate on qubits i, j, we apply two local depolarizing channels on qubits i, j. As two-qubit gates are more error-prone than one-qubit gates, we assign different error rates to single- and two-qubit gates, η_{1Q} and η_{2Q}, respectively. On IonQ, the average single- and two-qubit fidelities are reported to be 99.5% and 97.5%, respectively [42]. To adjust the error rate of the single-qubit gate, η_{1Q}, we follow the experimental procedure from [42] and simulate random benchmarking and measure the fidelity. We tune the error rate η_{1Q} such that we find a fidelity of 99.5%, resulting in $\eta_{1Q} = 0.0029$. Similarly, to find an appropriate value for η_{2Q}, we simulate partial state tomography on the Bell state and tune η_{2Q} resulting in $\eta_{2Q} = 0.0168$.

[1] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen *et al.*, Digitized adiabatic quantum computing with a superconducting circuit, Nature (London) **534**, 222 (2016).

[2] L. DiCarlo, J. Chow, J. Gambetta, L. S. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin *et al.*, Demonstration of two-qubit algorithms with a superconducting quantum processor, Nature (London) **460**, 240 (2009).

[3] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Demonstration of a small programmable quantum computer with atomic qubits, Nature (London) **536**, 63 (2016).

[4] M. Reagor, C. B. Osborn, N. Tezak, A. Staley, G. Prawiroatmodjo, M. Scheer, N. Alidoust, E. A. Sete, N. Didier, M. P. da Silva *et al.*, Demonstration of universal parametric entangling gates on a multi-qubit lattice, Sci. Adv. **4**, eaao3603 (2018).

[5] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[6] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM Press, New York, NY, 1996), pp. 212–219.

[7] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 1994), pp. 124–134.

[8] 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**, 4213 (2014).

[9] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[10] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, Continuous-variable quantum neural networks, Phys. Rev. Res. **1**, 033063 (2019).

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

[12] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem, arXiv:1412.6062.

[13] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv:1811.08419.

[14] M. Streif and M. Leib, Comparison of QAOA with quantum and simulated annealing, arXiv:1901.01903.

[15] M. Streif and M. Leib, Forbidden subspaces for level-1 quantum approximate optimization algorithm and instantaneous quantum polynomial circuits, Phys. Rev. A **102**, 042416 (2020).

[16] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, Quantum Inf. Proc. **19**, 197 (2020).

[17] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, Nat. Phys. **17**, 332 (2021).

[18] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin *et al.*, Quantum approximate optimization of the long-range ising model with a trapped-ion quantum simulator, Proc. Natl. Acad. Sci. USA **117**, 25396 (2020).

[19] J. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong *et al.*, Unsupervised machine learning on a hybrid quantum computer, arXiv:1712.05771.

[20] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, Phys. Rev. X **10**, 021067 (2020).

[21] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: a journey through digitalization, control, and hybrid quantum variational schemes, arXiv:1906.08948.

[22] A. Bapat and S. P. Jordan, Bang-bang control as a design principle for classical and quantum optimization algorithms, Quantum Inf. Comput. **19**, 424 (2019).

[23] M. B. Hastings, Classical and quantum bounded depth approximation algorithms, arXiv:1905.07047.

[24] S. Bravyi, D. Gosset, and R. Movassagh, Classical algorithms for quantum mean values, Nat. Phys. **17**, 337 (2021).

[25] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances, arXiv:1812.04170.

[26] M. Streif and M. Leib, Training the quantum approximate optimization algorithm without access to a quantum processing unit, Quantum Sci. Technol. **5**, 034008 (2020).

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

[28] Z. Jiang, E. G. Rieffel, and Z. Wang, Near-optimal quantum circuit for grover's unstructured search using a transverse field, Phys. Rev. A **95**, 062317 (2017).

[29] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard–even for logarithmically many qubits and free fermionic systems, arXiv:2101.07267.

[30] F. Barahona, On the computational complexity of ising spin glass models, J. Phys. A **15**, 3241 (1982).

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

[32] T. Epping, W. Hochstättler, and P. Oertel, Some results on a paint shop problem for words, Electron. Notes Discr. Math. **8**, 31 (2001).

[33] T. Epping, W. Hochstättler, and P. Oertel, Complexity results on a paint shop problem, Discr. Appl. Math. **136**, 217 (2004).

[34] A. Gupta, S. Kale, V. Nagarajan, R. Saket, and B. Schieber, The approximability of the binary paintshop problem, in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (Springer, Berlin, Heidelberg, 2013), pp. 205–217.

[35] S. Khot, On the power of unique 2-prover 1-round games, in *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing (STOC '02)* (ACM Press, New York, NY, 2002), pp. 767–775.

[36] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM **42**, 1115 (1995).

[37] S. D. Andres and W. Hochstättler, Some heuristics for the binary paint shop problem and their expected number of colour changes, J. Discr. Alg. **9**, 203 (2011).

[38] F. Meunier and B. Neveu, Computing solutions of the paintshop–necklace problem, Comput. Oper. Res. **39**, 2666 (2012).

[39] M. Streif, S. Yarkoni, A. Skolik, F. Neukart, and M. Leib, Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm (2021), 10.5281/zenodo.4451344.

[40] I. L. Markov and Y. Shi, Simulating quantum computation by contracting tensor networks, SIAM J. Comput. **38**, 963 (2008).

[41] W. Lechner, Quantum approximate optimization with parallelizable gates, IEEE Trans. Quantum Eng. **1**, 1 (2020).

[42] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins *et al.*, Benchmarking an 11-qubit quantum computer, Nat. Commun. **10**, 5464 (2019).

[43] Amazon Braket, available at https://aws.amazon.com/braket/.

[44] N. R. Draper and H. Smith, *Applied Regression Analysis*, 3th ed. (Wiley, New York, 1998).

[45] D. Maslov, Basic circuit compilation techniques for an ion-trap quantum machine, New J. Phys. **19**, 023035 (2017).

[46] IonQ Best Practices, available at https://ionq.com/best-practices.

[47] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Obstacles to State Preparation and Variational Optimization from Symmetry Protection, Phys. Rev. Lett. **125**, 260505 (2020).