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, Ungererstr. 69, 80805 München, Germany*
[2] *University Erlangen-Nürnberg (FAU), Institute of Theoretical Physics, Staudtstr. 7, 91058 Erlangen, Germany*
[3] *LIACS, Leiden University, Snellius Gebouw, Niels Bohrweg 1, 2333 CA Leiden, Netherlands*

(Dated: November 9, 2020)

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 and demonstrate that QAOA with constant depth is able to beat classical heuristics on average in the infinite size limit $n \to \infty$. For the BPSP, it is known that no classical algorithm can exist which approximates the problem in polynomial runtime. We introduce a BPSP instance which is hard to solve with QAOA, and numerically investigate its performance and discuss QAOA's ability to generate approximate solutions. We complete our studies by running first experiments of small-sized instances on a trapped-ion quantum computer through AWS Braket.

I. INTRODUCTION

Achievements in fabrication and control of two-level systems led to first quantum computers with tens of qubits [1–4] and recently culminated in the demonstration of a quantum computer solving a computational task 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 proven scaling advantages over classical algorithms, such as Grover's [6] or Shor's [7] algorithm, require fault-tolerant quantum computers. However the devices which will become available in the next 5 to 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 tailor-made to the quantum processing units' (QPU) characteristics. A promising class of NISQ algorithms is the class of variational quantum algorithms, which are parameterized ansätze optimized with classical learning loops. There exist various ideas how to tailor the ansatz for different tasks, such as the Variational Quantum 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 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 classical easy instances [21]. Establishing scaling comparisons between QAOA and classical methods for classically hard problems of relevant size thus belongs to the next important steps to find useful NISQ applications of QAOA.

In this present contribution, we take a step in this direction and present a combinatorial optimization problem from the automotive industry, the *binary paint shop problem* (BPSP). We show that the problem can be encoded in a constraint-free way and requires only 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 fixed degree and coupling strength, and thus bypasses the training procedure, using the method developed in [26]. We present numerical results and run first small-scale experiments on a trapped ion quantum computer. We show how QAOA performs against classical heuristics in the limit of infinite system sizes, $n \to \infty$, and discuss QAOA's ability to find approximate solutions.

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 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. V, 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 classical optimization problems.

To solve a combinatorial optimization problem with

QAOA, the first step is to reformulate it to 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 $E_\mathrm{P} = \sum_{(i,j)\in E} J_{i,j} s_i s_j$ that needs to be minimized. We note that finding the optimal solution of a spin glass is $\mathcal{NP}$-hard, thus there exist mappings with at most polynomial overhead from all $\mathcal{NP}$ problems to such a spin system, some of them shown in [29]. 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 an ansatz state $|\Psi(\{\beta_l, \gamma_l\})\rangle$,

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

Therefore we translate the spin system to its quantum version

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

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 closely inspired by quantum annealing techniques and is generated by the repeated application of the mixing and problem unitary, $U_\mathrm{M}(\beta_l) = e^{-i\beta_l H_\mathrm{M}}$ and $U_\mathrm{P}(\gamma_l) = e^{-i\gamma_l H_\mathrm{P}}$, 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, $H_\mathrm{M} = \sum_i \sigma_x^{(i)}$, and the problem Hamiltonian, see Eq. (2). The full ansatz state

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

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 is updating 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. Meaning, 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 which minimizes the number of color changes between adjacent cars in the sequence. This combinatorial optimization problem is called the *binary paint shop problem* (BPSP) [30–32]. In Fig. 1, we show a binary paint shop instance together with a sub-optimal and the optimal solution. A formal definition of the binary paint shop problem is given in Def. III.1.

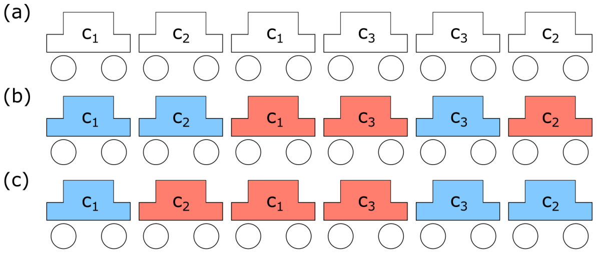

FIG. 1: (a) A binary paint shop instance with $n = 3$ cars $\{c_1, c_2, c_3\}$. (b) A valid but sub-optimal coloring with $\Delta_C = 3$ color changes. (c) An optimal coloring which only requires $\Delta_C = 2$ color changes to paint the sequence.

