Institut für Theoretische Physik
Leibniz Universität Hannover

Constraint Graph Model Analysis of the Quantum Alternating Operator Ansatz

Master's Thesis

Lennart Binkowski
10011560

Supervisor:

Prof. Dr. Tobias J. Osborne

August 20, 2022

Abstract

We study the quantum alternating operator ansatz (QAO) proposed by Hadfield et al. [Had+19], a class of variational quantum algorithms to find approximate solutions to constrained combinatorial optimization problems. An essential ingredient of a QAO instance is the problem-dependent mixer Hamiltonians. We refine the very definition of the QAO-mixing property based on a rigorous derivation from the quantum adiabatic algorithm [Far+00] to the quantum approximate optimization algorithm [FGG14] to the QAO. Thereby, we prove the convergence of all three classes of algorithms under comparatively few assumptions. Furthermore, we discuss the constraint graph model originating from the works of Leighton [Lei77] as an approach to studying feasible solutions to constrained problems. We also derive the connections between abstract groups acting on classical bit strings and their quantum mechanical operator analogs. For scheduling-type problems, we investigate bit string position permutations and characterize their mixing property in terms of transitive group actions on feasible vertex subsets of the constraint graph. We apply our new apparatus to free and flexible job-shop scheduling problems and find that the free case always admits suitable mixers. For the flexible case, we study examples that provide mixers and those that do not.

Contents

CHAPTER 1.

Introduction

The field of quantum computing arguably emerged in the eighties mainly due to Benioff's [Ben80] and Deutsch's [Deu85] pioneering work, bringing classical computer science and quantum mechanics together. In addition to purely theoretical questions, application attempts appeared early on, not least stimulated by Feynman's [Fey82] proposal to simulate quantum systems by other quantum systems because classical computers seemed and still seem to have pivotal difficulties with this task. His conjecture that quantum computers can, in principle, simulate any local quantum system was proven shortly after by Lloyd [Llo96]. These considerations paved the way for the application area of simulation of quantum systems; this is still one of the primary uses of quantum computers. However, the search for possible applications of quantum computers was not limited to this area. Indeed, Shor [Sho94] constructively showed, i.e. he wrote an algorithm, that quantum computers could efficiently solve the discrete logarithm problem and the problem of finding the prime factors of any integer. In contrast, until now, no classical algorithm is known to solve these problems efficiently. Grover [Gro96] made another groundbreaking discovery with his quantum algorithm for unstructured search, allowing a gentle speed-up of dozens of search-based procedures. Since then, countless quantum algorithms with numerous applications have been developed (cf. [Mon16] for a selection of particularly influential algorithms).

While the superiority of (theoretical) quantum computers over classical computers in the simulation of quantum systems is considered virtually proven, the extent to which quantum computers also have the upper hand for purely classical problems is more often the subject of current research. One particular exciting class of such problems is *combinatorial optimization problems* (COPs):

$$\text{optimize a Boolean function } C : \{0,1\}^N \to \mathbb{R} \text{ w.r.t. some constraints.}$$

These problems are abundant in industrial issues, so they are often of the utmost interest. At the same time, their treatment with classical computers is challenging.

Many representatives such as knapsack, traveling salesperson, or job shop scheduling are, in fact, NP-complete when considered as decision problems [Kar72]. Accordingly, the potential for algorithmic improvement is enormous here, and the quantum computational approach is considered promising. A massive catalog of classical analysis is already available for these problems, and the task now is to incorporate it into quantum algorithm design. This catalog includes, among others, the *constraint graph model*, which allows for a graph theoretical treatment of constraint optimization problems.

A limiting factor concerning quantum computing is still the physical realization of quantum computers. For example, Shor's algorithm requires millions of noiseless qubits (or even more noisy qubits using quantum error correction [LB13]) with a long coherence time to solve relevant problem instances [GE21]; IBM's latest quantum processor, however, only has 127 noisy qubits with moderate coherence time [Dia22], which is nonetheless an impressive achievement. Preskill [Pre18] coined the term noisy intermediate-scale quantum (NISQ) devices to describe these near-term quantum computers which do not incorporate quantum error correction.
Not all quantum algorithms developed so far are suitable for NISQ devices, but the so-called *variational quantum algorithms* (VQAs) represent a promising class of algorithms with already possible applications. They trace optimization tasks back to parameter adaptation. In this process, the parameters are updated on classical computers, and the quantum computer calculates the quantities required for the classical optimization rule. Clever implementation can thus significantly reduce the required coherence time and the number of simultaneously required qubits, making them executable on NISQ devices.

The prospect of near-term application has made the research field of VQAs flourish in a short time. The probably best-known VQA, the *quantum approximate optimization algorithm* (QAOA) [FGG14], is also a pioneer of its kind; to this day, it provides the basis for many other VQAs. The QAOA essentially consists of the alternating application of two parameterized unitaries to an initial state, the problem-independent "mixer" U_{M} and the problem-dependent "phase separator" U_{P}. Its simplicity makes it very tangible, and there is a wide range of works, e.g., on its universality [Llo18; MBZ20], performance on current NISQ devices [Qia+18; Pag+20; Har+21], and importance in the near future [GM19]. Meanwhile, the *quantum alternating operator ansatz* (QAO) [Had+19] has also caught much interest on its own. The QAO can be viewed as an extension of the QAOA that naturally implements problem constraints into the mixer unitary and thus may be better suited for optimization problems with many constraints, such as job shop scheduling. Many heuristic approaches to QAO mixer design exist for various problems; however, we propose a more rigorous analysis to extract mixing properties from the classical information about the optimization problem, mainly utilizing group theory and the constraint graph model.

The second chapter is a collection of basic concepts. Section 2.1 introduces appropriate notation for the whole thesis. Section 2.2 examines the possibility of calculating expectation values on quantum computers. As this can be regarded as the primary purpose of the quantum part of VQAs, this topic is crucial when analyzing VQAs. The notion of COPs is then formalized in Section 2.3. We also discuss certain transformations of these problems into each other and sketch the general strategy for implementing them on quantum computers. The chapter is concluded by the formal introduction to group actions on bit strings in Section 2.4. First and foremost, we draw the explicit connection between operators acting on the qubit space and the permutation representation, well-known in group representation theory. This allows us to characterize transitive group actions in terms of invariant coordinate subspaces. In addition, we embed the notion of feasibility preservation into this connection.

The third chapter discusses the motivation behind the QAOA and the QAO. In Section 3.1, we introduce the *quantum adiabatic algorithm* (QAA) [Far+00], a continuous-time quantum algorithm that utilizes properties of the adiabatic evolution of quantum systems. The algorithm aims to evolve an extremal eigenstate of an initial Hamiltonian H_{I} into an extremal eigenstate of a target Hamiltonian C which encodes optimal solutions to a COP. By invoking a more general adiabatic theorem as in [Far+00], we prove convergence of the QAA with arbitrary admissible target Hamiltonian C (Theorem 3.6). We then discuss the QAOA in Section 3.2 and highlight its connection to the QAA. Due to our results in Section 3.1, we can prove the convergence of the QAOA more generally and rigorously than in [FGG14] (Theorem 3.8). Finally, we motivate and introduce the QAO in Section 3.3. Our definitions occasionally differ somewhat from those in [Had+19]. Of particular importance is the accurate definition of the mixer property (Definition 3.12). This fine-tuning eventually allows the formulation of a convergence proof (Theorem 3.17) and a better translation into preceding and subsequent classical analyses.

In the fourth chapter, we introduce the constraint graph model as an approach to translating the issue of constructing suitable mixers for the QAO into the group- and graph-theoretical language. Section 4.1 contains the construction of the constraint graph for a generic COP. Subsequently, the basic notions are abstracted, allowing an unperturbed view of the mathematical benefits of this very construction. We discuss several immediate results and provide counterexamples whenever some properties are not generally valid. The connection to the QAO mixer construction is then made rigorous in Section 4.2. This eventually results in an equivalent characterization of possible mixers expressed in the properties of the constraint graph (Theorem 4.7). Our result immediately finds application in Section 4.3, where we study the constraint graphs of particular job-shop scheduling instances: free and flexible job-shop scheduling. This allows us to construct suitable mixers for some of the problem instances.

In addition to the QAOA and the QAO, we discuss other VQAs in the fifth chapter and mathematically derive the underlying principles that guarantee convergence. Firstly, Section 5.1 overviews the *variational quantum eigensolver* (VQE) [Per+14], another pioneering VQA. Like the QAOA, the VQE also lays the foundation for many more specialized algorithms. Section 5.2 then discusses the *variational quantum simulation of imaginary time evolution* (Var-QITE) [McA+19]. It is stricter in its specifications than the VQE and is directly introduced with a suitable variational method. Finally, the *filtering variational quantum eigensolver* (F-VQE) [Ama+21b] is introduced in Section 5.3. It is based upon the ideas of the VQE and the Var-QITE, and we highlight these connections in our discussion.

In the last chapter, we draw a conclusion and present an outlook on further extensions and investigations. We address open problems regarding the QAA and the QAOA and also comment on advanced questions regarding the constraint graph model.

The appendix consists of two parts. Appendix A entails a detailed introduction to graph theory based on the book by Diestel [Die17]. The content is trimmed to what is necessary for the analysis of the constraint graph model and the definition of completely non-diagonal matrices. In Appendix B, we collect the rather technical convergence proofs of Chapter 3.

Basic Concepts

2.1. Notation

We begin this chapter by fixing the required notations. The set of natural numbers $\mathbb{N}$ is defined without zero. Furthermore, we set $\mathbb{N}_0 = \{0\} \dot{\cup} \mathbb{N}$. For $N \in \mathbb{N}$, we abbreviate $[N] \coloneqq \{1, \dots, N\}$.

If not specified further, Ξ denotes a d-dimensional complex Hilbert space, $d \in \mathbb{N}$. Thus $\Xi \cong \mathbb{C}^d$ and we can choose an orthonormal basis (ONB) of Ξ, which we denote by $\mathcal{B}_\Xi \coloneqq \{|j\rangle \,:\, j = 0, \dots, d-1\}$. Linear operators acting on Ξ may be identified with complex $d \times d$-matrices by choosing an ONB $\mathcal{B}_\Xi$. This yields isomorphisms

- linear operators: $\mathcal{L}(\Xi) \cong \mathrm{Mat}(d, \mathbb{C})$.
- linear invertible operators: $\mathcal{L}is(\Xi) \cong \mathrm{GL}(d, \mathbb{C})$.
- unitary operators: $\mathcal{U}(\Xi) \cong \mathrm{U}(d, \mathbb{C})$.

For a matrix $A \in \mathrm{Mat}(d, \mathbb{C})$, the set of all invariant coordinate subspaces is

$$\mathfrak{I}(A) \coloneqq \{X = \mathrm{span}\{\boldsymbol{e}_i \,:\, i \in I\} \,:\, I \subseteq [d],\, A(X) \subseteq X\}. \tag{2.1.1}$$

These will be of particular interest throughout the thesis.

A single-qubit system is denoted by $\mathfrak{q} \cong \mathbb{C}^2$ and an N-qubit system accordingly by

$$\mathfrak{q}^{\otimes N} = \bigotimes_{n=1}^{N} \mathfrak{q} \cong \bigotimes_{n=1}^{N} \mathbb{C}^2 \cong \mathbb{C}^{2^N}, \quad N \in \mathbb{N}. \tag{2.1.2}$$

We abbreviate the set of all N bit strings by $Z(N) \coloneqq \{0,1\}^N$. The *computational basis* (CB) of $\mathfrak{q}^{\otimes N}$ is given as

$$\{|\boldsymbol{z}\rangle \coloneqq |z_1 z_2 \dots z_N\rangle \,:\, z_n \in \{0,1\}\} = \{|\boldsymbol{z}\rangle \,:\, \boldsymbol{z} \in Z(N)\}. \tag{2.1.3}$$

Moreover, we may also number the CB states consecutively with the prior introduced notation, i.e. writing the CB as $\{|z\rangle : z = 0, \dots, 2^N - 1\}$. Therefore we have, in the virtue of (2.1.2), identified

$$|z_1 z_2 \dots z_N\rangle \cong \bigotimes_{n=1}^{N} |z_n\rangle \cong |z\rangle\,, \quad \text{where } z = \sum_{n=1}^{N} z_n 2^{N-n}. \tag{2.1.4}$$

The Pauli matrices

$$\sigma^0 \coloneqq \mathbb{1} \mathrel{\widehat{=}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \sigma^1 \coloneqq \sigma^x \mathrel{\widehat{=}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$
$$\sigma^2 \coloneqq \sigma^y \mathrel{\widehat{=}} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \sigma^3 \coloneqq \sigma^z \mathrel{\widehat{=}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

together form a basis of $\mathrm{Mat}(2, \mathbb{C}) \cong \mathcal{L}(\mathfrak{q})$. Since $\mathcal{L}(\mathfrak{q}^{\otimes N}) \cong \bigotimes_{n=1}^{N} \mathcal{L}(\mathfrak{q})$, a canonical basis of $\mathcal{L}(\mathfrak{q}^{\otimes N})$ is given by all possible tensor products of single-qubit Pauli matrices. That is, every operator $A \in \mathcal{L}(\mathfrak{q}^{\otimes N})$ possesses a unique decomposition of the following form:

$$A = \sum_{\kappa_1=0}^{3} \dots \sum_{\kappa_N=0}^{3} \eta_{\kappa_1,\dots,\kappa_N} \bigotimes_{n=1}^{N} \sigma^{\kappa_n}, \quad \text{where } \eta_{\kappa_1,\dots,\kappa_N} \in \mathbb{C}\,. \tag{2.1.5}$$

It may be useful in time to arrange (2.1.5) by terms corresponding to different M-qubit operators, $M \leq N$. To avoid an overwhelming amount of tensor product symbols, we set

$$B_n \coloneqq \mathbb{1} \otimes \dots \otimes \mathbb{1} \otimes \underset{\substack{\uparrow \\ n\text{-th position}}}{B} \otimes \mathbb{1} \otimes \dots \otimes \mathbb{1} : \mathfrak{q}^{\otimes N} \to \mathfrak{q}^{\otimes N},$$

where $B : \mathfrak{q} \to \mathfrak{q}$ is a single-qubit operator. Then we may write (2.1.5) as

$$A = A_0^0 \mathbb{1} + \sum_{n=1}^{N} \sum_{\kappa=1}^{3} A_\kappa^n \, \sigma_n^\kappa + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \sum_{\kappa_1=1}^{3} \sum_{\kappa_2=1}^{3} A_{\kappa_1\kappa_2}^{n_1 n_2} \, \sigma_{n_1}^{\kappa_1} \sigma_{n_1}^{\kappa_2} + \dots =: \sum_{t=1}^{T} \check{A}_t \Sigma^t \tag{2.1.6}$$

with $A_{\boldsymbol{\kappa}}^{\boldsymbol{n}} \in \mathbb{C}$, alternating in the superscript, and κ now running through 1,2, and 3 (or equivalently x, y, and z).

The CB of $\mathfrak{q}^{\otimes N}$ is of such relevance that we will always consider elements of $\mathcal{L}(\mathfrak{q}^{\otimes N})$ as matrices w.r.t. the CB. Fixing the basis representation allows us to define matrix properties such as positivity or complete non-diagonality directly for the corresponding linear operators.

2.2. Expectation Value Calculation

The following scheme describes the possibility of computing expectation values of general normal operators/matrices on quantum computers, including Hermitian and unitary operators as a special case. It is fundamentally based on

Theorem 2.1 (Spectral theorem)**.** *Let $A \in \mathcal{L}(\Xi)$ be normal. Then A is diagonizable, i.e. there exists an ONB of Ξ consisting of eigenvectors of A.*

Interpreting a given normal operator A as matrix w.r.t. a given ONB $\mathcal{B}_\Xi$, the spectral theorem guarantees the existence of both a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ and a unitary matrix U such that

$$A = U^* \Lambda \, U \tag{2.2.1}$$

holds. The identity (2.2.1) is called *spectral decomposition*.

Consider a normal operator $A \in \mathcal{L}(\mathbb{q}^{\otimes N})$ with known spectral decomposition and a state $|\psi\rangle \in \mathbb{q}^{\otimes N}$. In addition, assume that the implementation of U as well as the preparation of $|\psi\rangle$ are known. Since

$$\langle\psi|A|\psi\rangle = \langle\psi|U^*\Lambda U|\psi\rangle = \sum_{j,k=1}^{2^N} \langle\psi|U^*|j\rangle \langle j|\Lambda|k\rangle \langle k|U|\psi\rangle = \sum_{j=1}^{2^N} \lambda_j P_j$$

with $P_j \coloneqq |\langle j|U|\psi\rangle|^2$ holds, the expectation value $\langle\psi|A|\psi\rangle$ can be approximately calculated via a repeated measurement of the state $U|\psi\rangle$ in the CB.

A more refined calculation method suggested by [Eke+02] is available for unitary operators. Consider $U \in \mathcal{U}(\mathbb{q}^{\otimes N})$ and a state $|\psi\rangle \in \mathbb{q}^{\otimes N}$ with known implementation. In order to evaluate $\langle\psi|U|\psi\rangle$ one introduces an ancilla qubit, initially set to $|0\rangle$, and implements the quantum circuit shown in Fig. 2.1. Its application yields

$$\begin{aligned}
|0\rangle|\psi\rangle &\mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|\psi\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + e^{i\varphi}|1\rangle)|\psi\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + e^{i\varphi}|1\rangle U|\psi\rangle) \\
&\mapsto \frac{1}{2}\Big((|0\rangle + |1\rangle)|\psi\rangle + e^{i\varphi}(|0\rangle - |1\rangle)U|\psi\rangle\Big) \\
&= \frac{1}{2}\Big(|0\rangle(|\psi\rangle + e^{i\varphi}U|\psi\rangle) + |1\rangle(|\psi\rangle - e^{i\varphi}U|\psi\rangle)\Big).
\end{aligned}$$

Measuring the ancilla qubit for varying relative phase shift φ in the $|0\rangle$-direction yields an interference pattern as in Fig. 2.2. From this pattern, one can now read off the

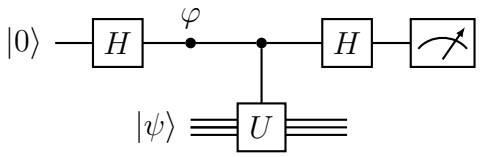

Figure 2.1.: Circuit encoding the expectation value into an ancilla qubit (compare [Eke+02, FIG. 1]). The ancilla qubit is initially set to $|0\rangle$. After applying a Hadamard gate and a variable phase gate, a controlled-U operation on the state $|\psi\rangle$ is performed. Another Hadamard gate is then applied to the ancilla qubit and after then it is measured. The quantum circuit was drawn using the quantikz package [Kay20].

visibility v as well as the phase shift α. As it is shown in [Sjö+00], this yields the desired expectation value via

$$\langle\psi|U|\psi\rangle = ve^{i\alpha}. \tag{2.2.2}$$

Here, the phase shift α is the phase difference between $U|\psi\rangle$ and $|\psi\rangle$, also known as the Pancharatnam phase (cf. [Pan56]). Provided that the decomposition (2.1.6) of a given operator $A \in \mathcal{L}(\mathbb{q}^{\otimes N})$ is known, this approach will also be used to calculate the expectation value $\langle\psi|A|\psi\rangle$.

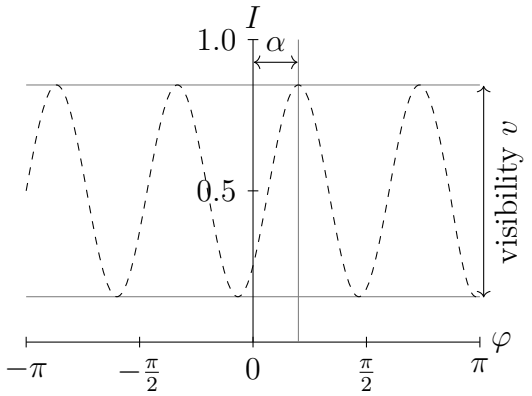

Figure 2.2.: Interference pattern from measurement (compare [Eke+02, FIG. 1]). The ancilla qubit is repeatedly measured in the $|0\rangle$-direction for varying relative phase shift φ. The resulting normalized intensity I is plotted on the vertical axis. The distance between the I-axis and the nearest argmax gives the phase shift α. The difference between maximum and minimum intensity yields the visibility v.

2.3. Combinatorial Optimization Problems

We give here a general definition of combinatorial optimization problems already tailored to a (quantum) computational treatment.

Definition 2.2. A *combinatorial optimization problem* is a quintuple

$$\mathrm{COP} \coloneqq \Big(N,\, \{c_a\}_{a=1}^{A},\, \{C_a\}_{a=1}^{A},\, \{D_b\}_{b=1}^{B},\, \mathrm{ext}\Big) \tag{2.3.1}$$

with $N \in \mathbb{N}$, $c_a \in \mathbb{R}_+$, $C_a : Z(N) \to \{0,1\}$, $D_b : Z(N) \to \{0,1\}$, respectively, and $\mathrm{ext} \in \{\min, \max\}$. It is called *unconstrained*, if $B = 0$, i.e. $\{D_b\}_{b=1}^{B} = \emptyset$.

