ACADEMIA QUANTUM

Research Article
Published: 2025-09-12
https://doi.org/10.20935/AcadQuant7900

Symmetry-based quantum algorithms for open-shop scheduling with hard constraints

Lennart Binkowski[1], Gereon Koßmann[2,3], Christian Tutschku[3], René Schwonnek[1,*]

Academic Editors: Misha Erementchouk, Yuhua Duan

Abstract

Encoding hard-constrained optimization problems into a variational quantum algorithm often turns out to be a challenging task. In this work, we provide a solution for the class of open-shop scheduling problems (OSSPs), which we achieve by rigorously employing the symmetries of the classical problem. An established approach for encoding the hard constraints of the closely related traveling salesperson problem (TSP) into mixer Hamiltonians was recently given by Hadfield et al.'s Quantum Alternating Operator Ansatz (QAOA). For the OSSP, which contains TSP as a special case, we show that desired properties of similarly constructed mixers can be directly linked to a purely classical object: the group of feasibility-preserving bit value permutations. We also outline a generic way to construct QAOA-like mixers for these problems. We further propose a new variational quantum algorithm that incorporates the underlying group structure more naturally and, as a proof of principle, implement our new algorithm for a small OSSP instance on an IBM Q System One. Unlike the generic QAOA, our algorithm allows for bounding the amount and the domain of parameters necessary to reach every feasible solution from above: If J jobs should be distributed, optimizing at most $J(J-1)^2/2$ parameters should suffice to reach the optimum with certainty.

Keywords: *quantum computing, open-shop scheduling, quantum approximate optimization, quantum alternating operator ansatz*

Citation: Binkowski L, Koßmann G, Tutschku C, Schwonnek R. Symmetry-based quantum algorithms for open-shop scheduling with hard constraints. *Academia Quantum* 2025;2. https://doi.org/10.20935/AcadQuant7900

1. Introduction

Logistic and scheduling tasks are a major branch within the collection of hard optimization problems for relevant industrial applications. A prominent representative of those is the *open-shop scheduling problem* $\mathrm{OSSP}(M,T,J)$, which we consider in this work: Given M machines with T time slots each, one has to distribute J jobs such that every job gets performed precisely once and no position is filled with more than one job. Not only is the OSSP at the mathematical core of many real-world problems, it also prominently incorporates the well-known *traveling salesperson problem* (TSP) as a subclass. More specifically, it holds that $\mathrm{TSP} = \mathrm{OSSP}(1,T,T)$.

Its practical relevance as well as the fact that solving an instance of OSSP can easily turn out to be a hard task for classical computers, make it an interesting target for the application of quantum algorithms. Confronted with the restricted capabilities of available quantum computer architectures, the class of *variational quantum algorithms* [1–5] (VQAs) is receiving a particular amount of attention, since it promises to yield tools for tackling computational challenges within the NISQ [6, 7] era, where qubit numbers are limited and the lack of error correction only allows for shallow circuits. In a nutshell, VQAs utilize parametrized quantum circuits to reliably explore a manifold of target states and to reduce the original optimization task to optimizing the circuits' parameters. The parameter updates themselves are determined on a classical computer while the quantum device solely serves as a sampling machine to estimate relevant expectation values. Outsourcing the parameter updates to the classical computer typically results in shallow circuits and low demands on the coherence time, as quantum states are measured immediately after their parametrized preparation.

In addition to the shortcomings of the current quantum hardware, there are additional, conceptual hurdles to overcome. The basic input for applying a VQA to an optimization problem is an encoding of an objective function f into a multi-qubit objective Hamiltonian H_f such that—in case of a minimization problem—optimal solutions correspond to the smallest expectation value attainable by a quantum state [8]. In contrast to other popular problems, like MAX-CUT or 3-Sat, the OSSP additionally demands us to not only consider the encoding of an objective function, but also the encoding of further constraints. As a consequence, a VQA ideally has to perform its optimization only on a subspace of 'allowed/feasible' quantum states, i.e., those that respect constraints. While in practice the presence of noise and gate infidelities prevents us from truly restricting the optimization to such a feasible subspace, the constraint-oriented quantum algorithmic design remains essential: Feasibility-preserving gate operations, even if imperfect, can concentrate just enough amplitude within the feasible subspace to study the objective function's structure among feasible states. This is especially important whenever the number of feasible solutions is significantly smaller than the number of all possible

[1]Institute for Theoretical Physics, AG Quantum Information, Leibniz University Hannover, Hannover, Germany.
[2]Institute for Quantum Information, RWTH Aachen University, Aachen, Germany.
[3]Fraunhofer Institute for Industrial Engineering IAO, Stuttgart, Germany.
*email: rene.schwonnek@itp.uni-hannover.de

variable assignments, as is the case for the OSSP. The major contribution of this work is a systematic analysis of symmetry structures in the OSSP, which will enable us to design a class of VQA algorithms whose logical components, by construction, restrict all parametrized states to the feasible subspace.

In general, there are two strategies for handling constrained optimization problems:

- *Softcoded* constraints. Any assignment is considered feasible, but the constraints enter the objective function as additional terms and penalize assignments that were originally infeasible. The modified objective function results in a more complex objective Hamiltonian. In exchange, the ground state search can be conducted in the entire qubit space without the necessity of preserving feasibility throughout the routine. A comprehensible introduction and case study using Qiskit and IBM quantum computers can be found in [9].
- *Hardcoded* constraints. The objective function (and thus the objective Hamiltonian) is left unmodified, while the ground state search is restricted to the subspace corresponding to feasible solutions. Preservation of feasibility is typically ensured by additional gates representing the classical constraints.

Most VQAs such as the *variational quantum eigensolver* [3] or the *quantum approximate optimization algorithm* [10] are originally formulated for unconstrained problems and are thus merely applicable to instances without or softcoded constraints. Countless use case studies have been executed on unconstrained problems like MAX-CUT (see, e.g., [11, 12]) and on constrained problems with softcoded constraints, including logistic problems like the TSP [13] and vehicle routing [14, 15], as well as various scheduling problems [16–18].

However, these and several other case studies indicate that softcoding the constraints often leads to suboptimal optimization landscapes or issues with feasibility [19–22]. This is why quantum algorithms with built-in possibilities for hardcoding constraints received increasing interest. Most notably, Hadfield et al. [23] extended the quantum approximate optimization algorithm to the *quantum alternating operator ansatz* (QAOA), which can also be applied to hardcoded constrained optimization problems. Unlike its predecessor, the QAOA is formulated with a problem-dependent feasibility-preserving mixer and is therefore sensitive to different feasibility structures. Hadfield et al. already gave explicit mixer constructions for several graph problems, the TSP, and single-machine scheduling. Later on, suitable mixers were also constructed for various other problem classes such as the knapsack problem [24], capacitated vehicle routing [25], and the facility locating problem [26].

Albeit there is a massive catalog [27] of classical constraint analysis available for these problems, concrete mixers were mainly heuristically designed (with a few exceptions like the Grover-mixer framework [28]). Often enough, the underlying feasibility structure was not completely exploited. One particular interesting construction for scheduling-type problems is the *constraint graph model* [29, 30]. Here the bit strings are identified with vertices, joint by edges corresponding to the constraints. This allows for investigating the feasibility structure with well-known results from graph theory. Most notably, one can identify graph automorphisms with feasibility-preserving bit permutations. With this work, we incorporate the entire structure into a more refined view on the QAOA and also come up with a new VQA design for OSSP instances.

In Section 2 we rigorously introduce the notion of combinatorial optimization problems, give a detailed formulation of the OSSP, cover the constraint graph model, which we will use extensively in our subsequent analysis, and describe general encoding strategies for tackling combinatorial optimization problems with the aid of quantum computers.

In Section 3 we present our theoretical and numerical results. We first apply the constraint graph model to the general OSSP, embedding the notions of solutions and solution-preserving functions into the graph-theoretical language. Utilizing this additional point of view, we fully characterize the group F of feasibility-preserving bit permutations. Furthermore, we uncover block structures within the OSSP-constraints. This ultimately reveals that F acts transitively on the set of all solutions. We then draw the connection between the classical description of F and specific VQA designs. Firstly, we review how the QAOA works in general and proceed with a discussion of its main ingredients. We give refined definitions for 'phase separator' and 'mixer' gates and detail a general construction for suitable mixers from elements of F. In particular, the transitive action of F is directly translated into substantial mixing properties. Second, we introduce a new VQA suitable for OSSP instances. It is fundamentally based on decomposing bit value permutations into products of transpositions. In contrast to the QAOA, we can bound the number of parameters necessary to reach every possible solution: while the number of OSSP solutions is $O(J!)$, only $O(J^3)$ parameters are necessary. We complement these theoretical results with two proof-of-principle numerical experiments. The first one consists of an extensive comparison between our VQA and standard QAOA with softcoded constraints on an OSSP(2,2,4) instance. In this noiseless simulation, we clearly observe the difficulties the standard approach has with the additional OSSP constraints; an issue that is, by design, circumvented by our VQA. The second experiment highlights the implementability of our VQA on today's quantum devices. For an OSSP(1,3,3) instance, we implement our VQA on the IBM Q System One, and, additionally, on a classical noisy simulator.

2. Material and methods

For the readers' convenience we will review the basic notion of combinatorial optimization problems (COPs), especially the open-shop scheduling problem. We also briefly introduce the constraint graph model and cover the very basics of problem encoding onto quantum computers. Throughout this work, we will use the shorthand $[N] \coloneqq \{1, \ldots, N\}$ where N is any natural number.

2.1. Constrained combinatorial optimization

In the following, we restrict to minimization problems, as maximization tasks may be considered analogously. Then a generic COP of *size N* with A constraints is of the form

$$\min_{z \in Z(N)} f(z) \quad \text{s.t. } c_a(z) = 1, \quad a \in [A], \tag{1}$$

where

- $Z(N) \coloneqq \{0,1\}^N$ denotes the set of all bit strings of length N;

- $f : Z(N) \to \mathbb{R}$ is the *objective function* to be minimized;
- $c_a : Z(N) \to \{0,1\}$ are the *constraints*.

Accordingly, a bit string z is said to fulfill a constraint c_a if $c_a(z) = 1$. In the following, we will refer to COPs formally as triples $(N, f, \{c_a\})$. For a given COP $\mathcal{C}$, we define its *solution set* as the set of all bit strings, fulfilling every constraint:

$$S(\mathcal{C}) \coloneqq \{z \in Z(N) \,:\, c_a(z) = 1, \quad a \in [A]\}. \tag{2}$$

Furthermore, the *optimal solution set* is the set of all solution bit strings minimizing f, i.e.,

$$S_{\min}(\mathcal{C}) \coloneqq \left\{ z \in S(\mathcal{C}) \,:\, f(z) = \min_{z' \in S(\mathcal{C})} f(z') \right\} \subseteq S(\mathcal{C}). \tag{3}$$

We now examine particular types of constraints: For two bit strings $z, z' \in z(N)$ the bit-wise *and* operation produces a new bit string $z \wedge z' \in z(N)$. Let $|z|$ denote the *Hamming weight* of z. For an index subset $I \subseteq [N]$, we define the bit string $z_I \in z(N)$ to fulfill $z_n = 1$ if $n \in I$. The *one-hot constraint* and the *at-most-one constraint* associated with the index set I are

$$\zeta_I(z) \coloneqq \begin{cases} 1, & \text{if } |z \wedge z_I| = 1 \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \tag{4}$$

$$\eta_I(z) \coloneqq \begin{cases} 1, & \text{if } |z \wedge z_I| \leq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2.2. Open-shop scheduling

Let us formally introduce the OSSP, depending on the three parameters

- M is the number of machines;
- T is the number of time slots per machine;
- J is the number of jobs.