Definition III.1. (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 $\mathcal{NP}$-hard optimization problems, thus there is no polynomial-time algorithm which 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 $\mathcal{APX}$-hard [31], i.e. it is as difficult to approximate as every problem in $\mathcal{APX}$. Additionally, if the Unique Games Conjecture (UGC) [33] holds, it is even not in $\mathcal{APX}$ and thus, there would not be a constant factor approximation algorithm for any α [32]. A constant factor approximation algorithm would be a polynomial-time algorithm which returns an approximate solution with at most αOPT color changes where OPT is the optimal number of color changes. This is a key difference to previous problem classes QAOA has been applied to, such as Max-Cut, where constant factor approximation algorithms are known [34].

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 [30, 31]. The greedy algorithm introduced in [30] 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. 1 for pseudo code of this algorithm. For $n \to \infty$ cars, this greedy algorithm finds a solution with an average number of color changes $\mathbb{E}_{\mathrm{G}}(\Delta_C) = n/2$ [35]. In Appendix A, we review two other greedy algorithms, the red first algorithm and the recursive greedy algorithm, yielding $\mathbb{E}_{\mathrm{RF}}(\Delta_C) = 2n/3$ and $\mathbb{E}_{\mathrm{RG}}(\Delta_C) = 2n/5$ color changes on average respectively [35]. Numerical results however suggest that the average number of optimal color changes is sub-linear in the number of cars n [36]. Moreover, for some instances, the greedy algorithms only find solutions far from the optimal solution [35]. For example, for the instance shown in Fig. 1(a), the greedy algorithm finds the solution given in (b) rather then the optimal solution shown in (c).

More general versions of the binary paint shop problem can be found in practice. Typically both 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 $\mathcal{NP}$-hard in both the number of cars and colors [31], 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 to be 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 pseudo code of this mapping in Fig. 2. 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 [29]. From the BPSP construction, we also know that a solution with Δ_C color changes is separated by an energy $\Delta E = 2$ from a solution with $\Delta_C + 1$ color changes.

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.

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 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

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.

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 2 times, each car has maximal 4 distinct neighbors. It follows that the nodes in the graph G representing the spin system also have 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. (9), 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 $\{-2, -1, 1, 2\}$. 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 anti-ferromagnetic coupling and that coupling strengths $|J_{ij}| = 2$ are suppressed in the infinite size limit.

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 $\{\beta_i^\mathrm{tree}, \gamma_i^\mathrm{tree}\}$ found with method [26], shown in Tab. II.

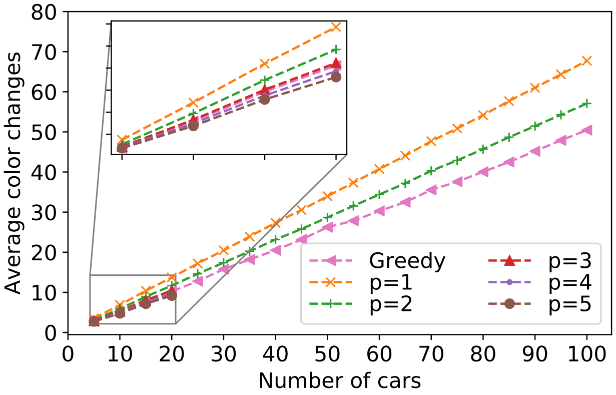

FIG. 4: Numerical results for the binary paint shop problem. The classical greedy algorithm is compared to QAOA with different levels p. Each data point is averaged over 100 randomly generated instances.

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 $n = 100$ cars in increments of 5 cars.

