

Exhaustive and feasible parametrisation with applications to the travelling salesperson problem

Marvin Schwiering*, Timo Ziegler*, Lennart Binkowski*, Benjamin Sambale†

*Institut für Theoretische Physik

Leibniz Universität Hannover, Hannover, Germany

Email: {marvin.schwiering, timo.ziegler, lennart.binkowski}@itp.uni-hannover.de

†Institut für Algebra, Zahlentheorie und Diskrete Mathematik,

Leibniz Universität Hannover, Hannover, Germany

Email: sambale@math.uni-hannover.de

Abstract—This paper introduces the concept of exhaustively parametrised, feasibility-respecting quantum circuits for constrained combinatorial optimisation problems. Such circuits can reach, given the right parameter values, every feasible solution with certainty—including the optimum—with a fixed number of parameters, while avoiding infeasible solutions altogether. This is in sharp contrast to conventional quantum alternating operator ansatz schemes, which are merely guaranteed to reach the optimum asymptotically. We introduce an abstract pipeline for constructing exhaustively parametrised, feasibility-respecting circuits from a transitive group action on a problem’s feasible set. Our constructions rely on the simple combination of the group action with group representation and the novel notion of generating sequences: group elements in fixed order, possibly with repetitions, that generate the entire group. That is, we trace expressivity of parametrised quantum circuits back to the most fundamental concepts of group theory. We apply this pipeline to two concrete examples for the travelling salesperson problem, thus showing that exhaustively parametrised, feasibility-respecting circuits are not an empty definition. Furthermore, we provide numerical proof-of-principles on instances with up to nine cities, comparing the suitability of our constructions for parameter optimisation purposes against established mixers.

Index Terms—combinatorial optimisation, hard constraints, QAOA, symmetric group, travelling salesperson problem

I. INTRODUCTION

The *quantum approximate optimisation algorithm* [1] and its generalisation to the *quantum alternating operator ansatz* (QAOA) [2] arguably constitute the most-studied class among (near-term) quantum algorithms for combinatorial optimisation. In a nutshell, QAOA consists of optimising the expectation value of an objective Hamiltonian, encoding the classical objective function, over a parameterised class of quantum states. This class is obtained by applying parameterised quantum circuits to a given initial state. The parameter optimisation itself is often outsourced to a classical optimiser, while the quantum computer is mainly used as a sampling machine, providing fast estimations of the objective function (and its gradients). Traditional classical algorithms for the QAOA parameter optimisation, such as COBYLA [3]

This work was supported by the DFG under Germany’s Excellence Strategy–EXC-2123 QuantumFrontiers, the Quantum Valley Lower Saxony and the BMFTR projects ATIQ and Quics.

or SLSQP [4], are usually (pseudo) gradient-based. More recently, also gradient-free approaches based on Monte Carlo tree search have been investigated [5]. An increasingly popular alternative to direct parameter optimisation is to transfer well-performing parameters between similar instances [6].

Irrespective of the utilised parameter optimisation technique, QAOA circuits typically face reachability issues in the sense that the class of parametrised states does not (fully) cover any optimal solution state. This is true for the original design for unconstrained and soft-constrained problems, but also often applies to refined versions which incorporate the problem’s feasibility structure directly into the parametrised circuit. While the original formulation and all versions derived from adiabatic protocols [7] can at least ensure exact reachability in the (arguably practically irrelevant) limit of infinitely many variational parameters [8], even this property may vanish for generic QAOA circuits for hard-constrained problems. In any case, this renders most QAOA circuits with a fixed number of parameters unreliable candidates for ever detecting the optimum of given optimisation problem.

Even though the heuristic nature of the QAOA might not require reliable sampling of optima in practice, we believe that the difficulty of finding good solutions with the QAOA should solely concentrate in the parameter optimisation, not in the coverage of the feasible space itself. To this end, we introduce the concept of *exhaustively parametrised* quantum circuits, which are guaranteed to reach every feasible solution state—including the optima—exactly and with a finite number of parameters. Analogous to the generalisation of the QAOA for constrained problems, such circuits should only produce (superpositions of) feasible states, restricting the search space to the feasible subspace and thus avoiding overhead through feasibility checks and penalty terms. Overall, our construction removes a representational obstruction. It does not, by itself, imply efficient parameter optimisation or an end-to-end quantum speedup.

While the concept of exhaustively parametrised, feasibility-respecting quantum circuits might be straightforward, constructing one for a concrete problem at hand poses a serious engineering challenge, requiring profound knowledge about

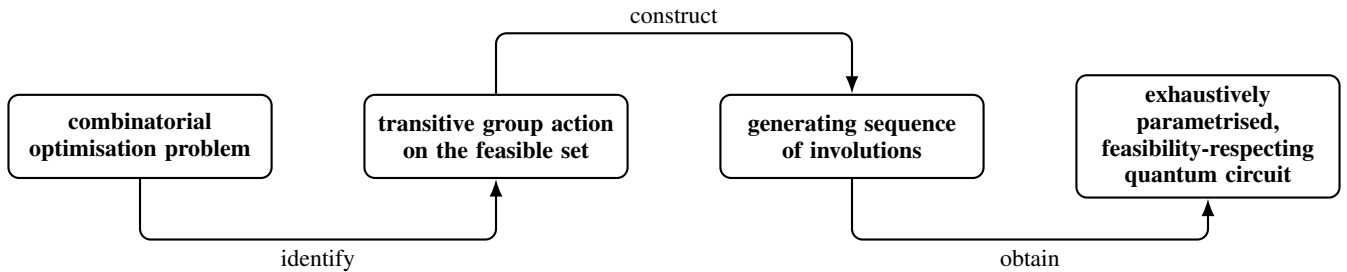


Fig. 1. High-level overview of our abstract pipeline for constructing exhaustively parametrised, feasibility-respecting quantum circuits for combinatorial optimisation problems. The task of exactly and exclusively preparing all feasible solutions via parametrised quantum circuits can be reduced to concepts of elementary group theory. The two key ingredients are a transitive group action on the problem’s feasible set, often available from decades of operational research studying symmetries in the feasible set, and a generating sequence of involutions. The output of the pipeline is an exhaustively parametrised, feasibility-respecting quantum circuit with a fixed number of parameters.

the problem’s feasibility structure. Our main contribution is the formalisation of an abstract pipeline, which essentially reduces the construction of such circuits to identifying a transitive group action on the feasible set. Intuitively, a transitive group action is able to map between every pair of arbitrary feasible elements. Representing the group (action) as multi-qubit operators is the conceptually easier part. Capturing the entire group with fixed number of elements requires the novel notion of a generating sequence.

We apply our pipeline to the *travelling salesperson problem* (TSP), one of the most paradigmatic NP-hard combinatorial optimisation problems with constraints. Its feasible set can be naturally identified with the symmetric group over all city indices, suggesting to consider the action of the symmetric group on its own, i.e., on the TSP’s feasible set, as entry point for our pipeline. We derive two classes of exhaustively parametrised, feasibility-respecting circuits based on the same group action, but on different generating sequences. The first example is inspired by the classical bubble sort algorithm. The latter’s ability to correctly sort an arbitrary input list can be reinterpreted as its building blocks—adjacency transpositions—constituting a generating sequence for the symmetric group. The bubble sort sequence for an n -city instance has $\mathcal{O}(n^2)$ elements and yields a circuit with the same number of parameters. The second design is based on a recursive construction of generating sequences, which only adds $\log(n)$ elements when increasing $n - 1 \mapsto n$. Applying this construction $n - 1$ times in a row yields a generating sequence with only $n \log(n)$ elements.

The TSP has been frequently in the scope of the QAOA and related methods, allowing a direct comparison between our approach and established ones. Concretely, the seminal work [2] enhancing the QAOA for constrained problems, also features concrete feasibility-preserving quantum circuits for the TSP. We utilise them as baseline for numerical experiments on TSP instances with up to nine cities. Despite increased performance of our approaches on the selected test instance, we refrain from predicting any immediate practical implications for solving realistic TSP instances. On the one hand, the unpredictably complex parameter landscapes permit any meaningful

extrapolations, neither to instances of a similar size, let alone to realistic instances. On the other, instances with thousands of variables have been classically solved to provable optimality [9]—a regime simply not matched by near-term quantum computing and tailored quantum algorithms. Instead, the TSP constructions and their numerical study are to be understood as proof-of-principles of our abstract pipeline, whose impact will grow the better we understand the feasibility structure of other, more challenging optimisation problems.