The J jobs should be distributed to the MT available positions such that

J represents every job gets performed precisely once;
P indicates that no position is filled with more than one job.

Note that OSSP$(1, T, T)$ has the exact same structure as the T-city TSP.

In order to bring OSSP $\coloneqq$ OSSP(M, T, J) into the form **(1)**, we introduce $N = MTJ$ bits, identify $[N] \cong [M] \times [T] \times [J]$, and set

$$z_{mtj} \coloneqq \begin{cases} 1, & \text{if job } j \text{ runs on machine } m \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The *job assignment constraints* **J** and the *position assignment constraints* **P** then read

$$\mathbf{J} : \sum_{m=1}^{M} \sum_{t=1}^{T} z_{mtj} = 1, \quad j \in [J] \quad \text{and} \tag{7}$$

$$\mathbf{P} : \sum_{j=1}^{J} z_{mtj} \leq 1, \quad (m,t) \in [M] \times [T]. \tag{8}$$

Thus **J** and **P** are one-hot and at-most-one constraints, respectively. Namely, we have

$$\mathbf{J} : c_j \coloneqq \zeta_{\Delta_{(j)}}, \, j \in [J] \quad \text{and} \tag{9}$$

$$\mathbf{P} : c_{m,t} \coloneqq \eta_{\Delta_{(m,t)}}, \, (m,t) \in [M] \times [T] \tag{10}$$

with *job blocks* $\Delta_{(j)} \coloneqq [M] \times [T] \times \{j\}$ and *position blocks* $\Delta_{(m,t)} \coloneqq \{m\} \times \{t\} \times [J]$. The constraints are equivalently captured in the coordinate relation

$$(m,t,j) \sim (m',t',j') :\Longleftrightarrow (m = m' \,\wedge\, t = t') \,\vee\, j = j'. \tag{11}$$

This means that if $(m,t,j) \sim (m',t',j')$ then any feasible solution z cannot have both bits z_{mtj} and $z_{m't'j'}$ set to one. Note that '$\sim$' is reflexive and symmetric, but not transitive. The OSSP solution set (expressed as a coordinate set) is explicitly given by

$$\bigcup \left\{ \{(m_j, t_j, j) : j \in [J]\} : (m_1, t_1) \neq \cdots \neq (m_J, t_J) \in [M] \times [T] \right\}. \tag{12}$$

Therefore, it possesses $(MT)!/(MT - J)!$ solutions.

Very common objectives of scheduling tasks are the minimization of the schedule's makespan or of the total completion time. For our formulation of the problem where jobs do not have to run on all machines and each job's completion takes up exactly one time slot, these two objective are, however, not of much interest as equally balancing the assignment of jobs over all machines at the earliest time slots readily minimizes both quantities. Instead, we introduce machine- and time-sensitive execution costs for each job, $\omega_{mtj} \in \mathbb{R}$, and impose the objective of minimizing the schedule's cumulative costs:

$$f : z(N) \to \mathbb{R}, \quad z \mapsto \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{j=1}^{J} \omega_{mtj} z_{mtj} =: \boldsymbol{\omega} \cdot z, \tag{13}$$

where $\boldsymbol{\omega} \in \mathbb{R}^N$ represents the cost vector. Therefore, the objective function is linear and reduces to evaluating the inner product of $\boldsymbol{\omega}$ with the bit string argument.

2.3. Constraint graph model

The solution set of a COP $\mathcal{C} = (N, f, \{c_a\})$ can be further studied by introducing the so-called *constraint graph* [29, 30]. First we identify $[N]$ with a vertex set $V = \{v_1, \ldots, v_N\}$. Moreover, the bit string z_I, associated to the subset $I \subseteq [N]$, is identified with the subset of vertices $V_I \coloneqq \{v_n \,:\, n \in I\} \subseteq V$.

The constraints enter as edges of the graph: Two vertices v_m and v_n are joint by an edge if there is a constraint that prohibits the two bits z_m and z_n from taking the value 1 at the same time. Since this construction is symmetric, the constraint graph $G = (V, E)$ is undirected. Thus, the edges are unordered pairs of vertices and we may address an edge $e \in E$ with its end points. Assuming that we have defined a coordinate relation '$\sim$' similar to **(11)**, we can express the set of edges simply as $E = \{(v_m, v_n) \,:\, m \sim n\}$.

For one-hot and at-most-one constraints we can be more concrete: If $\zeta_I \in \{c_a\}$ or $\eta_I \in \{c_a\}$ all vertices v_n with $n \in I$ are mutually connected, i.e., they form a clique in the constraint graph. **Figure 1** shows the constraint graph of an OSSP$(2, 2, 4)$-instance.

A solution to $\mathcal{C}$ does not violate any constraint. Thus it corresponds to an independent set (or coclique) in the associated constraint graph $G(\mathcal{C})$ (see **Figure 2** for such a graphical solution to the OSSP$(2, 2, 4)$ instance). If $\mathcal{C}$ incorporates J one-hot constraints and some additional at-most-one constraints (such as the OSSP), every solution has Hamming weight J. Accordingly, we call any vertex subset $W \subseteq V$ a *solution* to $\mathcal{C}$ if

(i) $|W| = J$;
(ii) W is an independent set.

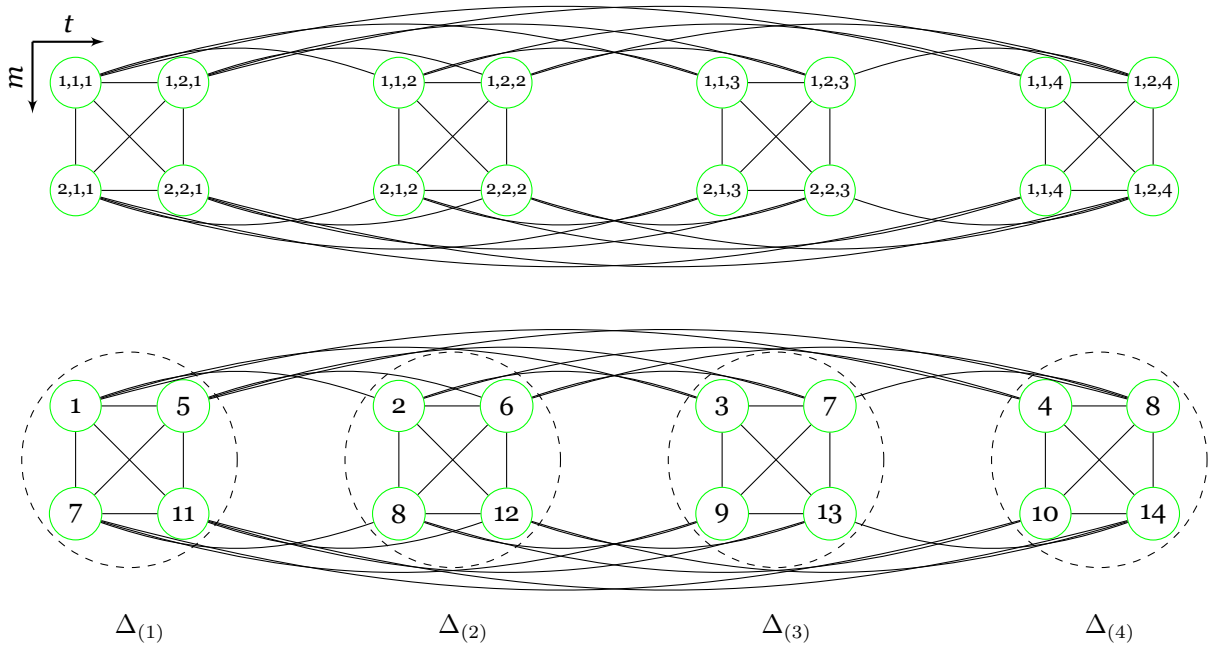

Figure 1 • OSSP(2,2,4) constraint graph. In the above graph, the vertices are labeled using the coordinate system (m,t,j). In addition, the three job blocks $\Delta_{(j)}, j \in [4]$ are depicted.

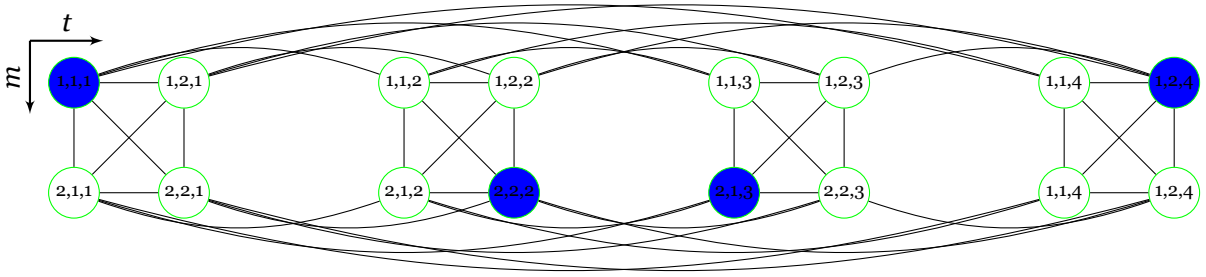

Figure 2 • OSSP(2,2,4) constraint graph with feasible solution. The colored vertices are a maximally independent set and thus constitute a feasible solution.

Moreover, we say that a permutation $\rho : V \to V$ *preserves feasibility* if $\rho(W)$ is again a solution whenever $W \subseteq V$ is a solution. We denote with F the set of all feasibility-preserving permutations. As the composition of two feasibility-preserving permutations preserves feasibility again and the identity also preserves feasibility, one readily deduces that F is a subgroup of $\mathrm{Sym}(V)$. Via the identification of bit strings with vertex subsets, F can also be interpreted as acting on the solution set $S(\mathcal{C})$, i.e., on a set of bit strings. However, the notions of solutions and of F can also be considered independently of any underlying COP.

We remark that we could alternatively have declared the complement graph of $G(\mathcal{C})$ as the actual constraint graph. Then, solutions would correspond to cliques instead of cocliques, but the underlying structure would not change since a graph and its complement have the same automorphism group. This equivalent point of view is noteworthy since many results in graph theory are formulated in terms of cliques. However, we proceed with our original definition.

2.4. Encoding on quantum computers

In the following, we recall the standard encoding procedure for addressing a general COP $\mathcal{C} = (N, f, \{c_a\})$ with the aid of a quantum computer. All relevant quantum computing essentials are covered in [31]. First, we identify each bit string z with a computational basis state $|z\rangle$ of the N-qubit space $\mathcal{H} := \mathbb{C}^{2^N}$. This induces a representation of functions over $Z(N)$ as linear operators on $\mathcal{H}$. Namely, the classical objective function f is mapped to an objective Hamiltonian, diagonal in the computational basis

$$f \mapsto C := \sum_{z \in Z(N)} f(z)\, |z\rangle\langle z| \,. \tag{14}$$

For unconstrained problems the minimization task is then equivalent to finding a computational basis state, which is a ground state of C. However, in the constrained case, the ground state search has to be restricted to the *solution space*

$$\mathcal{S} := \mathrm{span}\{|z\rangle \,:\, z \in S(\mathcal{C})\} \subsetneq \mathcal{H}. \tag{15}$$

The *optimal solution space*

$$\mathcal{S}_{\min} := \mathrm{span}\{|z\rangle \,:\, z \in S_{\min}(\mathcal{C})\} \tag{16}$$

then is a subspace of $\mathcal{S}$ and is the eigenspace of $C|_{\mathcal{S}}$, corresponding to its smallest eigenvalue. Unlike $\mathcal{H}$, $\mathcal{S}$ does not admit any favorable tensor product structure in general. This is precisely what makes constrained optimization more challenging on a quantum computer.