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 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,

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

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 $U^\mathrm{QAOA} = \prod_i^p U_\mathrm{M}(\beta_i) U_\mathrm{P}(\gamma_i)$ 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,

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

where $\mathrm{RCC}_{i,j}$ is the set of gates not commuting with $\sigma_z^{(i)} \sigma_z^{(j)}$ called the reverse causal cone of the correlation function, $\mathrm{supp}(\mathrm{RCC}_{i,j})$ is the support of the reverse causal cone, i.e. the minimal set of qubits the reverse causal cone acts on, and $|+_{\mathrm{supp}(\mathrm{RCC}_{i,j})}\rangle = \bigotimes_{l\in\mathrm{supp}(\mathrm{RCC}_{i,j})} 1/\sqrt{2}\left(|0\rangle_l + |1\rangle_l\right)$ 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 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 presented in Fig. 1. 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. 1 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. For a straightforward extension to all other cases, we refer to Appendix A. 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. same state for ferromagnetic- and 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 were 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 average energy of $E_{\mathrm{G}}/n=-1$, which results in solutions with color changes growing according to $\mathbb{E}_{\mathrm{G}}(\Delta_C)=n/2$.

Method	$-\mathbb{E}\left[\langle E\rangle/n\right]$	$\mathbb{E}\left[\Delta_C/n\right]$
p=1	0.325	0.675
p=2	0.432	0.568
p=3	0.497	0.503
p=4	0.538	0.462
p=5	0.568	0.432
Greedy algorithm, Fig. 1	0.500	0.500
Recursive Greedy algorithm	0.600	0.400
Red first algorithm	0.333	0.666

TABLE I: Average performance in terms of energy and color changes of QAOA with different levels p in comparison to greedy algorithms in the limit $n\to\infty$.

In comparison, in the limit of $n\to\infty$, the reverse causal cones of the two-local operators after p levels of QAOA only include qubits in graphs which resemble trees. As shown in [26], the tree topology can be exploited such that it is possible to calculate the QAOA expectation value using tensor networks for fixed p values for the infinite size problem, with average energies and color changes given in Fig. I. 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 $n\to\infty$ only, we have shown that the results on smaller systems (see Fig. 4) agree with these results.

We also compare the performance in the infinite size limit to two other heuristics, the red-first algorithm and the recursive greedy algorithm, see Appendix A. QAOA is better than the red-first algorithm on average with $p=2$ levels of QAOA, while a higher number of QAOA levels is needed to beat the recursive greedy algorithm. Unfortunately our classical computing power was not sufficient to calculate the QAOA expectation value for p higher than 4.

C. Experimental results

In this section, we execute QAOA circuits of binary paint shop instances with $p=1$ on a trapped-ion quantum computer, the IonQ device [37], provided by AWS Braket [38]. This device is composed of 11 fully-connected qubits with average single- and two-qubit fidelities of 99.5% and 97.5% respectively [37].

As 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 parameterized two-qubit rotation, $R_{\mathrm{XX}}(\alpha) = \exp[-\mathrm{i}\alpha\sigma_{\mathrm{x}}^{(i)}\sigma_{\mathrm{x}}^{(j)}/2]$ on qubits i and j, and a single qubit rotation, R,

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

which includes $R_{\mathrm{X}}(\theta) = \exp[-\mathrm{i}\theta/2\sigma_x^{(i)}] = R(\theta,0)$ and $R_{\mathrm{Y}}(\theta) = \exp[-\mathrm{i}\theta/2\sigma_y^{(i)}] = R(\theta,\pi/2)$ [39]. 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 parameterized two-qubit rotation $R_{\mathrm{ZZ}}(\gamma) = \exp[-i\gamma\sigma_{\mathrm{z}}^{(i)}\sigma_{\mathrm{z}}^{(j)}]$ on qubits i and j, parameterized 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 injecting Hadamard 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 higher p-value, the transformation is analogous and shown in Appendix D.

On IonQ devices, all gates are executed in sequence [40]. 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 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, $\mathcal{O}(d) \sim n$.

We execute the QAOA circuit with $p = 1$ for $N = 20$ randomly drawn binary paint shop instances from $n = 2$ to $n = 11$ cars. 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 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 instance-wise 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.