The natural number N of bits necessary to formulate the problem gives the *problem size*. A bit string is usually denoted $\boldsymbol{z} \coloneqq z_1 z_2 \dots z_N$. The Boolean functions C_a are the *clauses* which are *satisfied* (by a given bit string $\boldsymbol{z}$) if $C_a(\boldsymbol{z}) = 1$ and *unsatisfied* otherwise. The real numbers c_a are the respective *costs* or *priorities* of a clause C_a. The nomenclature of c_a depends on whether the COP is considered as a minimization problem, i.e. ext = min, or maximization problem, i.e. ext = max. The clauses are often summarized in a single *objective function*

$$C \coloneqq \sum_{a=1}^{A} c_a C_a. \tag{2.3.2}$$

Lastly, the Boolean functions D_b are the *constraints*. Analogoulsy to the clauses, they are *satisfied* if $D_b(\boldsymbol{z}) = 1$, and *unsatisfied* otherwise.

Definition 2.3. A bit string $\boldsymbol{z}^* \in Z(N)$ is a *feasible solution* to a combinatorial optimization problem $\mathrm{COP} = (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \mathrm{ext})$ if $D_b(\boldsymbol{z}^*) = 1$ holds for every $b = 1, \dots, B$. The set of feasible solutions to COP is denoted by COPsol. If $\mathrm{COPsol} = \emptyset$, COP is *infeasible*.
A feasible solution $\boldsymbol{z}^*$ is further an *optimal solution*, if it fulfills

$$\boldsymbol{z}^* = \operatorname*{ext\,arg}_{\boldsymbol{z} \in \mathrm{COPsol}} \sum_{a=1}^{A} c_a C_a(\boldsymbol{z}) = \operatorname*{ext\,arg}_{\boldsymbol{z} \in \mathrm{COPsol}} C(\boldsymbol{z}).$$

The set of optimal solutions to COP is denoted by COPopt.

In applications, it is often the case that constrained COPs are transformed into unconstrained ones by converting the constraints D_b into clauses with sufficiently high penalties/priorities $p_b \in \mathbb{R}_+$. The procedure goes as follows:

$$\mathrm{COP} \mapsto \mathrm{uCOP}(\boldsymbol{p}) \coloneqq \Big(N,\, \{\tilde{c}_a\}_{a=1}^{\tilde{A}},\, \{\tilde{C}_a\}_{a=1}^{\tilde{A}},\, \emptyset,\, \mathrm{ext}\Big), \tag{2.3.3}$$

where $\boldsymbol{p} \coloneqq (p_1, \ldots, p_B)$, $\tilde{A} \coloneqq A + B$, and

$$(\tilde{c}_a,\, \tilde{C}_a) \coloneqq \begin{cases} (c_a,\, C_a), & \text{if } 1 \leq a \leq A \\ (p_{a-A},\, D_{a-A}), & \text{if } A < a \leq A + B \text{ and } \mathrm{ext} = \max \\ (p_{a-A},\, 1 - D_{a-A}), & \text{if } A < a \leq A + B \text{ and } \mathrm{ext} = \min\,. \end{cases}$$

For the unconstrained problem, one has $\mathrm{uCOPsol}(\boldsymbol{p}) = Z(N)$. Thus both problems are generally not equivalent. But for sufficiently high p_b, the optimal solution to $\mathrm{uCOP}(\boldsymbol{p})$ may also lie within COPsol, assuming $\mathrm{COPsol} \neq \emptyset$. We will refer to the transformation (2.3.3) also as *softcoding the constraints*.

Minimization and maximization problems can be further transformed into each other by the following procedure:

$$\mathrm{COP} \mapsto \overline{\mathrm{COP}} \coloneqq \Big(N,\, \{c_a\}_{a=1}^{A},\, \{1 - C_a\}_{a=1}^{A},\, \{D_b\}_{b=1}^{B},\, \overline{\mathrm{ext}}\Big), \tag{2.3.4}$$

where $\overline{\min} = \max$ and $\overline{\max} = \min$. One readily verifies that COP and $\overline{\mathrm{COP}}$ are equivalent, that is $\mathrm{COPsol} = \overline{\mathrm{COP}}\mathrm{sol}$ and $\boldsymbol{z}^*$ is an optimal solution to COP if and only if $\boldsymbol{z}^*$ is an optimal solution to $\overline{\mathrm{COP}}$.

A general approach addressing COPs with quantum computers is considering the objective function C of a given $\mathrm{COP} = (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \mathrm{ext})$ as a Hamiltonian, acting on $\mathbb{q}^{\otimes N}$ and diagonal in the corresponding CB. That is, the CB states are identified with the classical bit strings and the objective Hamiltonian is defined via

$$C\,|\boldsymbol{z}\rangle \coloneqq C(\boldsymbol{z})\,|\boldsymbol{z}\rangle\,, \quad \boldsymbol{z} \in Z(N). \tag{2.3.5}$$

Due to the implementation (2.3.5), the problem of finding an optimal solution to COP is now equivalent to the problem of finding an extremal eigenstate of

$$C|_S, \quad S \coloneqq \mathrm{span}\{|\boldsymbol{z}\rangle \,:\, \boldsymbol{z} \in \mathrm{COPsol}\} \subseteq \mathbb{q}^{\otimes N}. \tag{2.3.6}$$

S is the so-called *solution space* or *feasible subspace* of COP. The eigenspace of $C|_S$ corresponding to the extremal eigenvalue is therefore given by

$$S_{\mathrm{opt}} \coloneqq \{|\boldsymbol{z}\rangle \,:\, \boldsymbol{z} \in \mathrm{COPopt}\} \subseteq S \tag{2.3.7}$$

and is called the *optimal solution space* of COP. Note that the decomposition (2.1.6) takes the form

$$H = H_0^0 \mathbb{1} + \sum_{n=1}^{N} H_3^n \sigma_n^z + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} H_{33}^{n_1 n_2} \sigma_{n_1}^z \sigma_{n_2}^z + \ldots \tag{2.3.8}$$

with $H_{\boldsymbol{3}}^{\boldsymbol{n}} \in \mathbb{R}$ for a Hamiltonian H diagonal in the CB such as (2.3.5).

2.4. Group Actions on Bit Strings

In the following, we embed basic notions of group representation theory into the translation between operations on the classical bit strings $Z(N)$ and corresponding linear operators on $\mathfrak{q}^{\otimes N}$. Firstly, we cover all relevant abstract definitions and concepts, mainly following [Sag01]. Throughout this thesis, all groups are assumed to be finite.

Definition 2.4. Let G be a group and let X be a set. A group homomorphism $G \to \mathrm{Sym}(X)$ is called a *left group action* of G on X. G is said to *act on X from the left* via the left group action $\phi : G \to \mathrm{Sym}(X)$.

We usually denote actions multiplicatively. That is, if G acts on X from the left via ϕ, we write $g \cdot x \coloneqq \phi(g)(x)$ for $x \in X$ and $g \in G$. In the following, we will refer to left group actions simply as actions.

There are countless notions concerning group action; we select those critical for our analysis. Thereby, let G denote an arbitrary group acting on a non-empty set X.

Definition 2.5. The *orbit* of an element $x \in X$ is

$$G \cdot x \coloneqq \{g \cdot x \,:\, g \in G\}.$$

The group action is *transitive* if there exists an $x \in X$ so that $X = G \cdot x$.

If a group action is transitive, it readily follows that $X = G \cdot x$ holds for all $x \in X$. Therefore, transitivity is equivalent to the statement that for all $x, y \in X$, there exists $g \in G$ so that $g \cdot x = y$.

Definition 2.6. For a subset $Y \subseteq X$, let

$$G \cdot Y \coloneqq \{g \cdot y \,:\, g \in G,\ y \in Y\} = \bigcup_{y \in Y} G \cdot y.$$

Y is *G-invariant* if $G \cdot Y = Y$. The G-invariant subsets $\emptyset$ and X are called *trivial*.

If Y is a non-empty G-invariant subset of X, restricting the action of G on X to Y defines again a group action.

We wish to embed the concept of actions into the theory of group representations. We first state the necessary definitions.

Definition 2.7. Let V be an $\mathbb{C}$-vector space and let G be a group. Then V is a *G-module* if there is a group homomorphism

$$\phi : G \to \mathcal{L}is(V).$$

The map ϕ is called a *representation* of G on V over the field $\mathbb{C}$, and $\deg \phi \coloneqq \dim_{\mathbb{C}} V$ is the *degree* of ϕ.

Definition 2.8. Let V be a G-module with corresponding representation ϕ. A *submodule* of V is a subspace W that is closed under $\phi(G)$. The submodules $\{0\}$ and V are called *trivial*.

Definition 2.9. A non-zero G-module V is *reducible* if it contains a non-trivial submodule, and *irreducible* otherwise.

For a set $X = \{x_1, \ldots, x_n\}$, we define its *formal $\mathbb{C}$-span* $\langle X \rangle_{\mathbb{C}}$ consisting of all the formal linear combinations $\alpha_1 x_1 + \ldots + \alpha_n x_n$, where $\alpha_i \in \mathbb{C}$ for all $i \in [n]$. We can turn $\langle X \rangle_{\mathbb{C}}$ into a $\mathbb{C}$-vector space by defining appropriate vector addition and scalar multiplication:

$$\begin{aligned}(\alpha_1 x_1 + \ldots + \alpha_n x_n) + (\beta_1 x_1 + \ldots + \beta_n x_n) &\coloneqq (\alpha_1 + \beta_1)x_1 + \ldots + (\alpha_n + \beta_n)x_n,\\ \alpha(\alpha_1 x_1 + \ldots + \alpha_n x_n) &\coloneqq (\alpha\alpha_1)x_1 + \ldots + (\alpha\alpha_n)x_n.\end{aligned}$$

Now suppose a group G acts on the set X. We can extend the action to an action on $\langle X \rangle_{\mathbb{C}}$ by linearity:

$$g \cdot (\alpha_1 x_1 + \ldots + \alpha_n x_n) \coloneqq \alpha_1 (g \cdot x_1) + \ldots + \alpha_n (g \cdot x_n).$$

Therefore, $\langle X \rangle_{\mathbb{C}}$ becomes a G-module of dimension $|X|$. The corresponding group homomorphism $\rho : G \to \mathcal{L}is(\langle X \rangle_{\mathbb{C}})$, $g \mapsto [v \mapsto g \cdot v]$ is called the *permutation representation* of G associated with X. By choosing a basis $\mathcal{B}$ of $\langle X \rangle_{\mathbb{C}}$, we further obtain a group isomorphism $\varphi_{\mathcal{B}} : \mathcal{L}is(\langle X \rangle_{\mathbb{C}}) \to \mathrm{GL}(|X|, \mathbb{C})$ and thus a group homomorphism $\rho_{\mathcal{B}} \coloneqq \varphi_{\mathcal{B}} \circ \rho : G \to \mathrm{GL}(|X|, \mathbb{C})$. The *standard basis* of $\langle X \rangle_{\mathbb{C}}$ is given by X.

Corollary 2.10. *Let G act on a non-empty set X and let $\rho : G \to \mathcal{L}is(\langle X \rangle_{\mathbb{C}})$ denote its permutation representation. Then the following hold:*

(i) $Y \subseteq X$ is G-invariant $\Leftrightarrow$ $\langle Y \rangle_{\mathbb{C}}$ is a submodule of $\langle X \rangle_{\mathbb{C}}$ $\Leftrightarrow$ $\langle Y \rangle_{\mathbb{C}} \in \bigcap_{g \in G} \mathfrak{I}(\rho_X(g))$.

(ii) G acts transitively on X $\Leftrightarrow$ $\bigcap_{g \in G} \mathfrak{I}(\rho_X(g)) = \{\{0\}, \langle X \rangle_{\mathbb{C}}\}$.

In fact, the invariant subsets of a group G are determined by the generators of G.

Proposition 2.11. *Let S, $0 < |S| < \infty$, be a generating set of a group G, i.e. for every $g \in G$, there exist powers $\{r_s\}_{s\in S} \subset \mathbb{N}_0$ so that $g = \prod_{s\in S} s^{r_s}$. Furthermore, let G act on a non-empty set X and let $\rho : G \to \mathcal{L}is(\langle X\rangle_{\mathbb{C}})$ denote its permutation representation. Then the following hold:*

(i) $Y \subseteq X$ is G-invariant $\Leftrightarrow \langle Y\rangle_{\mathbb{C}} \in \bigcap_{s\in S} \mathfrak{I}(\rho_X(s))$.

(ii) G acts transitively on $X \Leftrightarrow \bigcap_{s\in S} \mathfrak{I}(\rho_X(s)) = \{\{0\}, \langle X\rangle_{\mathbb{C}}\}$.

Proof. It suffices to show that

$$\bigcap_{g\in G} \mathfrak{I}(\rho_X(g)) = \bigcap_{s\in S} \mathfrak{I}(\rho_X(s)). \tag{2.4.1}$$

Since $S \subseteq G$, the inclusion "$\subseteq$" is trivial. Conversely, let $W \in \bigcap_{s\in S} \mathfrak{I}(\rho_X(s))$ and let $g \in G$ be arbitrary. As $\rho_X : G \to \mathrm{GL}(|X|, \mathbb{C})$ is a group homomorphism, it follows that

$$\begin{aligned}\rho_X(g)(W) &= \rho_X\Big(\prod_{s\in S} s^{r_s}\Big)(W) = \prod_{s\in S} \rho_X(s)^{r_s}(W)\\ &= \rho_X(s_1)^{r_1}\Big(\rho_X(s_2)^{r_2}\Big(\dots\Big(\rho_X(s_{|S|})^{r_{|S|}}(W)\Big)\Big)\Big) \subseteq W\end{aligned}$$

That is, also the inclusion "$\supseteq$" holds. □

Now consider $\mathrm{COP} = (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \{D_b\}_{b=1}^B, \mathrm{ext})$ and a group action

$$\phi : G \to \mathrm{Sym}(Z(N)), \quad \pi \mapsto [\boldsymbol{z} \mapsto \pi \cdot \boldsymbol{z}].$$

Definition 2.12. A subgroup F is called COP-*feasibility-preserving* if COPsol is an F-invariant subset of $Z(N)$. In this case, F also acts on COPsol.

We extend the group action to the formal $\mathbb{C}$-span of COPsol and $Z(N)$, respectively. However, $\langle\mathrm{COPsol}\rangle_{\mathbb{C}}$ and $\langle Z(N)\rangle_{\mathbb{C}}$ just correspond to S and $\mathbb{q}^{\otimes N}$, where the respective standard bases are $\{|\boldsymbol{z}\rangle : \boldsymbol{z} \in \mathrm{COPsol}\}$ and the full CB. This eventually yields the permutation representation $\rho : F \to \mathcal{L}is(\mathbb{q}^{\otimes N})$. Since F is COP-feasibility-preserving, S is a submodule of $\mathbb{q}^{\otimes N}$ and composing ρ with the restriction $A \mapsto A|_S$ yields the permutation representation $\rho_S : F \to \mathcal{L}is(S)$. Recall that we always consider elements of $\mathcal{L}is(\mathbb{q}^{\otimes N}) \subset \mathcal{L}(\mathbb{q}^{\otimes N})$ as matrices w.r.t. the CB. If we continue to do so for all coordinate subspaces, we eventually obtain canonical group homomorphisms, the *permutation matrix representations*

$$\mathcal{P} : F \to \mathrm{U}(2^N, \mathbb{C}) \cong \mathcal{U}\big(\mathbb{q}^{\otimes N}\big), \tag{2.4.2}$$

$$\mathcal{P}_S : F \to \mathrm{U}(|\mathrm{COPsol}|, \mathbb{C}) \cong \mathcal{U}(S). \tag{2.4.3}$$

From Adiabatic Evolution to Alternating Operators

The quantum adiabatic algorithm (QAA) proposed by Farhi et al. [Far+00] is a continuous-time quantum algorithm for solving unconstrained COPs. It is based on adiabatic evolution of extremal eigenstates of a time-dependent Hamiltonian. We review the mathematical background in Section 3.1 and provide a convergence proof in Theorem 3.6 more general as discussed in [Far+00] and [FGG14]. Most importantly, we highlight the key features of the quantities involved that guarantee convergence.

We then introduce the quantum approximate optimization algorithm (QAOA) proposed by Farhi et al. [FGG14] in Section 3.2. It is, in contrast, a variational quantum algorithm (compare Chapter 5). We again provide a convergence proof in Theorem 3.8 that is heavily based on our result for the convergence of the QAA; that is, we draw a rigorous derivation of the QAOA from the QAA. We also discuss issues of the QAOA with (hardcoded) constrained problems.

Section 3.3 concludes this chapter with the introduction of the quantum alternating operator ansatz (QAO) proposed by Hadfield et al. [Had+19]. We motivate its introduction by discussing several case studies of the QAOA performance for a variety of constrained optimization problems. The QAO is more a family of variational quantum algorithms than a single one; it emphasizes the properties of the building blocks of the QAOA and generalizes them so that a broader class of problems is treatable. A detailed description is also given in [Had18]. Due to this "conservation of concepts", we can even provide a convergence proof in Theorem 3.17 for this more abstract setting.

3.1. Quantum Adiabatic Algorithm

The QAA incoporates the idea of quantum adiabatic evolution, that is, of arbitrarily slowly evolving a given quantum system. Adiabatic theorems describe how system properties are preserved during the process. Thereby, Farhi et al. [Far+00] referred to a particular version of adiabatic theorems: those with a gap condition. This kind of theorem is, e.g., well-formulated in [Sim17, Theorem 17.2].

Theorem 3.1 (Adiabatic Theorem)**.** *Let* $\{H(t) : 0 \leq t \leq 1\} \subseteq \mathcal{L}(\Xi)$ *be a family of self-adjoint operators such that* $H(\,\cdot\,) \in C^2([0,1], \mathcal{L}(\Xi))$*. For* $T > 0$*, let* $\tilde{U}_T$ *be the solution of*

$$\frac{\mathrm{d}}{\mathrm{d}s}\tilde{U}_T(s) = -iH(s/T)\tilde{U}_T(s), \quad 0 \leq s \leq T; \quad \tilde{U}_T(0) = \mathbb{1}$$

and set $U_T(t) \coloneqq \tilde{U}_T(tT)$*,* $0 \leq t \leq 1$*. Furthermore, let* $\lambda(t)$ *be an eigenvalue of* $H(t)$*, respectively, such that*

$$\alpha \coloneqq \inf_{0 \leq t \leq 1} \operatorname{dist}(\lambda(t), \sigma(H(t)) \setminus \{\lambda(t)\}) > 0 \tag{3.1.1}$$

holds. Let $\mathbb{P}(t)$ *denote the spectral projection for* $\lambda(t)$*, respectively. Then*

$$\lim_{T \to \infty} (\mathbb{1} - \mathbb{P}(t)) U_T(t) \mathbb{P}(0) = 0 \tag{3.1.2}$$

uniformly in t *in* $[0,1]$*.*

The condition (3.1.1) ensures that the spectral gap stays finite; no level-crossing is allowed. Thus, the adiabatic theorem states that evolving sufficiently slowly with respect to the time-varying Hamiltonian $H(s)$ preserves the eigenspaces as long as they do not mutually intersect and the Hamiltonian itself is sufficiently smoothly (C^2) varied with time. We remark that in the infinite-dimensional case, the assignment $s \mapsto \lambda(s)$ is required to be C^2 additionally. However, since we only deal with finite-dimensional Hilbert spaces, this directly follows from the requirement that $H(\,\cdot\,)$ is C^2. In the following, we will always denote with U_T the time evolution defined in Theorem 3.1 and call it *quasi-adiabatic evolution* for finite $T > 0$.

Now consider an unconstrained COP $= (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \emptyset, \mathrm{ext})$. W.l.o.g. we assume that ext $=$ max. Therefore, we seek to find a highest energy state of the objective Hamiltonian (2.3.5). The idea is then to consider a second Hamiltonian $H_\mathrm{I} \in \mathcal{L}(\mathbb{q}^{\otimes N})$ with at least one known and easy to construct highest energy state $|\iota\rangle$. From H_I and C, a time-dependent Hamiltonian $H(s)$ is constructed via linear interpolation:

$$H_{\mathrm{lin}(H_\mathrm{I}, C)}(t) \coloneqq (1-t)H_\mathrm{I} + tC, \quad 0 \leq t \leq 1. \tag{3.1.3}$$

Then $H_{\mathrm{lin}(H_\mathrm{I},C)}(\,\cdot\,)$ fulfills the C^2 criterion. In fact, $H_{\mathrm{lin}(H_\mathrm{I},C)}(\,\cdot\,)$ is even analytic. If the curve of instantaneous largest eigenvalues of (3.1.3) also fulfills (3.1.1), starting in $|\iota\rangle$ and evolving with the unitary $U_T(1)$ for sufficiently large $T > 0$ yields a state close to a highest energy state of C. That is, the QAA consists of the following steps:

1. Prepare the initial state $|\iota\rangle$ on the quantum computer.

2. Specify an evolution time $T > 0$ and apply the quasi-adiabatic evolution U_T w.r.t. (3.1.3).

3. Repeatedly measure the outcome $U_T(1)\,|\iota\rangle$ in the CB to obtain a distribution of optimal solution approximations.