3. Results

In this section we first investigate the constraint graph of an OSSP instance. Our focus lies on determining the feasibility-preserving subgroup F and its properties. For brevity, we will not distinguish between the COP and its constraint graph in what follows.

Subsequently, we elaborate on the QAOA method for tackling combinatorial optimization with the aid of quantum computers. Here, we connect our notion of feasibility-preserving groups to desirable properties of QAOA circuits. These theoretical considerations cumulate in the formulation of our new VQA, specifically tailored to the OSSP, with its guaranteed upper bound on variational parameters required to reach the optimal solution.

Finally, as a proof of principle, we numerically test our proposed VQA on a 16-qubit OSSP instance in a noiseless simulation as well as on a 9-qubit instance on the IBM Q System One, i.e., on real quantum hardware. The latter experiment is accompanied by a classical simulation with a noise model.

3.1. Feasibility-preserving graph automorphisms

Let $G = (V, E)$ be a graph. Recall that a graph automorphism is a bijection $\varphi : V \to V$ such that φ as well as φ^{-1} preserve adjacency. One can actually prove that any bijective graph homomorphism between G and itself is already a graph automorphism. We start with the general observation that graph automorphisms are always feasibility-preserving.

Proposition 1 ([32]). *Let $G = (V, E)$ be a graph with solution set S. It holds that* $\mathrm{Aut}(G) \subseteq F$.

Proof. Let $\varphi \in \mathrm{Aut}(G)$ and let $W \in S$ be arbitrary. Since $\varphi : V \to V$ is bijective, it holds that $|\varphi(W)| = |W| = J$. Let $\varphi(v), \varphi(w) \in \varphi(W)$. Since W is an independent set, it holds that $vw \notin E$. With the isomorphism property of φ^{-1}, it follows that $\varphi(v)\varphi(w) \notin E$; hence $\varphi(W)$ is again an independent set. This shows that $\mathrm{Aut}(G) \subseteq F$. □

For general graphs G, F will be a strict superset of $\mathrm{Aut}(G)$. In case of the OSSP graph, however, equality holds. In order to prove this we utilize a simple auxiliary result.

Proposition 2 ([32]). *Let $G = (V, E)$ be a graph with solution set S. If for all non-adjacent $v, w \in V$ there exists $W \in S$ such that $v, w \in W$, then* $\mathrm{Aut}(G) = F$.

Proof. By Proposition 1 it suffices to show that $F \subseteq \mathrm{Aut}(G)$. Let $\rho \in F$ and $v, w \in V$ with $\rho(v)\rho(w) \notin E$. Then, there exists $W \in S$ so that $\rho(v), \rho(w) \in W$. Since F is a group, ρ^{-1} is also feasibility-preserving, hence $v, w \in \rho^{-1}(W) \in S$ are non-adjacent. Thus the bijective map ρ fulfills $(\rho(v)\rho(w) \notin E \Rightarrow vw \notin E)$, which is an equivalent characterization of a graph automorphism. □

Consider again the OSSP solution set **(12)**. Note that any pair (m, t, j), (m', t', j') corresponding to non-adjacent vertices, i.e., $(m, t) \neq (m', t')$ and $j \neq j'$, can be augmented to form a solution to the OSSP. Therefore, we can apply 2 and conclude that $\mathrm{Aut}(\mathrm{OSSP}) = F$.

3.2. Determining F

Since the relation **(11)** is divided in two logically disjoint parts, one concludes that the constraint graph is given by the Cartesian product [33] of the two graphs K^P and K^J for $P := MT$, where K^n is the complete graph with n vertices. With some effort one can show the following theorem which we owe to personal correspondence with Benjamin Sambale.

Theorem 1. *Let G be a graph and $H_1, \ldots, H_k$ be representatives of the isomorphism types of indecomposable components [33] of G.*
Let m_i be the number of indecomposable components of G that are isomorphic to H_i. Then, it holds

$$\mathrm{Aut}(G) \cong \bigtimes_{i=1}^{k} \big(\underbrace{[\mathrm{Aut}(H_i) \times \cdots \times \mathrm{Aut}(H_i)]}_{m_i\text{- times}} \rtimes S_{m_i} \big). \tag{17}$$

Theorem 1 states that knowing the indecomposable components of a graph reduces the problem of determining its automorphism group to the symmetries of the indecomposable components. In the situation of the OSSP we are mostly done since it is not difficult to show that complete graphs are indecomposable. The automorphism group of complete graphs simply is the whole symmetric group over its vertex set because all vertex permutations are graph automorphisms. Therefore, the automorphism group of the OSSP is given by

$$F = \mathrm{Aut}(\mathrm{OSSP}) \cong \begin{cases} (S_J \times S_J) \rtimes S_2 & P = J, \\ S_P \times S_J & P > J. \end{cases} \tag{18}$$

We observe that the *busy* case ($P = J$) differs structurally from the *non-busy* case ($P > J$). For the latter case we simply obtain an automorphism group that is the direct product of *position permutations* and *job permutations*. However, in the busy case, we obtain a *wreath product* [34] structure. There is an additional non-trivial automorphism that interchanges between position and jobs. In our subsequent analysis, however, we will consider the subgroup

$$F' := S_J \times S_J \leq (S_J \times S_J) \rtimes S_2 \tag{19}$$

in order to treat both cases similarly. The concrete action of an element $(\sigma, \tau) \in S_P \times S_J$ on a coordinate tuple (m, t, j) is given by

$$^{(\sigma,\tau)}(m, t, j) = (\sigma(m, t), \tau(j)). \tag{20}$$

We now argue that $S_P \times \{\mathrm{id}\} \leq F$ acts transitively on the solution set **(12)**. Let $s, s' \subset [M] \times [T] \times [J]$ be two solutions, hence there exist $m_1, \ldots, m_J, m'_1, \ldots, m'_J \in [M]$ and $t_1, \ldots, t_J, t'_1, \ldots, t'_J \in [T]$ such that

$$s = \{(m_1, t_1, 1), \ldots, (m_J, t_J, J)\} \text{ and} \tag{21}$$

$$s' = \{(m'_1, t'_1, 1), \ldots, (m'_J, t'_J, J)\}. \tag{22}$$

Since S_P acts P-transitively on $[M] \times [T]$ and $P \geq J$, there exists $\sigma \in S_P$ such that

$$^{\sigma}(m_j, t_j) = (m'_j, t'_j) \tag{23}$$

holds for all $j \in [J]$, i.e., $^{(\sigma,\mathrm{id})}s = s'$. Thus we have just concluded

Theorem 2 ([32, 35]). *The action of F on the solution set S is transitive for the OSSP.*

3.3. Block structure

We further characterize the group action of F (resp. F') via block systems. Given a group G acting on some set X, a subset $\Delta \subset X$ with $1 < |\Delta| < |X|$ is called a *block* [36] of G if for all $g \in G$ it holds that ${}^g\Delta = \Delta \vee {}^g\Delta \cap \Delta = \emptyset$. A partition of X into blocks of G is then called a *block system*.

It is immediately clear that the collection of job blocks $\Delta_{(j)}$ and of position blocks $\Delta_{(m,t)}$ each form a block system of F (resp. F'). Furthermore, one readily verifies that F (resp. F') acts transitively on $[M]\times[T]\times[J]$. Recall that the job and position blocks result from one-hot and at-most-one constraints and are therefore cliques in the constraint graph. That is, all vertices in a block are adjacent, which implies that a solution is a subset $s \subset [M] \times [T] \times [J]$ such that each element in s belongs to exactly one block in each of the two partitions into position blocks and job blocks. In both cases this yields a bijection between elements in s and the job blocks. In the busy case there is an additional bijection between elements in s and the position block while in the non-busy case s only occupies a subset of all position blocks.

Since each element of a solution exactly corresponds to one position and to one job block, we can capture the action of F (resp. F') on the solution set S equivalently as its action on the blocks. Here, ([36], Proposition 1.37) states that if the blocks are maximal with respect to inclusion (which they are here), then the block-wise action of F is primitive, i.e., it does not possess blocks on its own.

We lastly focus on the busy case. Since the normal action of S_J on J elements is *sharply transitive*, for every two blocks there is exactly one element in S_J that maps between them. The solutions are now precisely the $J!$ permutations of these blocks. Furthermore, each of the two copies of S_J is a stabilizer for one of the block structures and is also a normal subgroup of F'. Therefore we can identify the solutions with one of those subgroups and obtain a *regular* [36] group action. Note that for the non-busy case the unique identification of solutions with elements of S_P or S_J is not possible as one has

$$|S_J| = J! < \frac{P!}{(P-J)!} < P! = |S_P|. \tag{24}$$

We leave it as an open problem to find and characterize subgroups of S_P that are in bijection with the solution set.

3.4. Relation to the QAOA

We now utilize our knowledge about feasibility-preserving permutations to study and design VQAs. The underlying connection is due to the fact that classical operations on bit strings may be considered as quantum permutation operators acting on the associated qubit space $\mathcal{H}$: Consider a COP $\mathcal{C}$. Any group G that acts on $Z(N)$ (resp. on $S(\mathcal{C})$) also acts on the computational basis (resp. on the feasible computational basis states) via ${}^g|z\rangle := |{}^g z\rangle$, $g \in G$. By linearity, we can extend this action to the whole space $\mathcal{H}$ (resp. $\mathcal{S}$), yielding linear operators $\rho(g) \in \mathcal{L}(\mathcal{H})$ (resp. $\rho_S(g) \in \mathcal{L}(\mathcal{S})$), which are simply permutation matrices in the computational basis. In group representation theory this construction is known as the *permutation representation* [37].

We start by connecting our group-theoretic picture for mixing operations with the established QAOA framework. Consider a COP with encoded objective Hamiltonian C, solution space $\mathcal{S} \subseteq \mathcal{H}$, and optimal solution space $\mathcal{S}_{\min} \subseteq \mathcal{S}$. Starting from a *feasible* initial state $|\iota\rangle \in \mathcal{S}$, the QAOA prescribes to alternately apply parametrized *phase separator* gates $U_{\mathrm{P}}(\gamma)$ and *mixer* gates $U_{\mathrm{M}}(\beta)$ p times, where p is the circuit depth. This yields a parametrized trial state

$$\left|\vec{\beta},\vec{\gamma}\right\rangle := \left(\prod_{o=1}^{p} U_{\mathrm{M}}(\beta_o)U_{\mathrm{P}}(\gamma_o)\right)|\iota\rangle\,. \tag{25}$$

After preparing $\left|\vec{\beta},\vec{\gamma}\right\rangle$, one can now estimate

$$F_p(\vec{\beta},\vec{\gamma}) := \left\langle \vec{\beta},\vec{\gamma}|C|\vec{\beta},\vec{\gamma}\right| \left|\vec{\beta},\vec{\gamma}|C|\vec{\beta},\vec{\gamma}\right\rangle \tag{26}$$

via repeated sampling on the quantum computer and pass this quantity to a classical optimizer, which updates the parameters $\vec{\beta}$, $\vec{\gamma}$ in order to minimize $F_p(\vec{\beta},\vec{\gamma})$. On real quantum hardware, the required finite coherence time as well as infidelities of the parametrized quantum circuits will accumulate noise, which will impact the quality of the measured estimate in addition to the *shot noise* obtain from the finite sample size. Several approaches for QAOA-specific error mitigation techniques exist that can markedly improve the estimate's quality. One prominent approach is to leverage symmetries within the classical objective function, which corresponds to joint degeneracies of the objective Hamiltonian's eigenspaces. Symmetrizing the trial states $\left|\vec{\beta},\vec{\gamma}\right\rangle$ with respect to these degeneracies can increase the estimate's accuracy [38, 39]. A second approach is centered around post-selecting directly on the COP's constraints in order to project back to the feasible subspace [40]. Furthermore, and more generally, redundancy encoding of classical variables also enables error mitigation schemes with applications also in combinatorial optimization [41].