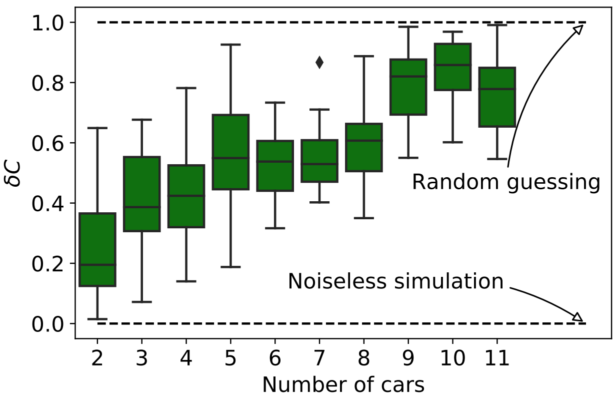

FIG. 5: Performance of the QAOA experiments using an IonQ QPU in comparison to random guessing and an ideal (noiseless) simulation, as function of the number of cars. Results are presented at each system size for $N = 20$ randomly drawn instances, averaged over $M = 10^5$ measurements.

In Fig. 5, we plot the distribution of δC over all N instances for increasing system size. As 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.

D. Approximation Ratio

QAOA was designed to find approximate solutions of combinatorial optimization problems. However, results show that there is no classical polynomial-time algorithm for the binary paint shop problem which finds an approximate solution within any constant factor α from the optimal solution for all instances [32]. This directly raises the question whether QAOA is able to find a constant approximation with polynomially increasing resources. However, since QAOA is a randomized algorithm we first have to adjust the definition of a constant factor approximation algorithm accordingly: we demand from QAOA to provide us with a solution within a factor α of the optimal solution with probability decreasing at most polynomially and level p growing polynomially in the number of cars. Derived from this definition there are two orthogonal strategies to obtain solutions within the performance bound given by α: either we increase the number of samples or we increase the number of QAOA levels p.

In this section, we discuss how the number of samples required to find a constant factor approximation solu-

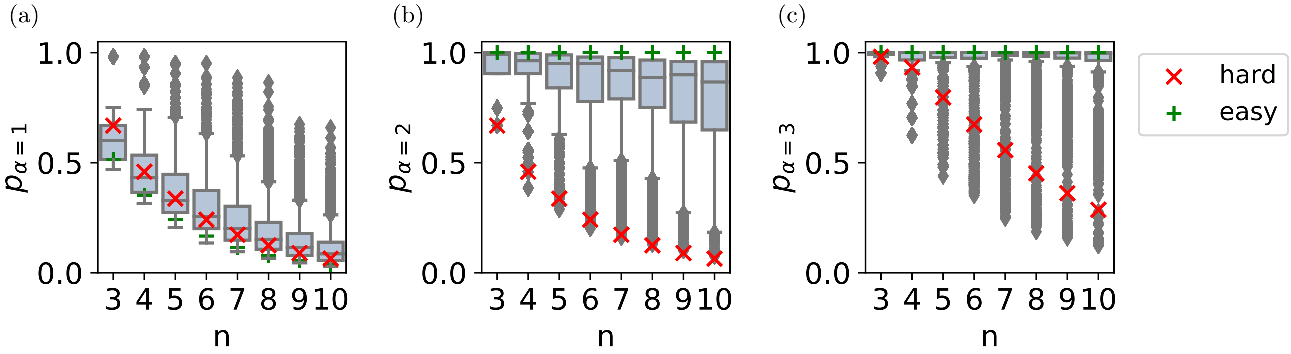

FIG. 6: The distribution of the probability to observe an α-approximation p_α, see Eq. (9), for 5000 random instances of the BPSP and a QAOA circuit with $p = 1$. (a) $\alpha = 1$ (b) $\alpha = 2$ (c) $\alpha = 3$

tion must scale for a fixed number of QAOA levels. To this end we introduce the probability to observe an α-approximation as

$$p_\alpha = \sum_k |a_k|^2 \; \forall \; k \text{ if } E_k < E_\alpha \;, \tag{9}$$

where E_α is the instance-dependent energy required to achieve an α-approximation, a_k the coefficients of the QAOA output state in the computational basis and E_k the corresponding energies $H_\mathrm{P} \, |k\rangle = E_k \, |k\rangle$. For a randomized constant factor approximation algorithm, the probability is only allowed to decrease at most polynomially when increasing the system size. We note that this has to hold for all possible instances and not only on average. Thus, we check how the number of required samples scales in the worst-case scenario. For this we generate 5000 random binary paint shop instances for system sizes from $n = 3$ up to $n = 10$ cars and calculate p_α for various values of α. In Fig. 8, the distribution of p_α is shown as function of the number of cars n. The hardest instances, i.e. the instances with lowest probability p_α, decay exponentially as a function of the number of cars. As there are exponentially many different possible BPSP instances, we cannot carry out such a study for arbitrary large problem sizes. To investigate the scaling for larger problem sizes, we use a proxy instance defined on n cars by the sequence