II. PREVIOUS WORK

The TSP is arguably one of the most prominent combinatorial optimisation problems and motivated the development of several, more generally applicable classical frameworks. For example, in their seminal paper [10], Dantzig, Fulkerson, and Johnson already utilised cutting planes [11] and branch-and-bound [12] methods. Bellman [13] developed the concept of dynamic programming and studied its application to the TSP. Also the Held-Karp algorithm [14], which enjoys the best asymptotic worst-case runtime among all known classical exact algorithms for the TSP, is based on dynamic programming.

In the field of quantum(-assisted) optimisation, the original formulation of the QAOA by Farhi *et al.* [1] is arguably the best-studied algorithm. It is universally applicable to combinatorial optimisation. However, constrained problems, such as the TSP, require a reformulation as (usually quadratic) unconstrained binary optimisation problems (QUBOs) [15]. This reformulation technique predates the field of quantum optimisation—for example, the QUBO formulation of the TSP can be traced back to early works of Hopfield and Tank [16]—but regained massive popularity due to the QAOA. Most notably, Lucas [17] published a comprehensive list of QUBO formulations for various combinatorial optimisation problems, including Hopfield and Tank’s formulation that was previously rediscovered by Warren [18]. More space-efficient encoding, resulting in higher-order binary optimisation (HOBQ) formulations, have been discussed as well [19]. Especially the QUBO formulation of the TSP has recently come under increasing criticism due to well-documented shortcomings in QUBO-based classical methods [20].

As a promising alternative for constrained problems, Hadfield *et al.* [2] enhanced the QAOA to directly incorporate constraints in a problem-specific mixer. They designed concrete mixers for several problem classes, including the TSP, while Ruan *et al.* [21] considered pre-compiling suitable QAOA mixers for TSP. Alternative mixers, derived via similar group-theoretical considerations as in this work, have been proposed for the open-shop scheduling problem [22], containing the TSP as special case. The group-theoretical treatment of constraints complements the analysis in [23], studying the effect of symmetries in the objective function on the QAOA dynamics.

Besides variational methods, there exist other hybrid and quantum approaches to the TSP. Most notably, the application of the closely related framework of quantum annealing has been proposed [24], extensively studied [25], and fine-tuned for the TSP [26]. Another interesting proposal encodes city coordinates directly on the Bloch sphere of a single qubit [27]. Furthermore, Grover’s famous search algorithm [28] has been adapted to the TSP, both, for feasible state preparation and subsequent search [29], based on a generalised Grover oracle [30]. Bärtschi and Eidenbenz [31], alongside their proposal of the Grover-mixer QAOA, provided an efficient preparation routine for the uniform superposition of all TSP states, rendering Grover’s algorithm obsolete for this task. Their construction can, despite the claims in [29], be extended to the more compact HOB0 encoding [32, Corollary 3]. More recently, heuristic, non-uniform state preparation routines for the TSP have been also discussed [33].

III. PRELIMINARIES

Throughout this article, we use the notational short-hands $\log(n) := \lceil \log_2(n) \rceil$, as well as $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$, $\mathbb{b} := \{0, 1\}$ for the bit, and $\mathbb{q} := \mathbb{C}^2$ for the qubit. For $\mathbf{x} \in \mathbb{b}^n$, $|\mathbf{x}|$ denotes its Hamming weight, i.e., the number of bits of \mathbf{x} set to 1.

A. Travelling salesperson problem

Abstractly, an instance of the *Travelling Salesperson Problem* (TSP) on n cities is defined on a complete weighted digraph $K_n = (V, E, w)$ (without self-loops) with $|V| = n$ and edge weights $w: E \rightarrow \mathbb{R}_{>0}$. Feasible solutions correspond to *Hamiltonian cycles*, i.e., cycles that visit every vertex exactly once; K_n has $(n-1)!$ such cycles. The goal is to find a Hamiltonian cycle of minimum total edge weight.

Instead of the linear formulation used by Dantzig, Fulkerson, and Johnson [10], let us consider an alternative encoding which allows for a cleaner transfer to the quantum-mechanical treatment. It was popularised by Lucas [17] for the Ising/QUBO formulation of the TSP and utilises n^2 bits, each indexed by the city $u \in [n]$ and the time $t \in [n]$. Then, $x_{u,t} = 1$ means that city u is visited at time t . Accordingly, if $x_{u,t} = x_{v,t+1} = 1$ (or $x_{u,n} = x_{v,1} = 1$), the tour traverses the

edge $(u, v) \in E$, giving rise to a quadratic objective function, mapping n^2 -bit strings \mathbf{x} to

$$\sum_{u=1}^n \sum_{v=1}^n w(u, v) \left(x_{u,n} x_{v,1} + \sum_{t=1}^{n-1} x_{u,t} x_{v,t+1} \right),$$

summing up the weights of all traversed edges. Note that not all bit strings $\mathbf{x} \in \mathbb{b}^{n^2}$ are valid tours. We have to enforce explicitly that each city is visited exactly once and that this happens at distinct time slots:

$$\sum_{t=1}^n x_{u,t} = 1 \quad \forall u \in [n] \quad \text{and} \quad \sum_{u=1}^n x_{u,t} = 1 \quad \forall t \in [n].$$

Interpreting these n^2 -bit strings as $n \times n$ Boolean matrices, the feasible solutions exactly correspond to permutation matrices, i.e., matrices with exactly one 1 in each row and each column. Especially, there are $n!$ feasible assignments in this encoding so that a single Hamiltonian cycle is necessarily represented by more than just one bit string.

This approach effectively consists of two one-hot encodings for city and time slot. It can be straightforwardly compactified by encoding one of the indices in binary instead, reducing the bit count to $n \log(n)$ [19]. We choose here to keep the one-hot time index and to assign visited cities in $\log(n)$ -subregisters. For now, this complicates the evaluation of the objective function, but will later result in fewer operations to traverse the feasible solutions. Instead of checking two bits $x_{u,t}$ and $x_{v,t+1}$, we have to check which values the two $\log(n)$ -bit substrings \mathbf{x}_t and \mathbf{x}_{t+1} hold. Similarly, for every $u \in [n]$, the first constraint has to evaluate all subregisters \mathbf{x}_t , $t \in [n]$, and check that exactly one of them holds the value of u in binary. Equivalently, feasible states are those where the cities encoded in the time-subregisters form a valid permutation of $[n]$. The second constraint becomes redundant in this encoding, as every time t , by construction, carries a visited city in its associated subregister \mathbf{x}_t . We already remark that the complicated objective function will not have any impact on our circuit constructions.

Both variants introduce a n -fold degeneracy for encoding Hamiltonian cycles as bit strings, stemming from the fact that a time coordinate introduces an artificial “starting city”, i.e., the city visited at time $t = 1$. Choosing any of the n cities along a Hamiltonian cycle as starting city does not alter the cycle itself, but corresponds to different bit strings. We can remove this degeneracy by removing one city—say the n -th city—and one time slot from the variable space, and by implicitly connecting the cities visited at times $t \in [n-1]$ with this city. In the twofold one-hot encoding, the resulting objective function \tilde{c} maps $(n-1)^2$ -bit strings \mathbf{x} to

$$\sum_{u=1}^{n-1} w(n, u) x_{u,1} + w(u, n) x_{u,n-1} + \sum_{v=1}^{n-1} \sum_{t=1}^{n-2} w(u, v) x_{u,t} x_{v,t+1}.$$

The two constraints stay the same, except that they only run over $[n-1]$ in both indices. These changes to objective function and constraints are entirely analogous in the compactified encoding.

B. Elementary group theory and the symmetric group

The encoded n -city TSP can be understood as an optimisation over the symmetric group S_n of all permutations of n elements, or $n - 1$ elements if one starting city is fixed. That is, each feasible solution $\mathbf{x} \in \mathbb{b}^{n^2}$ —resp. $\mathbb{b}^{n \log(n)}$ —can be uniquely identified with an element $\sigma \in S_n$. The most natural such identification is given via the representation of S_n on \mathbb{F}_2^n as permutation matrices which, when flattened to an array, exactly correspond to n^2 -bit strings, as discussed before. For the $n \log(n)$ -encoding, a similarly natural connection can be established: identify a feasible $n \log(n)$ -bit string \mathbf{x} with $\sigma \in S_n$ with $\text{bin}(\sigma(i)) = \mathbf{x}_i$ for all $i \in [n]$. Special elements of S_n which will become important later are the transpositions $\tau_{i,j} \in S_n$ with $i, j \in [n]$ and $i \neq j$, exchanging the i -th and j -th element. They have the useful property of being involutions, i.e., it holds that $\tau_{i,j}^2 = \text{id}$.