However, without any additional information about the minimum spectral gap α, one cannot quantify how large T has to be taken to approximate a highest energy state of C within a certain error. See, e.g., [Dua20, Theorem 2] for such quantitative estimates. Furthermore, a direct consequence of Theorem 3.1 is that the geometric multiplicity of $\lambda(s)$ remains invariant throughout the adiabatic evolution. This, in turn, means that to ensure an adiabatic transition of the highest energy states according to Theorem 3.1, one necessary condition is that H_I must be chosen so that the geometric multiplicity of its largest eigenvalue is equal to that of the largest eigenvalue of C. Since the latter is generally unknown, constructing a suitable H_I is also not readily possible. Nevertheless, the operator

$$B \coloneqq \sum_{n=1}^{N} \sigma_n^x \tag{3.1.4}$$

has prevailed as a popular candidate for the initial Hamiltonian H_I. Its lowest and highest eigenvalue are $\pm N$ and non-degenerate, respectively. The uniform superposition

$$|+\rangle \coloneqq |+\rangle_N \coloneqq \bigotimes_{n=1}^{N} |+\rangle_1 \coloneqq \bigotimes_{n=1}^{N} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^N}} \sum_{\boldsymbol{z}\in Z(N)} |\boldsymbol{z}\rangle \tag{3.1.5}$$

is its easy to construct highest energy state while

$$|-\rangle \coloneqq |-\rangle_N \coloneqq \bigotimes_{n=1}^{N} |-\rangle_1 \coloneqq \bigotimes_{n=1}^{N} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2^N}} \sum_{\boldsymbol{z}\in Z(N)} (-1)^{\sharp \boldsymbol{z}} |\boldsymbol{z}\rangle \tag{3.1.6}$$

is its also easy to construct ground state. Here, $\sharp\boldsymbol{z}$ denotes the number of ones within the bitstring $\boldsymbol{z}$ modulo 2.

For COPs with exactly one optimal solution, Theorem 3.1 is sufficient to prove that the quasi-adiabatic evolution U_T w.r.t.

$$H_{\mathrm{lin}(B,C)}(t) = (1-t)B + tC, \quad 0 \leq t \leq 1 \tag{3.1.7}$$

yields the optimal solution in the adiabatic limit $T \to \infty$ if the initial state is taken to be $|+\rangle$. The proof is sketched in [FGG14]. However, the same statement also holds true if COP has more than one optimal solution. In this case, the spectral gap necessarily approaches 0 for $t \to 1$. Thus, we need a version of the adiabatic theorem without a gap condition. Precisely this topic was treated by Avron and Elgart [AE99]. However, we state and use the more refined version of their theorem given by Teufel [Teu01].

Theorem 3.2 (Adiabatic Theorem v2)**.** *Let $\{H(t) : 0 \leq t \leq 1\} \subseteq \mathcal{L}(\Xi)$ be a family of self-adjoint operators such that $H(\,\cdot\,) \in C^2([0,1], \mathcal{L}(\Xi))$. Let $\lambda(t)$ be an eigenvalue of $H(t)$, respectively, with corresponding spectral projection $\mathbb{P}(t)$. Furthermore, let $P(\,\cdot\,) \in C^2([0,1], \mathcal{L}(\Xi))$ such that for every $0 \leq t \leq 1$, $P(t)$ is a projection with $H(t)P(t) = \lambda(t)P(t)$. In addition, $P(t) = \mathbb{P}(t)$ should hold for almost all $t \in [0,1]$. Then*

$$\lim_{T\to\infty} (\mathbb{1} - P(t))U_T(t)P(0) = 0 \tag{3.1.8}$$

uniformly in t in $[0,1]$.

Thus, Theorem 3.2 allows to treat the case of level crossings, as long as the spectral projections can be continued in a C^2 way through the crossings. As we will use in Theorem 3.6, this is indeed the case for the adiabatic evolution w.r.t. the linear interpolation (3.1.3), since this assignment is analytic in t. However, to remain within the eigenspace of the largest eigenvalue, we must ensure that the level crossing supports this. To argue that this is the case for (3.1.7), we recall the notion of positive, non-negative, and completely non-diagonal matrices and already introduce appropriate notation.

Definition 3.3. A matrix $A = (a_{ij}) \in \mathrm{Mat}(d, \mathbb{C})$ is…

(i) …*positive* if $a_{ij} > 0$ holds for all $1 \leq i, j \leq d$.

(ii) …*non-negative* if $a_{ij} \geq 0$ holds for all $1 \leq i, j \leq d$.

Definition 3.4. $A \in \mathrm{Mat}(d, \mathbb{C})$ is *completely non-diagonal* if one of the following equivalent properties hold:

(i) $\mathfrak{I}(A) = \{\{0\}, \mathbb{C}^d\}$.

(ii) The directed network G_A with adjacency matrix A is strongly connected (cf. Definition A.3 and Definition A.19).

For a proof of equivalency of these two properties, consult, e.g., [HJ12]. These matrices are usually called *irreducible*. However, as this term is already used in group representation theory and has a different meaning, we decided not to use it here. The second definition immediately gives rise to the following result as the connectivity of a graph/network is determined by the off-diagonal elements of its adjacency matrix.

Corollary 3.5. *Let $A \in \mathrm{Mat}(d, \mathbb{C})$ be completely non-diagonal and $B \in \mathrm{Mat}(d, \mathbb{C})$ be diagonal. Then $A + B$ is completely non-diagonal.*

With these preliminaries, we can prove that the QAA with initial Hamiltonian (3.1.4) converges for an arbitrary objective Hamiltonian C.

Theorem 3.6 (Convergence of QAA)**.** *Let* COP $= (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \emptyset, \max)$ *with objective Hamiltonian C. Furthermore, let S_{opt} denote the subspace spanned by optimal solution states, i.e. the eigenspace of the largest eigenvalue of C. Then it holds that*

$$\lim_{T\to\infty} U_T(1)\,|+\rangle \in S_{\mathrm{opt}}, \tag{3.1.9}$$

where U_T is the quasi-adiabatic evolution w.r.t. (3.1.7)*.*

The proof of Theorem 3.6 is located at Appendix B. It can be boiled down to three essential parts:

(i) $H(\,\cdot\,) \in C^2\Big([0,1], \mathcal{L}(\mathbb{q}^{\otimes N})\Big)$ such that $H(t)$ is symmetric for all $0 \le t \le 1$,

(ii) there is an eigenvalue curve $\lambda : [0,1] \to \mathbb{R}$ such that $\lambda(0) = \lambda_{\max}(0)$ and $\lambda(1) = \lambda_{\max}(1)$, and

(iii) the corresponding spectral projection curve possesses a C^2-continuation on $[0,1]$.

In our proof, the crucial property (ii) is ensured by the fact that B is non-negative and completely non-diagonal. While the former is not really a restriction, complete non-diagonality is not that easy to achieve. However, conserving complete non-diagonality for more complicated (i.e., constrained) COPs will be of utmost importance. Indeed, the following proposition shows that complete non-diagonality is a necessary property of the initial Hamiltonian if one wants to ensure convergence in the general case. The proof is again located at Appendix B.

Proposition 3.7. *Let H_I have a proper invariant coordinate subspace $\mathcal{R}$ and let $|\iota\rangle \in \mathfrak{q}^{\otimes N} \setminus \{0\}$ be arbitrary. Then there exists a* COP $= (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \emptyset, \max)$ *with objective Hamiltonian C and optimal solution subspace S_{opt} such that*

$$\lim_{T\to\infty} U_T(1)\,|\iota\rangle \notin S_{\text{opt}}, \tag{3.1.10}$$

where U_T is the quasi-adiabatic evolution w.r.t. (3.1.3).

Furthermore, we used analyticity of the linear interpolation for (i) as well as for (iii). However, we want to emphasize that (i) does not, in general, imply (iii). Consider the following counterexample by Rellich [Rel37] (also see [Kat95, Example 5.3]):

$$H(0) = 0, \quad H(t) = e^{-1/t^2} \begin{bmatrix} \cos\left(\frac{2}{t}\right) & \sin\left(\frac{2}{t}\right) \\ \sin\left(\frac{2}{t}\right) & -\cos\left(\frac{2}{t}\right) \end{bmatrix}, \text{ for } t \in \mathbb{R} \setminus \{0\}.$$

Then $[t \mapsto H(t)] \in C^\infty\left(\mathbb{R}, \mathbb{C}^{2\times 2}\right)$ and the eigenvalues are given by

$$\lambda_\pm(t) = \begin{cases} 0, & t = 0, \\ \pm e^{-1/t^2}, & t \neq 0. \end{cases}$$

That is, even $[t \mapsto \lambda_\pm(t)] \in C^\infty(\mathbb{R}, \mathbb{R})$. However, for $t \neq 0$, the spectral projections are given by

$$\mathbb{P}_+(t) = \begin{bmatrix} \cos^2\left(\frac{1}{t}\right) & \cos\left(\frac{1}{t}\right)\sin\left(\frac{1}{t}\right) \\ \cos\left(\frac{1}{t}\right)\sin\left(\frac{1}{t}\right) & \sin^2\left(\frac{1}{t}\right) \end{bmatrix} \quad \text{and}$$
$$\mathbb{P}_-(t) = \begin{bmatrix} \sin^2\left(\frac{1}{t}\right) & -\cos\left(\frac{1}{t}\right)\sin\left(\frac{1}{t}\right) \\ -\cos\left(\frac{1}{t}\right)\sin\left(\frac{1}{t}\right) & \cos^2\left(\frac{1}{t}\right) \end{bmatrix},$$

both of which have not even continuous continuations through $t = 0$.

3.2. Quantum Approximate Optimization Algorithm

The QAOA is a variational quantum algorithm based on the QAA. In contrast to the other variational quantum algorithms introduced in Chapter 5, it primarily describes the parametrization of the unitary gates but is not very restrictive in other aspects. Assuming an integer-valued objective function C, the parameters may be restricted

to be angles $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q) \in [0, \pi)^q$ and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_q) \in [0, 2\pi)^q$ with $q \in \mathbb{N}$. For general real-valued objective functions, $\boldsymbol{\gamma}$ has to be taken from whole $\mathbb{R}^q$. The corresponding trial states $|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle$ are built using specific unitaries

$$U_{\mathrm{M}}(\beta) \coloneqq e^{-i\beta B} = \prod_{n=1}^{N} e^{-i\beta\sigma_n^x} \quad \text{and} \tag{3.2.1}$$

$$U_{\mathrm{P}}(\gamma) \coloneqq e^{-i\gamma C} = \prod_{a=1}^{A} e^{-i\gamma c_a C_a}. \tag{3.2.2}$$

Note that the indices M for "mixing" and P for "phase separation" already anticipate the ideas of Hadfield et al. [Had18] for the generalization of the QAOA.
The uniform superposition $|+\rangle$ is chosen as the initial state. Further trial states are constructed via

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle \coloneqq V(\boldsymbol{\beta}, \boldsymbol{\gamma}) |+\rangle \coloneqq \left(\prod_{o=1}^{q} U_{\mathrm{M}}(\beta_o) U_{\mathrm{P}}(\gamma_o) \right) |+\rangle . \tag{3.2.3}$$

In addition, set

$$F_q(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \boldsymbol{\beta}, \boldsymbol{\gamma} | C | \boldsymbol{\beta}, \boldsymbol{\gamma} \rangle \tag{3.2.4}$$

and

$$M_q = \max_{\boldsymbol{\beta}, \boldsymbol{\gamma}} F_q(\boldsymbol{\beta}, \boldsymbol{\gamma}). \tag{3.2.5}$$

For a given $q \in \mathbb{N}$ and initial parameter guesses $\boldsymbol{\beta}_0$, $\boldsymbol{\gamma}_0$, the QAOA iteratively performs the following steps, starting with $i = 0$:

1. Prepare the initial state $|+\rangle$ on the quantum computer, apply QAOA-gates to obtain $|\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i\rangle$, and evaluate $F_q(\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i)$ via repeated measurement.

2. Adjust the parameters $\boldsymbol{\beta}_i \mapsto \boldsymbol{\beta}_{i+1}$, $\boldsymbol{\gamma}_i \mapsto \boldsymbol{\gamma}_{i+1}$ using a classical optimization routine.

3. Repeat 1 and 2 with the adjusted parameters until some termination condition is satisfied (maximum iterations, vanishing parameter updates, etc.).

4. Repeatedly measure the final outcome $|\boldsymbol{\beta}_{i_{\mathrm{end}}}, \boldsymbol{\gamma}_{i_{\mathrm{end}}}\rangle$ in the CB to obtain a distribution of optimal solution approximations.

From the adiabatic perspective, the unitaries U_{M} and U_{P} approximate the adiabatic time evolution of $|+\rangle$ w.r.t. (3.1.7), where larger q may increase the approximation quality. Thus, the QAOA can be viewed as a parametrized version of the QAA with fixed initial Hamiltonian B.

Theorem 3.8 (Convergence of QAOA)**.** *Let* COP $= (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \emptyset, \max)$ *with corresponding objective Hamiltonian* C *and optimal solution space* S_{opt}*. For all* $\varepsilon > 0$*, one can choose finitely many angles* $\boldsymbol{\beta}$ *and* $\boldsymbol{\gamma}$ *such that*

$$\operatorname{dist}\Big(|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle, S_{\text{opt}}\Big) < \varepsilon. \tag{3.2.6}$$

The proof can be found in Appendix B. As an immediate consequence of Theorem 3.6, one obtains the following [FGG14, (10)]:

Corollary 3.9. *Let* COP $= (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \emptyset, \max)$ *with objective function* C*. Then*

$$\lim_{q\to\infty} M_q = \max_{\boldsymbol{z}\in Z(N)} C(\boldsymbol{z}). \tag{3.2.7}$$

The proof of Theorem 3.8 is constructive in the sense that for any $\varepsilon > 0$, it specifies angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ so that (3.2.6) holds. However, these do not have to be the angles chosen in any way optimal.

This concludes the rigorous derivation of the QAOA from the QAA. We emphasize that, until now, we merely treated unconstrained COPs. However, constrained problems may also be considered due to the softcoding transformation (2.3.3), but not as hardcoded instances. In order to argue why the QAOA generally fails to solve hard constrained COPs, consider COP $= (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \text{ext})$ with non-trivial solution space $\{0\} \neq S \subsetneq \mathbb{q}^{\otimes N}$ and optimal solution space S_{opt}. Applying the QAOA to this problem merely approximates an extremal eigenstate of the objective Hamiltonian C but does not respect the feasibility condition: in contrast to the unconstrained case, the eigenspace corresponding to the extremal eigenvalue of C need not be S_{opt}. In fact, their intersection might be trivial. Since C and thus the phase separator $U_{\text{P}}(\gamma)$, $\gamma \in [0, 2\pi)$, is diagonal in the CB, it leaves the coordinate subspace S invariant. Therefore, the problem occurs in the mixing process. Indeed the mixer $U_{\text{M}}(\beta)$ has no non-trivial invariant coordinate subspace for $\beta \neq 0$, especially not S. That means that the complete non-diagonality property of B, which, on the one hand, guarantees convergence in the unconstrained case, is, on the other hand, responsible for the failure of the QAOA in the constrained case.

3.3. Quantum Alternating Operator Ansatz

The QAO abstracts the basic design of the QAOA to general design specifications for variational quantum algorithms. This step finally allows us also to treat constrained

COPs with hardcoded constraints. Several case studies indicate the necessity of introducing hard constraints (or some other techniques) because softcoding often either leads to suboptimal optimization landscapes (if the penalties are too high) or issues with feasibility (if the penalties are too low). Firstly, we mention an extensive performance discussion of different classical optimizers for the QAOA applied to soft- and hardcoded portfolio optimization by Baker and Radha [BR22]. While their work focuses on the quantitative behavior of different classical optimization rules, one can also read off the trend that many of the classical optimizers considered perform better in hardcoded instances. We especially refer to Figure S2-S5. Meanwhile, van Dam et al. [Dam+21] as well as de la Grand'rive and Hullo [GH19] analyzed the performance of the QAOA with soft constraints applied to certain Knapsack instances. They also propose penalty-free approaches, which outperform the naive softcoding of constraints, indicating again that softcoding the constraints is generally not very powerful. However, we shall also remark that the situation is not always so clear. For example, Radzihovsky et al. [RMS19] could not find any significant advantage of hardcoding over softcoding the constraints in their analysis of the QAOA performance on the 4-city traveling salesperson problem. Nevertheless, we see the prior studies as reason enough to examine the implementation of hardcoded constraints closely.

The QAO encompasses the problem of finding suitable initial states, mixing operators, and phase separation operators, especially for COPs with non-trivial solution space. It contains the QAOA as the particular case when $S = \mathbb{q}^{\otimes N}$. We mainly follow [Had18, Section 6.2] for the introductory part. However, we directly define our phase separators and mixers to be (products of) unitary groups, and we use different nomenclature. We start with the easy concept of phase seperators and proceed with an extensive discussion of the mixers. The concept of an initial state then is relatively immediate.

In the following, let COP $= (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \text{ext})$ be arbitrary with solution space S and optimal solution space S_{opt}.

Definition 3.10. A non-negative, diagonal $H \in \mathcal{L}(\mathbb{q}^{\otimes N})$ is a COP-*phase separator Hamiltonian* if the extremal eigenspace of $H|_S$ is S_{opt}. Then

$$U_{\text{P}}(H, \cdot\,) : \mathbb{R} \to \mathcal{U}\big(\mathbb{q}^{\otimes N}\big), \quad \gamma \mapsto e^{-i\gamma H} \tag{3.3.1}$$

is the corresponding (*parametrized*) *phase separator*.

The canonical choice for a COP-phase separator Hamiltonian is the objective Hamiltonian C. However, there might be decent approximations of C which are simpler to implement and still preserve the optimal solution space. Our definition precisely matches the definition for phase separation unitaries given in [Had18, (6.6)]. Again, if the objective function C is assumed to be integer-valued, the parameter space may be restricted to $[0, 2\pi)$.

Definition 3.11. $T \in \mathcal{L}(\mathbb{q}^{\otimes N})$ is COP-*feasibility-preserving* if $T(S) \subseteq S$.

Recall that $\mathfrak{I}(T)$ denotes the set of all invariant coordinate subspaces of $T \in \mathcal{L}(\mathbb{q}^{\otimes N})$.

Definition 3.12. A family of operators $\{A_i\}_{i\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$, $0 < |I| < \infty$, is COP-*mixing* if it is COP-feasibility-preserving and fulfills

$$X \in \bigcap_{i\in I} \mathfrak{I}(A_i) \;\Rightarrow\; X_1 = \{0\} \text{ or } X_1 = S, \tag{3.3.2}$$

whereby $X = X_1 \oplus X_2 \subseteq S \oplus S^{\perp}$.

The essence of Definition 3.12 is that the part X_1 of a common invariant coordinate subspace X lying in the solution space S should be trivial, that is, no proper subspace, while the part X_2 lying in $S^{\perp}$ could be arbitrary. The condition of being coordinate subspaces automatically prohibits any mixing between $X_1 \subseteq S$ and $X_2 \subseteq S^{\perp}$. Since every element of a COP-mixing family is COP-feasibility-preserving, its restriction to S defines an element in $\mathcal{L}(S)$. This yields an equivalent characterization.

Corollary 3.13. *A* COP-*feasibility-preserving family* $\{A_i\}_{i\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$ *is* COP-*mixing if and only if*

$$\bigcap_{i\in I} \mathfrak{I}\big(A_i|_S\big) = \{\{0\}, S\}. \tag{3.3.3}$$

We eventually obtain the following property of COP-mixing families of Hamiltonians.