$$(w_1, w_2, w_3, \ldots w_n, w_n, w_{n-1}, w_{n-2}, \ldots, w_1) \;. \tag{10}$$

We first argue why this instance is notoriously hard to approximate with QAOA for all system sizes, which is why we refer to it as a *hard instance*. This instance can be painted with a single color change when painting the first n cars with the first color and the second n cars with the second color or vice versa. We note that the classical greedy algorithm is able to find the optimal coloring for all problem sizes. After mapping the instance on an Ising Hamiltonian (cf. Fig. 2), we find a 1D chain of n spins with coupling strengths $J = -2$ for all interactions, cf. Fig. 7(b). The ferromagnetic ground state of this spin system has an energy of $E_\mathrm{GS}^\mathrm{(hard)} = -2n + 2$ and thus an α-approximation requires an energy

$$E_\alpha^\mathrm{(hard)} = (-2n + 1) + 2(\alpha - 1) \;. \tag{11}$$

To keep the approximation factor constant, only a constant energy shift is allowed when increasing the system size. This means that QAOA must lower the expectation value of each edge (i, j) when increasing the system size. This behaviour is reflected in Fig. 6, where the scaling of the *hard instance* shows similar scaling properties compared to the hardest random instances. Additionally, [41] argues that QAOA needs to see the whole graph, meaning that the support of reverse causal cones has to be the entire qubit register. As the *hard instance* is a 1D chain, the reverse causal cones only increase linearly with the QAOA level, and as such QAOA requires a high value of p to see the whole graph.

To further illustrate the hardness of the introduced instance, we contrast to the closely related sequence

$$(w_1, w_1, w_2, w_2, \ldots, w_n, w_n) \;, \tag{12}$$

which we refer to as an *easy instance*. This sequence also maps to a 1D spin chain of n coupled spins, but with coupling strengths $J = 1$, as pictured in Fig. 7(a). The anti-ferromagnetic ground state ($\alpha = 1$) with energy of $E_\mathrm{GS}^\mathrm{(easy)} = -n + 1$ represents n color changes and for an α-approximation the energy required is

$$E_\alpha^\mathrm{(easy)} = (-n + 1) + 2(\alpha - 1)n \;. \tag{13}$$

For any constant α, this means that the energy gap is allowed to grow as fast as the energy scale of the problem. If we choose $\alpha = 3/2$, it follows $E_{3/2}^\mathrm{(easy)} = 1$. This energy can be easily achieved with random guessing, i.e. even QAOA with $p = 1$ and $(\beta_1, \gamma_1) = (0, 0)$ achieves a constant factor approximation for this instance, independent of its size, which is in stark contrast to the *hard instance*. This behaviour is also visible in Fig. 6(b) and (c), where we see that we only need to sample once from

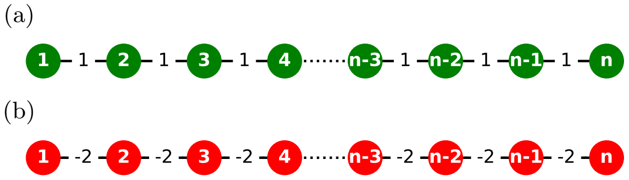

FIG. 7: Spin systems corresponding to the (a) easy instance given in Eq. (12) and the (b) hard instance given in Eq. (10).

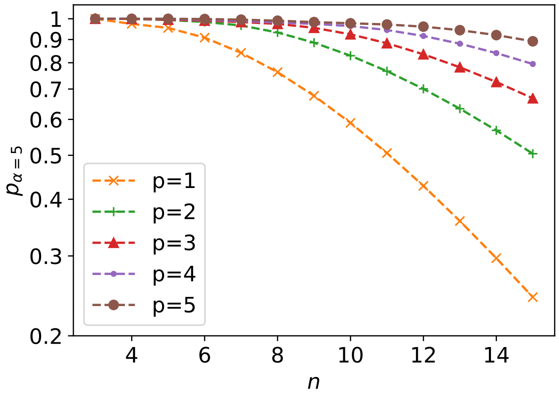

FIG. 8: The probability to see at least approximation with $\alpha = 5$ of the hard instance, see Eq. 12, as function of the system size n.

the QAOA output state to find an α-approximation for $\alpha = 2$ or $\alpha = 3$.