3.4.1. Phase separator

The unitary phase separator is supposed to render the classical objective function's behavior but is technically merely required to be diagonal in the computational basis. We want to be more precise and suggest the following definition.

Definition 1. *A Hamiltonian H is called a phase separator Hamiltonian if it fulfills the following two conditions:*

(i) H is diagonal in the computational basis.
(ii) The eigenspace of $H|_{\mathcal{S}}$ corresponding to its smallest eigenvalue is $\mathcal{S}_{\min}$.

Then

$$U_{\mathrm{P}}(H,\gamma) := e^{-i\gamma H} \tag{27}$$

is the corresponding (parametrized) phase separator.

The canonical choice for a phase separator Hamiltonian is the objective Hamiltonian C. However, there might be decent approximations of C that are easier to implement and still preserve the optimal solution space (e.g., in threshold-based QAOA [42]). Since the phase separator is itself diagonal in the computational basis it trivially leaves the solution space $\mathcal{S}$ invariant.

3.4.2. Mixer

The unitary mixer is supposed to preserve and explore the solution space $\mathcal{S}$. Preservation of $\mathcal{S}$ simply means that $U_{\mathrm{M}}(\beta)(\mathcal{S}) \subseteq \mathcal{S}$ should hold for all $\beta \in \mathbb{R}$. The exploring condition is defined as follows: for all $z, z' \in \mathcal{S}$, there should exist a power $r \in \mathbb{N}$ and a parameter value $\beta \in \mathbb{R}$ so that $\langle z|U_{\mathrm{M}}^{r}(\beta)|z'|z|U_{\mathrm{M}}^{r}(\beta)|z'\rangle \neq 0$.
Our aim for a refined definition is now to mimic the two properties of the original QAOA mixer Hamiltonian

$$B = \sum_{n=1}^{N} \sigma_x^{(n)} \tag{28}$$

for unconstrained problems, but tailored to the constrained case. B, considered as a matrix in the computational basis, is component-wise non-negative and irreducible. Irreducibility means that the matrix B does not leave any non-trivial *coordinate subspace* invariant. That is, the only two subspaces of $\mathcal{H}$ that are a linear span of computational basis states and are left invariant under B are $\{0\}$ and $\mathcal{H}$. For brevity, we will address every linear span of computational basis states as a *coordinate subspace*. Our crucial observation is that the concept of irreducibility is indeed a fundamental mixing property that should be preserved in the constrained case [43].

In order to establish also 'sequential' mixers [23], we utilize the following result, which can be proved by considering each Hamiltonian as the adjacency matrix of a graph. More precisely, in a first step, one can interpret the matrix representation of each Hamiltonian as the adjacency matrix of a weighted directed graph. However, since we are not interested in actual weights and all matrices are hermitian, it suffices to consider simply a graph. Furthermore, since all components are non-negative, no cancellation happens when summing up all the matrices [32].

Proposition 3. *Let $\{H_i\}_{i\in I} \subset \mathcal{L}(\mathcal{H})$, $0 < |I| < \infty$, be a family of Hamiltonians, component-wise non-negative in the computational basis, such that $H_i(\mathcal{S}) \subseteq \mathcal{S}$ holds for all $i \in I$. Then the following two statements are equivalent:*

(i) Any coordinate subspace $X \subseteq \mathcal{S}$ that is left invariant under every $H_i|_{\mathcal{S}}$, is trivial.
(ii) $\left(\sum_{i\in I} H_i\right)|_{\mathcal{S}} \in \mathcal{L}(\mathcal{S})$ is irreducible in the computational basis.

Definition 2. *A family of Hamiltonians $\mathrm{H} = \{H_i\}_{i\in I} \subset \mathcal{L}(\mathcal{H})$ fulfilling the conditions in 3 is called a mixing family. The corresponding (parametrized) simultaneous mixer is defined as*

$$U_{\mathrm{M},0}(\mathrm{H}, \beta) := e^{-i\beta \sum_{i\in I} H_i}. \tag{29}$$

Specifying a permutation $\sigma \in S(I)$, the corresponding (parametrized) sequential mixer is defined as

$$U_{\mathrm{M},\sigma}(\mathrm{H}, \beta) := \prod_{i\in I} e^{-i\beta H_{\sigma(i)}}. \tag{30}$$

We present here a general method for obtaining suitable mixers from the feasibility-preserving subgroup F. Following the permutation representation, we identify each $g \in F$ via its action on $Z(N)$ with a linear operator $\rho(g) \in \mathcal{L}(\mathcal{H})$, and via its restricted action on $S(\mathcal{C})$ with a linear operator $\rho_{\mathcal{S}}(g) \in \mathcal{L}(\mathcal{S})$. Both representations yield permutation matrices in the computational basis. In the same way we have identified $S(\mathcal{C})$ with $\mathcal{S}$, we may also identify every subset of solutions with coordinate subspaces of $\mathcal{S}$. Then it readily follows that F acts transitively on $S(\mathcal{C})$ if the only coordinate subspaces of $\mathcal{S}$ that are left invariant by every $\rho_{\mathcal{S}}(g)$, $g \in F$, are $\{0\}$ and $\mathcal{S}$. Starting from the unitary operators $\rho(F) = \{W_g\}_{g\in F}$, we construct a family of Hamiltonians by taking suitable matrix logarithms $\{iL(W_g)\}_{g\in F}$. Since we only have finitely many unitary matrices W_g, we can always find a common branch L of the complex logarithm for the union of all their eigenvalues. A direct calculation yields that $A \subseteq \mathcal{H}$ is a W-invariant subspace if A is an $L(W)$-invariant subspace. Thus, the constructed family $\{iL(W_g)\}_{g\in F}$ admits the just introduced mixing property whenever F acts transitively on $S(\mathcal{C})$ and can therefore be used to build simultaneous and sequential mixers.

As we have shown before, the action of F on $S(\mathrm{OSSP})$ is indeed transitive. In addition, F consists of bit (value) permutations. Thus, the operator analogs $\rho(F)$ respect the tensor product structure of the qubit space $\mathcal{H}$, namely

$$\rho(g) \bigotimes_{n=1}^{N} |\psi_n\rangle = \bigotimes_{n=1}^{N} \left|\psi_{g^{-1}(n)}\right\rangle. \tag{31}$$

Therefore, our formalism yields particularly suitable mixers for the OSSP. It is, however, definitely not restricted to scheduling-type problem as it solely hinges on the identification and transitive action of a feasibility-preserving group. Other potential applications are perfect matching with flipping edges along even-length alternating cycles, XOR-SAT with affine transformations, and conservation-constrained network flow with flow in- and decrements along directed cycles. In general, the notion of a feasibility-preserving group seems to favor problem classes with equality constraints. Inequality constraints, on the other hand, seem to not induce sufficient symmetries to allow for a group-theoretical treatment. Take, for example, the knapsack problem where the only constraint is that an inner product of the binary variable vector with some non-negative weight vector should not exceed some positive capacity. There is no apparent symmetry and hence no apparent invertible operation, which always maps feasible assignments to feasible assignments. Problems of these types are therefore not suited to profit from our approach.

3.5. A quantum group optimization algorithm

After embedding our group-theoretic approach into the established QAOA framework, we now derive an alternative VQA design specifically tailored to busy OSSP instances [35]. Unlike the QAOA, it does not require a phase separator and, most importantly, comes with a natural upper bound for the number of parameters necessary to reach every feasible solution (especially the optimum) with certainty.

According to the previous section, we can describe the entire solution set S of this problem with a bit-to-qubit mapping. Namely, the set of all the solutions is identified with the symmetric group S_J. Fixing one solution $s \in S$, we can consider the problem of optimizing f equivalently as optimizing

$$\tilde{f}: S_J \to \mathbb{R}; \quad g \mapsto f({}^{g}s). \tag{32}$$

Thus busy OSSP instances (this also includes TSP) can be cast to optimization problems over symmetric groups. The underlying group structure can be exploited in the following way: Consider an arbitrary $\sigma \in S_J$. There is a well-known representation as a product of at most $\frac{J(J-1)}{2}$ transpositions [44]. We can further write every transposition as a product of some of the $J-1$ specific transpositions $\tau_1 = (1,2), \ldots, \tau_{J-1} = (J-1, J)$ in S_J. In summary,

we find for $\sigma \in S_J$ a binary vector $r^{(\sigma)} \in \{0,1\}^{\frac{J(J-1)^2}{2}}$ such that

$$\sigma = \prod_{k=1}^{\frac{J(J-1)}{2}} (\tau_1^{r_{k_1}^{(\sigma)}} \cdots \tau_{J-1}^{r_{k_{J-1}}^{(\sigma)}}). \tag{33}$$

Recall that the elements of S_J simultaneously act on multiple vertices, namely those lying in different position blocks, but having the same job coordinate. The representation of a transposition τ_i as elements in $\mathcal{L}(\mathcal{H})$ thus yields a product of disjoint SWAP gates, with each SWAP gate corresponding to one position block:

$$\tau_i \mathrel{\widehat{=}} \prod_{j=1}^{J} \mathrm{SWAP}^{(j)}_{(i,i+1)} =: B_i. \tag{34}$$

Here, $\mathrm{SWAP}^{(j)}_{(i,i')}$ interchanges the i-th and the i'-th qubit in the j-th position block. Since all SWAP gates are also hermitian, the operator B_i is again hermitian. Introducing $J(J-1)^2/2$ parameters and exponentiating each transposition operator B_i then yields the following implementation of σ as a parameterized quantum circuit:

$$U(\vec{\beta}) = \prod_{k=1}^{\frac{J(J-1)}{2}} e^{-i\beta_{k_1} B_1} \cdots e^{-i\beta_{k_{J-1}} B_{J-1}}. \tag{35}$$

Due to the fact that $e^{-i\beta \mathrm{SWAP}}$ is equal to SWAP with a global phase i for $\beta = -\pi/2$ and equal to the identity $\mathbb{1}$ for $\beta = 0$, we can precisely implement the action of each permutation $\sigma \in S_J$ with the right choice for the $J(J-1)^2/2$ parameters values. Therefore, we can directly transfer Theorem 2 to the gate implementation of permutations in S_J to conclude the following theorem:

Theorem 3. *Let* $\mathrm{OSSP}(M,T,J)$ *be a busy OSSP instance, i.e.,* $MT = J$*. Starting from a feasible computational basis state* $|\iota\rangle \in \mathcal{S}$*, there exist values* $\vec{\beta} \in [0, \frac{\pi}{2}]^{\frac{J(J-1)^2}{2}}$ *so that applying* $U(\vec{\beta})$*, as defined in* ***(35)****, to* $|\iota\rangle$ *yields an optimal solution to* $\mathrm{OSSP}(M,T,J)$*. Therefore, sampling from the resultant state returns the optimum with certainty.*

Theorem 3 constitutes a strong theoretical performance guarantee and has to be contrasted with substantially weaker guarantees for the generic QAOA. Here, only in the limit $p \to \infty$ one is guaranteed to reach an optimal solution, but also subject to selecting (in this case an entire sequence of) suitable parameter values [43]. However, the practical implications of Theorem 3 are ultimately limited by our ability to indeed find optimal parameter values and by quantum hardware limitations such as limited numbers of available qubits, missing qubit connectivity, decoherence, and gate infidelities. Nevertheless, such theoretical guarantees, even if not entirely transferable to a realistic setup, provide strong guidance for the formulation of practically useful heuristics. In this particular case, the above result inspires the following VQA:

1. Start in a solution state $|\iota\rangle \in \mathcal{S}$.
2. Apply the parameterized quantum circuit **(35)**.
3. Variationally optimize $\vec{\beta} \in [0, \frac{\pi}{2}]^{\frac{J(J-1)^2}{2}}$ using a classical optimization rule.
4. Measure the final outcome state in the computational basis.