Proposition 3.14. *Let* $\{H_i\}_{i\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$, $0 < |I| < \infty$, *be a family of non-negative Hamiltonians. The following two statements are equivalent:*

(i) $\{H_i\}_{i\in I}$ *is* COP-*mixing.*

(ii) $(\sum_{i\in I} H_i)|_S \in \mathcal{L}(S)$ *is completely non-diagonal.*

Proof. Let G_i and G_I denote the directed networks with adjacency matrix $H_i|_S$, $i \in I$, and $H_I|_S \coloneqq (\sum_{i\in I} H_i)|_S$, respectively. Since all matrices are Hermitian, w.l.o.g. consider G_i, $i \in I$, and G_I simply as graphs. In addition, all matrices are non-negative which implies that

$$\langle \boldsymbol{z}|H_{i_0}|\boldsymbol{z}'\rangle \neq 0 \;\Rightarrow \sum_{i\in I} \langle \boldsymbol{z}|H_i|\boldsymbol{z}'\rangle \neq 0 \tag{3.3.4}$$

holds for all $i_0 \in I$ and $\boldsymbol{z}, \boldsymbol{z}' \in \mathrm{COPsol}$. Now (3.3.4) means that G_I contains an edge e whenever there is an $i \in I$ so that G_i contains e, i.e.

$$G_I = \bigcup_{i \in I} G_i.$$

Naming the vertices $V(G_i) = V(G_I) = \mathrm{COPsol}$, $i \in I$, establishes a bijection between the coordinate subspaces of S and the vertex subsets of V by identifying each coordinate subspace with its unique basis of COPsol-elements. For $A \in \{H_i|_S\}_{i \in I} \cup \{H_I|_S\}$, it follows that

$$\langle \boldsymbol{z}_k \,:\, k \in K \rangle_{\mathbb{C}} \in \mathfrak{I}(A) \;\Leftrightarrow\; \mathrm{nbhd}_{G_A}\Big(\{\boldsymbol{z}_k \,:\, k \in K\}\Big) = \emptyset.$$

Therefore, $\{H_i\}_{i \in I}$ is COP-mixing if and only if any subset of vertices has at least one neighbor in one of the graphs G_i. This, in turn, is equivalent to G_I having no non-trivial neighborless vertex subset which is an equivalent characterization of connectivity (cf. Proposition A.23). □

Given a COP-mixing family of Hamiltonians, we can enforce non-negativity by taking the absolute value of each element. Therefore, we will w.l.o.g. assume that each COP-mixing family of Hamiltonians has said property.

Definition 3.15. For a COP-mixing family of Hamiltonians $\mathcal{H} = \{H_i\}_{i \in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$, the corresponding (*parametrized*) *simultaneous mixer* is defined as

$$U_{\mathrm{M},0}(\mathcal{H}, \cdot) : \mathbb{R} \to \mathcal{U}\Big(\mathbb{q}^{\otimes N}\Big), \quad \beta \mapsto e^{-i\beta \sum_{i \in I} H_i}. \tag{3.3.5}$$

Specifying a permutation $\sigma \in \mathrm{Sym}(I)$, the corresponding (*parametrized*) *sequential mixer* is given by

$$U_{\mathrm{M},\sigma}(\mathcal{H}, \cdot) : \mathbb{R} \to \mathcal{U}\Big(\mathbb{q}^{\otimes N}\Big), \quad \beta \mapsto \prod_{i \in I} e^{-i\beta H_{\sigma(i)}}. \tag{3.3.6}$$

Note that we have defined both simultaneous and sequential mixers in the same manner as Hadfield et al. By the demands placed on COP-mixing families and Proposition 3.14, the mixers meet the two criteria established by them:

- *preserve the feasible subspace*: for all $\beta \in \mathbb{R}$ and $\sigma \in \mathrm{Sym}(I) \cup \{0\}$, $U_{\mathrm{M},\sigma}(\mathcal{H}, \beta)$ is COP-feasibility preserving.

- *explore the feasible subspace*: for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathrm{COPsol}$, there exists $\beta \in \mathbb{R}$ so that $\langle \boldsymbol{z} | U_{\mathrm{M},0}(\mathcal{H}, \beta) | \boldsymbol{z}' \rangle \neq 0$.

The latter also holds true for any sequential mixer as one can derive from Theorem 3.17. As pointed out by Hadfield et al., one significant advantage of sequential mixers over simultaneous ones is their implementation costs; implementing single Hamiltonians H_i and thus unitaries $\exp(-i\beta H_i)$ is generally easier than implementing the whole $\mathcal{H}$-sum. In addition, if two Hamiltonians H_i and H_j act on disjoint qubits, they may be implemented in parallel. Thus, choosing a suitable permutation $\sigma \in \mathrm{Sym}(I)$ may result in a drastically decreased circuit depth for implementing $U_{\mathrm{M},\sigma}(\mathcal{H}, \cdot)$ compared to implementing $U_{\mathrm{M}}(\mathcal{H}, \cdot)$. In general, the parameter space can not be restricted any further since the mixing Hamiltonians may have very complicated spectra.

Definition 3.16. Let $\{H_i\}_{i\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$ be a COP-mixing family of Hamiltonians. Any highest energy state of $\sum_{i\in I} H_i|_S$ is a COP-*initial state*.

A QAO instance is defined by the choice of an initial state, a phase separator, a mixing family, and a mixing procedure (simultaneous or sequential). Thus, we may consider a given instance as a quadrupel

$$\left(|\iota\rangle\,,\, H_{\mathrm{P}},\, \{H_i\}_{i\in I},\, \sigma\right)$$

with

- initial state: $|\iota\rangle \in \mathbb{q}^{\otimes N}$,
- phase separator $H_{\mathrm{P}} \in \mathcal{L}(\mathbb{q}^{\otimes N})$,
- mixing family $\{H_i\}_{i\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$, and
- mixing procedure: $\sigma \in \mathrm{Sym}(I) \cup \{0\}$.

It is then called a COP-*QAO instance*, if initial state, phase separator, and the mixing family are appropriately chosen. For a given instance and quality $q \in \mathbb{N}$, we use the same notation for the trial states, i.e.

$$|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle \coloneqq V(\boldsymbol{\beta}, \boldsymbol{\gamma})\,|\iota\rangle \coloneqq \left(\prod_{o=1}^{q} U_{\mathrm{M},\sigma}(\beta_o)U_{\mathrm{P}}(\gamma_o)\right)|\iota\rangle\,. \tag{3.3.7}$$

The algorithmic procedure is then analogous to the QAOA, that is, for a given $q \in \mathbb{N}$ and initial parameter guesses $\boldsymbol{\beta}_0$ and $\boldsymbol{\gamma}_0$, the QAO iteratively performs the following steps, starting with $i = 0$:

1. Prepare the initial state $|\iota\rangle$ on the quantum computer, apply QAO-gates to obtain $|\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i\rangle$, and evaluate $F_q(\boldsymbol{\beta}_i, \boldsymbol{\gamma}_i)$ via repeated measurement.

2. Adjust the parameters $\boldsymbol{\beta}_i \mapsto \boldsymbol{\beta}_{i+1}$, $\boldsymbol{\gamma}_i \mapsto \boldsymbol{\gamma}_{i+1}$ using a classical optimization routine.

3. Repeat 1 and 2 with the adjusted parameters until some termination condition is satisfied (maximum iterations, vanishing parameter updates, etc.).

4. Repeatedly measure the final outcome $|\boldsymbol{\beta}_{i_{\text{end}}}, \boldsymbol{\gamma}_{i_{\text{end}}}\rangle$ in the CB to obtain a distribution of feasible optimal solution approximations.

Note that in the case $S = \mathbb{q}^{\otimes N}$, $|+\rangle$ is again a valid initial state. Furthermore, $\{\sigma_n^x\}_{n=1}^N$ is a mixing family, yielding the complete non-diagonality of B. Thus, the QAO really containts the QAOA as a special case. Indeed, we can prove the convergence of any appropriately designed QAO instance. For the proof, consult Appendix B.

Theorem 3.17 (Convergence of QAO)**.** *Let* $(|\iota\rangle, H_P, \{H_i\}_{i\in I}, \sigma)$ *be a* COP-*QAO instance for* $\text{COP} = (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \{D_b\}_{b=1}^B, \max)$ *with non-trivial solution space* S *and optimal solution space* $S_{\text{opt}} \subseteq S$*. For all* $\varepsilon > 0$*, one can choose finitely many angles* $\boldsymbol{\beta}$ *and* $\boldsymbol{\gamma}$ *such that*

$$\text{dist}\Big(|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle, S_{\text{opt}}\Big) < \varepsilon. \tag{3.3.8}$$

Furthermore, we wish to discuss the construction of parametrized mixers from a family of (unparametrized) unitaries $\{W_i\}_{i\in I} \subset \mathcal{U}(\mathbb{q}^{\otimes N})$. We can trace this back to the previous considerations utilizing matrix logarithms.

Definition 3.18. Let $A = \text{diag}(\lambda_1, \ldots, \lambda_d) \in \text{Mat}(d, \mathbb{C})$ with $\lambda_j \neq 0$ for all $j \in [d]$. A matrix $B = \text{diag}(\mu_1, \ldots, \mu_d) \in \text{Mat}(d, \mathbb{C})$ is a *logarithm* of A if there exists a branch L of the complex logarithm so that $\mu_j = L(\lambda_j)$ holds for all $j \in [d]$. In this case, write $B = L(A)$.
Let $A \in \text{GL}(d, \mathbb{C})$ be normal. By the spectral theorem, there exist $U \in \text{U}(d, \mathbb{C})$ and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d) \in \text{GL}(d, \mathbb{C})$ so that $A = U^{-1}\Lambda U$. A matrix $B \in \text{Mat}(d, \mathbb{C})$ is a *logarithm* of A if $B = U^* L(\Lambda) U$ for a branch L of the complex logarithm. In this case, write again $B = L(A)$.

One property of the matrix exponential is that $e^{V^{-1}AV} = V^{-1}e^A V$ holds for any $A \in \text{Mat}(d, \mathbb{C})$ and $V \in \text{GL}(d, \mathbb{C})$. Therefore, if B is a logarithm of A with U and Λ as in Definition 3.18, then

$$e^B = e^{U^{-1}L(\Lambda)U} = U^{-1}e^{L(\Lambda)}U = U^{-1}\Lambda U = A$$

because $e^{L(\lambda)} = \lambda$ holds for all branches L of the complex logarithm. An invertible normal matrix A always possesses a (non-unique) logarithm since one can always find

a ray $\Gamma_\alpha \coloneqq \{re^{i\alpha} : r \geq 0\} \subset \mathbb{C}$, $\alpha \in [0, 2\pi)$, so that no eigenvalue of A lies on Γ_α; this allows to define a branch of the complex logarithm on $\{\lambda_j : j \in [d]\} \subset \mathbb{C} \setminus \Gamma_\alpha$.

An important feature of the matrix exponential and the matrix logarithm is that both preserve invariant subspaces of invertible normal matrices.

Lemma 3.19. *Let $A \in \mathrm{GL}(d, \mathbb{C})$ be normal. Furthermore, let $L(A)$ be a logarithm of A. Then $W \subseteq \mathbb{C}^d$ is an A-invariant subspace if and only if W is an $L(A)$-invariant subspace.*

Proof. Let U and Λ be as in Definition 3.18. The partition of its diagonal entries of a diagonal matrix into mutually distinct ones completely determines its invariant subspaces. Therefore, Λ and $L(\Lambda)$ have the same invariant subspaces. Furthermore, it holds that

$$\begin{aligned} A(W) \subseteq W &\Leftrightarrow \Big(U^{-1}\Lambda U\Big)(W) \subseteq W \Leftrightarrow \Lambda\Big(U(W)\Big) \subseteq U(W) \\ &\Leftrightarrow L(\Lambda)\Big(U(W)\Big) \subseteq U(W) \Leftrightarrow \Big(U^{-1}L(\Lambda)U\Big)(W) \subseteq W \\ &\Leftrightarrow L(A)(W) \subseteq W. \end{aligned}$$

□

In particular, taking the logarithm preserves the invariant coordinate subspaces. For a given family of unitaries $\{W_j\}_{j\in I} \subset \mathcal{U}(\mathbb{q}^{\otimes N})$ and suitable branches L_j of the complex logarithm, we consider the family of Hamiltonians $\{iL_j(W_j)\}_{j\in I} \subset \mathcal{L}(\mathbb{q}^{\otimes N})$, which is, by Lemma 3.19, COP-mixing if and only if $\{W_j\}_{j\in I}$ is COP-mixing.

Constraint Graph Model

We introduce the constraint graph model to provide an alternative approach to constrained optimization problems. The core idea originates from [Lei77] and is also well-presented in [FQ85]. In addition, [RBW06, Chapter 5] provides a context for this model within more general approaches to constrained problems. The constraint graph model should help better understand the set of feasible solutions of COPs with certain symmetric constraints. It offers two major advantages: an alternative mathematical approach and a visualization basis.

We are mainly interested in COPs of scheduling type (cf. Definition 4.1) and discuss the concrete graph construction process for these problems in Section 4.1. Moreover, we abstract notions such as feasible solutions (Definition 4.2) and feasibility-preservation (Definition 4.3) to general graphs. Furthermore, we conceptually place graph automorphisms in the context of feasibility-preservation.

We investigate COPs of scheduling type further in Section 4.2 and examine concrete candidates for mixer Hamiltonians. Their mixing properties are translated into the just established machinery. More precisely, Theorem 4.7 gives an alternative characterization of the mixing property in terms of a transitive group action on vertex subsets of the associated constraint graph corresponding to feasible solutions.

In Section 4.3, we utilize this alternative characterization of mixer properties to verify that the so-called free job-shop scheduling problems (Definition 4.8) always admit suitable mixers (Corollary 4.10). In addition, we discuss the more complex flexible job-shop scheduling problems (Definition 4.11). We do not obtain a universal statement about mixer properties but rather show that several possibilities exist by considering two examples.

4.1. Construction and Properties

Consider COP $= (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \text{ext})$. Each of the N bits is identified with a vertex. That is, COP induces a set of vertices V with $|V| = N$. We label the vertices accordingly as $V = \{v_1, \ldots, v_N\}$. That is, v_n represents the n-th bit in the bit string $\boldsymbol{z}$. Now, any bit string can be represented graphically by coloring each vertex corresponding to a bit set to one. This *indexing mapping* is formally expressed as

$$\iota : [N] \to V, \quad n \mapsto v_n. \tag{4.1.1}$$

We can identify bit strings with subsets of vertices via the *identification mapping*

$$\kappa : Z(N) \to 2^V, \quad z_1 \ldots z_N \mapsto \{\iota(n) \,:\, z_n = 1\}. \tag{4.1.2}$$

Meanwhile, the constraints determine the set of edges E, a number $J \in \mathbb{N}$ of vertices to be colored, and a possible update of the set of vertices. The main idea is to connect two vertices with an edge whenever it can be ruled out that both associated bits can be set to one. The following list contains treatable constraints and their implementation. It is sorted in the order in which the constraints should be implemented. In the following, let $I \subseteq [N]$ be arbitrary.

- **none:** All bits z_n, $n \in I$, must take the value zero. The corresponding vertices are deleted from the graph, yielding a new set of vertices V' with $|V'| = N - |I|$.

- **at-most-one:** At most one of the bits z_n, $n \in I$, can be set to one. The result is that all associated vertices are connected to form a clique, i.e. for all $m, n \in I$ with $m \neq n$, it should apply that $v_m v_n \in E$.

- **one-hot:** Assume $|I| > 1$. Exactly one of the bits z_n, $n \in I$, must take the value one. The result is again that all associated vertices are connected to form a clique. In addition, the number J is increased by one.

- **all:** All bits z_n, $n \in I$, must take the value one. The corresponding vertices as well as all vertices connected to them are deleted from the graph. In addition, the number J is reduced by $|I|$.

- **all-equal:** All bits z_n, $n \in I$, must take the same value. The corresponding vertices in $V_I \coloneqq \{v_n \,:\, n \in I\}$ are joined together to form a new vertex v_I, yielding a new set of vertices V'. The vertex is also weigthed by the weight $|I|$, which means that coloring v_I counts as coloring $|I|$ vertices. The set of edges is also updated: if for any $v \in V \setminus V_I$, there is an $n \in I$ such that $v v_n \in E$ (in the original graph), then it should apply that $v v_I \in E'$.

Therefore, the number J specified by the constraints indicates how many bits of a feasible solution must take the value one. We then seek to color J vertices which are mutually unconnected. In the following, we are mostly interested in COPs of a specific type.

Definition 4.1. A COP $= (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \{D_b\}_{b=1}^B, \text{ext})$ is of *scheduling type* if

(i) there exists exactly one partition $\{I_1, \dots, I_J\}$ of $[N]$, $N_j \coloneqq |I_j| > 1$ for every $j \in [J]$, so that every I_j is subject to a one-hot constraint.

(ii) all other $B - J$ constraints are at-most-one constraints.

We wish to mathematically concretize the construction of the constraint graph for scheduling problems. Utilizing the just established mappings ι and κ, we can describe the translation from one-hots and at-most-ones, regarded as Boolean functions, to sets of edges of the constraint graph. We begin with a convenient notation for both constraint types. For two bit strings $\boldsymbol{z}, \boldsymbol{z}' \in Z(N)$, the *and* operation reads

$$\boldsymbol{z} \wedge \boldsymbol{z}' \coloneqq (z_1 \wedge z_1') \dots (z_N \wedge z_N') \in Z(N). \tag{4.1.3}$$

Furthermore, let

$$|\boldsymbol{z}| \coloneqq |\{n \,:\, z_n = 1\}| = |\kappa(\boldsymbol{z})| \tag{4.1.4}$$

denote the *Hamming weight* of $\boldsymbol{z} \in Z(N)$. For $I \subseteq [N]$, write

$$\boldsymbol{z}_I \coloneqq \kappa^{-1}\Big(\iota(I)\Big) = \kappa^{-1}(\{v_n\}_{n \in I}).$$

Then, the one-hot constraint and the at-most-one constraint associated with the index set I are

$$\zeta_I(\boldsymbol{z}) \coloneqq \begin{cases} 1, & \text{if } |\boldsymbol{z} \wedge \boldsymbol{z}_I| = 1 \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \eta_I(\boldsymbol{z}) \coloneqq \begin{cases} 1, & \text{if } |\boldsymbol{z} \wedge \boldsymbol{z}_I| \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.1.5}$$

Both constraints are incorporated as cliques (complete subgraphs) into the corresponding constraint graph: if $\zeta_I \in \{D_b\}_{b=1}^B$ or $\eta_I \in \{D_b\}_{b=1}^B$, then

$$E_I \coloneqq \{\iota(m)\iota(n) \,:\, m, n \in I, m \neq n\} = \{v_m v_n \,:\, m, n \in I, m \neq n\} \in E.$$

This formalism results in a concrete construction rule for the constraint graph of scheduling problems. Let COP $= (N, \{c_a\}_{a=1}^A, \{C_a\}_{a=1}^A, \{D_b\}_{b=1}^B, \text{ext})$ be of scheduling type and let $G_0 = (V, E_0) \coloneqq (\{v_n \,:\, n \in [N]\}, \emptyset)$ be the initial constraint graph. Furthermore, set $J_0 \coloneqq 0$ and $b \coloneqq 1$. Then iteratively perform the following steps:

1. If $D_b = \zeta_I$, set $(E_b, J_b) \coloneqq (E_{b-1} \cup E_I, J_{b-1} + 1)$. Otherwise, if $D_b = \eta_I$, set $(E_b, J_b) \coloneqq (E_{b-1} \cup E_I, J_{b-1})$.

2. If $b < B$, increase $b \mapsto b + 1$ and repeat step 1. Otherwise, end the construction.

In Fig. 4.1, we provide the constraint graph of a T-city traveling salesperson problem as an example.

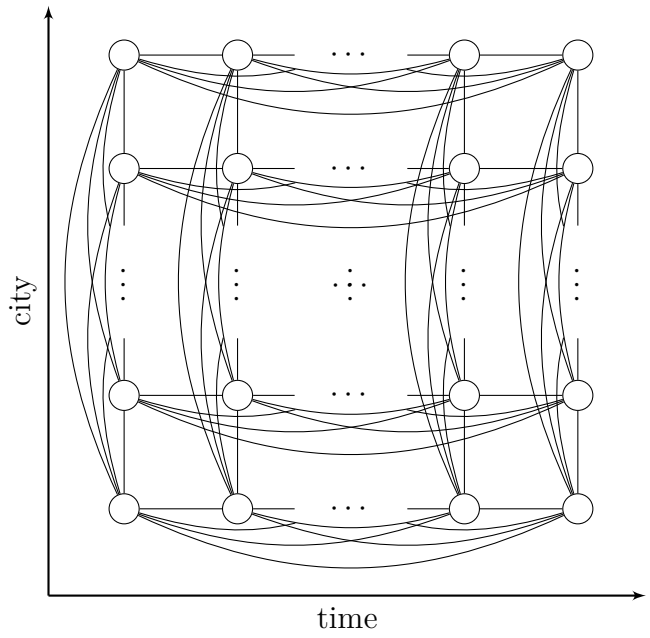

Figure 4.1.: Traveling salesperson constraint graph. Given $T \in \mathbb{N}$ cities, the set of vertices V has a total of $N = T^2$ elements representing the visit to a specific city at a specific time, respectively. It is more convenient to label the vertices with two indices specifying time and city. Then, the coloring of vertex $v_{i,k}$, $i, k \in [T]$, means that city k is visited at time i. The amount of vertices to be colored is T (visit each city exactly once). The latter constraint also determines the sets of edges $E = \bigcup_{i \in [T]} \{v_{i,k} v_{i,k'} : k, k' \in [T],\, k \neq k'\} \cup \bigcup_{k \in [T]} \{v_{i,k} v_{i',k} : i, i' \in [T],\, i \neq i'\}$.