Another important, but more abstract concept is that of a group action. The elements of a group G can act on a set X , mapping elements from X again to X . One imposes that this action respects the group structure of G in the sense that for all $x \in X$, it holds that $e_G \cdot x = x$, where e_G is G 's neutral element, and for all $g, h \in G$, it is $g \cdot (h \cdot x) = (gh) \cdot x$. A group can also act on itself via (left or right) multiplication, i.e., simply via $g \cdot h = gh$ for $g, h \in G$. This self-action has the important property of being (*sharp*) *transitive*: for every $h_1, h_2 \in G$, there exists (exactly) one $g \in G$ so that $g \cdot h_1 = h_2$.

The last important concept is that of the permutation representation of a group which is not necessarily S_n . Given a group G acting on a finite set X , one can consider the formal \mathbb{C} -span of X , $\langle X \rangle_{\mathbb{C}}$, consisting of formal \mathbb{C} -linear combinations of elements in X . Then, $\langle X \rangle_{\mathbb{C}}$ is a $|X|$ -dimensional complex vector space with a fixed reference basis provided by the elements of X . The action of G on X can be naturally extended to an action on $\langle X \rangle_{\mathbb{C}}$ simply via linear extension, i.e.,

$$g \cdot \sum_{x \in X} \alpha_x x := \sum_{x \in X} \alpha_x g \cdot x.$$

By construction, this action respects the space's linear structure and therefore represents the elements of G as linear operators on $\langle X \rangle_{\mathbb{C}}$, which are, expressed in the natural reference basis, simple permutation matrices.

C. Quantum combinatorial optimisation

The standard way of mapping combinatorial optimisation problems with objective function $c : \mathbb{b}^m \rightarrow \mathbb{R}$ to a (qubit-based) quantum computer is to simply replace every bit with a qubit and to introduce the diagonal objective Hamiltonian $C \in \mathcal{L}(\mathbb{q}^{\otimes m})$ via $C|\mathbf{x}\rangle = c(\mathbf{x})|\mathbf{x}\rangle$. Then, by construction, ground states of C exactly correspond to (superpositions of) minimisers of c . However, if the problem is constrained, such as the TSP, the ground states of C need not correspond to feasible solutions.

The arguably most prominent approaches for tackling combinatorial ground state search are Farhi *et al.*'s Quantum Approximate Optimisation Algorithm [1] and Hadfield *et al.*'s

generalisation to the Quantum Alternating Operator Ansatz [2] (QAOA). In a nutshell, an instantiation of QAOA prescribes an initial state $|\iota\rangle$, a depth $p \in \mathbb{N}$, and two parametrised quantum circuits: the *phase separator* $U_P(\gamma)$ and the *mixer* $U_M(\beta)$. The (approximate) ground state search is then restricted to the parametrised states of the form

$$|\beta, \gamma\rangle := U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1)|\iota\rangle$$

and can thus be translated to a minimisation task over the $2p$ parameters $\beta = (\beta_1, \dots, \beta_p)$ and $\gamma = (\gamma_1, \dots, \gamma_p)$.

Usually, the phase separator is of the form $U_P(\gamma) = e^{-i\gamma C}$ and is therefore itself diagonal in the computational basis $\{|\mathbf{x}\rangle \in \mathbb{q}^{\otimes m} : \mathbf{x} \in \mathbb{b}^m\}$. For unconstrained or soft-constrained problems¹, a common choice for the initial state and the mixer is the combination of uniform superposition of all computational basis states $|\iota\rangle = |+\rangle$ and $U_M(\beta) = e^{-i\beta B}$, where B is the m -fold sum over all single-qubit Pauli- X operators. $|+\rangle$ is the unique ground state of $-B$. For (hard-)constrained problems with feasible set $F \subset \mathbb{b}^m$, Hadfield *et al.* require the mixer to leave the *feasible subspace* $\mathcal{F} := \text{span}\{|\mathbf{x}\rangle : \mathbf{x} \in F\}$ invariant and to provide transitions between all pairs of feasible states. The latter condition is formalised as

$$\forall \mathbf{x}, \mathbf{y} \in F \exists r \in \mathbb{N} \exists \beta \in \mathbb{R} : \langle \mathbf{x} | U_M^r(\beta) | \mathbf{y} \rangle \neq 0. \quad (1)$$

Note that since the usual phase separator is diagonal in the computational basis, it trivially leaves the feasible subspace invariant. Therefore, choosing a feasible initial state $|\iota\rangle \in \mathcal{F}$ effectively restricts all parametrised states $|\beta, \gamma\rangle$ to the feasible subspace.

Note that, in the language of [Subsection III-B](#), $\mathbb{q}^{\otimes m}$ is nothing but the formal \mathbb{C} -span of \mathbb{b}^m with its computational basis as natural reference basis. Analogously, the feasible subspace \mathcal{F} may be viewed as the formal \mathbb{C} -span of the feasible subset F with the feasible computational basis states as reference basis elements.

We conclude the preliminaries with (a simplified version of) Hadfield *et al.*'s mixer construction for the TSP. They consider the n^2 -encoding of the TSP, but their construction can be readily adapted to the more compact $n \log(n)$ -encoding. First, for an unordered pair of times $\{s, t\}$ and an unordered pair of cities $\{u, v\}$, define the swap partial Hamiltonian

$$H_{\text{PS}, \{s, t\}, \{u, v\}} := S_{u, s}^+ S_{v, t}^+ S_{u, t}^- S_{v, s}^- + S_{u, s}^- S_{v, t}^- S_{u, t}^+ S_{v, s}^+$$

with $S^+ = |1\rangle\langle 0|$ and $S^- = |0\rangle\langle 1|$. This Hamiltonian clearly swaps cities u and v between time slots s and t . Hence, $H_{\text{PS}, t, \{u, v\}} := H_{\text{PS}, \{t, t+1\}, \{u, v\}}$ swaps cities u and v between subsequent time slots t and $t+1$. By summing over all unordered city pairs $\{u, v\}$, we obtain $H_{\text{PS}, t}$, swapping

¹A soft constraint enters as penalty term within the objective function. Assignments violating a soft constraint are still considered feasible, but—depending on the penalty's magnitude—might have worse objective values than strictly feasible solutions in the narrower sense.

whatever cities are visited on subsequent time slots t and $t+1$. The sequential ordering swap mixer is now defined as

$$U_{\text{seq-PS}}(\beta) := \prod_{t=1}^n e^{-i\beta H_{\text{PS},t}}.$$

Hadfield *et al.* also consider the simultaneous version of $U_{\text{seq-PS}}$ with an exponentiated sum of swap operators instead of a product of individually exponentiated swaps. Since $H_{\text{PS},s}$ and $H_{\text{PS},t}$ do not commute if $|t-s|=1$, these two notions are not equivalent.

IV. METHODS

As a consequence of the adiabatic theorem, the original QAOA for unconstrained problems is guaranteed to reach the optimum in the limit of $p \rightarrow \infty$ [1]. That is, for every $\varepsilon > 0$, there exists a $p \in \mathbb{N}$ and a parameter vector $(\beta, \gamma) \in \mathbb{R}^{2p}$ so that $|\langle \beta, \gamma | x^* \rangle| \geq 1 - \varepsilon$, where $x^* \in \mathbb{b}^n$ is the problem's optimal solution. This even holds true when there are multiple optimal solutions [8]. This asymptotic reachability guarantee can be extended to the QAOA for constrained problems by demanding stronger mixing properties [22]. However, these guarantees are of little practicality; neither can current quantum devices execute such deep circuits reliably, nor would their execution result in practical runtime advantages. As a recourse, we formulate here the concept of exhaustively parametrised quantum circuits with non-asymptotic reachability guarantees.

Definition 1. [34, Definition 5.9] Let COP be a combinatorial optimisation problem with feasible subset $F \subset \mathbb{b}^m$. A parametrised quantum circuit $V(\theta)$, $\theta \in \mathbb{R}^d$, is called *exhaustively parametrised* if for every $x \in F$ there exist parameter values θ_x so that $|\langle x | V(\theta_x) | \mathbf{0} \rangle| = 1$.