Furthermore, the parameter space is always the same and especially compact, regardless of the objective function f. This has to be seen in contrast to general QAOA-mixers for which no comparable restriction of the necessary parameter space is possible. Thus, in our case, sampling methods for parameter adaptation like the sampled gradient descent (compare Section 3.7) are particularly powerful since they eventually grant access to the entire compact parameter space. However, we highlight that global parameter optimization quickly becomes infeasible for larger values of J due to the effect of shot and hardware noise [45], and because of the general hardness of multi-dimensional, non-convex optimization [46]. More precisely, the parameter optimization landscape typically suffers from exponentially flat regions—a phenomenon called *barren plateaus* [47]—and from a large number of local minima [8]. On the bright side, there are already several frameworks for mitigating the effect of barren plateaus [48, 49], enhancing parameter optimization strategies [50, 51], and formulating the parameter optimization as an easier problem [52, 53] that are applicable to the general VQAs, including our algorithm. Moreover, note that the aforementioned error mitigation techniques are also applicable to our algorithmic scheme: redundancy encoding of binary variables is a generally valid strategy to introduce error mitigation techniques. The notion of feasibility is easily updated for redundantly encoded variables, merely resulting in additional SWAP gates within B_i circuits. While our linear OSSP objective function **(13)** will typically not have symmetries to exploit for error mitigation purposes, the notion of OSSP feasibility definitely falls within the range of feasibility-based error mitigation protocols considered in [40].

The identification of the set of feasible solutions with S_J further suggests a dynamic programming [54] ansatz: Iteratively optimize over subgroup series of the form

$$1 = \langle \tau_i : i \in I_0 \rangle \leq \ldots \leq \langle \tau_i : i \in I_n \rangle = S_J, \tag{36}$$

with ascending index sets $I_0 \subset \ldots \subset I_n$, by restricting to the set of corresponding mixers B_i with $i \in I_k$, $k \in [n]$.

Lastly, let us discuss a concrete decomposition scheme of the compound gate $U(\vec{\beta})$ into logical components. It consists of a product of $J(J-1)^2/2$ exponentials of the Hamiltonians B_i such that it suffices to focus on the decomposition of a single exponential. The entire implementation of $U(\vec{\beta})$ is then given by sequential execution of each individually decomposed exponential. A single B_i consists of J SWAP gates on mutually disjoint qubit pairs. SWAP gates are involuntary, i.e., $\mathrm{SWAP}^2 = \mathbb{1}$, which allows us to exactly implement their exponential with the aid of a single ancilla qubits via the circuit depicted in **Figure 3**; a thorough proof can be found in ([55], Theorem 5.25). All components of this circuit except for the controlled B_i-operations are elementary single-qubit gates, hence adding only a constant overhead in both gate count and circuit depth. The entire construction essentially boils down to implementing a singly controlled product of SWAP gates, i.e., a product of Fredkin gates. Even though all SWAP gates contained in B_i act on mutually disjoint pairs of qubits and are thus parallelizable (assuming the quantum hardware supports parallel execution of gates on disjoint qubits), this does not hold anymore for the controlled version, as all SWAP gates are controlled on the same qubit. Furthermore, Fredkin gates do not usually belong to a quantum computer's native gate set. They can be logically decomposed into a Toffoli gate and two enclosing CNOT gates. The Toffoli gate, in turn, has an optimal decomposition into nine single-qubit gates and six CNOT gates [56], introducing only a constant implementation overhead. In summary, decomposing the entire circuit $U(\beta)$ into single-qubit and CNOT gates requires $\mathcal{O}(J^4)$ elementary gates.

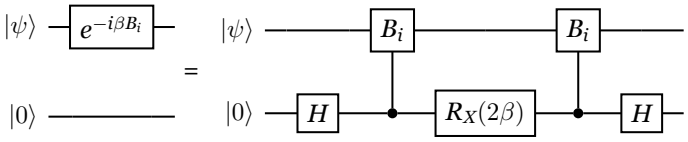

Figure 3 • Quantum circuit implementing the parametrized exponential of B_i. This construction requires a $|0\rangle$-initialized temporary working qubit, that is, the additional qubit returns to the $|0\rangle$ with certainty. The parameter β is introduced via a parametrized X-rotation on the ancilla qubit. The entire construction only depends on B_i being involuntary, i.e., $B_i^2 = \mathbb{1}$.

3.6. Numerical results: exact simulation

First, we demonstrate our just introduced VQA in a noiseless (neither shot- nor hardware noise) classical simulation on an OSSP(2,2,4) instance. **Figure 1** displays this instance's constraint graph. The group of job permutations is thus given by $S_J = S_4$. After assigning a bit to each of the 16 elements in $[2] \times [2] \times [4]$ via $(m,t,j) \mapsto 4(m-1) + 2(t-1) + j$, we explicitly generate S_J with τ_1, τ_2, and τ_3:

$$S_J = \langle (1,2)(5,6)(9,10)(13,14), (2,3)(6,7)(10,11)(14,15), (3,4)(7,8)(11,12)(15,16) \rangle. \quad (37)$$

Consequently, we have to implement three distinct mixer Hamiltonians B_1, B_2, B_3. As a concrete example, consider the permutation $(1,2)(5,6)(9,10)(13,14)$, which corresponds to the Hamiltonian

$$B_1 = \sum_{p=1}^{4} \mathrm{SWAP}^{(p)}_{(1,2)} = \mathrm{SWAP}_{(1,2)}\mathrm{SWAP}_{(5,6)}\mathrm{SWAP}_{(9,10)}\mathrm{SWAP}_{(13,14)}\,. \quad (38)$$

As a concrete feasible initial state, we choose $|z_0\rangle = |1000010000100001\rangle$.

Since $J = 4$, our variational circuit **(35)** contains 18 parameters to be optimized. At this scale, global black-box optimization techniques such as differential evolution [57] already becomes unpractical and we have to resort to local optimization strategies. Concretely, we utilize the L-BFGS-B algorithm [58] as implemented in scipy. However, in order to overcome the local optimizer's tendency to get stuck in a local minimum and to avoid the risk of initializing the parameters within a barren plateau, we enhance the parameter optimization in the following two ways: First, we iteratively optimize first $2q$ parameters, $1 \leq q \leq 9$ with the first $2(q-1)$ parameters being warm-started from the prior iteration. For $q = 9$ this eventually covers all 18 parameters, but with quantitative warm starts for the first 16 parameters, rather than random initialization. At each iteration step, we include two additional parameters into the scope of optimization. Our second enhancement consists of initializing the additional parameters with several value pairs, stemming from a shared grid. We conduct the subsequent optimization for each chosen pair of initial parameter values and only keep the best performing one.

We further compare the performance of our VQA with the standard QAOA and softcoded constraints. Recall, that softcoding the constraints means to alter the objective function f in order to penalize infeasible solutions. We adapt the standard penalization strategy for Hamiltonian cycles [59] with penalty function

$$g(z) \coloneqq \sum_{m=1}^{M}\sum_{t=1}^{T}\left(1 - \sum_{j=1}^{J} z_{mtj}\right)^2 + \sum_{j=1}^{J}\left(1 - \sum_{m=1}^{M}\sum_{t=1}^{T} z_{mtj}\right)^2. \quad (39)$$

The adapted objective function then reads $f + \alpha g$, where α is chosen so that no infeasible bit string receives a value lower or equal to the minimum of f over the feasible solutions. In reality, one can only guess or give non-tight upper bounds for α. For our toy example, however, we compute the optimal, i.e., lowest possible value for α by looping through all 2^{16} bit strings and determining the smallest value for α, which lifts all infeasible bit strings above the optimal feasible one. The QAOA circuit consists of an alternating application of the exponentiated standard mixer **(28)** and the usual phase separator, and is applied to $|+\rangle$ as initial state. For better comparability, we implement the QAOA with depth 9 and optimize its 18 variational parameters with the exact same strategy as described above for our feasibility-preserving VQA.

The results of our noiseless are depicted in **Figure 4**. Note that the approximation ratios are always with respect to the chosen α value. Accordingly, the initial state $|+\rangle$ (the state obtained after 0 iterations) exhibits different approximation values. Since our VQA stays within the feasible subspace, different values for α do not affect its achieved approximation ratios. We clearly observe that the standard approach with softcoded constraints fails to reliably deliver high-quality, feasible solutions. For the tested range of α-values, it merely achieves approximation ratios up to 30%. In comparison, starting from a feasible computational basis state and hardcoding the constraints within our VQA yields better ratios across all iterations. In fact, for this specific example we observe that optimizing the first 12 parameters already suffices to reach the global minimum. For larger values of J, jointly optimizing all $J(J-1)^2/2$ parameters in the last step will eventually become unpractical. However, optimizing smaller subsets of parameters in our VQA still holds the potential to significantly improve the initial solution. The drastic performance differences between our VQA with hardcoded constraints and the standard QAOA with softcoded constraints have two simple explanations, which also generalize to larger instance sizes: First, our design comes with a guaranteed upper bound of parameters needed to achieve all possible solutions with certainty, including the optimum (see Theorem 3), while for the generic QAOA there are no such guarantees. Second, the feasible subspace is of much lower dimension than the entire Hilbert space. For $J = 24$, there are $4! = 24$ feasible solutions, but $2^{4^2} = 65536$ different bit strings in total, and the ratio of feasible solutions to arbitrary bit strings quickly decreases further for larger numbers of J. Accordingly, the standard QAOA initial state $|+\rangle$ only has very small overlap with the feasible states and the guidance via softcoded constraints is not sufficient to concentrate enough amplitude within the feasible subspace.