We now introduce these lines of thought in more abstract terms. To treat the facts as general as possible, we consider an arbitrary graph $G = (V, E)$ and an arbitrary number $J \in \mathbb{N}$ with $J \leq N \coloneqq |V|$. For simplicity, we do not treat weighted vertices.

Definition 4.2. A subset of vertices $W \subseteq V$ is a *feasible solution* if

1. $|W| = J$ and

2. W is an independent set (cf. Definition A.8).

Let $\mathcal{S} \coloneqq \mathcal{S}_J$ denote the set of feasible solotions

Definition 4.3. A function $\rho : V \to V$ *preserves feasibility* if $\rho(W) \in \mathcal{S}$ holds for all $W \in \mathcal{S}$.

It is reasonable to demand surjectivity of ρ to reach every possible feasible solution. Since V is a finite set, such an ρ is automatically injective, too, hence a permutation. We obtain the following property.

Proposition 4.4. *The family $\mathcal{F}$ of all feasibility-preserving permutations is a subgroup of the symmetric group* $\mathrm{Sym}(V) \cong \mathrm{Sym}([N])$.

Proof. The composition of two feasibility-preserving permutations trivially preserves feasibility again. Clearly, the identity preserves feasibility, hence $\mathrm{id}_V \in \mathcal{F}$. Thus, $\mathcal{F}$ is a subset of the finite group $\mathrm{Sym}(V)$ containing the identity and closed under composition of functions, and hence a subgroup. □

Proposition 4.5. *Every graph automorphism preserves feasibility, i.e.* $\mathrm{Aut}(G) \subseteq \mathcal{F}$. *Furthermore, if for all non-adjacent $v, w \in V$, there exists $W \in \mathcal{S}$ so that $v, w \in W$, then* $\mathrm{Aut}(G) = \mathcal{F}$.

Proof. Let $\varphi \in \mathrm{Aut}(G)$ and let $W \in \mathcal{S}$ 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, it follows that $\varphi(v)\varphi(w) \notin E$; hence $\varphi(W)$ is again an independent set. This shows that $\mathrm{Aut}(G) \subseteq \mathcal{F}$. Now assume that for an arbitrary pair of non-adjacent vertices $v, w \in V$, there exists $W \in \mathcal{S}$ so that $v, w \in W$, and let $\rho \in \mathcal{F}$. Since $\rho : V \to V$ is bijective, Proposition A.12 implies that it suffices to show its homomorphism property:

$$\Big(vw \in E \;\Rightarrow \rho(v)\rho(w) \in E\Big) \;\Leftrightarrow\; \Big(\rho(v)\rho(w) \notin E \;\Rightarrow vw \notin E\Big).$$

Let $v, w \in V$ so that $\rho(v)\rho(w) \notin E$. Then by assumption, there exists $W \in \mathcal{S}$ with $\rho(v), \rho(w) \in W$. Since also ρ^{-1} is feasibility-preserving by Proposition 4.4, one obtains that $v, w \in \rho^{-1}(W) \in \mathcal{S}$. Therefore, also $vw \notin E$ holds. □

Note that the statement $\mathcal{F} \subseteq \mathrm{Aut}(G)$ is, in general, false. If, e.g., $J = 1$, we would always have that $\mathcal{F} = \mathrm{Sym}(V)$. Another non-trivial counterexample is shown in Fig. 4.2. Meanwhile, in the particular case of $J = 2$, Proposition 4.5 already implies that $\mathrm{Aut}(G) = \mathcal{F}$.

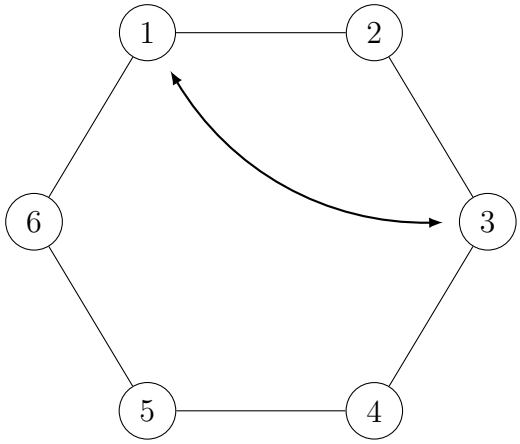

Figure 4.2.: Graph with feasibility-preserving non-automorphisms. Its automorphism group is isomorphic to D_{12}, the dihedral group of order 12. For $J = 3$, the only two feasible solutions are $W_1 = \{1, 3, 5\}$ and $W_2 = \{2, 4, 6\}$. Therefore, the transposition (13) is feasibility-preserving, but is not an element of $\mathrm{Aut}(G)$.

Every subgroup of $\mathrm{Sym}(V)$ canonically acts on 2^V via

$$\Psi : \mathrm{Sym}(V) \to \mathrm{Sym}\left(2^V\right), \quad \rho \mapsto [U \mapsto \rho \cdot U]$$

with

$$\rho \cdot U \coloneqq \rho(U). \tag{4.1.6}$$

As $\mathcal{F}$ consists of feasibility-preserving elements, the restricted action $\Psi : \mathcal{F} \to \mathrm{Sym}(\mathcal{S})$ is well-defined. We say that $\mathcal{F}$ *connects all feasible solutions* if it acts transitively on $\mathcal{S}$. We can define this more generally for families of feasibility-preserving functions.

Definition 4.6. A family of feasibility-preserving functions $\{\rho_i : V \to V\}_{i \in I}$ *connects all feasible solutions* if for all $W, W' \in \mathcal{S}$, there exists $i \in I$ such that $\rho_i(W) = W'$.

Note that, in general, $\mathcal{F}$ does not connect all feasible solotions as the minimal counterexample graph drawn in Fig. 4.3 shows.

4.2. Connection to the QAO

We continue to consider scheduling problems. An immediate consequence of the scheduling structure is that feasible solutions must have the same Hamming weight

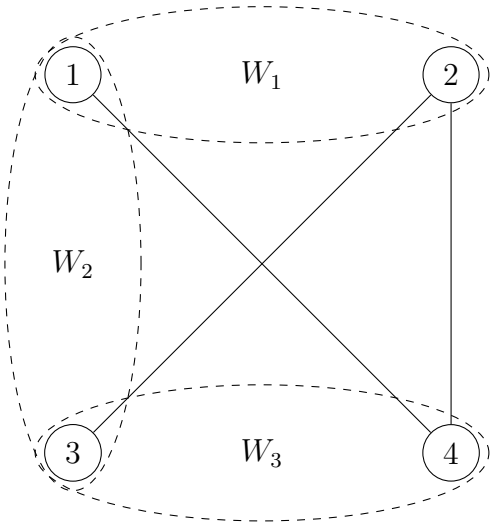

Figure 4.3.: Graph with unconnected feasbile solutions. For $J = 2$, the feasible solutions are $\mathcal{S} = \{W_1, W_2, W_3\}$ drawn in. The subgroup of feasibility-preserving permutations is given by $\mathcal{F} = \{\mathrm{id}, (13)(24)\}$ which does not connect all feasible solutions.

(4.1.4) J. Since leaving the Hamming weight invariant is, in a sense, the fundamental property of bit value permutations, it is advisable to study the latter in terms of feasibility preservation and mixing properties. Namely, we consider the group action

$$\Phi : \mathrm{Sym}([N]) \to \mathrm{Sym}(Z(N)), \quad \tau \mapsto [\boldsymbol{z} \mapsto \tau \cdot \boldsymbol{z}]$$

with

$$\tau \cdot z_1 \dots z_n \dots z_N \coloneqq z_{\tau(1)} \dots z_{\tau(n)} \dots z_{\tau(N)}. \tag{4.2.1}$$

By linear extension

$$\lambda : \mathrm{Sym}(Z(N)) \to \mathcal{U}\left(\mathbb{q}^{\otimes N}\right), \quad \lambda(\pi) : \sum_{\boldsymbol{z} \in Z(N)} \alpha_{\boldsymbol{z}} \left|\boldsymbol{z}\right\rangle \mapsto \sum_{\boldsymbol{z} \in Z(N)} \alpha_{\pi^{-1}(\boldsymbol{z})} \left|\boldsymbol{z}\right\rangle, \tag{4.2.2}$$

we obtain the permutation (matrix) representation $\mathcal{P} = \lambda \circ \Phi : \mathrm{Sym}([N]) \to \mathcal{U}(\mathbb{q}^{\otimes N})$. It is compatible with the tensor product structure of $\mathbb{q}^{\otimes N}$:

$$\mathcal{P}(\tau) \bigotimes_{n=1}^{N} \left|\psi_n\right\rangle = \bigotimes_{n=1}^{N} \left|\psi_{\tau(n)}\right\rangle. \tag{4.2.3}$$

Let $G = (V, E)$ be the constraint graph of a $\mathrm{COP} = (N, \{c_a\}_{a=1}^{A}, \{C_a\}_{a=1}^{A}, \{D_b\}_{b=1}^{B}, \mathrm{ext})$ of scheduling type. We specify an ordering $V = \{v_1, \dots, v_N\}$ of the vertices and thereby consider the following group action

$$\Pi : \mathrm{Sym}([N]) \to \mathrm{Sym}\left(2^V\right), \quad \tau \mapsto [U \mapsto \tau \cdot U]$$

with

$$\tau \cdot \{v_n\}_{n\in I} \coloneqq \{v_{\tau(n)}\}_{n\in I}. \tag{4.2.4}$$

We now provide a link between (4.2.1) and (4.2.4): recall that $Z(N)$ and 2^V are in bijection via the identification mapping (4.1.2). This induces another bijection

$$\tilde{\kappa} : \mathrm{Sym}(Z(N)) \to \mathrm{Sym}\Big(2^V\Big), \quad \pi \mapsto \kappa \circ \pi \circ \kappa^{-1}. \tag{4.2.5}$$

As in Section 4.1, we write $\boldsymbol{z}_I = \kappa^{-1}(\{v_n\}_{n\in I})$. That is, $\boldsymbol{z}_n = 1$ whenever $n \in I$. For every $I \subseteq [N]$ and every $\tau \in \mathrm{Sym}([N])$, it holds that

$$\Pi(\tau)\Big(\{v_n\}_{n\in I}\Big) = \{v_{\tau(n)}\}_{n\in I} = \kappa\Big(\boldsymbol{z}_{\tau(I)}\Big) = \Big(\kappa \circ \Phi(\tau)\Big)(\boldsymbol{z}_I) = \Big(\kappa \circ \Phi(\tau) \circ \kappa^{-1}\Big)\Big(\{v_n\}_{n\in I}\Big).$$

Therefore, we have shown that $\Pi = \tilde{\kappa} \circ \Phi$. Recall that $[N]$ and V are in bijection via the indexing mapping (4.1.1). This induces a bijection via conjugation

$$\tilde{\iota} : \mathrm{Sym}([N]) \to \mathrm{Sym}(V), \quad \tau \mapsto \iota \circ \tau \circ \iota^{-1}. \tag{4.2.6}$$

Let $\Psi : \mathrm{Sym}(V) \to \mathrm{Sym}(2^V)$ be as in (4.1.6). Then we obtain that for every $n \in [N]$, $I \subseteq [N]$, and $\tau \in \mathrm{Sym}([N])$,

$$\begin{aligned} &\Big(\iota \circ \tau \circ \iota^{-1}\Big)(v_n) = \Big(\iota \circ \tau\Big)(n) = \iota\Big(\tau(n)\Big) = v_{\tau(n)} \\ \Rightarrow\ &(\Psi \circ \tilde{\iota})(\tau)\Big(\{v_n\}_{n\in I}\Big) = \Psi(\iota \circ \tau \circ \iota^{-1})\Big(\{v_n\}_{n\in I}\Big) = \{v_{\tau(n)}\}_{n\in I} = \Pi(\tau)\Big(\{v_n\}_{n\in I}\Big). \end{aligned}$$

Thus, also $\Pi = \Psi \circ \tilde{\iota}$ holds. This allows us to relate the properties of the permutation representation of subgroups of $\mathrm{Sym}([N])$ to their actions on vertex subsets of the associated constraint graph, yielding a commutative diagram (see Fig. 4.4).

Let $\mathcal{F} \subseteq \mathrm{Sym}(V)$ be as in Section 4.1. By construction of the constraint graph model, $\kappa(\boldsymbol{z}) \in \mathcal{S}$ if and only if $\boldsymbol{z} \in \mathrm{COPsol}$. Thus, also $\tilde{\kappa} : \mathrm{Sym}(\mathrm{COPsol}) \to \mathrm{Sym}(\mathcal{S})$ is a bijection. $\Psi \circ \tilde{\iota} = \tilde{\kappa} \circ \Phi$ now implies that $F \coloneqq \tilde{\iota}^{-1}(\mathcal{F}) \subseteq \mathrm{Sym}([N])$ is COP-feasibility-preserving in the sense of Definition 2.12. Therefore, we may also consider the restricted actions $\Psi : \mathcal{F} \to \mathrm{Sym}(\mathcal{S})$, $\Pi : F \to \mathrm{Sym}(\mathcal{S})$, and $\Phi : F \to \mathrm{Sym}(\mathrm{COPsol})$. We also obtain the restricted permutation (matrix) representation $\mathcal{P}_S = \lambda \circ \Phi : F \to \mathcal{U}(S)$. The restriction to feasibility-preserving mappings also yields a commutative diagram, which is displayed in Fig. 4.5.

We eventually combine the graph- and group-theoretical, and the linear algebraic approach to test $\mathcal{P}(F)$ for its mixer properties.

Theorem 4.7. *Let* COP *be of scheduling type, and let $G = (V, E)$ be its associated constraint graph. Then $\mathcal{P}(F)$ is* COP*-mixing if and only if $\mathcal{F}$ connects all feasible solutions.*

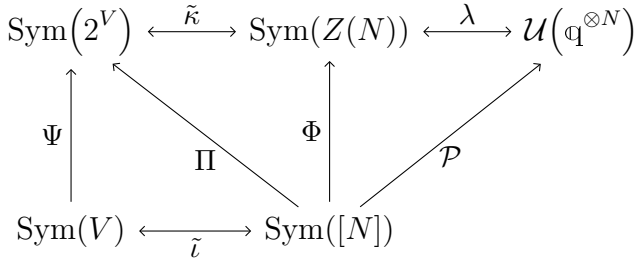

Figure 4.4.: Commutative diagram of bit permutations. $\mathrm{Sym}([N])$ acts on the vertex subsets of the associated constraint graph via Π. It consists of a vertex indexing and the group action Ψ. Alternatively, Π can be factorized into $\Phi : \mathrm{Sym}([N]) \to \mathrm{Sym}(Z(N))$ and the identification of vertex subsets with bit strings. The permutation (matrix) representation $\mathcal{P}$ factorizes into Φ and the subsequent linear extension to $\mathbb{q}^{\otimes N}$.

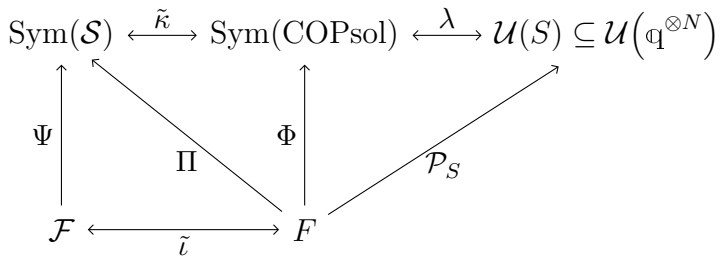

Figure 4.5.: Commutative diagram of feasibility-preserving bit permutations. F acts on the feasible solutions of the associated constraint graph via Π. It consists of a vertex indexing and the group action Ψ of $\mathcal{F}$. Alternatively, Π can be factorized into $\Phi : F \to \mathrm{Sym}(\mathrm{COPsol})$ and the identification of vertex subsets with bit strings. This identification preserves feasibility by construction of the constraint graph. The restricted permutation (matrix) representation $\mathcal{P}_S$ factorizes into Φ and the subsequent linear extension to S.

Proof. By Corollary 3.13 and Corollary 2.10, $\mathcal{P}(F)$ is COP-mixing if and only if F acts transitively on COPsol via Φ. Now it holds that

$$\tilde{\kappa} \circ \Phi = \Pi = \Psi \circ \tilde{\iota} \;\Leftrightarrow\; \Phi = \tilde{\kappa}^{-1} \circ \Psi \circ \tilde{\iota}.$$

Since $\tilde{\kappa}^{-1} : \mathrm{Sym}(\mathcal{S}) \to \mathrm{Sym}(\mathrm{COPsol})$ and $\tilde{\iota} : F \to \mathcal{F}$ are group isomorphisms, Φ is transitive if and only if Ψ is transitive. □

Therefore, if $\mathcal{F}$ connects all feasible solutions, we can construct parametrized mixers from elements of $\mathcal{P}(F)$ by suitably taking logarithms (cf. Lemma 3.19) and taking their absolute value (cf. Proposition 3.14). Invoking (2.4.1), it then suffices to consider merely a set of generators of F to construct parametrized mixers.

4.3. Applications

We start with the prototypical class of scheduling problems: free job-shop scheduling problems. One is given a list of J jobs and M machines with T time slots each and is asked to distribute the jobs to the slots such that

J: every job gets performed precisely once and

S: no slot is filled with more than one job.

The cost function can be manifold: machines may perform different jobs with certain costs, performing specific jobs in order may reduce or increase costs, etc. In the subsequent description of these problems, we therefore focus on the constraints rather than the cost function.

Definition 4.8. Let $J, M, T \in \mathbb{N}$. A *free job-shop scheduling problem of type* (J, M, T) is a COP

$$\mathrm{FJSP}(J, M, T) \coloneqq \Big(JMT,\ \{c_a\}_{a=1}^{A},\ \{C_a\}_{a=1}^{A},\ \mathbf{J} \cup \mathbf{S}, \mathrm{ext}\Big)$$

with arbitrary costs/priorities c_a and clauses C_a, *job assignment constraints*

$$\begin{aligned} \mathbf{J} &\coloneqq \{\zeta_{\mathsf{X}(j)} \,:\, j \in [J]\} \quad \text{with} \\ \mathsf{X}(j) &\coloneqq \mathcal{E}\Big(\{j\} \times [M] \times [T]\Big), \quad j \in [J], \end{aligned} \tag{4.3.1}$$

and *slot assignment constraints*

$$\begin{aligned} \mathbf{S} &\coloneqq \{\eta_{\mathsf{Y}(m,t)} \,:\, (m,t) \in [M] \times [T]\} \quad \text{with} \\ \mathsf{Y}(m,t) &\coloneqq \mathcal{E}\Big([J] \times \{m\} \times \{t\}\Big), \quad (m,t) \in [M] \times [T]. \end{aligned} \tag{4.3.2}$$

Here,

$$\begin{aligned} \mathcal{E} : [J] \times [M] \times [T] &\to [JMT] \\ (j, m, t) &\mapsto MT(j-1) + T(m-1) + t \end{aligned} \tag{4.3.3}$$

denotes the *canonical encoding function* for the free job-shop scheduling problem.

The job assignment constraints ensure that any feasible bit string allocates every job $j \in [J]$ exactly once. Furthermore, the slot assignment constraints guarantee that it also gives any slot at most one job. Thus, the constraints imposed directly incorporate the problem framework. In addition, we already see that the problem is

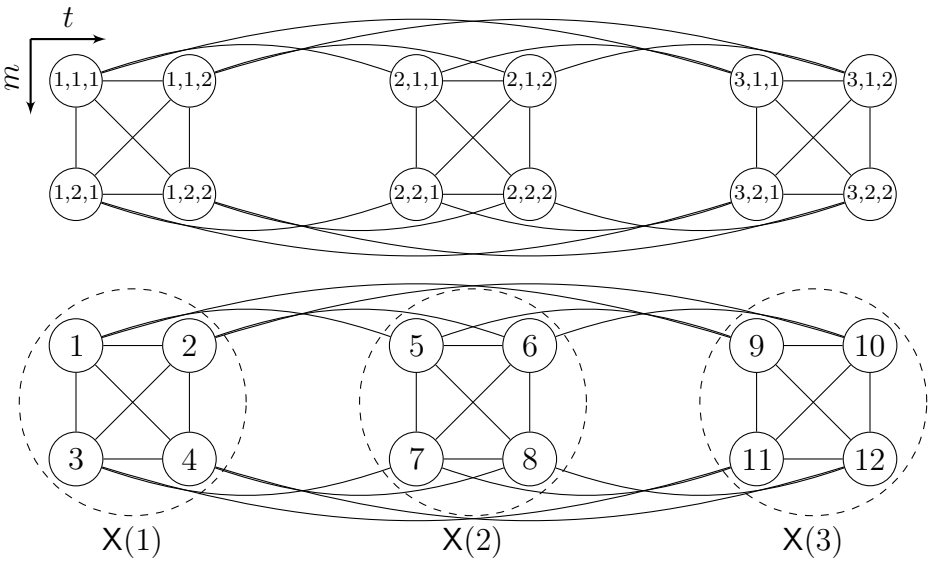

Figure 4.6.: FJSP(3, 2, 2) constraint graph. In the above graph, the vertices are labeled using the coordinate system (j, m, t), while in the graph below, they are labeled via the canoncial encoding function $\mathcal{E}$. In addition, the three job blocks $\mathsf{X}(j)$, $j \in [3]$, are drawn in.

infeasible if $J > MT$; therefore, we will w.l.o.g. assume that $J \leq MT$. Note that the case $(J, M, T) = (T, 1, T)$ precisely gives the traveling salesperson problem (compare Fig. 4.1). The constraint graph of a FJSP(3, 2, 2) instance is shown in Fig. 4.6.