We therefore conjecture that the *hard instance* is actually hard to approximate with QAOA independently from the problem size and thus provides a good proxy to study the worst-case scaling of QAOA's ability to approximately solve the BPSP.

Because of this, we now solely focus on the approximation QAOA can provide for the *hard instance*. In Fig. 8, we show numerical results for $\alpha = 5$ and various values of p. As clearly visible, for fixed p, starting from certain critical problem sizes QAOA requires exponentially many samples to find an α-approximation, and as such QAOA with fixed p is no constant factor approximation algorithm. However, as mentioned earlier, instead of increasing the number of samples, we could also increase the number of QAOA levels. The numerical results suggest that increasing the QAOA levels polynomially is already sufficient to find a constant approximation. We here want to connect with recent results of QAOA on ferromagnetic or anti-ferromagnetic rings for arbitrary number of QAOA levels [21]. In [42], it was shown that such instances can be solved exactly when allowing the number of QAOA levels p to increase linearly in the system size. Due to the broken translational symmetry in our instances, such an approach is not possible. However, the numerical results suggest a similar scaling for the *hard instance*. If true, QAOA would be able to generate a constant factor approximation in polynomial time for this instance.

However, a general statement for all possible instances is absent and thus, an interesting question for the future is how the number of QAOA levels would need to scale for a constant factor approximation for all instances.

V. CONCLUSION

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 a noiseless quantum computation and discussed whether QAOA can achieve approximations to the problem in polynomial time, a task a classical computer cannot achieve.

The experimental results of this paper highlight the deterioration of the quantum algorithms' 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 adaptions 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 [43] or the encoding of QAOA into a lattice gauge model [44] might provide improvements. Furthermore, providing a definite answer on the question whether QAOA is a constant factor approximation algorithm could open up new room for quantum advantage.

As mentioned, in real-world industrial settings, coloring car bodies requires more than two colors. Finding a suitable mapping for the generalized paint shop problem is another task for the future.

ACKNOWLEDGMENTS

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

[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 com-

puting with a superconducting circuit, Nature **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 **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 **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, Science advances **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 **574**, 505 (2019).

[6] L. K. Grover, A fast quantum mechanical algorithm for database search, arXiv preprint quant-ph/9605043 (1996).

[7] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings 35th annual symposium on foundations of computer science* (IEEE, 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, Nature communications **5**, 4213 (2014).

[9] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review 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, Physical Review Research **1**, 033063 (2019).

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

[12] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem, arXiv preprint arXiv::1412.6062 (2014).

[13] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem, arXiv preprint arXiv:1811.08419 (2018).

[14] M. Streif and M. Leib, Comparison of QAOA with quantum and simulated annealing, arXiv preprint arXiv:1901.01903 (2019).

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

[16] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, Benchmarking the quantum approximate optimization algorithm, Quantum Information Processing **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, arXiv preprint arXiv:2004.04197 (2020).

[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, Proceedings of the National Academy of Sciences **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 preprint arXiv:1712.05771 (2017).

[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, Physical Review X **10**, 021067 (2020).

[21] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: a journey through digitalitalization, control, and hybrid quantum variational schemes, arXiv preprint arXiv:1906.08948 (2019).

[22] A. Bapat and S. Jordan, Bang-bang control as a design principle for classical and quantum optimization algorithms, arXiv preprint arXiv:1812.02746 (2018).

[23] M. B. Hastings, Classical and quantum bounded depth approximation algorithms, arXiv preprint arXiv:1905.07047 (2019).

[24] S. Bravyi, D. Gosset, and R. Movassagh, Classical algorithms for quantum mean values, arXiv preprint arXiv:1909.11485 (2019).

[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 preprint arXiv:1812.04170 (2018).

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

[27] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, arXiv preprint arXiv:1602.07674 (2016).

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

[29] A. Lucas, Ising formulations of many np problems, Frontiers in Physics **2**, 5 (2014).