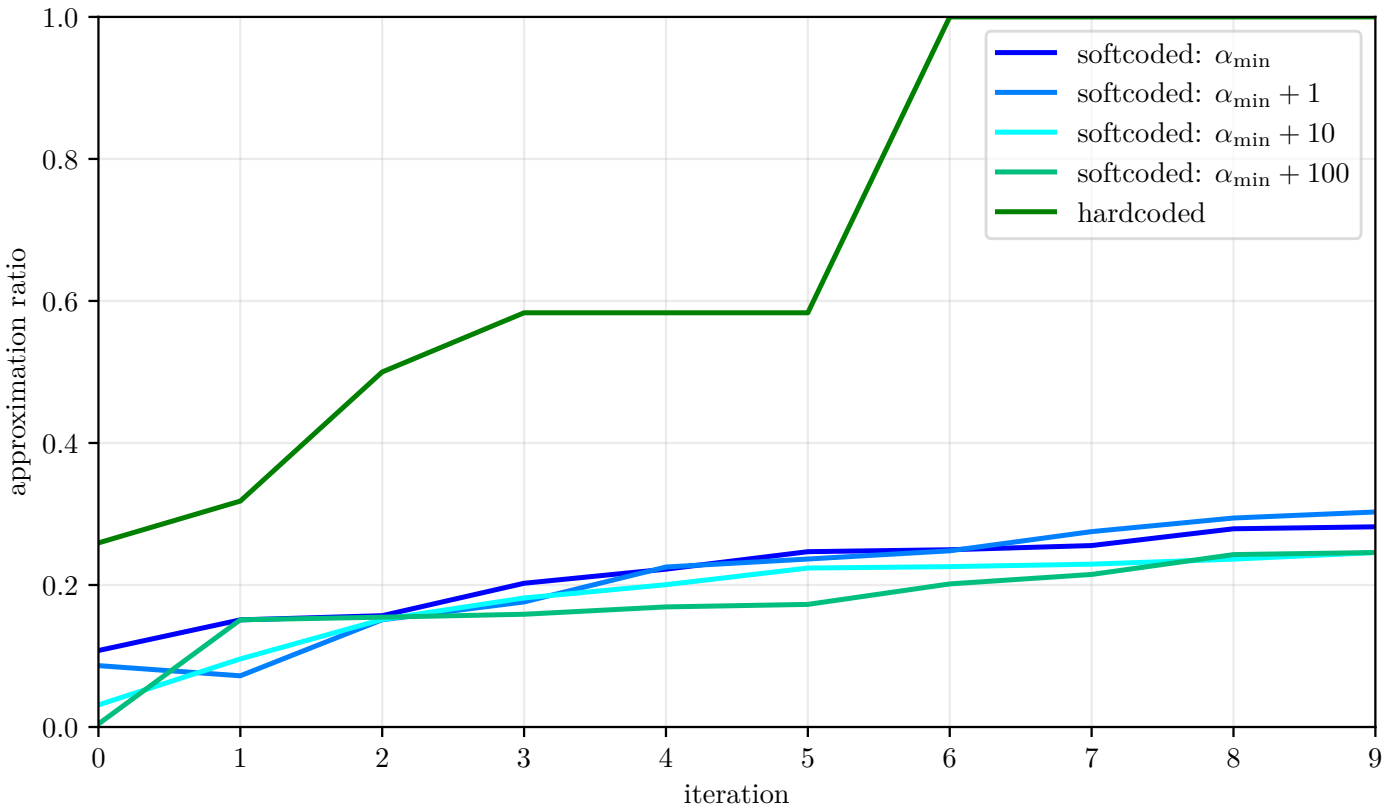

Figure 4 • Numerical experiments for the OSSP(2,2,4) instance. The plots depicts the approximation ratios (theoretical minimum divided by expectation value) after each iteration of parameter optimization for the standard QAOA with four different penalties as well as our VQA with hardcoded constraints. We observe that our VQA is able to reach the optimum (approximation ratio of 100%) already after six iterations, which corresponds to 12 optimized parameters. In contrast, the standard QAOA is not able to produce approximation ratios above 30%, even when having access to all 18 variational parameters.

3.7. Numerical results: real hardware implementation

As another proof of principle, we consider an OSSP(1,3,3) instance, i.e., a TSP instance with three cities, and demonstrate our VQA on a real (noisy) quantum hardware: the IBM Q System One. For the same problem instance, we also conduct a noisy simulation of the VQA on a classical computer.

The group of job permutations is given by $S_J = S_3$. Since we only have one machine, we simply drop the machine coordinate and consider the enumeration $(t,j) \mapsto 3(t-1)+j$. Thus, S_J is generated by two elements $\tau_1 = (1,2)(4,5)(7,8)$ and $\tau_2 = (2,3)(5,6)(8,9)$. We further consider the objective function

$$f\colon \{0,1\}^9 \to \mathbb{R}, \quad z \mapsto \sum_{t,j} \omega_{tj} z_{tj} \tag{40}$$

$$\text{with weight matrix } (\omega)_{tj} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 2 & 3 \\ 1 & 2 & 2 \end{pmatrix}. \tag{41}$$

In addition to the two mixer Hamiltonians B_1 and B_2, we also implement the QAOA-phase separator, resulting in the parametrized quantum circuit

$$U(\vec{\beta},\vec{\gamma}) = \underbrace{e^{-i\beta_1 B_1} e^{-i\beta_2 B_2} e^{-i\gamma_1 C}}_{1} \underbrace{e^{-i\beta_3 B_1} e^{-i\beta_4 B_2} e^{-i\gamma_2 C}}_{2} \underbrace{e^{-i\beta_5 B_1} e^{-i\beta_6 B_2} e^{-i\gamma_3 C}}_{3}. \tag{42}$$

However, this circuit is yet too deep to be fully implemented on the quantum device. Therefore, we restrict our circuit to the first factor of **(42)**. Choosing the feasible initial state $|z_0\rangle = |100010001\rangle$, we can predict which feasible states are actually accessible in this setting. Restricting to one factor in **(42)** classically corresponds to having only access to four group elements: id, τ_1, τ_2, and $\tau_1\tau_2$. Their application to z_0 yields

$$^{\mathrm{id}}z_0 = z_0,\ ^{\tau_1}z_0 = 010100001,\ ^{\tau_2}z_0 = 100001010, \tag{43}$$

$$^{\tau_1\tau_2}z_0 = 010001100, \tag{44}$$

where the colored bit string $^{\tau_1\tau_2}z_0$ is the optimal accessible feasible solution. In contrast, the global minimum 001010100 requires at least access to the first two factors as it is readily given by $^{\tau_2\tau_1\tau_2}z_0$.

For the actual parameter adaptation we use sampled gradient descent:

1. Construct a ball $\mathbb{B}_i \subset [0,\pi/2]^3$ around some parameter values $\vec{\beta}_i$.
2. Sample new parameter values uniformly from $\mathbb{B}_i$.
3. Apply the correspondingly parametrized circuit to the initial state and estimate the expectation value of C.
4. Choose parameter values $\vec{\beta}_{i+1}$ from the sample minimizing the expectation value and repeat step 1 with $i \mapsto i+1$.

The size of the constructed ball $\mathbb{B}_i$ is adapted in each step i: The steeper the drop in expectation values, the smaller the radius becomes. For our numerical execution we choose random initial parameters and a sample size of 40 in each step.

Figures 5 and **6** show the sampling results obtained after each round of the sampled gradient descent. After five iterations we indeed observe a dominating sampling of the local optimum as well as increased overlap with computational basis states that have a small Hamming distance to the local optimum. There remains, however, a noisy background both in the simulation as well as on the IBM Q System One. While the local optimum is clearly distinguishable in the simulation, the real quantum device introduces so much noise that the noisy background dominates the slight, but visible improvements in sampling the local optimum. The mismatch in magnitude of noise indicates a too optimistic classical noise model. At the same time, however, the data obtained from this simulation hints at what capabilities can be unlocked with improved quantum hardware components.

bit string	value
001010100	5
001100010	6
010001100	6
010100001	7
100001010	8
100010001	8

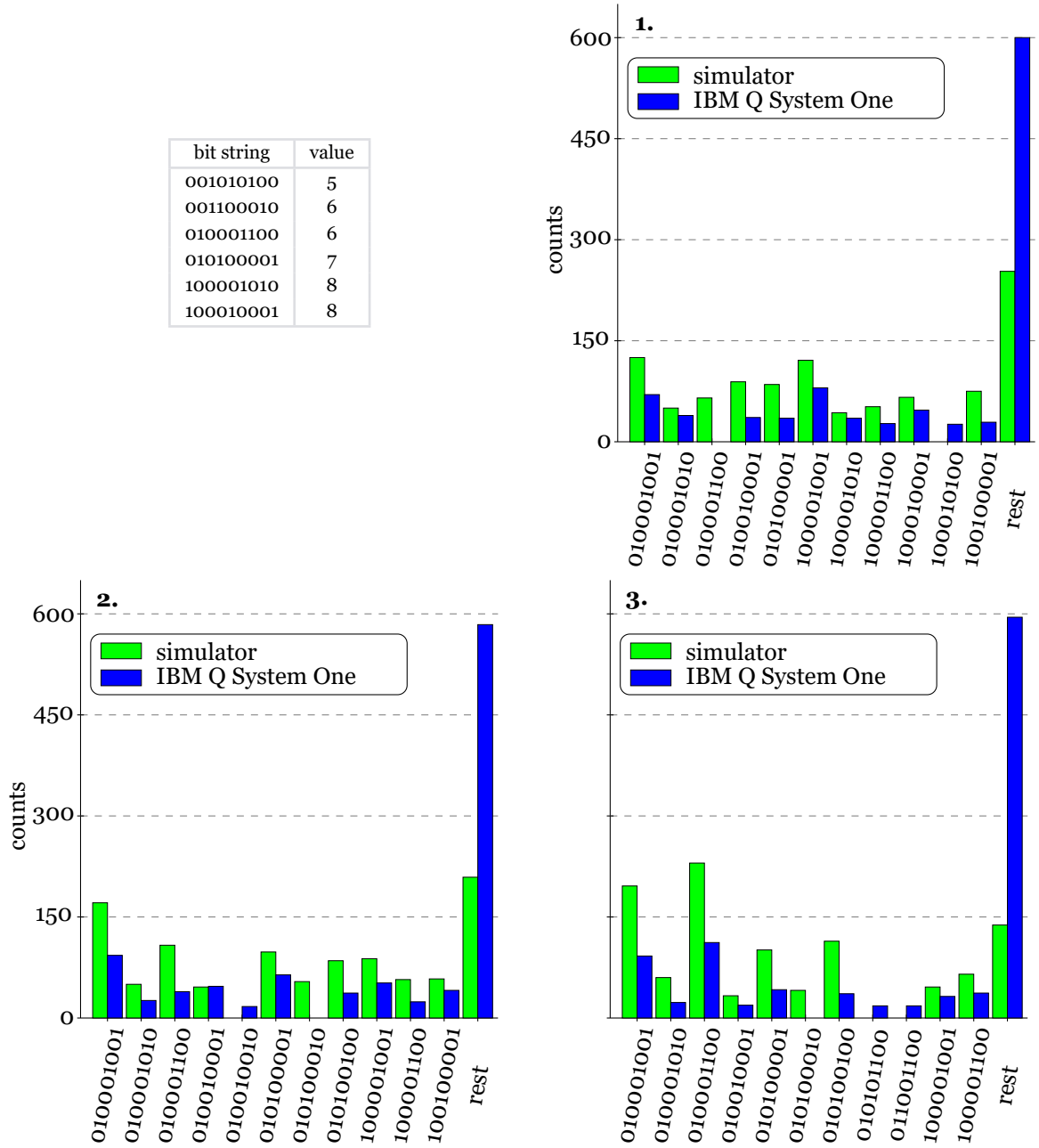

Figure 5 • Numerical experiment for the OSSP(1,3,3) instance. The table enumerates all feasible solutions. The three subplots show the distribution of sampling 1024 times from the quantum computer, resp. the simulated state, after one, two, and three iterations of sampled gradient descent. In each plot, we depict the counts for the ten most sampled bit strings in lexicographical order and collect all remaining counts into a "rest" state. The VQA quickly leaves the initial state $z_0 = 100010001$ already in the **1.** step The initial state is barely sampled already after the **2.** step, neither on the quantum device nor in the noisy simulation, and reduces further in frequency with the **3.** step. In contrast, the overlap with the local minimum steadily increases from step to step in both cases.