In the following, we write $v_{j,m,t} \coloneqq \iota\Big(\mathcal{E}(j, m, t)\Big)$ to denote the vertex associated with the coordinates (j, m, t). The solutions to FJSP $\coloneqq$ FJSP(J, M, T) (as vertex subsets) are precisely given by

$$\begin{aligned}\mathcal{S}_{\mathrm{FJSP}} &= \bigcup\Big\{\{v_{j,m_j,t_j} \,:\, j \in [J]\} \,:\, (t_1, m_1), \ldots, (t_J, m_J) \in [M] \times [T]\Big\} \\ &=: \bigcup_{([M]\times[T])^{\underline{J}}} \Big\{\{v_{1,m_1,t_1}, \ldots, v_{J,m_J,t_J}\}\Big\}.\end{aligned} \tag{4.3.4}$$

Here, "$\bigcup_{X^{\underline{n}}}$" means the union over all n-subsets of X, that is, we adopted the falling factorial notation. Indeed, assigning the first job to a slot (m_1, t_1) leaves $MT - 1$ slots open for further job assignments; the assignment of the second job then reduces the number of available slots to $MT - 2$, and so on. Accordingly, the number of solutions is given by

$$|\mathcal{S}_{\mathrm{FJSP}}| = \frac{(MT)!}{(MT - J)!} =: (MT)^{\underline{J}}. \tag{4.3.5}$$

For $J = 1$, the slot assignment constraints are meaningless since in this case, $|\mathsf{Y}(m,t)| = 1$ holds. Thus, the only proper constraint is the remaining job assignment constraint; the constraint graph equals K^{MT} with its autmorphism group $\mathrm{Sym}([MT])$.

Now, let $G(J,M,T) = (V,E)$ denote the constraint graph of $\mathrm{FJSP}(J,M,T)$ for $2 \leq J \leq MT$. Take $(j,m,t\,),(j',m',t') \in [J] \times [M] \times [T]$ so that $v_{j,m,t}\, v_{j',m',t'} \notin E$. Then $j \neq j'$ and $(m,t) \neq (m',t')$ necessarily hold. Since $J \leq MT$, we can augment the set $\{(m,t),(m',t')\}$ to a J-subset

$$\{(m_1,t_1),\ldots,(m_J,t_J)\} \subseteq [M] \times [T] \quad \text{with } (m_j\,,t_j\,) = (m,t) \text{ and } (m_{j'},t_{j'}) = (m',t').$$

But then $\{v_{j,m_j,t_j} : j \in [J]\} \in \mathcal{S}_{\mathrm{FJSP}}$ holds. Thus, invoking Proposition 4.5, we obtain the following result.

Corollary 4.9. *Let $J,M,T \in \mathbb{N}$ with $J \leq MT$, and let G be the constraint graph of $\mathrm{FJSP}(J,M,T)$. Then $\mathcal{F} = \mathrm{Aut}(G)$.*

In addition, the problem indeed admits suitable mixers. Let $S,S' \in \mathcal{S}_{\mathrm{FJSP}}$. By (4.3.4), there exist $m_1,\ldots,m_J,m'_1,\ldots,m'_J \in [M]$ and $t_1,\ldots,t_J,t'_1,\ldots,t'_J \in [T]$ so that

$$S = \left\{v_{1,m_1,t_1}\,,\ldots,v_{J,m_J,t_J}\right\} \quad \text{and} \quad S' = \left\{v_{1,m'_1,t'_1},\ldots,v_{J,m'_J,t'_J}\right\}$$

hold. Since $\mathrm{Sym}([M] \times [T])$ acts (MT)-transitively on $[M] \times [T]$, there exists a $\sigma = (\sigma_1,\sigma_2) \in \mathrm{Sym}([M] \times [T])$ so that

$$\sigma(m_j,t_j) = (\sigma_1(m_j),\sigma_2(t_j)) = (m'_j,t'_j) \quad \forall j \in [J].$$

Define

$$\hat{\sigma} : V \to V, \quad v_{j,m,t} \mapsto v_{j,\sigma_1(m),\sigma_2(t)}.$$

Then $\hat{\sigma}$ is bijective and feasibility-preserving, hence $\hat{\sigma} \in \mathcal{F}$. Furthermore, $\hat{\sigma}$ is constructed to satisfy $\hat{\sigma}(S) = S'$. This proves the next result.

Corollary 4.10. *Let $J,M,T \in \mathbb{N}$ with $J \leq MT$, and let G be the constraint graph of $\mathrm{FJSP}(J,M,T)$. Then $\mathcal{F}$ connects all feasible solutions.*

In conclusion, the solutions to FJSP admit a nice and simple structure that allows a complete traversing of all solutions with feasibility-preserving permutations. We have explicitely constructed an embedding

$$\hat{} : \mathrm{Sym}([M] \times [T]) \to \mathrm{Sym}(V), \quad \sigma \mapsto \hat{\sigma} \tag{4.3.6}$$

that already yields a mixing subgroup. However, this embedding does generally not yield the whole group $\mathcal{F}$. For example, a similar embedding

$$\check{} : \mathrm{Sym}([J]) \to \mathrm{Sym}(V), \quad \rho \mapsto \check{\rho} \coloneqq [v_{j,m,t} \mapsto v_{\rho(j),m,t}] \tag{4.3.7}$$

produces different feasibility-preserving permutations. As we have already constructed suitable mixers for FJSP in the form of a proper subgroup of $\mathcal{F}$, we will not investigate $\mathcal{F}$ and thus $\mathrm{Aut}(G(J, M, T))$ further.

Within the framework of FJSP, the machines and the time slots are treated equally by the constraints. Therefore, we could have replaced the set $[M] \times [T]$ simply with a set of slots $[S]$, and all results would remain valid. Especially, the case where each machine has an individual amount of time slots is not any more complicated than the symmetric case discussed. The supposed restriction we have made stems from simple visualization reasons and the fact that the time slots are given their "correct" meaning in the following.

One possible step to impede the free job-shop scheduling problem is subdividing the jobs into operations that must be executed in a non-descending order, thus conceptually separating the time slots from the machines. The problem is then called a flexible job-shop scheduling problem. One is given a list of J jobs, each consisting of O ordered operations, and M machines with T time slots each and is now asked to distribute the operations to the slots such that

J: every operation of every job gets performed precisely once,

S: no slot is filled with more than one operation, and

P: subsequent operations of the same job are performed in non-descending order.

The cost functions can again be diverse: machines may perform different operations with certain costs, performing all operations of one job on the same machine may reduce or increase costs, additional orders may reduce or increase the costs, certain jobs may be of higher priority than others, etc. We collect all these possibilities by merely specifying the constraints.

Definition 4.11. Let $J, O, M, T \in \mathbb{N}$. A (*flexible*) *job-shop scheduling problem of type* (J, O, M, T) is a COP

$$\mathrm{JSP}(J, O, M, T) \coloneqq \Big(JOMT,\ \{c_a\}_{a=1}^{A},\ \{C_a\}_{a=1}^{A},\ \mathbf{J} \cup \mathbf{S} \cup \mathbf{P}, \mathrm{ext}\Big)$$

with arbitrary costs/priorities c_a and clauses C_a, *job assignment constraints*

$$\begin{aligned} \mathbf{J} &\coloneqq \{\zeta_{\mathsf{X}(j,o)} \,:\, (j,o) \in [J] \times [O]\} \quad \text{with} \\ \mathsf{X}(j,o) &\coloneqq \mathcal{E}\Big(\{j\} \times \{o\} \times [M] \times [T]\Big), \quad j \in [J],\ o \in [O], \end{aligned} \tag{4.3.8}$$

slot assignment constraints

$$\begin{aligned} \mathbf{S} &\coloneqq \{\eta_{\mathsf{Y}(m,t)} \,:\, (m,t) \in [M] \times [T]\} \quad \text{with} \\ \mathsf{Y}(m,t) &\coloneqq \mathcal{E}\Big([J] \times [O] \times \{m\} \times \{t\}\Big), \quad (m,t) \in [M] \times [T], \end{aligned} \tag{4.3.9}$$

and the *precedence constraints*

$$\begin{aligned} \mathbf{P} &\coloneqq \bigcup_{(j,o,m,t)} \mathbf{P}_{j,o,m,t} \\ \mathbf{P}_{j,o,m,t} &\coloneqq \{\eta_{\mathsf{Z}(j,o;\,m,m',t,t')} \,:\, m \in [M], 1 \le t' < t\} \quad \text{with} \\ \mathsf{Z}(j,o;\,m,m',t,t') &\coloneqq \begin{cases} \mathcal{E}\Big(\{(j,o,m,t),(j,o+1,m',t')\}\Big), & \text{if } 1 \le o < O \\ \emptyset, & \text{if } o = O. \end{cases} \end{aligned} \tag{4.3.10}$$

Here,

$$\begin{aligned} \mathcal{E} : [J] \times [O] \times [M] \times [T] &\to [JOMT] \\ (j,o,m,t) &\mapsto OMT(j-1) + MT(o-1) + T(m-1) + t \end{aligned} \tag{4.3.11}$$

denotes the *canonical encoding function* for the flexible job-shop scheduling problem.

Similar to the free job-shop case, the job assignment constraints assure that any feasible bit string allots every operation of every job $(j,o) \in [J] \times [O]$ exactly once and the slot assignment constraints make sure that every slot is assigned at most one operation. Thus, the problem is infeasible if $JO > MT$. Lastly, the precedence constraints enforce the execution of the operation of the same job in non-descending order. In the following, let $G(J,O,M,T)$ denote the constraint graph of $\mathrm{JSP}(J,O,M,T)$. We will especially focus on the case where $J = 1$. Then the problem is similar to the free job-shop scheduling problem, with the operations $[O]$ being an ordered substitute for the jobs in the free case. Namely, we have the following graph splitting into a free job-shop part and a precedence constraint part:

$$G(1,O,M,T) = G(O,M,T) \cup G_{\mathbf{P}}(O,M,T), \tag{4.3.12}$$

whereby

$$V(G_{\mathbf{P}}(O,M,T)) = V(G(O,M,T))$$

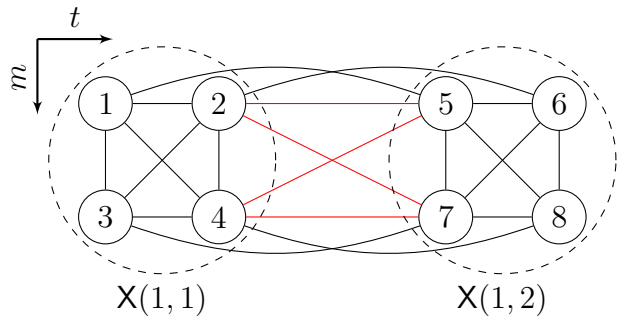

Figure 4.7.: JSP$(1, 2, 2, 2)$ constraint graph. The vertices are labled via the canoncial encoding function $\mathcal{E}$. In addition, the two operation blocks $\mathsf{X}(1, o)$, $o \in [2]$, are drawn in. The black edges correspond to job assignment constraints $\mathbf{J}$ and slot assignment constraints $\mathbf{S}$, while the red ones correspond to the precedence constraints $\mathbf{P}$. Neglecting the red edges thus reproduces $G(2, 2, 2)$, while ignoring the black ones yields $G_{\mathbf{P}}(2, 2, 2)$.

and

$$E(G_{\mathbf{P}}(O, M, T)) = \Big\{ v_{o,m,t}\, v_{o+1,m',t'} \,:\, o \in [O-1],\, m, m' \in [M],\, t, t' \in [T],\, t' < t \Big\}.$$

Next, we investigate the extent to which analogs of Corollary 4.9 and Corollary 4.10 apply to JSP$(1, O, M, T)$. We start with JSP$(1, 2, 2, 2)$ and its constraint graph shown in Fig. 4.7. Since its feasible solutions consist of two elements, respectively, Proposition 4.5 implies that in this particular case $\mathcal{F} = \mathrm{Aut}(G(1, 2, 2, 2))$ holds. Henceforth we will directly identify each vertex $v_i = (\iota \circ \mathcal{E})(j, o, m, t)$ with its label $i = \mathcal{E}(j, o, m, t)$. We obtain

$$\begin{aligned}\mathrm{Aut}(G(1, 2, 2, 2)) = \{&\mathrm{id}, (1\,3)(5\,7), (2\,4)(6\,8), (1\,3)(2\,4)(5\,7)(6\,8), (1\,6)(2\,5)(3\,8)(4\,7),\\ &(1\,8)(2\,7)(3\,6)(4\,5), (1\,6\,3\,8)(2\,7\,4\,5), (1\,8\,3\,6)(2\,5\,4\,7)\}\\ = \langle &(1\,6\,3\,8)(2\,7\,4\,5), (1\,3)(2\,4)(5\,7)(6\,8)\rangle.\end{aligned}$$

Thus, $\mathrm{Aut}(G(1, 2, 2, 2)) \cong \mathrm{D}_8$, the dihedral group of order 8. Meanwhile, the feasible solutions are given by

$$\mathcal{S} = \Big\{\{1, 6\}, \{1, 7\}, \{1, 8\}, \{2, 8\}, \{3, 5\}, \{3, 6\}, \{3, 8\}, \{4, 6\}\Big\}.$$

As one can verify, $\mathcal{F} = \mathrm{Aut}(G(1, 2, 2, 2))$ does not act transitively on $\mathcal{S}$. For example, there is no element in $\mathcal{F}$ that maps $\{1, 7\}$ to $\{1, 8\}$. The permutation $(34)(78)$, which is feasibility-preserving in $G(2, 2, 2)$ and would do the transition, does not respect the precedence constraints as it maps the feasible solution $\{3, 8\}$ to the non-solution $\{4, 7\}$ and is thus not an element of $\mathcal{F}$ in $G(1, 2, 2, 2)$. We display the dihedral symmetry of $G(1, 2, 2, 2)$ by a suitable symmetric graph drawing in Fig. 4.8.

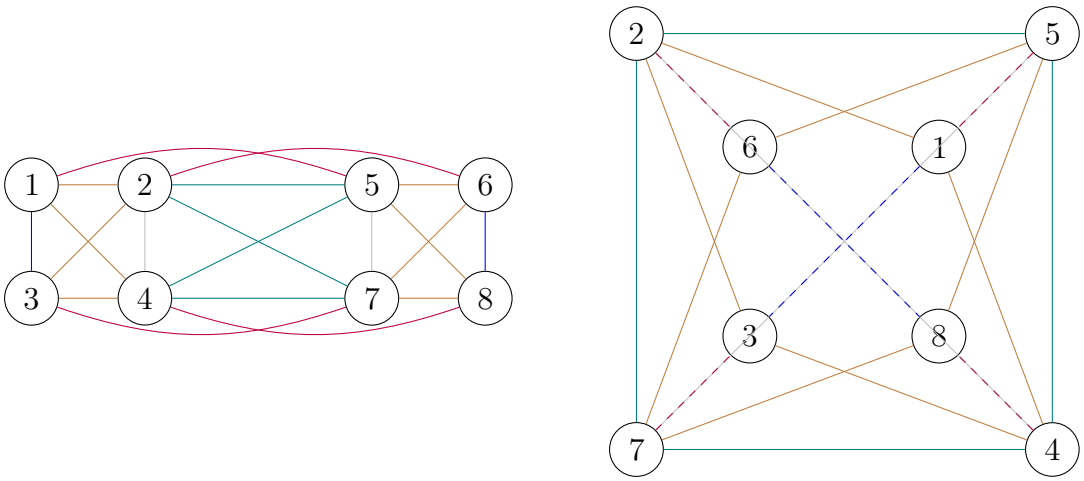

Figure 4.8.: Dihedral symmetry display of the JSP$(1,2,2,2)$ constraint graph. The vertices are divided into "inner" vertices $\{1,3,6,8\}$ and "outer" vertices $\{2,4,5,7\}$. The swap of 1 and 6 as well as of 4 and 7 unravels the graph and displays its full symmetry.

We proceed with JSP$(1,3,2,2)$ and its constraint graph shown in Fig. 4.9. This time, the solutions consist of three elements, respectively. Therefore, Proposition 4.5 is not readily applicable. In fact, we find that $\mathcal{F} \neq \mathrm{Aut}(G(1,3,2,2))$.

For the autmorphism group of the constraint graph, we obtain

$$\begin{aligned}\mathrm{Aut}(G(1,3,2,2)) = \{&\mathrm{id}, (1\,3)(5\,7)(9\,11), (2\,4)(6\,8)(10\,12),\\ &(1\,3)(2\,4)(5\,7)(6\,8)(9\,11)(10\,12),\\ &(1\,10)(2\,9)(3\,12)(4\,11)(5\,6)(7\,8),\\ &(1\,12)(2\,11)(3\,10)(4\,9)(5\,8)(6\,7),\\ &(1\,10\,3\,12)(5\,6\,7\,8)(2\,11\,4\,9),\\ &(1\,12\,3\,10)(5\,8\,7\,6)(2\,9\,4\,11)\}\\ = \langle &(1\,3)(2\,4)(5\,7)(6\,8)(9\,11)(10\,12), (1\,10\,3\,12)(5\,6\,7\,8)(2\,11\,4\,9)\rangle.\end{aligned}$$

The constraint graph again has the square symmetry, that is, $\mathrm{Aut}(G(1,3,2,2)) \cong \mathrm{D}_8$. An appropriate symmetric graph drawing is shown in Fig. 4.10. The feasible solutions are given by

$$\mathcal{S} = \Big\{\{1,6,12\},\{1,7,10\},\{1,7,12\},\{1,8,10\},\{3,5,10\},\{3,5,12\},\{3,6,12\},\{3,8,10\}\Big\}.$$

One can verify that $\mathrm{Aut}(G(1,3,2,2))$ already acts transitively on $\mathcal{S}$. However, the whole group $\mathcal{F}$ of feasibility preserving permutations is even larger. As a direct consequence of the precedence constraints, the vertices $2, 4, 9$, and 11 are not contained in any solution:

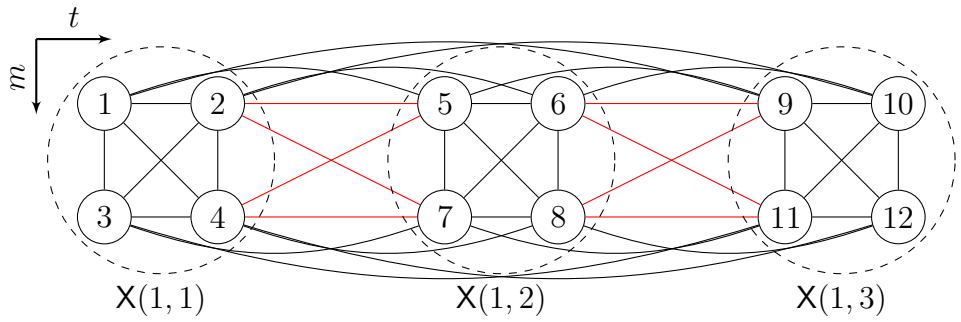

Figure 4.9.: JSP$(1, 3, 2, 2)$ constraint graph. The vertices are labeled via the canonical encoding function $\mathcal{E}$. In addition, the three operation blocks $\mathsf{X}(1, o)$, $o \in [3]$, are drawn in. The black edges correspond to job assignment constraints $\mathbf{J}$ and slot assignment constraints $\mathbf{S}$, while the red ones correspond to the precedence constraints $\mathbf{P}$. Neglecting the red edges thus reproduces $G(3, 2, 2)$, while ignoring the black ones yields $G_{\mathbf{P}}(3, 2, 2)$.

performing the first operation on the second time slot would not leave enough space for the remaining two operations. Analogously, assigning the third operation to the first time slot would force one of the remaining two operations to be performed on the second slot and thus after the third operation. Therefore, the property of a permutation being feasibility-preserving is determined by its action on the proper vertex subset $\{1, 3, 5, 6, 7, 8, 10, 12\}$. In contrast, the "global" automorphism property depends on the action on all vertices. In this light, any modification of an automorphism on the subset $\{2, 4, 9, 11\}$ yields again an element in $\mathcal{F}$ but, in general, not an automorphism. Take, e.g., $(2\,4)(6\,8)(10\,12) \in \mathrm{Aut}(G(1, 3, 2, 2)) \subset \mathcal{F}$. Then the modification $(6\,8)(10\,12)$ is again in $\mathcal{F}$ but not in $\mathrm{Aut}(G(1, 3, 2, 2))$.