[30] T. Epping, W. Hochstättler, and P. Oertel, Some results on a paint shop problem for words, Electronic Notes in Discrete Mathematics **8**, 31 (2001).

[31] T. Epping, W. Hochstättler, and P. Oertel, Complexity results on a paint shop problem, Discrete Applied Mathematics **136**, 217 (2004).

[32] 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, 2013) pp. 205–217.

[33] 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* (2002) pp. 767–775.

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

[35] S. D. Andres and W. Hochstättler, Some heuristics for

the binary paint shop problem and their expected number of colour changes, Journal of Discrete Algorithms **9**, 203 (2011).

[36] F. Meunier and B. Neveu, Computing solutions of the paintshop–necklace problem, Computers & operations research **39**, 2666 (2012).

[37] K. Wright, K. Beck, S. Debnath, J. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. Pisenti, M. Chmielewski, C. Collins, *et al.*, Benchmarking an 11-qubit quantum computer, Nature communications **10**, 1 (2019).

[38] *AWS Braket*, available at `https://aws.amazon.com/braket/`.

[39] D. Maslov, Basic circuit compilation techniques for an ion-trap quantum machine, New Journal of Physics **19**, 023035 (2017).

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

[41] E. Farhi, D. Gamarnik, and S. Gutmann, The quantum approximate optimization algorithm needs to see the whole graph: A typical case, arXiv preprint arXiv:2004.09002 (2020).

[42] W. W. Ho and T. H. Hsieh, Efficient variational simulation of non-trivial quantum states, SciPost Phys **6**, 029 (2019).

[43] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Obstacles to state preparation and variational optimization from symmetry protection, arXiv preprint arXiv:1910.08980 (2019).

[44] W. Lechner, Quantum approximate optimization with parallelizable gates, arXiv preprint arXiv:1802.01157 (2018).

Appendix A: Classical heuristics

1. Greedy algorithm

For an instance of the binary paint shop instance of finite size, the greedy algorithm exactly solves a spin system which is different from the spin system presented in Sec. III A. It is generated in the following way: for each first occurrence of a car i in the sequence, an interaction J_{ij} is added between the qubit c_i representing the car i and the qubit j representing the car c_j previous to car i in the sequence. If car j also appears for the first time in the sequence, the interaction is $J_{ij} = -1$, and otherwise $J_{ij} = +1$. The new spin system includes $n-1$ interactions. Moreover, as each first occurrence of a car only has a single predecessor, the graph representing the new spin system is cycle-free and thus can be exactly solved in linear time, which the greedy algorithm also does.

2. 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 proven average performance for $n \to \infty$ of

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

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

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

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

3. 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 colored optimally. After that, the occurrences of the last deleted car are added back to the sequence. While keeping the already colored cars fixed, the new car is colored optimally. This is repeated until the whole sequence is colored.

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

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

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

As single color change yields an increase of energy by 2, we use the results on the red-first heuristic to determine the average energy to

$$\lim_{n\to\infty} \mathbb{E}_{\mathrm{RG}}(E) = -\frac{6n}{5}\ .$$

Appendix B: Characteristics of the spin system

In this section of the 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. 2.

a. Coupling strengths We show 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 finding an anti-ferromagnetic coupling ($J = 1$) and that the probability of finding coupling strengths $J = -2$ and $J = 2$ is vanishing when $N \to \infty$. By construction, a ferromagnetic coupling is generated whenever two cars in the sequence are neighbors and both occur for the first time or both already have occurred, i.e. the probability of a ferromagnetic coupling can be expressed by

$$P(J=-1) = N \sum_{<ij>} P(00) + P(11),$$

where $P(00)$ is the probability that both cars occur the first time, and $P(11)$ that both cars appear the second time. This can be reformulated as

$$P(J=-1) = N \sum_{<ij>} \left[\left(1 - \frac{i-1}{N}\right)\left(1 - \frac{j-1}{N}\right)\frac{1}{N^2} + \frac{i-1}{N}\frac{j-1}{N}\frac{1}{N^2} \right]$$

For $N \to \infty$,

$$2 \lim_{N\to\infty} \sum_{i}^{N} \frac{i^2}{N^3} = 2\int_0^1 \mathrm{d}x x^2 = \frac{2}{3},$$