Especially, every optimal solution $|x^*\rangle$ can be exactly reached by applying $V(\theta_{x^*})$ once to $|\mathbf{0}\rangle$, which in sharp contrast to the usual asymptotic guarantee. The choice of $|\mathbf{0}\rangle$ as universal initial state is not restrictive, as any state preparation unitary can be absorbed into $V(\theta)$.

Definition 1 deliberately does not restrict the parametrised states to the feasible subspace. We introduce this as a separate definition, although all examples constructed in this work conform to both requirements.

Definition 2. [34, Definition 5.9] Let COP be a combinatorial optimisation problem with feasible subset $F \subset \mathbb{b}^m$. A parametrised quantum circuit $V(\theta)$, $\theta \in \mathbb{R}^d$, is called *feasibility-respecting* if for every $x \in \mathbb{b}^m \setminus F$ and every $\theta \in \mathbb{R}^d$ it holds that $\langle x | V(\theta) | \mathbf{0} \rangle = 0$.

Next, we formulate a central ingredient for constructing exhaustively parametrised circuits for problems with group-based constraints, such as the TSP.

Definition 3. [35, Definition 5.1] Let G be a group and $d \in \mathbb{N}$. A sequence $(h_i)_{i \in [d]} \in G^d$ is a *generating sequence* (for G) if for every $g \in G$, there exists $\mathbf{b} \in \mathbb{b}^d$ so that $g = h_d^{b_d} \cdots h_2^{b_2} h_1^{b_1}$.

That is, unlike in the usual notion of group generators, elements within a generating sequence have to be applied in

a fixed order and occur with a fixed multiplicity. We give concrete examples for generating sequences for the symmetric group S_n in the Section V.

For now, consider a generic problem COP with feasible set $F \subset \mathbb{b}^m$ and a group G acting on F . The bit-to-qubit mapping for COP naturally yields the formal \mathbb{C} -spans of \mathbb{b}^m ($\mathbb{q}^{\otimes m}$) and F (\mathcal{F}) and hence a permutation representation of G on \mathcal{F} . As permutation matrices, they are especially unitary and can be extended to unitaries on $\mathbb{q}^{\otimes m}$, i.e., to valid m -qubit quantum gates. For us, it will not be important how the extended representations act outside of the feasible subspace, so that we may pick whatever extensions is easiest to implement. Every such unitary extension readily leaves the feasible subspace invariant.

As a final ingredient, we require a simple, well-known identity for involutory operators. Let $A \in \mathcal{L}(\mathbb{q}^{\otimes m})$ be a (not necessarily unitary) involution, i.e., it is $A^2 = \mathbb{1}$. Then, it holds for every $\theta \in \mathbb{R}$ that

$$\begin{aligned} e^{-i\theta A} &= \sum_{j=0}^{\infty} \frac{(-i)^{2j} \theta^{2j} A^{2j}}{(2j)!} + \sum_{j=0}^{\infty} \frac{(-i)^{2j+1} \theta^{2j+1} A^{2j+1}}{(2j+1)!} \\ &= \sum_{j=0}^{\infty} \frac{(-1)^j \theta^{2j}}{(2j)!} - i \sum_{j=0}^{\infty} \frac{(-1)^j \theta^{2j+1}}{(2j+1)!} A \\ &= \cos(\theta) - i \sin(\theta) A. \end{aligned} \quad (2)$$

We are now in the position to state and prove our main abstract result.

Theorem 4. Let COP be a combinatorial optimisation problem with feasible subset $F \subset \mathbb{b}^m$, let G be a group acting transitively on F , let $(h_i)_{i \in [d]} \in G^d$ be a generating sequence for G of involutions, and, for every $i \in [d]$, let $H_i \in \mathcal{L}(\mathbb{q}^{\otimes m})$ be the involutory unitary so that $H_i|_{\mathcal{F}}$ is the permutation representation of h_i . Then, for every unitary $W \in \mathcal{L}(\mathbb{q}^{\otimes m})$ with $W|\mathbf{0}\rangle \propto |x\rangle$ for some $x \in F$, the unitary

$$V(\theta) := e^{-i\theta_d H_d} \cdots e^{-i\theta_2 H_2} e^{-i\theta_1 H_1} W$$

is an exhaustively parametrised, feasibility-respecting quantum circuit for COP.

Proof. Since h_i is an involution, so is its permutation representation $\hat{h}_i \in \mathcal{L}(\mathcal{F})$. Therefore, each \hat{h}_i can be extended to an involutory unitary $H_i \in \mathcal{L}(\mathbb{q}^{\otimes m})$ —e.g., canonically by $H_i|_{\mathcal{F}^\perp} := \mathbb{1}$. Now, fix $x \in F$ and let $y \in F$ be arbitrary. Since G acts transitively on F , there exists some $g \in G$ so that $g \cdot x = y$. By the definition of a generating sequence, there exists $\mathbf{b} \in \mathbb{b}^d$ so that $g = h_d^{b_d} \cdots h_2^{b_2} h_1^{b_1}$. As shown in (2), H_i being involutory implies that $e^{-i\theta H_i} = \cos(\theta)\mathbb{1} - i \sin(\theta)H_i$ for all $\theta \in \mathbb{R}$. Especially, it holds that $e^{-i0H_i} = \mathbb{1}$ and $e^{-i\pi H_i/2} = -iH_i$. Let $\theta_y := \pi \mathbf{b}/2$, then

$$\begin{aligned} V(\theta_y) |\mathbf{0}\rangle &= (-iH_d)^{b_d} \cdots (-iH_2)^{b_2} (-iH_1)^{b_1} W |\mathbf{0}\rangle \\ &= e^{i\xi} (-i)^{|\mathbf{b}|} H_d^{b_d} \cdots H_2^{b_2} H_1^{b_1} |x\rangle = e^{i\xi} (-i)^{|\mathbf{b}|} |y\rangle. \end{aligned}$$

That is, $|\langle y | V(\theta_y) | \mathbf{0} \rangle| = |e^{i\xi} (-i)^{|\mathbf{b}|} \langle y | y \rangle| = 1$. Furthermore, since $\mathbb{1}$ and all H_i , $i \in [d]$, leave \mathcal{F} invariant, and $W|\mathbf{0}\rangle \in \mathcal{F}$, $V(\theta)$ is readily feasibility-respecting. \square

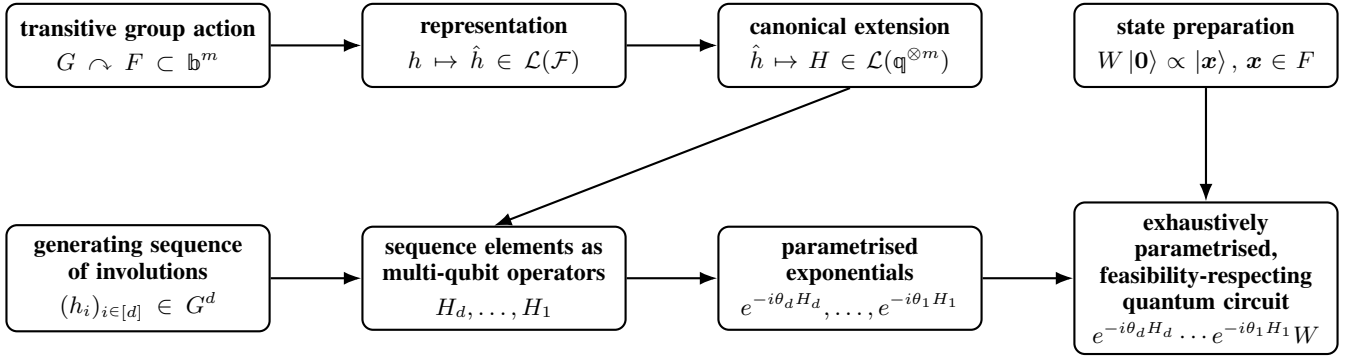


Fig. 2. Abstract pipeline for constructing an exhaustively parametrised, feasibility-respecting quantum circuit from a transitive group action on the feasible set $F \subset \mathbb{b}^m$. Central to this construction is the existence of a generating sequence of involutions, whose elements can be represented as m -qubit operators. Exponentiating these operators, introducing real parameters in the exponent, and prepending the state preparation of some feasible computational basis state $|\mathbf{x}\rangle$, completes the construction.

Theorem 4 establishes an abstract pipeline for obtaining exhaustively parametrised, feasibility-preserving quantum circuits for problems with a group acting transitively on their feasible subset and for which a generating sequence of involutions is known. This pipeline is also schematically depicted in [Figure 2](#). Importantly, it only ensures exact reachability and not that the parameters realising any optimal solution can be found efficiently.