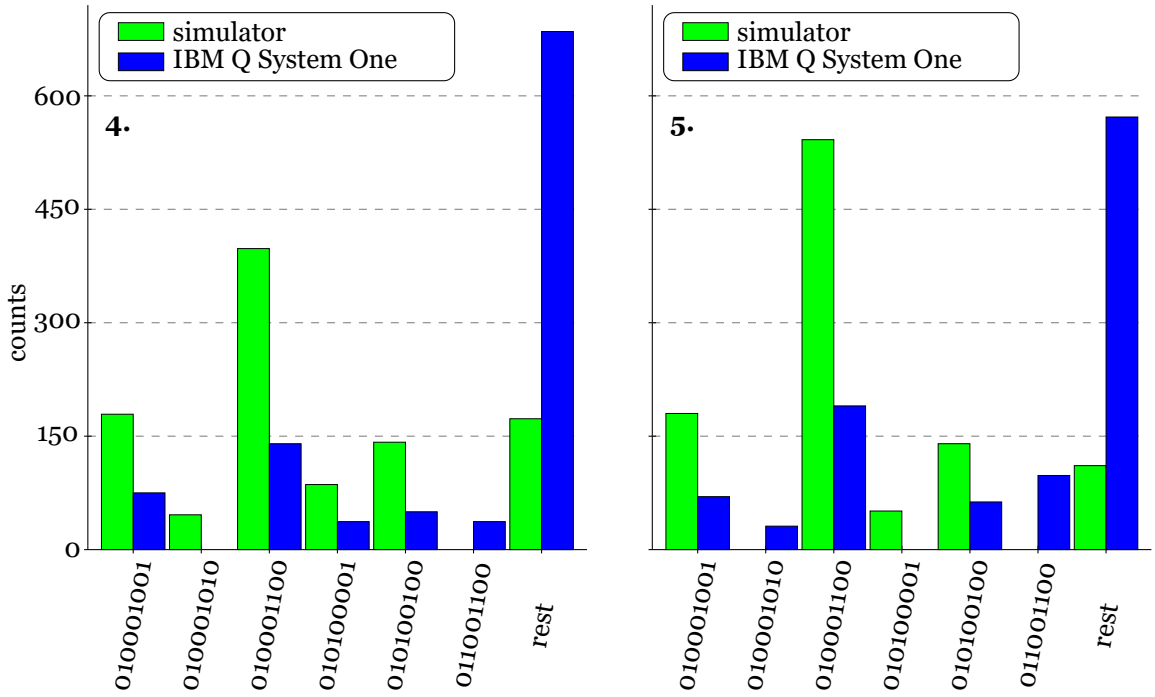

Figure 6 • Continued numerical experiment for the OSSP(1,3,3) instance. The two subplots show the distribution of sampling 1024 times from the quantum computer, resp. the simulated state, after four and five iterations of sampled gradient descent. Each plot depicts the counts for the six most sampled bit strings in lexicographical order and collects all remaining counts into a "rest" state. The overlap with the local minimum continues to increase with the **4.** VQA-step. This trend is stronger in the noisy simulation where more than half of all samples after the final **5.** step are from the optimal state. On the quantum device, the optimal state is at least the most sampled state of all computational basis states, but fails to concentrate more than roughly five percent of all samples in itself.

While the performance of the IBM Q System One does not suffice to tackle large-scale OSSP instances—neither in terms of available qubits nor in terms of feasible circuit depths—this proof of principle showcases that all the components of our VQA are already implementable and executable on today's quantum hardware. Quantitatively extrapolating to larger and less error-prone quantum devices based on a single numerical experiment is, of course, not meaningful. However, we can discuss the effect of improved qubit counts and gate fidelities versus larger instance sizes qualitatively: First of all, the number of available qubits is a hard upper limit for the accessible problem sizes. In general, encoding an OSSP(M, T, J) instance requires MTJ qubits; the busy case $MT = J$ therefore requires J^2 qubits. The quadratically increasing instance size is well matched by the recent efforts to increase the number of qubits such that larger instances with tens of jobs can be encoded. Second, coherence time and gate fidelities massively influence the performance of any quantum algorithm. Both factors generally do not respect the problem's feasibility structure so that long noisy circuits tend to exit the feasible subspace if its orthogonal complement is large. This is typically the case for hard constrained COPs such as the OSSP. Concretely for the busy case, the number of bit strings and hence the Hilbert space dimension of the encoded problem are given by 2^{J^2}. In comparison, the feasible set "only" comprises $J!$ solutions. For large J, random noise is therefore far more likely to steer the parametrized states outside of the feasible subspace.

4. Conclusion

In this paper we presented a general approach for characterizing the feasibility structure of open-shop scheduling problems. We utilized the constraint graph model as a general interplay between graph and group theory for determining symmetries for certain problem instances. This additional perspective allowed us to find feasibility-preserving mappings as graph automorphisms. We calculated the entire group of such feasibility-preserving functions and proved that it is isomorphic to the automorphism group of the constraint graph. This is a very strong statement which unfortunately is not transferable to other types of job-shop scheduling problems. For example, for a generic *flexible* job-shop instance, where each job is subdivided into an ordered sequence of operations, these ordering constraints introduce additional edges that break certain symmetries, rendering the constraint graph's automorphism group significantly smaller. The remaining symmetries can still be incorporated into QAOA-mixers but have to be supplemented with additional elements that do not correspond to classical bit permutations.

Let us emphasize one more time that the busy OSSP with J jobs is equivalent to optimizing over the symmetric group S_J. We addressed this type of problem by representing the symmetric group with (products of) parametrized exponentials of SWAP gates. Here, the discussed transitive action of the group on the solution set guarantees that all possible solutions can be reached by suitable parameter adjustments. Via a decomposition of arbitrary group elements into a sequence of $J(J-1)^2/2$ simple transitions, we were able to prove that $J(J-1)^2/2$ variational parameters are sufficient to reach all feasible OSSP solutions, including the optimum. In principle, we can generalize this procedure to an arbitrary (finite) group G, acting transitively on a given set of solutions S. Then, the permutation representation would still yield permutation operators on the qubit space, which, however, will not correspond to actual qubit permutations anymore. It would

be interesting to characterize in the future the obtained operators and the performance of our algorithm for more exotic cases such as the mentioned perfect matching problem, XOR-SAT, and conservation-constrained network flow. However, operators that do not respect the tensor product structure of $\mathcal{H}$ will be generally very difficult to implement. Additionally, problems with inequality constraints seem to have significantly less exploitable feasibility structure so that the group of (unconditionally) feasibility-preserving operations often ends up to be trivial. For those kind of problems, additional techniques may be employed to complement our approach, which works well for equality constraints. The ultimate goal would be to create a toolbox for hardcoding arbitrary constraints and to extend those methods also to mixed-integer and continuous optimization problems. This would allow to also tackle more complicated, but also more realistic logistic problems like transit scheduling [60, 61].

Moreover, while the cubic upper bound on the number of required variational parameters constitutes a strong theoretical results, its practical implications ultimately hinge on the employed classical optimizer's ability to find high-quality parameter values. For larger problem instances, simultaneous optimization of cubically many parameters will quickly decline in performance due to exponentially increasing numbers of local minima and flatness of the parameter landscape (barren plateaus). However, established frameworks for enhancing the parameter optimization such as warm starts, layer-wise optimization schedules, and reformulations as easier problems are also applicable to our method. Studying their impact on the practicality of our proposed algorithmic blueprint remains an interesting open research question.

Acknowledgments

We thank Michael Cuntz, Tim Heine, Max Hess, Jan Rasmus Holst, Andreea-Iulia Lefterovici, Lauritz van Luijk, Tobias J. Osborne, Lilly Palackal, Leonhard Richter, Marco Tomamichel, and Timo Ziegler for helpful discussions. The authors especially thank and are deeply indebted to Benjamin Sambale for many long and fruitful discussions.

Funding

This research was sponsored by the BMBF under the projects ATIQ and QuBRA, by the state of Lower Saxony and the Volkswagen Foundation under the Quantum Valley Lower Saxony, and by the state of Baden-Württemberg under the project SEQUOIA.

Author contributions

Conceptualization, L.B. and G.K.; methodology, L.B. and G.K.; software, L.B. and G.K.; validation, L.B., G.K., C.T. and R.S.; formal analysis, L.B. and G.K.; investigation, L.B. and G.K.; resources, C.T.; data curation, L.B.; writing—original draft preparation, L.B., G.K., C.T. and R.S.; writing—review and editing, L.B.; visualization, L.B.; supervision, C.T. and R.S.; project administration, C.T. and R.S.; funding acquisition, C.T. and R.S. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare that they have no competing interests.

Data availability statement

All data supporting the findings of this publication are available within this article.

Additional information

Received: 2025-04-11

Accepted: 2025-08-29

Published: 2025-09-12

Academia Quantum papers should be cited as *Academia Quantum 2025*, ISSN 3064-979X, https://doi.org/10.20935/AcadQuant7900. The journal's official abbreviation is *Acad. Quant.*

Publisher's note

Copyright

References

1. Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, et al. Variational quantum algorithms. Nat Rev Phys. 2021;3(9):625–44. doi: 10.1038/s42254-021-00348-9

2. Marshall J, Wudarski F, Hadfield S, Hogg T. Characterizing local noise in QAOA circuits. IOP SciNotes. 2020; 1(2):025208. doi: 10.1088/2633-1357/abb0d7

3. Peruzzo A, McClean J, Shadbolt P, Yung MH, Zhou XQ, Love PJ, et al. A variational eigenvalue solver on a photonic quantum processor. Nat Commun. 2014;5(1):4213. doi: 10.1038/ncomms5213

4. Barkoutsos PK, Nannicini G, Robert A, Tavernelli I, Woerner S. Improving variational quantum optimization using CVaR. Quantum. 2020;4:256. doi: 10.22331/q-2020-04-20-256

5. Larkin J, Jonsson M, Justice D, Guerreschi GG. Evaluation of QAOA based on the approximation ratio of individual samples. Quantum Sci Technol. 2022;7(4):045014. doi: 10.1088/2058-9565/ac6973

6. Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018;2:79. doi: 10.22331/q-2018-08-06-79

7. Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, et al. Noisy intermediate-scale quan-tum algorithms. Rev Mod Phys. 2022;94(1):015004. doi: 10.1103/revmodphys.94.015004