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

$$\begin{aligned} P(J=+1) &= N \sum_{<ij>} P(01) + P(10) \\ &= \left[\left(\frac{i-1}{N}\right)\left(1 - \frac{j-1}{N}\right)\frac{1}{N^2} + \frac{i-1}{N}\left(1 - \frac{j-1}{N}\right)\frac{1}{N^2} \right], \end{aligned}$$

which for $N \to \infty$ leads to

$$\lim_{N\to\infty} = 2 \lim_{N\to\infty} \sum_{i}^{N} \frac{i}{N^2} - \frac{i^2}{N^3} = 2\int_0^1 \mathrm{d}x(x - x^2) = \frac{1}{3}.$$

Thus, the spin system formulation of the binary paint shop problems have (for $N \to \infty$) integer ferromagnetic or anti-ferromagnetic couplings with probabilities 2/3 and 1/3 respectively.

b. Average degree In Fig. 9, we show the average degree of 1000 randomly drawn binary paint shop instances while increasing the system size. As can be seen, the average degree approaches 4.

Appendix C: Extension of tree-QAOA

In this section, 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,

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

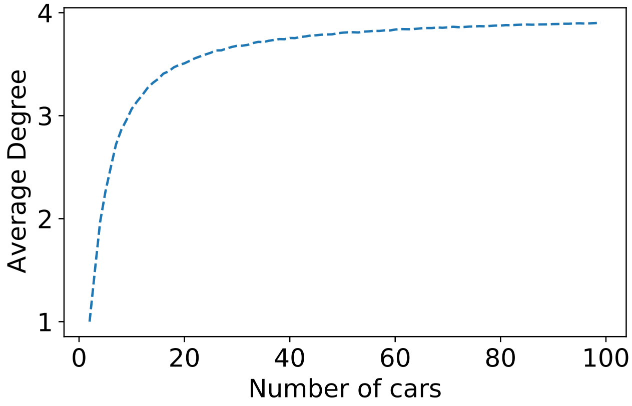

FIG. 9: Average degree of the binary paint shop instances. Averaged over 1000 randomly drawn instances for each system size.

QAOA level p	γ_1	β_1	γ_2	β_2	γ_3	β_3	γ_4	β_4	γ_5	β_5
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

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

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 exist a transformation

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

with $U_{\mathrm{T}} = \bigotimes_j^k X_j$ 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

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

This signifies 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 $p = 1$ circuit such that only native gates are used. In this section, we show how this can be done for an arbitrary number of QAOA levels p. By inserting Hadamard gates, the circuits transforms to

$$\begin{aligned} |\Psi\rangle^{p=1}_{\mathrm{QAOA}} &= U_{\mathrm{X}}(\beta_p)U_{\mathrm{ZZ}}(\gamma_p)\ldots U_{\mathrm{X}}(\beta_2)U_{\mathrm{ZZ}}(\gamma_2)U_{\mathrm{X}}(\beta_1)U_{\mathrm{ZZ}}(\gamma_1)\,|+\rangle \\ &= U_{\mathrm{X}}(\beta_p)\mathrm{HH}U_{\mathrm{ZZ}}(\gamma_p)\mathrm{HH}\ldots\mathrm{HH}U_{\mathrm{X}}(\beta_2)\mathrm{HH}U_{\mathrm{ZZ}}(\gamma_2)\mathrm{HH}U_{\mathrm{X}}(\beta_1)\mathrm{HH}U_{\mathrm{ZZ}}(\gamma_1)\mathrm{H}\,|0\rangle \\ &= U_{\mathrm{X}}(\beta_p)\mathrm{H}U_{\mathrm{XX}}(\gamma_p)\ldots U_{\mathrm{Z}}(\beta_2)U_{\mathrm{XX}}(\gamma_2)U_{\mathrm{Z}}(\beta_1)U_{\mathrm{XX}}(\gamma_1)\,|0\rangle \\ &= U_{\mathrm{X}}(\beta_p-\pi)U_{\mathrm{Y}}(\pi/2)U_{\mathrm{XX}}(\gamma_p)\ldots U_{\mathrm{Z}}(\beta_2)U_{\mathrm{XX}}(\gamma_2)U_{\mathrm{Z}}(\beta_1)U_{\mathrm{XX}}(\gamma_1)\,|0\rangle\ , \end{aligned}$$

including only native gates.