Before deriving concrete examples in [Section V](#), let us first make the general case more practical. Suitable unitaries W that prepare the initial feasible state $|\mathbf{x}\rangle$ can be simple Pauli-X layers, applying a bit flip to the i -th qubit whenever $x_i = 1$. Furthermore, since $e^{-i\theta H_i}$ is 2π -periodic in θ , and $e^{-i(\theta+\pi)H_i} = e^{-i\theta H_i} = -e^{i\theta H_i}$, the parameter range for θ can be effectively restricted to $[0, \pi)$. Moreover, note that the pipeline does, unlike the QAOA, not feature the concept of a phase separator at all, but solely focusses on the feasibility structure. This makes it simpler to transfer a construction between problems with similar feasibility features, but different objective functions, and protects against more complicated objective functions incurred by a denser encoding, as presented for the TSP. Lastly, recall the standard circuit construction for implementing an exponentiated involutory unitary $U \in \mathcal{L}(\mathbb{q}^{\otimes m})$ using one ancilla qubit. The circuit is depicted in [Figure 3](#). Concretely, if U is not only unitary, but also an involution, then

$$e^{-i\theta U} = (|0\rangle_{\text{H}} \otimes \mathbb{1})C(U)(\text{RX}(2\theta) \otimes \mathbb{1})C(U)(\text{H}|0\rangle \otimes \mathbb{1})$$

That is, the parametrised exponential can be implemented with an additional $|0\rangle$ -initialised ancilla qubit and with two Hadamard gates, a parametrised rotational X-gate and two controlled versions of U .

V. RESULTS

We present now two concrete constructions of exhaustively parametrised circuits for the TSP. Both will follow the abstract pipeline established by [Theorem 4](#) and the subsequent comments. Consider an n -city instance in either the twofold

one-hot encoding or the more compact $n \log(n)$ -encoding. For simplicity, we stick to the non-reduced case with $n!$ (instead of $(n-1)!$) feasible solutions, but the reduced case follows the same logic. The feasible solutions directly correspond to the elements S_n , e.g., via the identification of n^2 -bit strings with $n \times n$ permutation matrices or the interpretation of $n \log(n)$ -bit strings as one-line notations of permutations in binary. Symbolically, any of these identifications establishes a bijection $\epsilon: S_n \rightarrow F$, where F is the set of feasible bit strings. The (sharp) transitive group action of S_n on itself via left multiplication thereby extends to a (sharp) transitive action on F via $\sigma \cdot \mathbf{x} := \epsilon(\sigma \epsilon^{-1}(\mathbf{x}))$, where $\mathbf{x} \in F$ and $\sigma \in S_n$.

As the initial feasible state we may, without loss of generality, choose the bit string representing the identity permutation. In the n^2 -encoding, it is given by $x_{i,i} = 1$ and $x_{i,j} = 0$ for all $i, j \in [n]$ with $i \neq j$. Preparing this state from $|0\rangle$ thus requires n Pauli-X gates. In the $n \log(n)$ -encoding, the bit string is given by $x_i = \text{bin}(i-1)$ for all $i \in [n]$. This requires the application of $\approx n \log(n)/2$ Pauli-X gates to $|0\rangle$, since the average Hamming weight of the first n integers' binary representations is $\log(n)/2$.

A. Bubble sort sequence

The given transitive group action on the feasible set of the TSP is somewhat natural. However, there is no canonical candidate for a generating sequence of involutory elements. Our first example is derived from the (optimised) bubble sort algorithm [36]. Bubble sort iterates through an array of length n element by element, comparing neighbouring values and swapping them if they are in the wrong order. It repeats this procedure, skipping comparison with the last element since it is already in its necessarily already at the correct position. That is, the second iteration effectively operates on an array of length $n-1$. In each iteration, the effective length reduces by one, until it reaches 0, meaning that the entire array is sorted. In summary, the i -th iteration consists of $n-i$ adjacency comparisons and subsequent conditional swaps. On an abstract

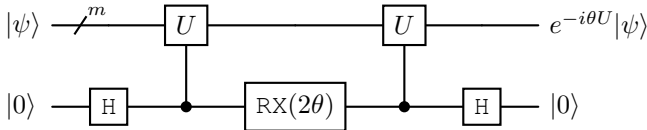


Fig. 3. Implementation of $e^{-i\theta U}$, where U is a Hermitian unitary.

level, and by reorganising the iterations $i \mapsto n - i$, bubble sort is able to apply all the permutations

$$\prod_{i=1}^{n-1} \left(\tau_i^{b_{i,i}} \circ \dots \circ \tau_1^{b_{i,1}} \right), \quad (3)$$

where $\tau_i := \tau_{i,i+1}$ is an adjacency transposition, and the bits $b_{i,j}$, $i \in [n-1]$ and $j \in [i]$, determines whether two neighbouring elements j and $j+1$ are swapped in the i -th iteration ($b_{i,j} = 1$) or not ($b_{i,j} = 0$). That bubble sort can sort every possible input array—or, equivalently, can invert every possible permutation of an initially sorted array—corresponds exactly to the fact that the *bubble sort sequence* of adjacency transpositions in the order given by (3) is a generating sequence for S_n . We denote the bubble sort sequence for S_n by Δ_n and observe that

$$|\Delta_n| = \frac{n(n-1)}{2} \in \mathcal{O}(n^2). \quad (4)$$

Its elements are involutions so that [Theorem 4](#) is applicable. Furthermore, by fixing a starting city, we can simply exchange n with $n-1$ in the above derivation.

Let us further discuss the representation of τ_i , $i \in [n]$, as m -qubit operator, where $m \in \{n^2, n \log(n)\}$. In both cases, τ_i also acts naturally on \mathbb{b}^m (and not only the feasible subset) by swapping the i -th and $(i+1)$ -th substrings.² Defining $r := m/n \in \{n, \log(n)\}$, the i -th substring consists of the r bit positions $(i-1)r+1, \dots, ir$. That is, τ_i acts as r individual, disjoint transpositions of bit positions. Each of these corresponds to a SWAP-gate on the m -qubit space, yielding the representation of τ_i as

$$\hat{\tau}_i = \prod_{k=1}^r \text{SWAP}_{(i-1)r+k}^{ir+k}. \quad (5)$$

By indexing the qubits in “column-major” format, i.e., by distributing the i -th subregister over the qubits $(k-1)n+i$, $k \in [r]$, all SWAP-gates only act on nearest neighbours in an m -qubit chain. However, implementing the parametrised exponentials $e^{-i\theta \hat{\tau}_i}$ via the circuit construction shown in [Figure 3](#) requires one additional ancilla qubit with full connectivity to all other qubits, or, alternatively, $m-1$ ancilla qubits, each of which is connected to two subsequent data qubits. The controlled version of $\hat{\tau}_i$ decomposes into the r -fold product of controlled swaps—i.e., Fredkin gates—where the target qubits are nearest neighbours.

²To be more precise, swapping subregisters in the $n \log(n)$ -encoding corresponds to a right action. For the reachability argument, this is not important, as also right self-actions of groups are transitive.

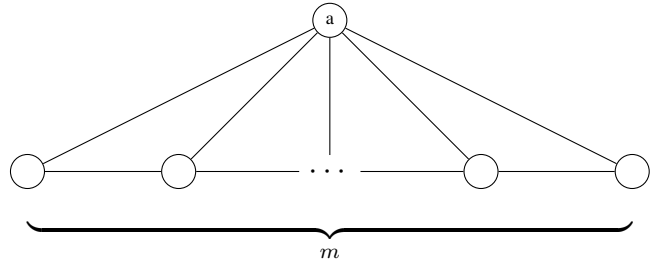


Fig. 4. Qubit connectivity required for implementing the exhaustively parametrised quantum circuit based on the bubble sort sequence. The m data qubits, encoding the TSP solutions, can be aligned on a chain. The ancilla qubit, serving as control for the Fredkin gates, has to be connected to all m data qubits. Alternatively, one can introduce $m-1$ ancilla qubits, each connected to two contiguous data qubits.