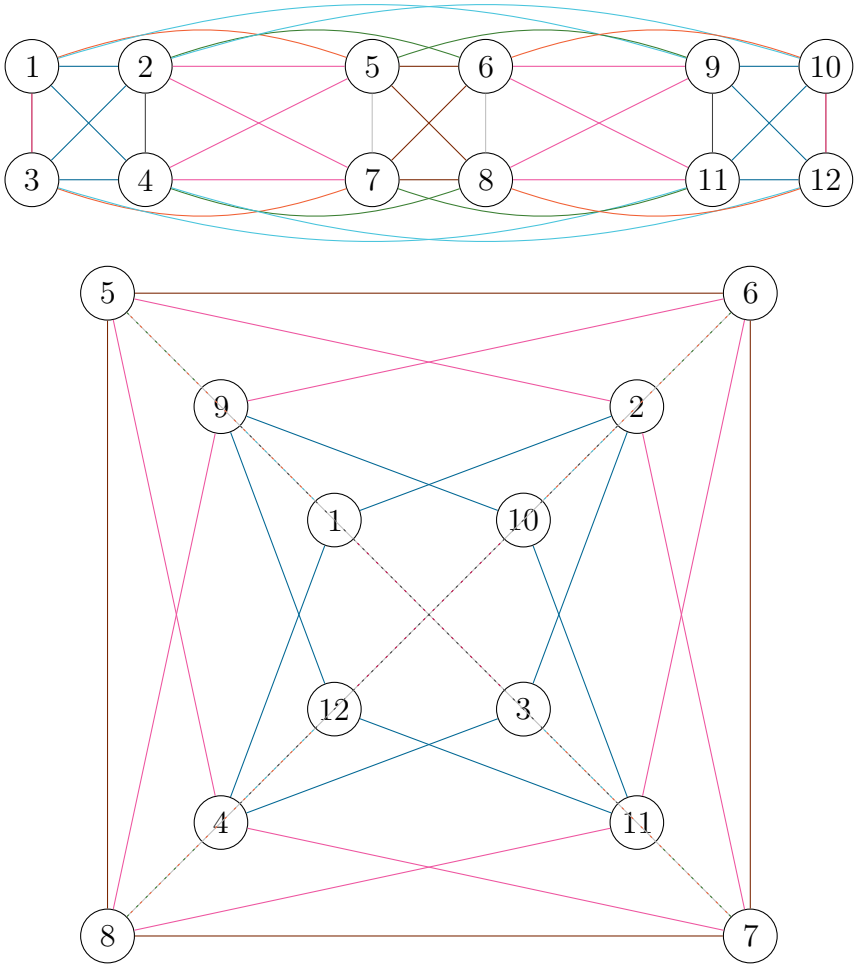

Figure 4.10.: Dihedral symmetry display of the JSP$(1, 3, 2, 2)$ constraint graph. The vertices are divided into "inner", "middle", and "outer" vertices. The middle vertices 2, 4, 9, and 11 are not contained in any feasible solutions. The swap of 2 and 9, 3 and 12, and 7 and 8 displays the full symmetry.

Other Variational Quantum Algorithms

Besides the QAOA and its generalization to the QAO, there exist countless other variational quantum algorthims. While QAOA prescribes the parametrized unitaries quite rigidly for the generic case, many other algorithms rely on more problem-specific approaches. Bharti et al. [Bha+22] provide an excellent overview of various quantum algorithms and their applications. They also clearly define the term *variational quantum algorithm* (VQA), which we state mutatis mutandis.

A VQA consists of four parts:

- an objective function to be minimized/maximized,
- a parametrized quantum circuit (PQC), i.e. parametrized unitaries,
- a measurement scheme to estimate expectation values, and
- a classical optimizer to obtain optimal circuit parameters.

For COPs, the generic implementation of the Boolean objective function C is given by the procedure (2.3.5) discussed in Section 2.3. Meanwhile, in quantum chemistry, another primary scope of VQAs, the objective Hamiltonian arises directly from a physical description of the system and generally contains non-diagonal elements representing interactions.

In the QAOA, the PQC consists of the p-fold alternation of parametrized mixers U_{M} and parametrized phase separators U_{P}. While the phase separator directly incorporates the objective Hamiltonian, the mixer is problem-independent. This rigidity is precisely what is being removed in the transition from the QAOA to the QAO.

In Section 5.1, we will briefly introduce the variational quantum eigensolver. In essence, it merely specifies which quantities are to be computed on the quantum computer; it does not prescribe a general parameterization rule or a classical optimizer. However, this flexibility is the reason for numerous generalizations of this algorithm. The variational quantum simulation of imaginary time evolution as discussed in Section 5.2 is another (incomplete) VQA but with a more concrete instruction for the interaction of classical and quantum computing. Both algorithms are eventually generalized by the filtering variational quantum eigensolver which we discuss in Section 5.3.

We emphasize that our focus entirely lies on quantum algorithmic methods and not on the classical optimizers. We are well aware that the influence of the classical optimization rule is everything but negligible and is, as well as the quantum part of a VQA, highly problem-dependent. However, as we mainly aim to understand future quantum algorithms better, we must cut the line somewhere. In this sense, performance studies that compare VQAs with different classical optimizers should also be taken with care. It cannot be excluded that alleged performance differences are only due to the different classical optimization rules.

5.1. Variational Quantum Eigensolver

The variational quantum eigensolver (VQE) was first proposed by Peruzzo et al. [Per+14] as an approach to calculate the eigenvalues of a given Hamiltonian based on the Rayleigh-Ritz variational method [Str11; Rit09].

Lemma 5.1 (Rayleigh-Ritz inequality)**.** *For $H \in \mathcal{L}(\Xi)$ self-adjoint and $|\psi\rangle \in \Xi \setminus \{0\}$, it holds that*

$$\min \sigma(H) \leq \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}. \tag{5.1.1}$$

Proof. Let $H \in \mathcal{L}(\Xi)$ be self-adjoint with eigenvalues $E_1 \leq E_2 \leq \ldots \leq E_d$. Let $\mathcal{B}_\Xi$ be an ONB consisting of eigenvectors of H, that is, $H|j\rangle = E_j|j\rangle$ for all $j = 1, \ldots, d$. Furthermore, let $|\psi\rangle \in \Xi$ be arbitrary. Then it holds that

$$\langle\psi|H|\psi\rangle = \sum_{j=1}^{d} \langle\psi|H|j\rangle\langle j|\psi\rangle = \sum_{j=1}^{d} E_j \langle\psi|j\rangle\langle j|\psi\rangle \geq E_1 \sum_{d=1}^{d} \langle\psi|j\rangle\langle j|\psi\rangle = E_1\langle\psi|\psi\rangle.$$

For $|\psi\rangle \neq 0$, one obtains (5.1.1) since $E_1 = \min \sigma(H)$. □

The quantity on the right-hand side of (5.1.1) is also called the Rayleigh quotient. One readily verifies from the proof of Lemma 5.1 that (5.1.1) becomes an equality if $|\psi\rangle$ is a ground state of H. This suggests a variational approach: consider a family of parameterized states $|\boldsymbol{\theta}\rangle$ with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_S)$, $S \leq d$, and iteratively determine the minimum of Rayleigh quotients for these states to obtain an approximation of the ground state and its energy E_1.

Using the decomposition (2.1.6), the expectation value of H can be calculated by

$$\begin{aligned}\langle\boldsymbol{\theta}|H|\boldsymbol{\theta}\rangle &= H_0^0 + \sum_{n=1}^{N}\sum_{\kappa=1}^{3} H_\kappa^n \langle\boldsymbol{\theta}|\sigma_n^\kappa|\boldsymbol{\theta}\rangle + \sum_{n_1=1}^{N}\sum_{n_2=1}^{N}\sum_{\kappa_1=1}^{3}\sum_{\kappa_2=1}^{3} H_{\kappa_1\kappa_2}^{n_1 n_2} \langle\boldsymbol{\theta}|\sigma_{n_1}^{\kappa_1}\sigma_{n_2}^{\kappa_2}|\boldsymbol{\theta}\rangle + \ldots \\ &= \sum_{t=1}^{T} \check{H}_t \langle\boldsymbol{\theta}|\Sigma^t|\boldsymbol{\theta}\rangle\,.\end{aligned} \tag{5.1.2}$$

Since H is Hermitian, the coefficients $\check{H}_t$ have to be real-valued. The trial states $|\boldsymbol{\theta}\rangle$ are produced in the PQC, that is, by applying parametrized unitary gates:

$$|\boldsymbol{\theta}\rangle \coloneqq V(\boldsymbol{\theta})\,|\mathbf{0}\rangle \coloneqq U_S(\theta_S)\cdots U_1(\theta_1)\,|\mathbf{0}\rangle\,. \tag{5.1.3}$$

In practice, only the expectation values of the Pauli terms are calculated on a QPU while multiplying by the coefficients $\check{H}_t$ and adding is performed on a CPU. A general VQE-circuit is sketched in Fig. 5.1. To obtain $\langle\boldsymbol{\theta}|H|\boldsymbol{\theta}\rangle$ it is necessary to derive said coefficients in advance. However, this is always possible, although often cumbersome.

The procedure can be extended in order to find excited states, following the folded spectrum method (cf. [Mac34]). One replaces H with $H_\lambda \coloneqq (H-\lambda)^2$ and scans through different values of $\lambda \in \mathbb{R}$. The variational method then converges to the eigenstate with eigenvalue closest to the shift parameter λ. Note that this consideration increases the number of terms T in (5.1.2) only quadratically.

As highlighted by Peruzzo et al., the strength of the VQE lies in two facts: firstly, expectation values can be efficiently approximated on a quantum computer (cf. [Ort+01]). Secondly, the state $|\boldsymbol{\theta}\rangle$ will be measured immediately after its implementation. Thus, the required coherence time is comparably short. In contrast, the quantum phase estimation (QPE) proposed by Kitaev [Kit95] requires larger circuit depths and a long coherence time. However, the amount of measurements in QPE necessary to reach a certain precision is comparably small. There is an attempt to interpolate between both VQE und QPE proposed by Wang et al. [WHB19], called the α-VQE.

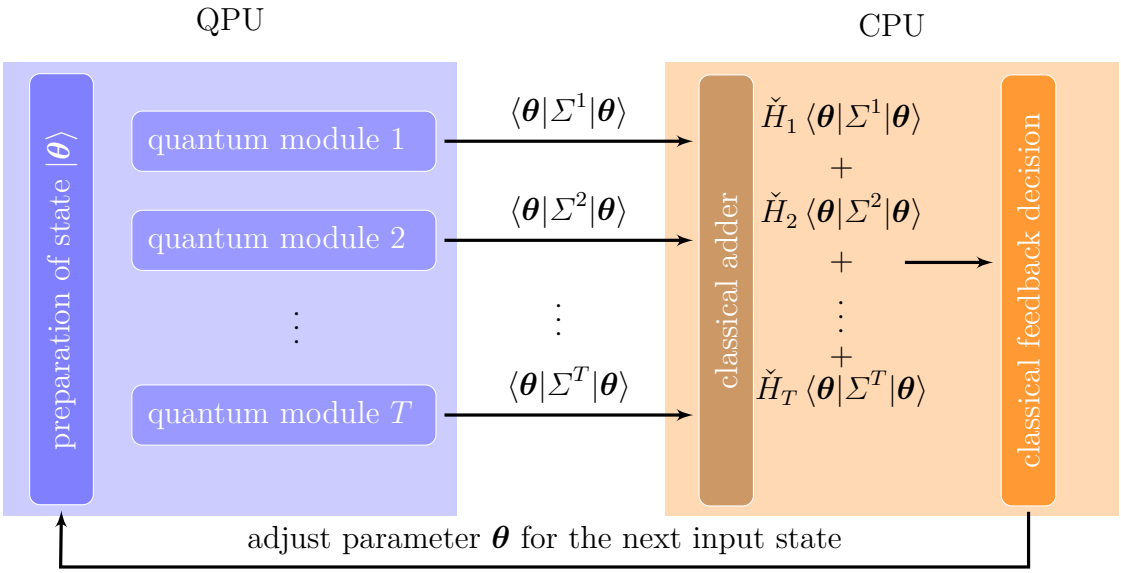

Figure 5.1.: Architecture of the VQE (compare [Per+14, FIG. 1]). The initial state $|\boldsymbol{\theta}\rangle$ is prepared on the QPU and fed into the quantum modules which compute the expectation values $\langle\boldsymbol{\theta}|\Sigma^t|\boldsymbol{\theta}\rangle$, for $t = 1, \ldots, T$. These are multiplied by $\check{H}_t$, respectively, and added together on the CPU to obtain $\langle\boldsymbol{\theta}|H|\boldsymbol{\theta}\rangle$. This quantity is, in turn, used for the Ritz optimization method. The new state parameters are eventually fed back to the QPU.

5.2. Variational Quantum Simulation of Imaginary Time Evolution

Propagating a wave function in imaginary time is a mathematical trick often used to determine a Hamiltonian's ground state. This method, also known as *Wick rotation* [Wic54], is justified by

Lemma 5.2. *For $H \in \mathcal{L}(\Xi)$ self-adjoint and $|\psi\rangle \in \Xi \setminus \{0\}$, it holds that*

$$\lim_{\tau\to\infty} \frac{e^{-H\tau}\,|\psi\rangle}{\sqrt{\langle\psi|e^{-2H\tau}|\psi\rangle}} = \frac{\mathbb{P}\,|\psi\rangle}{\sqrt{\langle\psi|\mathbb{P}|\psi\rangle}}, \tag{5.2.1}$$

where $\mathbb{P}$ denotes the projection onto the eigenspace of H corresponding to the smallest eigenvalue such that $|\psi\rangle$ is not in its orthogonal complement.

Proof. This follows as a special case from Lemma 5.3 with $f = 1/\exp$. □

Starting with an initial state $|\psi\rangle$ with non-zero overlap with a ground state of H, one may approximate the latter by simulating the imaginary time evolution of $|\psi\rangle$. Since $e^{-H\tau}$ is not a unitary operator, it cannot be implemented on a quantum computer and is thus outsourced to a classical computer. Following [McA+19], it is implemented as a variational method based on McLachlan's variational principle (cf. [McL64]), yielding the variational quantum simulation of imaginary time evolution (Var-QITE). We consider τ-dependent parameters $\boldsymbol{\theta}(\tau) = (\theta_1(\tau), \ldots, \theta_S(\tau))$ and trial states

$$|\boldsymbol{\theta}(\tau)\rangle \coloneqq V(\boldsymbol{\theta}(\tau))\,|\mathbf{0}\rangle \coloneqq U_S(\theta_S(\tau)) \cdots U_1(\theta_1(\tau))\,|\mathbf{0}\rangle\,. \tag{5.2.2}$$

Considering the decomposition (2.1.6) of H, the application of McLachlan's variational principle to the trial states (5.2.2) yields a differential equation for the parameters:

$$\sum_{s=1}^{S} \eta_{r,s}\dot{\theta}_s(\tau) = \xi_r, \tag{5.2.3}$$

where

$$\eta_{r,s}(\tau) = \mathrm{Re}\left(\frac{\partial\,\langle\boldsymbol{\theta}(\tau)|}{\partial\,\theta_r(\tau)}\frac{\partial\,|\boldsymbol{\theta}(\tau)\rangle}{\partial\,\theta_s(\tau)}\right) \quad \text{and} \quad \xi_r(\tau) = \mathrm{Re}\left(-\sum_{t=1}^{T}\check{H}_t\frac{\partial\,\langle\boldsymbol{\theta}(\tau)|}{\partial\,\theta_r(\tau)}\Sigma^t\,|\boldsymbol{\theta}(\tau)\rangle\right).$$

Assuming that for all $s = 1, \ldots, S$, it holds that

$$\frac{\partial U_s(\theta_s(\tau))}{\partial\theta_s(\tau)} = \sum_{i=1}^{I_s} f_{i,s}U_s(\theta_s(\tau))W_{i,s}$$

with amplitudes $f_{i,s} \in \mathbb{C}$ and unitaries $W_{i,s} \in \mathcal{L}(\mathbb{q}^{\otimes N})$, we obtain

$$\begin{aligned}\frac{\partial\,|\boldsymbol{\theta}(\tau)\rangle}{\partial\,\theta_s(\tau)} &= \sum_{i=1}^{I_s} f_{i,s}U_S(\theta_S(\tau)) \cdots U_i(\theta_i(\tau))W_{i,s}U_{i-1}(\theta_{i-1}(\tau)) \cdots U_1(\theta_1(\tau))\,|\mathbf{0}\rangle \\ &=: \sum_{i=1}^{I_s} f_{i,s}\tilde{V}_{i,s}(\boldsymbol{\theta}(\tau))\,|\mathbf{0}\rangle\,.\end{aligned}$$

This eventually yields the following form of the coefficients:

$$\begin{aligned}\eta_{r,s}(\tau) &= \mathrm{Re}\left(\sum_{i=1}^{I_r}\sum_{j=1}^{J_s} f_{i,r}^* f_{j,s}\,\langle\mathbf{0}|\tilde{V}_{i,r}^*(\boldsymbol{\theta}(\tau))\tilde{V}_{j,s}(\boldsymbol{\theta}(\tau))|\mathbf{0}\rangle\right) \qquad \text{and} \\ \xi_r(\tau) &= \mathrm{Re}\left(-\sum_{i=1}^{I_r}\sum_{t=1}^{T} f_{i,r}^*\check{H}_t\,\langle\mathbf{0}|\tilde{V}_{i,r}^*(\boldsymbol{\theta}(\tau))\Sigma^t V(\boldsymbol{\theta}(\tau))|\mathbf{0}\rangle\right).\end{aligned} \tag{5.2.4}$$

The resulting circuit is sketched in Fig. 5.2.

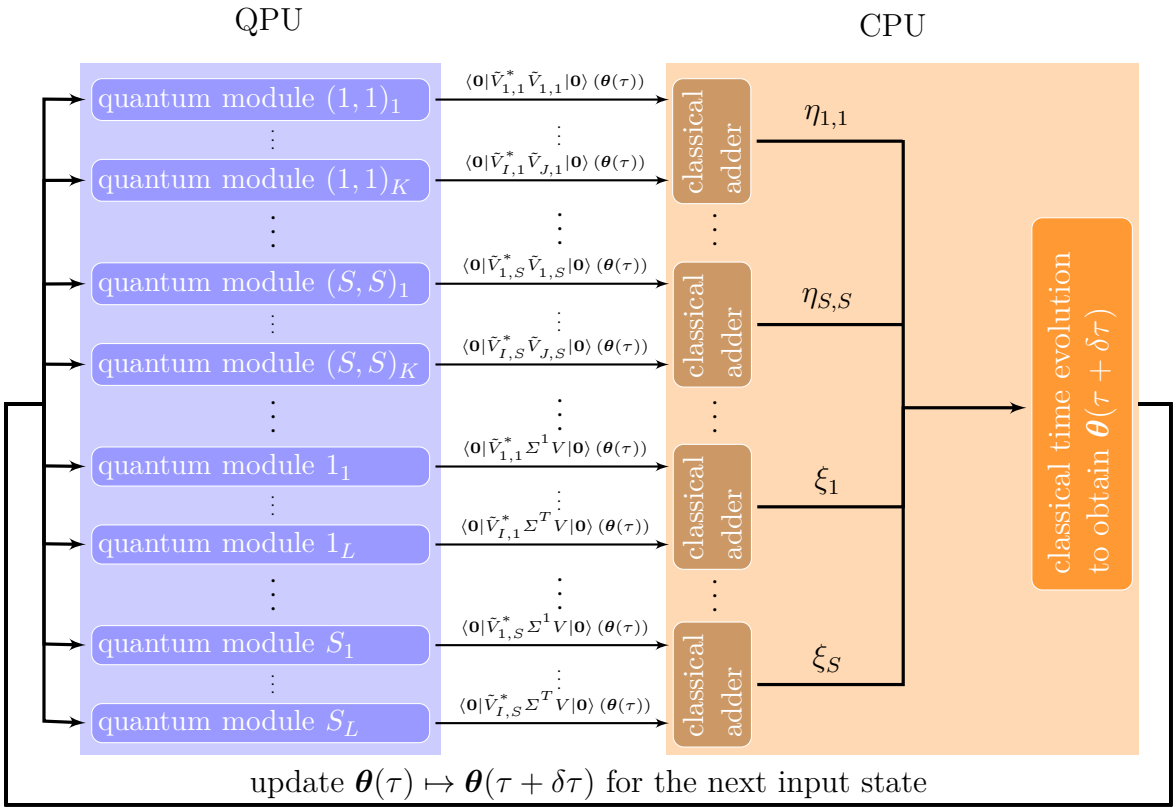

Figure 5.2.: Architecture of the Var-QITE. The unitary gates are set up in the corresponding quantum modules which compute their expectation value in the initial state $|\mathbf{0}\rangle$, respectively. From the results, the coefficients $\eta_{r,s}$ and ξ_r are calculated on the CPU according to (5.2.4). These are then used to simulate the imaginary time evolution of $\boldsymbol{\theta}$ over a small interval $\delta\tau$. The updated parameters are eventually fed back to the QPU.