8. Koßmann G, Binkowski L, van Lujik L, Ziegler T, Schwonnek R. Deep-circuit QAOA. arXiv. 2022. arXiv:2210.12406v2

9. Sturm A. Theory and implementation of the quantum approximate optimization algorithm: a comprehensible introduction and case study using qiskit and IBM quantum computers. arXiv. 2023. arXiv:2301.09535

10. Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. arXiv. 2014. arXiv:1411.4028v1

11. Willsch M, Willsch D, Jin F, De Raedt H, Michielsen K. Benchmarking the quantum approximate optimization algorithm. Quantum Inf Process. 2020;19(7):197. doi: 10.1007/s11128-020-02692-8

12. Harrigan MP, Sung KJ, Neeley M, Satzinger KJ, Arute F, Arya K, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nat. Phys. 2021;17(3):332–6. doi: 10.1038/s41567-20-01105-y

13. Sato R, Gordon C, Saito K, Kawashima H, Nikuni T, Watabe S. Two-step quantum search algorithm for solving traveling salesman problems. IEEE Trans Quantum Eng. 2025;6:3100712. doi: 10.1109/TQE.2025.3548706

14. Azad U, Behera BK, Ahmed EA, Panigrahi PK, Farouk A. Solving vehicle routing problem using quantum approximate optimization algorithm. IEEE Trans Intell Transp Syst. 2023;24(7):7564–73. doi: 10.1109/TITS.2022. 3172241

15. Palackal L, Poggel B, Wulff M, Ehm H, Lorenz JM, Mendl CB. Quantum-assisted solution paths for the capacitated vehicle routing problem. Proceedings of the 2023 IEEE International Conference on Quantum Computing and Engineering (QCE); 2023 Sep 17–22; Bellevue, WA, USA. doi: 10.1109/QCE57702.2023.00080

16. Tran T, Do M, Rieffel E, Frank J, Wang Z, O'Gorman B, et al. A hybrid quantum-classical approach to solving scheduling problems. Proceedings of the International Symposium on Combinatorial Search; 2016 Jul 6–8; Tarrytown, NY, USA. p. 98–106. doi: 10.1609/socs.v7i1.18390

17. Kurowski K, Pecyna T, Slysz M, Różycki R, Waligóra G, Weglarz J. Application of quantum approximate optimization algorithm to job shop scheduling problem. Eur J Oper Res. 2023;310(2):518–28. doi: 10.1016/j.ejor.2023.03.013

18. Dalal A, Montalban I, Hegade NN, Cadavid AG, Solano E, Awasthi A, et al. Digitized counterdiabatic quan-tum algorithms for logistics scheduling. Phys Rev Appl. 2024;22(6):064068. doi: 10.1103/PhysRevApplied.22.064068

19. de la Grand'rive PD, Hullo JF. Knapsack Problem variants of QAOA for battery revenue optimisation. arXiv. 2019. arXiv:1908.02210v2

20. van Dam W, Eldefrawy K, Genise N, Parham N. Quan-tum optimization heuristics with an application to knapsack problems. Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE Computer Society; 2021 Oct 17–22; Los Alamitos, CA, USA. p. 160–70. doi: 10.1109/QCE52317.2021.00033

21. Baker JS, Radha SK. Wasserstein solution quality and the quantum approximate optimization algorithm: a portfolio optimization case study. arXiv. 2022. arXiv:2202.06782

22. Awasthi A, Bär F, Doetsch J, Ehm H, Erdmann M, Hess M, et al. Quantum computing techniques for multi-knapsack problems. Intelligent computing. Cham: Springer Nature Switzerland; 2023. p. 264–84. doi: 10.1007/978-3-031-37963-5_19

23. Hadfield S, Wang Z, O'Gorman B, Rieffel E, Venturelli D, Biswas R. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. Algorithms. 2019;12(2):34. doi: 10.3390/a12020034

24. Christiansen P, Binkowski L, Ramacciotti D, Wilkening S. Quantum tree generator improves QAOA state-of-the-art for the knapsack problem. arXiv. 2025. arXiv:2411.00518

25. Xie N, Lee X, Cai D, Saito Y, Asai N, Lau HC. A feasibility-preserved quantum approximate solver for the capacitated vehicle routing problem. Quantum Inf Process. 2024;23(8):291. doi: 10.1007/s11128-024-04497-5

26. Nakada H, Tanahashi K, Tanaka S. Inductive construction of variational quantum circuit for constrained combinatorial optimization. IEEE Access. 2025;13:73096–108. doi: 10.1109/ACCESS.2025.3563960

27. Rossi F, van Beek P, Walsh T. Handbook of constraint programming. Amsterdam: Elsevier Science Inc.; 2006.

28. Bärtschi A, Eidenbenz S. Grover mixers for QAOA: shifting complexity from mixer design to state preparation. Proceed-ings of the 2020 IEEE International Conference on Quan-tum Computing and Engineering (QCE); 2020 Oct 12–16; Online. p. 72–82.

29. Leighton FT. A graph coloring algorithm for large scheduling problems. J Res Natl Bur Stand. 1977;84(6):489–506. doi: 10.6028/jres.084.024

30. Freuder EC, Quinn MJ. Taking advantage of stable sets of variables in constraint satisfaction problems. Proceed-ings of the 9th International Joint Conference on Artificial Intelligence–Volume 2. IJCAI'85; 1985 Aug 18–23; Los Angeles, CA, USA. p. 1076–8.

31. Nielsen MA, Chuang IL. Quantum computation and quantum information. Cambridge: Cambridge University Press; 2010. doi: 10.1017/CBO9780511976667

32. Binkowski L. Constraint graph model analysis of the quantum alternating operator ansatz. Hannover: Leibniz Universität Hannover; 2022. [accessed on 2025 Apr 1]. Available from: https://www.itp.uni-hannover.de/fileadmin/itp/qinfo/Team_Tobias_Osborne/Masters_Theses/Lennart_Binkowski_Masters_Thesis.pdf.

33. Imrich W, Klavzar S. Product graphs. New York (NY): Wiley-Interscience; 2000.

34. Roman JJ. An introduction to the theory of groups. New York (NY): Springer; 1995. doi: 10.1007/978-1-4612-4176-8

35. Koßmann G. A quantum algorithm with group theory. Hannover: Leibniz Universität Hannover; 2023.

36. Sambale B. Endliche permutationsgruppen. Berlin and Heidelberg: Springer Fachmedien Wiesbaden; 2017. doi: 10.1007/978-3-658-17597-9

37. Sagan BE. The symmetric group. New York (NY): Springer; 2001. doi: 10.1007/978-1-4757-6804-6

38. Shaydulin R, Galda A. Error mitigation for deep quantum optimization circuits by leveraging problem symmetries. Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE); 2021 Oct 17–22; Broomfield, CO, USA. p. 291–300. doi: 10.1109/QCE52317.2021.00046

39. Kakkar A, Larson J, Galda A, Shaydulin R. Characterizing error mitigation by symmetry verification in QAOA. Proceedings of the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE); 2022 Sep 18–23; Broomfield, CO, USA. p. 635–45. doi: 10.1109/qce53715.2022.00086

40. Botelho L, Glos A, Kundu A, Miszczak JA, Salehi O, Zimborás Z. Error mitigation for variational quantum algorithms through mid-circuit measurements. Phys Rev A. 2022;105(2):022441. doi: 10.1103/PhysRevA.105.022441

41. Weidinger A, Mbeng GB, Lechner W. Error mitigation for quantum approximate optimization. Phys Rev A. 2023;108(3):032408. doi: 10.1103/PhysRevA.108.032408

42. Golden J, Bärtschi A, O'Malley D, Eidenbenz S. Threshold-based quantum optimization. Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE Computer Society; 2021 Oct 17–22; Los Alamitos, CA, USA. p. 137–47. doi: 10.1109/QCE52317.2021.00030

43. Binkowski L, Koßmann G, Ziegler T, Schwonnek R. Elementary proof of QAOA convergence. New J Phys. 2024; 26(7):073001. doi: 10.1088/1367-2630/ad59bb

44. Humphreys JE. Reflection groups and coxeter groups. Cambridge: Cambridge University Press; 1992 [accessed on 2025 Apr 1]. Available from: https://www.ebook.de/de/product/4286037/james_e_humphreys_reflection_groups_and_coxeter_group.html.

45. Pellow-Jarman A, McFarthing S, Sinayskiy I, Park DK, Pillay A, Petruccione F. The effect of classical optimizers and Ansatz depth on QAOA performance in noisy devices. Sci Rep. 2024;14(1):16011. doi: 10.1038/s41598-024-66625-6

46. Bittel L, Kliesch M. Training variational quantum algorithms is NP-Hard. Phys Rev Lett. 2021;127(12):120502. doi: 10.1103/PhysRevLett.127.120502

47. Larocca M, Thanasilp S, Wang S, Sharma K, Biamonte J, et al. Barren plateaus in variational quantum computing. Nat Rev Phys. 2025;7(4):174–89. doi: 10.1038/s42254-25-00813-9

48. Patti TL, Najafi K, Gao X, Yelin SF. Entanglement de-vised barren plateau mitigation. Phys Rev Res. 2021;3(3): 033090. doi: 10.1103/PhysRevResearch.3.033090

49. Sack SH, Medina RA, Michailidis AA, Kueng R, Serbyn M. Avoiding barren plateaus using classical shadows. PRX Quantum. 2022;3(2):020365. doi: 10.1103/PRXQuantum.3.020365

50. Egger DJ, Mareček J, Woerner S. Warm-starting quantum optimization. Quantum. 2021;5:479. doi: 10.22331/q-2021-6-17-479

51. Lee X, Yan X, Xie N, Cai D, Saito Y, Asai N. Iterative layerwise training for the quantum approximate optimization algorithm. Phys Rev A. 2024;109(5):052406. doi: 10.1103/PhysRevA.109.052406

52. Binkowski L, Koßmann G, Osborne TJ, Schwonnek R, Ziegler T. From barren plateaus through fertile valleys: Conic extensions of parameterised quantum circuits. arXiv. 2025. arXiv:2310.04255

53. Binkowski L, Osborne TJ, Schwiering M, Schwonnek R, Ziegler T. One for all: universal quantum conic programming framework for hard-constrained combinatorial optimization problems. arXiv. 2024. arXiv:2411.00435

54. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. 4th ed. Cambridge (MA): MIT Press; New York (NY): McGrwa-Hill; 2022.

55. Schwiering M. Quantum optimization algorithms for the traveling salesman problem. Hannover: Leibniz Universiä Hannover; 2024 [accessed on 2025 Apr 1]. Available from: https://www.itp.uni-hannover.de/fileadmin/itp/qinfo/Team_Tobias_Osborne/Masters_Theses/Marvin_Schwiering_Masters_Thesis.pdf.

56. Shende VV, Markov IL. On the CNOT-cost of TOFFOLI gates. Quantum Inf Comput. 2009;9(5):461–86.

57. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59. doi: 10.1023/A:1008202821328

58. Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput. 1995;16(5):1190–208. doi: 10.1137/0916069

59. Lucas A. Ising formulations of many NP problems. Front Phys. 2014;2:5. doi: 10.3389/fphy.2014.00005

60. Owais M, Osman MK, Moussa G. Multi-objective transit route network design as set covering problem. IEEE Trans Intell Transp Syst. 2016;17(3):670–9. doi: 10.1109/TITS.2015.2480885

61. Owais M, Ahmed AS. Frequency based transit assignment models: graph formulation study. IEEE Access. 2022;10:62991–3003. doi: 10.1109/ACCESS.2022.3182046