Corollary 5. *For the n -city TSP, there exists an exhaustively parametrised quantum circuit with $(n-1)(n-2)/2$ parameters, even on star-augmented nearest-neighbour architectures (see [Figure 4](#)).*

B. Binary insertion sequence

Among generating sequences, only containing adjacency transpositions, the bubble sort sequence is already of optimal length.³ However, we can find shorter generating sequences, containing also other involutions than adjacency transpositions. Our construction relies on grouping together disjoint (non-adjacency) transpositions so that their product is again an involution. The approach is similarly inductive on n as the construction of the bubble sort sequence Δ_n . The latter arises from Δ_{n-1} by embedding it into S_n via the index increment $\tilde{\sigma}(i) = \sigma(i-1) + 1$ and $\tilde{\sigma}(1) = 1$ for $\sigma \in S_{n-1}$, and by subsequently appending the $n-1$ adjacency transpositions $\tau_i \in S_n$, $i \in [n-1]$. These $n-1$ additional elements ensure that 1 can be mapped to every other position $j \in [n] \setminus \{1\}$. However, by grouping together multiple transpositions in a single permutation and by sacrificing adjacency, one can achieve the same with less elements.

Lemma 6. *Let $(\rho_i)_{i \in [t]}$ be a generating sequence for S_{n-1} and define, for $\ell \in [\log(n)]$, $\pi_\ell \in S_n$ so that $\pi_\ell(j) = j + 2^{\ell-1}$ and $\pi_\ell(j + 2^{\ell-1}) = j$ for all $j \in [\min\{n - 2^{\ell-1}, 2^{\ell-1}\}]$. Then, $(\tilde{\rho}_1, \dots, \tilde{\rho}_t, \pi_1, \dots, \pi_{\log(n)})$ is a generating sequence for S_n .*

Proof. Let $\beta \in S_n$ be arbitrary and let $\mathbf{p} \in \mathbb{b}^{\log(n)}$ be the binary representation of $\beta(1) - 1$. Furthermore, define the permutation $\pi^{\mathbf{p}} := \pi_{\log(n)}^{p_{\log(n)}} \dots \pi_1^{p_1}$. We claim that

$$\left((\pi^{\mathbf{p}})^{-1} \beta \right) (1) = 1. \quad (6)$$

³Especially, the element $\mu \in S_n$ with $\mu(i) = n+1-i$ for all $i \in [n]$ has to be generated. In the language of inversion numbers [37]: μ has inversion number $n(n-1)/2$, while id has inversion number 0. Multiplying a permutation with an adjacency transposition increases or decreases its inversion number by 1. Hence, reaching μ from id only with multiplications by adjacency transpositions requires at least $n(n-1)/2$ of them.

Provided (6) holds, $(\rho_i)_{i \in [t]}$ being a generating sequence for S_{n-1} implies that there exists $\mathbf{q} \in \mathbb{b}^t$ so that

$$\tilde{\rho}_t^{q_t} \cdots \tilde{\rho}_1^{q_1} = (\boldsymbol{\pi}^{\mathbf{P}})^{-1} \beta \implies \beta = \pi_{\log(n)}^{p_{\log(n)}} \cdots \pi_1^{p_1} \tilde{\rho}_t^{q_t} \cdots \rho_1^{q_1}.$$

Since $\beta \in S_n$ was chosen arbitrarily, this already proves the assertion. \square

In summary, [Lemma 6](#) establishes that generating sequences for S_{n-1} can be extended to generating sequence for S_n with only $\log(n)$ additional elements. In addition, these elements are involutions. Starting from the empty sequence $()$, which is (by convention) generating for $S_1 = \{\text{id}\}$, we can now simply apply [Lemma 6](#) $n-1$ times in order to obtain a sequence Ξ_n —which we name *binary insertion sequence*—for S_n of length

$$|\Xi_n| = \sum_{i=2}^n \log(i) \in \mathcal{O}(n \log(n)), \quad (7)$$

only consisting of involutions. Therefore, [Theorem 4](#) is again applicable. Notably, since the n -city TSP has $\mathcal{O}(n!)$ feasible solutions, it follows from Stirling’s formula that the length of Ξ_n is asymptotically optimal.

We again discuss the representation of the sequence elements π_ℓ . They are comprised by $\min\{n - 2^{\ell-1}, 2^{\ell-1}\}$ individual, disjoint transpositions $\tau_{j, j+2^{\ell-1}}$. Analogous to the adjacency transpositions, each of those will be represented as a subregister swap, i.e., by an r -fold product of SWAP-gates. Since the transpositions $\tau_{j, j+2^{\ell-1}}$ act on mutually distinct elements, their fully decomposed representation consists of $r \min\{n - 2^{\ell-1}, 2^{\ell-1}\}$ disjoint SWAP-gates:

$$\hat{\pi}_\ell = \prod_{j=1}^{\min\{n-2^{\ell-1}, 2^{\ell-1}\}} \prod_{k=1}^r \text{SWAP}_{(j-1)r+k}^{(j+2^{\ell-1}-1)r+k} \quad (8)$$

Since the elements π_ℓ in Ξ_n do not correspond to adjacency transpositions anymore, it requires better qubit connectivity than the one depicted in [Figure 4](#), which suffices to implement the bubble sort version.

Corollary 7. *For the n -city TSP, there exists an exhaustively parametrised quantum circuit with $\log(2) + \dots + \log(n)$ parameters.*

C. Numerical comparison

After having established exact reachability guarantees for the circuits based on the bubble sort and the binary insertion sequence, we numerically study how well their parameters can be optimised for a concrete example. We consider a 9-city TSP instance whose weight matrix can be found in the [companion code base](#). Using the $n \log(n)$ -encoding, we require $(9 - 1) \log(9 - 1) = 24$ qubits for representing the instance, effectively reducing the number of cities to $n = 8$. We benchmark our two constructions against the QAOA with Hadfield *et al.*’s TSP mixers and the standard phase separator. For our proof-of-principle, we assume all qubit and circuit components to be noiseless, all qubits to be fully connected, and work with exact expectation values rather

than sample-based approximations. We conduct the simulation using Ziegler’s low-level quantum simulator Orkan [\[38\]](#).

Recalling our two constructions for the TSP, one can easily verify that both circuits, despite differing numbers of parameters, have the exact same circuit length. The binary insertion sequence simply groups together multiple transpositions in one element. One QAOA layer, in turn, contains only $n - 1$ exponentiated transpositions and the phase separator. As a benevolent assumption, we assume the phase separator to have the same circuit length as one exponentiated transposition. Hence $(n - 1)/2$ QAOA layers have roughly the same circuit length as the two exhaustively parametrised circuits with their $n(n - 1)/2$ exponentiated transpositions. Prescribing the circuit length in the above fashion, the QAOA thus has $n - 1$ variational parameters, the binary insertion circuit has $\log(2) + \dots + \log(n)$ parameters, and the bubble sort circuit has $n(n - 1)/2$ parameters. The parameter optimisation is carried out by the COBYLA optimiser [\[3\]](#), a pseudo gradient-based optimiser. For all three methods, the objective Hamiltonian’s expectation value serves as the objective to be optimised and recorded. If the numerical gradient of several subsequent iterations falls short of a small, predefined threshold, we terminate the procedure.

[Figure 5](#) depicts the curves of obtained expectation values, normalised by the optimal value to yield approximation ratios. Besides our two exhaustively parametrised circuits, we tested QAOA with two different initial states: a single feasible computational basis state and the uniform superposition of all feasible states. For the chosen instance, both QAOA variants fail to deliver notably improved approximation ratios, while the two exhaustive constructions perform significantly better. The bubble sort-based circuit plateaus at an approximation ratio around 0.83, while the circuit based on binary insertion even reaches a value of roughly 0.91. That is, despite the exact reachability guarantee of both circuits, the optimal parameters are (unsurprisingly) not found by the (local) optimiser. We attest the better performance of the binary insertion-based variant over the bubble sort version to the former’s lower-dimensional parameter space, allowing the optimiser to navigate more efficiently.

VI. CONCLUSION

In this work, we have introduced the concept of exhaustively parametrised, feasibility-respecting quantum circuits, which, by design, are able to exactly reach all feasible solutions of a given optimisation problem with finitely many parameters, including optimal states, while avoiding infeasible states entirely. These non-asymptotic reachability guarantees are unmatched by generic QAOA circuits. We have further established an abstract pipeline for promoting a transitive group action on a problem’s feasible set to such an exhaustively parametrised, feasibility-respecting circuit. The construction is centred around the simple, yet powerful group-theoretical concept of generating sequences. We showed that generating sequences yield natural candidates for highly expressive quantum circuits. Additionally demanding the sequence elements

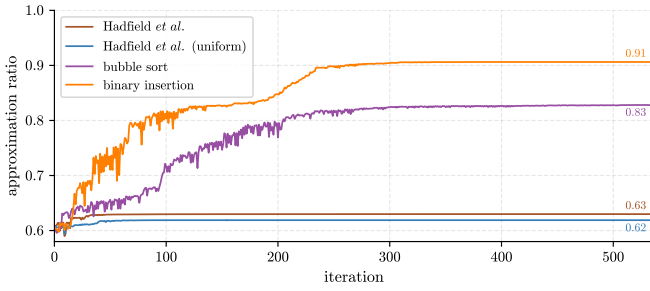


Fig. 5. Approximation ratios obtained after each COBYLA iteration, using different parametrised quantum circuits on a 9-city TSP instance. The QAOA with Hadfield *et al.*'s mixers was not able to significantly improve initial state's approximation ratio, neither if a single computational basis state, nor the uniform superposition of all feasible states, was used. In contrast, the exhaustively parametrised circuit based on bubble sort was able to push the initial ratio of ≈ 0.60 to 0.83, while the construction based on the binary insertion sequence even reached a final approximation ratio of 0.91. Not depicted here: the bubble sort-inspired approach did not terminate before 2300 steps, but was not able to significantly improve its approximation ratio during this long runtime. In contrast, the binary insertion variant and the QAOA initialised in a single computational basis state terminated after roughly 600 steps. The QAOA with uniform initial state already terminated after roughly 150 iterations, but failed to deliver a high-quality approximation ratio.

to be involutions allows us to continuously parametrised these circuits, reproducing exact application or omission of individual sequence elements.

For the n -city Travelling Salesperson Problem (TSP), whose feasible set can be naturally identified with the symmetric group S_n , we have introduced two concrete classes of exhaustively parametrised, feasibility-respecting circuits. Both are based on the (sharply) transitive group action of S_n on itself, and only differ in the chosen generating sequence. The first example consists of $\mathcal{O}(n^2)$ adjacency transpositions and is inspired by the bubble sort algorithm. The second example is more involved, but only contains $\mathcal{O}(n \log(n))$ elements, drastically reducing the number of parameters necessary for exact reachability.

As a proof-of-principle, we have tested both constructions against the QAOA with established TSP-mixers on a 9-city instance. For our concrete example, we observed improved performance of both exhaustively parameterised circuits over a 7-layer QAOA circuit. Moreover, the circuit with $\mathcal{O}(n \log(n))$ parameters performed notably better than the bubble sort-based circuit with its $\mathcal{O}(n^2)$ parameters, which we attribute to the classical solver's ability to navigate better in lower-dimensional parameter spaces. However, none of the two methods was able—despite the theoretical reachability guarantee—to recover the optimum exactly. This underscores once more that reachability does not imply efficient trainability. Ultimately, our proof-of-principle does not substitute for extensive benchmarking, and the investigating the implications of exact reachability to the selection and adaptation of the classical solver remains an interesting open research question.

We highlight that the pipeline utilised to find concrete circuits for the TSP is general enough to be applicable to other

problems as well. Abstractly, any group acting transitively on a problem's feasible set and has a generating sequence of involutions can be promoted to an exhaustively parametrised, feasibility-respecting quantum circuit. A promising research direction will be to identify such group actions for other, more challenging combinatorial optimisation problems and to apply the pipeline to those problems. Here, versions of facility location problems and vehicle routing seem to have richer symmetries in their feasible sets than packing problems. Possible applications of our framework also arise when a group action only covers part of the feasible set. In such cases, the pipeline might be enhanceable by additional, conditional operations to complete coverage of the entire feasible set.

APPENDIX

Proof of (6). For $\ell \in [\log(n)]$, define $\pi_r^{\mathbf{p}} := \pi_\ell^{p_\ell} \dots \pi_1^{p_1}$ so that $\pi_{\log(n)}^{\mathbf{p}} = \pi^{\mathbf{p}}$. We prove that

$$\pi_\ell^{\mathbf{p}}(1) = \sum_{i=1}^{\ell} p_i 2^{i-1} + 1$$

for all $\ell \in [\log(n)]$ via (finite) induction on ℓ . For $\ell = \log(n)$, we then obtain that

$$\pi^{\mathbf{p}}(1) = \sum_{i=1}^{\log(n)} p_i 2^{i-1} + 1 = \beta(1) - 1 + 1 = \beta(1),$$

since \mathbf{p} is the binary representation of $\beta(1) - 1$. This identity already establishes (6) via multiplication with $(\pi^{\mathbf{p}})^{-1}$ from the left.

Base case. Let $\ell = 1$. Then it is $\pi_1^{\mathbf{p}}(1) = \pi_1^{p_1}(1)$. Since $\min\{n - 2^{\ell-1}, 2^{\ell-1}\} = \min\{n - 1, 1\} = 1$ for all $n \geq 2$, it is $\pi_1(1) = 2$, $\pi_1(2) = 1$, and $\pi_1(j) = j$ for all $j \geq 3$. Hence,

$$\pi_1^{p_1}(1) = 1 + p_1 1 = p_1 2^0 + 1 = \sum_{i=1}^1 p_i 2^{i-1} + 1.$$

Induction step. Let $\ell > 1$ and assume that

$$\pi_{\ell-1}^{\mathbf{p}}(1) = \sum_{i=1}^{\ell-1} p_i 2^{i-1} + 1$$

holds. If $p_\ell = 0$, then the claim is trivially true also for ℓ . Therefore, consider only the case of $p_\ell = 1$. It is

$$\pi_\ell^{\mathbf{p}}(1) = (\pi_\ell^{p_\ell} \pi_{\ell-1}^{\mathbf{p}})(1) = \pi_\ell \left(\sum_{i=1}^{\ell-1} p_i 2^{i-1} + 1 \right) =: j_0,$$

where we have already used the induction hypothesis. Observe that $j_0 \leq 2^{\ell-1}$ (with equality only if $p_i = 1$ for all $i \in [\ell-1]$). Furthermore, it is

$$j_0 + 2^{\ell-1} = j_0 + p_\ell 2^{\ell-1} = \sum_{i=1}^{\ell} p_i 2^{i-1} + 1 = \beta(1) \leq n.$$

In summary, $j_0 \leq \min\{n - 2^{\ell-1}, 2^{\ell-1}\}$, so that π_ℓ acts non-trivially on j_0 by incrementing it by $2^{\ell-1} = p_\ell 2^{\ell-1}$, yielding the correct result. \square

ACKNOWLEDGMENT

We thank Tim Heine, Tobias J. Osborne, Debora Ramacciotti, and Sören Wilkening for helpful discussions.

Data and code availability statement. The depicted data can be found at https://github.com/Timo59/qce26_numerics.

REFERENCES

- [1] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm,” 2014, [arXiv preprint [arXiv:1411.4028](https://arxiv.org/abs/1411.4028)].
- [2] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli *et al.*, “From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz,” *Algorithms*, vol. 12, no. 2, p. 34, 2019. [Online]. Available: <https://doi.org/10.3390/a12020034>
- [3] M. J. D. Powell, “Direct search algorithms for optimization calculations,” *Acta Numer.*, vol. 7, pp. 287–336, 1998. [Online]. Available: <https://doi.org/10.1017/s0962492900002841>
- [4] D. Kraft, “Algorithm 733: TOMP–Fortran modules for optimal control calculations,” *ACM Trans. Math. Softw.*, vol. 20, no. 3, pp. 262–281, 1994. [Online]. Available: <https://doi.org/10.1145/192115.192124>
- [5] A. Agirre, E. van Nieuwenburg, and M. M. Wauters, “A Monte Carlo Tree Search approach to QAOA: finding a needle in the haystack,” *New J. Phys.*, vol. 27, no. 4, p. 043014, 2025. [Online]. Available: <https://doi.org/10.1088/1367-2630/adc765>
- [6] J. Wurtz and D. Lykov, “Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs,” *Phys. Rev.*, vol. 104, no. 5, 2021. [Online]. Available: <https://doi.org/10.1103/physreva.104.052419>
- [7] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, “Quantum Computation by Adiabatic Evolution,” 2000, [arXiv preprint [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106)].
- [8] L. Binkowski, G. Koßmann, T. Ziegler, and R. Schwonnek, “Elementary proof of QAOA convergence,” *New J. Phys.*, vol. 26, no. 7, p. 073001, 2024. [Online]. Available: <https://doi.org/10.1088/1367-2630/ad59bb>
- [9] D. L. Applegate, R. E. Bixby, V. Chvátal, W. Cook, D. G. Espinoza *et al.*, “Certification of an optimal TSP tour through 85,900 cities,” *Oper. Res. Lett.*, vol. 37, no. 1, pp. 11–15, 2009. [Online]. Available: <https://doi.org/10.1016/j.orl.2008.09.006>
- [10] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a Large-Scale Traveling-Salesman Problem,” *J. Oper. Res. Soc. Am.*, vol. 2, no. 4, pp. 393–410, 1954. [Online]. Available: <https://doi.org/10.1287/opre.2.4.393>
- [11] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, “Cutting planes in integer and mixed integer programming,” *Discret. Appl. Math.*, vol. 123, no. 1-3, pp. 397–446, 2002. [Online]. Available: [https://doi.org/10.1016/s0166-218x\(01\)00348-1](https://doi.org/10.1016/s0166-218x(01)00348-1)
- [12] A. H. Land and A. G. Doig, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, vol. 28, no. 3, p. 497, 1960. [Online]. Available: <https://doi.org/10.2307/1910129>
- [13] R. Bellman, “Dynamic Programming Treatment of the Travelling Salesman Problem,” *J. ACM*, vol. 9, no. 1, pp. 61–63, 1962. [Online]. Available: <https://doi.org/10.1145/321105.321111>
- [14] M. Held and R. M. Karp, “A Dynamic Programming Approach to Sequencing Problems,” *J. Soc. Ind. Appl. Math.*, vol. 10, no. 1, pp. 196–210, 1962. [Online]. Available: <https://doi.org/10.1137/0110015>
- [15] G. Kochenberger, J. Hao, F. Glover, M. Lewis, Z. Lü *et al.*, “The unconstrained binary quadratic programming problem: a survey,” *J. Comb. Optim.*, vol. 28, no. 1, pp. 58–81, 2014. [Online]. Available: <https://doi.org/10.1007/s10878-014-9734-0>
- [16] J. J. Hopfield and D. W. Tank, ““Neural” computation of decisions in optimization problems,” *Biol. Cybern.*, vol. 52, no. 3, pp. 141–152, 1985. [Online]. Available: <https://doi.org/10.1007/bf00339943>
- [17] A. Lucas, “Ising formulations of many NP problems,” *Front. Phys.*, vol. 2, p. 5, 2014. [Online]. Available: <https://doi.org/10.3389/fphy.2014.00005>
- [18] R. H. Warren, “Adapting the traveling salesman problem to an adiabatic quantum computer,” *Quantum Inf. Process.*, vol. 12, no. 4, pp. 1781–1785, 2012. [Online]. Available: <https://doi.org/10.1007/s11128-012-0490-8>
- [19] A. Glos, A. Krawiec, and Z. Zimborás, “Space-efficient binary optimization for variational computing,” 2020, [arXiv preprint [arXiv:2009.07309](https://arxiv.org/abs/2009.07309)].
- [20] K. A. Smith-Miles, H. H. Hoos, H. Wang, T. Bäck, and T. J. Osborne, “The travelling salesperson problem and the challenges of near-term quantum advantage,” *Quantum Sci. Technol.*, vol. 10, no. 3, p. 033001, 2025. [Online]. Available: <https://doi.org/10.1088/2058-9565/add61d>
- [21] Y. Ruan, S. Marsh, X. Xue, Z. Liu, and J. Wang, “The Quantum Approximate Algorithm for Solving Traveling Salesman Problem,” *Comput. Mater. & Contin.*, vol. 63, no. 3, pp. 1237–1247, 2020. [Online]. Available: <https://doi.org/10.32604/cmcc.2020.010001>
- [22] L. Binkowski, G. Koßmann, C. Tutschku, and R. Schwonnek, “Symmetry-based quantum algorithms for open-shop scheduling with hard constraints,” *Acad. Quantum*, vol. 2, no. 3, 2025. [Online]. Available: <https://doi.org/10.20935/acadquant7900>
- [23] R. Shaydulin, S. Hadfield, T. Hogg, and I. Safro, “Classical symmetries and the Quantum Approximate Optimization Algorithm,” *Quantum Inf. Process.*, vol. 20, no. 11, 2021. [Online]. Available: <https://doi.org/10.1007/s11128-021-03298-4>
- [24] R. Martoňák, G. E. Santoro, and E. Tosatti, “Quantum annealing of the traveling-salesman problem,” *Phys. Rev.*, vol. 70, no. 5, 2004. [Online]. Available: <https://doi.org/10.1103/physreve.70.057701>
- [25] R. H. Warren, “Solving the traveling salesman problem on a quantum annealer,” *SN Appl. Sci.*, vol. 2, no. 1, 2019. [Online]. Available: <https://doi.org/10.1007/s42452-019-1829-x>
- [26] Y. Chen and P. Zhang, “Optimized annealing of traveling salesman problem from the nth-nearest-neighbor distribution,” *Phys. A: Stat. Mech. its Appl.*, vol. 371, no. 2, pp. 627–632, 2006. [Online]. Available: <https://doi.org/10.1016/j.physa.2006.04.052>
- [27] K. Goswami, G. Anekonda Veereshi, P. Schmelcher, and R. Mukherjee, “Solving the travelling salesman problem using Bloch sphere encoding,” *Quantum Sci. Technol.*, vol. 11, no. 1, p. 015007, 2025. [Online]. Available: <https://doi.org/10.1088/2058-9565/ae1e9a>
- [28] L. K. Grover, “A fast quantum mechanical algorithm for database search,” 1996, [arXiv preprint [arXiv:quant-ph/9605043](https://arxiv.org/abs/quant-ph/9605043)].
- [29] R. Sato, C. Gordon, K. Saito, H. Kawashima, T. Nikuni *et al.*, “Two-Step Quantum Search Algorithm for Solving Traveling Salesman Problems,” *IEEE Trans. Quantum Eng.*, vol. 6, pp. 1–12, 2025. [Online]. Available: <https://doi.org/10.1109/tqe.2025.3548706>
- [30] J. Bang, J. Ryu, C. Lee, S. Yoo, J. Lim *et al.*, “A quantum heuristic algorithm for the traveling salesman problem,” *J. Korean Phys. Soc.*, vol. 61, no. 12, pp. 1944–1949, 2012. [Online]. Available: <https://doi.org/10.3938/jkps.61.1944>
- [31] A. Bartschi and S. Eidenbenz, “Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation,” in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2020, pp. 72–82. [Online]. Available: <https://doi.org/10.1109/qce49297.2020.00020>
- [32] L. Binkowski and M. Schwiering, “Quantum Fisher-Yates shuffle: Unifying methods for generating uniform superpositions of permutations,” 2025, [arXiv preprint [arXiv:2504.17965](https://arxiv.org/abs/2504.17965)].
- [33] M. Hess, L. Palackal, A. Awasthi, P. J. Eder, M. Schnaus *et al.*, “Grover Adaptive Search with Problem-Specific State Preparation,” 2026, [arXiv preprint [arXiv:2602.08418](https://arxiv.org/abs/2602.08418)].
- [34] L. Binkowski, “Quantum algorithms for combinatorial optimisation,” 2026. [Online]. Available: https://www.itp.uni-hannover.de/fileadmin/itp/qinfo/Team_Tobias_Osborne/Doctoral_Theses/Lennart_Binkowski_Doctoral_Thesis.pdf
- [35] M. Schwiering, “Quantum Optimization Algorithms for the Traveling Salesman Problem,” 2024. [Online]. Available: https://www.itp.uni-hannover.de/fileadmin/itp/qinfo/Team_Tobias_Osborne/Masters_Theses/Marvin_Schwiering_Masters_Thesis.pdf
- [36] E. H. Friend, “Sorting on Electronic Computer Systems,” *J. ACM*, vol. 3, no. 3, pp. 134–168, 1956. [Online]. Available: <https://doi.org/10.1145/320831.320833>
- [37] H. Manilla, “Measures of Presortedness and Optimal Sorting Algorithms,” *IEEE Trans. Comput.*, vol. C-34, no. 4, pp. 318–325, 1985. [Online]. Available: <https://doi.org/10.1109/tc.1985.5009382>
- [38] T. Ziegler, “Orkan: Cache-friendly simulation of quantum operations on hermitian operators,” 2026, [arXiv preprint [arXiv:2604.15765](https://arxiv.org/abs/2604.15765)].