Since the quantum modules respectively calculate expectation values of unitaries, the quantum circuit shown in Fig. 2.1 could be used to achieve this goal. However, since the considered unitaries are all of a specific form, a more efficient measurement circuit is proposed in [LB17, Fig. 2].

5.3. Filtering Variational Quantum Eigensolver

Building on the idea of the Var-QITE, Amaro et al. [Ama+21b] proposed the filtering variational quantum eigensolver (F-VQE). The core idea is to reduce the imaginary time evolution to its properties leading to convergence and abstracting them.

Lemma 5.3. *For $H \in \mathcal{L}(\Xi)$ self-adjoint, $|\psi\rangle \in \Xi \setminus \{0\}$, and $f : \sigma(H) \to \mathbb{R}_+$ strictly decreasing, it holds that*

$$\lim_{\tau\to\infty} \frac{f^\tau(H)\,|\psi\rangle}{\sqrt{\langle\psi|f^{2\tau}(H)|\psi\rangle}} = \frac{\mathbb{P}\,|\psi\rangle}{\sqrt{\langle\psi|\mathbb{P}|\psi\rangle}}, \tag{5.3.1}$$

where $\mathbb{P}$ denotes the projection onto the eigenspace of H corresponding to the smallest possible eigenvalue such that $|\psi\rangle$ is not in its orthogonal complement.

Proof. Let $H \in \mathcal{L}(\Xi)$ be self-adjoint with eigenvalues $E_1 \leq E_2 \leq \ldots \leq E_d$ and let $\mathcal{B}_\Xi$ be an ONB consisting of eigenvectors of H, that is, $H\,|j\rangle = E_j\,|j\rangle$ for all $j = 1, \ldots, d$. Let $|\psi\rangle \in \Xi \setminus \{0\}$ and consider its decomposition

$$|\psi\rangle = \sum_{j=1}^{d} \eta_j\,|j\rangle\,, \quad \eta_j \coloneqq \langle j|\psi\rangle\,.$$

Since $|\psi\rangle \neq 0$, it holds that $\{1 \leq j \leq d : \eta_j \neq 0\} \neq \emptyset$. Let k denote the minimum of this set. Then the decomposition simplifies to

$$|\psi\rangle = \sum_{j=k}^{d} \eta_j\,|j\rangle\,.$$

Furthermore, let $\mathrm{Eig}(H, E_k)$ denote the corresponding eigenspace. W.l.o.g. assume that it is one dimensional, i.e. $\mathrm{Eig}(H, E_k) = \mathrm{span}\{|k\rangle\}$. Let $f : \sigma(H) \to \mathbb{R}_+$ be strictly decreasing. For all $j > k$, it follows that

$$E_j > E_k \;\Rightarrow\; \frac{f(E_j)}{f(E_k)} < 1 \;\Rightarrow\; \frac{f^\tau(E_j)}{f^\tau(E_k)} \xrightarrow{\tau\to\infty} 0.$$

Now, the application of $F \coloneqq f^\tau(H)$ yields

$$\begin{aligned}
&F\,|\psi\rangle = F\sum_{j=k}^{d} \eta_j\,|j\rangle = \sum_{j=k}^{d} \eta_j f^\tau(E_j)\,|j\rangle \\
\Rightarrow\; &f^{-\tau}(E_k)F\,|\psi\rangle = \eta_k\,|k\rangle + \sum_{j=k+1}^{d} \eta_j \frac{f^\tau(E_j)}{f^\tau(E_k)}\,|j\rangle \xrightarrow{\tau\to\infty} \eta_k\,|k\rangle \\
\Rightarrow\; &f^{-2\tau}(E_k)\,\langle\psi|F^2|\psi\rangle = |\eta_k|^2 + \sum_{j=k+1}^{d} |\eta_j|^2 \left(\frac{f^\tau(E_j)}{f^\tau(E_k)}\right)^2 \xrightarrow{\tau\to\infty} |\eta_k|^2 \\
\Rightarrow\; &\lim_{\tau\to\infty} \frac{F\,|\psi\rangle}{\sqrt{\langle\psi|(F)|\psi\rangle}} = \lim_{\tau\to\infty} \frac{f^{-\tau}(E_k)F\,|\psi\rangle}{\sqrt{f^{-2\tau}(E_j)\,\langle\psi|F^2|\psi\rangle}} = \frac{\eta_k}{|\eta_k|^2}\,|k\rangle = \frac{\mathbb{P}\,|\psi\rangle}{\sqrt{\langle\psi|\mathbb{P}|\psi\rangle}}
\end{aligned}$$

as in this case $\mathbb{P} = |k\rangle\,\langle k|$. □

As already mentioned in Section 5.2, choosing $f = 1/\exp$ proves Lemma 5.2. Further examples for positive Hamiltonians H are $f = 1/\mathrm{id}$ and $f = -\ln$. The quantity $F = f^\tau(H)$ is called a *filtering operator*. Note that Amaro et al. demand that f is defined on the whole energy interval $[E_1, E_d]$ of H, and that f^2 should be strictly decreasing instead of f. However, it suffices to merely specify the function values of f on $\sigma(H)$. Furthermore, a non-negative f is strictly decreasing if and only if f^2 is strictly decreasing.

One can interpret Lemma 5.3 in the following two ways: a good approximation of a ground state can be conceived either by applying a filtering operator for a sufficiently long "time" τ or by repeatedly applying it for a moderate τ sufficiently often. As the filtering operator usually is not unitary, it cannot be implemented directly on a quantum computer; the F-VQE therefore utilizes a variational approach to approximate the application of filtering operators. Starting with an initial state $|\iota\rangle$ which has finite overlap with a ground state of H, further trial states are constructed by a PQC:

$$|\boldsymbol{\theta}\rangle \coloneqq V(\boldsymbol{\theta})\,|\iota\rangle \coloneqq U_S(\theta_S)\cdots U_1(\theta_1)\,|\iota\rangle\,. \tag{5.3.2}$$

Let $|\psi_{t-1}\rangle$ denote the state obtained after $t-1$ iterations. An application of a filtering operator F_t and appropriate normalization would produce the state

$$|F_t\psi_{t-1}\rangle \coloneqq \frac{F_t\,|\psi_{t-1}\rangle}{\sqrt{\langle\psi_{t-1}|F_t^2|\psi_{t-1}\rangle}} = \frac{F_t\,|\psi_{t-1}\rangle}{\sqrt{\langle F_t^2\rangle_{\psi_{t-1}}}}.$$

This state is now approximated by a state $|\psi_t\rangle = |\boldsymbol{\theta}_t\rangle$ by minimizing the Euclidean distance between both states

$$\begin{aligned}\mathcal{C}_t(\boldsymbol{\theta}) &\coloneqq \frac{1}{2}\big\||\boldsymbol{\theta}\rangle - |F_t\psi_{t-1}\rangle\big\|^2 \\ &= 1 - \frac{\mathrm{Re}(\langle\psi_{t-1}|F_t|\boldsymbol{\theta}\rangle)}{\sqrt{\langle F_t^2\rangle_{\psi_{t-1}}}}.\end{aligned} \tag{5.3.3}$$

Assuming each parameter corresponds to exactly one rotation gate, the partial derivatives of (5.3.3) are given by

$$\frac{\partial\mathcal{C}_t(\boldsymbol{\theta})}{\partial\theta_s} = -\frac{\mathrm{Re}(\langle\psi_{t-1}|F_t|\,|\boldsymbol{\theta}+\pi\boldsymbol{e}_s\rangle)}{2\sqrt{\langle F_t^2\rangle_{\psi_{t-1}}}}, \quad s\in[S]. \tag{5.3.4}$$

Invoking the parameter shift rule (cf. [Sch+19]), evaluating the partial derivatives at the old parameters yields

$$\frac{\partial\mathcal{C}_t(\boldsymbol{\theta})}{\partial\theta_s}\Big|_{\boldsymbol{\theta}_{t-1}} = -\frac{\langle F_t\rangle_{\psi_{t-1}^{s+}} - \langle F_t\rangle_{\psi_{t-1}^{s-}}}{4\sqrt{\langle F_t^2\rangle_{\psi_{t-1}}}} \tag{5.3.5}$$

with $|\psi_{t-1}^{s\pm}\rangle := |\boldsymbol{\theta}_{t-1} \pm \frac{\pi}{2}\boldsymbol{e}_s\rangle$, $s \in [S]$. These quantities are relatively easy to compute since all three states $|\psi_{t-1}\rangle$, $|\psi_{t-1}^{s\pm}\rangle$ are generated by the same PQC but with different parameters. (5.3.5) is then used to perform a single gradient-descent update for the parameters:

$$\boldsymbol{\theta}_t := \boldsymbol{\theta}_{t-1} - \eta \sum_{s=1}^{S} \frac{\partial \mathcal{C}_t(\boldsymbol{\theta})}{\partial \theta_s}\Big|_{\boldsymbol{\theta}_{t-1}} \boldsymbol{e}_s, \tag{5.3.6}$$

where $\eta > 0$ is the so-called learning rate. To prevent the gradient $\nabla \mathcal{C}_t(\boldsymbol{\theta})|_{\boldsymbol{\theta}_{t-1}}$ from vanishing, the "time" τ may be dynamically adapted. Hence the filtering operator may change in each iteration step and is thus labeled with a t.

Amaro et al. [Ama+21a] conducted a case study about VQA performances on a specific job shop scheduling problem instance which is not covered by our discussion in Chapter 4. They compare the performance of QAOA with quality $q = 2$, VQE, Var-QITE, and F-VQE on a 5-qubits instance with softcoded constraints. The convergence behavior improves drastically in ascending order of the algorithms mentioned: while the QAOA overall shows very poor performance, the VQE "converges" after 120 iterations, the Var-QITE after 20 iterations, and the F-VQE after 10-15 iterations.

Conclusion and Outlook

First and foremost, we have drawn a rigorous connection between the QAA, the QAOA, and the QAO. An essential part of this derivation is the concrete convergence proofs for all three algorithms. Utilizing a more general adiabatic theorem without a gap condition (Theorem 3.2), we could prove the convergence of the QAA with fixed initial Hamiltonian B and initial state $|+\rangle$ even for COPs with multiple optimal solutions (Theorem 3.6), thus extending the convergence discussion in [FGG14]. Furthermore, we have highlighted the importance of B being completely non-diagonal by proving its necessity for convergence in the generic case (Proposition 3.7). By invoking elementary matrix inequalities (Lemma B.1 and Lemma B.2), we have extended the convergence of the QAA to the QAOA (Theorem 3.8). Subsequently, we have retraced the generalization of the QAOA to the QAO but obtained a refined version of the QAO-mixing property (Definition 3.12 and Corollary 3.13) that allows for an analogous convergence proof (Theorem 3.17).

We wish to discuss several possible generalizations and open problems in this regard:
1. We did not address the rate of convergence for the universal QAA. Since this will highly depend on the usually unknown spectral properties of the objective Hamiltonian C, there will be nothing like a "universal" rate of convergence. However, the curve $\lambda_{\max}(t)$, $0 \leq t \leq 1$, corresponding to the instantaneous largest eigenvalues of the interpolating Hamiltonian $H_{\mathrm{lin}(B,C)}(t)$ always admits a peculiar behavior: it is non-degenerate except possibly at the endpoint $t = 1$. On the one hand, this might actually reduce the dependency of the convergence speed on C's spectral properties and, on the other hand, accelerates the convergence in comparison to cases with multiple level crossings. A more detailed application of adiabatic theorems to this exceptional case thus might provide additional insights.
2. We used the non-negativity of B and C as well as the complete non-diagonality of B to conclude that $\lambda_{\max}(t)$ is non-degenerate for each $0 \leq t < 1$, thereby invoking the Perron-Frobenius theorem. By Proposition 3.7, complete non-diagonality of B is necessary for convergence. It now stands to reason whether it is also sufficient, that is, whether one could drop the assumption $B, C \geq 0$ and still show convergence. For example, an equal spectral shift of both operators resulting in some negative or even

complex matrix elements should produce the same time evolution as the unshifted case and thus preserve the convergence. In the same spirit, it would be valuable to reformulate the proof of Theorem 3.6 for minimization problems. Any new insights could then be directly incorporated into the QAO-mixing property.
3. In the proof of Lemma B.2, we have explicitly constructed finitely many angles to approximate the quasi-adiabatic time evolution $U_{\mathrm{T}}(t)$ with QAOA-gates $e^{-i\beta B}$ and $e^{-i\gamma C}$. A more refined version of the proof respecting explicit error bounds w.r.t. the properties of B and C as well as the chosen time scale $T > 0$ would help assess the QAOA's ability to approximate the QAA.
4. The matrix properties of B and C are always w.r.t. the CB, simply due to the encoding of the classical objective function into an operator diagonal in the CB. However, as discussed in Chapter 5, simulating quantum systems generally leads to objective Hamiltonians with off-diagonal entries in the CB. Diagonalizing this Hamiltonian then entails a different orthonormal basis $\mathcal{B}'$ of $\mathbb{q}^{\otimes N}$, and all matrix properties of B and C now must be valid w.r.t. $\mathcal{B}'$. Any knowledge about the structure of $\mathcal{B}'$ can be used to construct suitable initial/mixer Hamiltonians for those non-classical problems.

Furthermore, we have detailed the translation of group actions on bit strings to operators acting on the qubit space (Section 2.4). We also provided an additional perspective on this issue with the constraint graph model. Both approaches allow for an alternative characterization of the refined QAO-mixing property in terms of the properties of the problem instance (Theorem 4.7): a family of operators fulfills the QAO-mixing property if and only if their classical analogs act transitively on the set of feasible bit strings. In this context, we have studied permutations of bit string values as possible mixers for scheduling-type problems. For free job-shop scheduling problems, we have shown that the feasibility-preserving permutations admit the QAO-mixing property (Corollary 4.10). For the more complex flexible job-shop scheduling problems, we have provided one counterexample (Fig. 4.8) and one example (Fig. 4.10) for feasibility-preserving permutations fulfilling the QAO-mixing property.

This new formalism may now be applied to numerous problem instances. Besides, there are still some open questions regarding the abstract constraint graph model:
5. There are countless notions related to group theory, which could also be transferred to operators acting on the qubit space. An overview of the quantum mechanical correspondences of the classical group action concepts, including their interpretation, would, in any case, be informative.
6. By Proposition 4.5, every automorphism of a given constraint graph is feasibility-preserving, but the converse is not generally true. For example, we have shown that $\mathrm{Aut}(G) \subsetneq \mathcal{F}$ for the graphs drawn in Fig. 4.2 and Fig. 4.9. A crucial observation is that $\mathrm{Aut}(G)$ already acts transitively on the feasible solutions $\mathcal{S}$ in both cases. Thus, $\mathrm{Aut}(G) \setminus \mathcal{F}$ does not "increase the mixing property" of $\mathcal{F}$. The question of whether $\mathrm{Aut}(G)$ acts transitively on $\mathcal{S}$ if and only if $\mathcal{F}$ does is still unanswered.

7. With the introduction of the constraint graph model, we have merely scratched the surface of graph theory. A primary goal would be to characterize further the constraint graphs of, e.g., free and flexible job-shop scheduling problems, thereby extracting the actual graph properties necessary and sufficient for transitively acting $\mathcal{F}$.
8. For the general free job-shop scheduling problem, we constructed a proper subgroup of $\mathrm{Aut}(G) = \mathcal{F}$ via (4.3.6), which already acts transitively on $\mathcal{S}$. As for this problem, the whole group $\mathcal{F}$ might generally be unnecessarily large. Thus, the task will be to identify minimal subgroups $\mathcal{G} \subseteq \mathcal{F}$ that still act transitively on $\mathcal{S}$. Another possible quality measure is the implementability of the operator analogs of elements of $\mathcal{G}$ and, particularly, of its generators.
9. We have considered $\mathrm{JSP}(1, 2, 2, 2)$ and $\mathrm{JSP}(1, 3, 2, 2)$, and already found very different behavior of $\mathcal{F}$ in these two cases. The computation of $\mathcal{F}$ for a general $\mathrm{JSP}(J, O, M, T)$ instance is undoubtedly indispensable for further investigations of the underlying structure.

Appendices

A. Basic Concepts of Graph Theory

In this appendix, we present the basic concepts of graph theory necessary to characterize completely non-diagonal matrices (cf. Definition 3.4) and to analyze the constraint graph model (Chapter 4). In this regard, we mainly follow [Die17].

For a given set A and natural number $k \in \mathbb{N}$, let $[A]^k$ denote the set of all k-element subsets (k-subsets) of A.

Definition A.1. A *graph* is a pair $G = (V, E)$ such that $E \subseteq [V]^2$ and $V \cap E = \emptyset$. $V(G) \coloneqq V$ is the *vertex set* of G, and $E(G) \coloneqq E$ is its *edge set*. Furthermore, set $|G| \coloneqq |V(G)|$ (*order* of G) and $\|G\| \coloneqq |E(G)|$.

Definition A.2. A *directed graph* is a pair $G = (V, E)$ such that $E \subseteq V \times V$. All other notions are identical to those from Definition A.1.

Definition A.3. A (*directed*) *network* is a pair $G = (H, f)$ such that H is a (directed) graph, and $f : E(H) \mapsto \mathbb{C} \setminus \{0\}$ is the *weight function*. All other notions are identical to those from Definition A.1.

Note that a graph in the sense of Definition A.1 is often called an undirected graph. However, since we mainly consider undirected graphs, we call them graphs right away, even though the concept of directed graphs is more universal since any graph $G = (V, E)$ can be transformed into a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ via

$$\tilde{V} \coloneqq V, \quad \tilde{E} \coloneqq \bigcup_{\{v,w\} \in E} \{(v, w), (w, v)\}. \tag{A.1}$$

Moreover, any (directed) graph can be conceived as a (directed) network with weight function $f \equiv 1$. This distinguishes some directed graphs and networks that are, in a sense, undirected.

Definition A.4. A directed graph H is *symmetric* if there exists a graph G so that $H = \tilde{G}$ in the sense of (A.1). A directed network (H, f) is *duplex* if H is symmetric. It is further *symmetric* if, in addition, $f((v, w)) = f((w, v))$ for all $(v, w) \in E(H)$.

The following definitions have a more general version valid for directed graphs or even networks, but we will only state them if necessary. In the following, let $G = (V, E)$ denote an arbitrary graph.

Definition A.5. A vertex $v \in V$ is *incident* with an edge $e \in E$ if $v \in e$. The two vertices incident with an edge are its *ends*. Moreover, $E(v) \coloneqq \{e \in E : v \in e\}$ denotes the set of all edges in E the vertex v is incident with.

Since an edge $e \in E$ is uniquely determined by its ends, we may also write $vw \coloneqq wv \coloneqq e$ if $v, w \in V$ are the ends of e.

Definition A.6. Two vertices $v, w \in W$ are *adjacent* or *neighbors* if $vw \in E$. Two edges $e, f \in E$ are *adjacent* if they have exactly one end in common.

The notion of adjacency produces two special cases:

Definition A.7. If all vertices $v \in V$ are pairwise adjacent, G is *complete* and is denoted by $K^{|G|}$.

Definition A.8. A subset of vertices $U \subseteq V$ or a subset of edges $F \subseteq E$ is *independent* if no two of its elements are adjacent.

Often, the neighbors of a specified vertex are of particular interest.

Definition A.9. $\operatorname{nbhd}(v) \coloneqq \operatorname{nbhd}_G(v) \coloneqq \{w \in V : vw \in E\}$ is the *neighborhood* of the vertex $v \in V$. More generally, the *neighborhood* of $U \subseteq V$ is given by

$$\operatorname{nbhd}(U) \coloneqq \operatorname{nbhd}_G(U) \coloneqq \left(\bigcup_{v \in U} \operatorname{nbhd}(v) \right) \setminus U.$$

Mappings between graphs are also of significant importance. Since the structure of a graph is fully determined by its edges, homomorphisms are defined so that they preserve adjacency.

Definition A.10. Let $G' = (V', E')$ be another graph. A map $\varphi : V \to V'$ is an *homomorphism* from G to G' if $vw \in E$ implies that $\varphi(v)\varphi(w) \in E'$. If φ is also bijective and its inverse φ^{-1} is also a homomorphism, φ is an *isomorphism* between G and G'. In this case, G and G' are *isomorphic*, $G \cong G'$. An isomorphism between G and G is an *automorphism* of G, and $\operatorname{Aut}(G)$ denotes the set of all automorphisms of G.

A natural question that arises is how to characterize isomorphisms further. Clearly, bijectivity is inevitable. However, the following example shows that this does not imply that the edge sets are also in bijection: consider the graphs $G_1 = (\{1,2\}, \emptyset)$ and $G_2 = (\{1,2\}, \{12\})$. Then the identity is a bijective homomorphism between G_1 and G_2, but $\|G_1\| \neq \|G_2\|$ holds. In general, we obtain that injective homomorphisms do not decrease the amount of edges.

Lemma A.11. *Let φ be an injective homomorphism between two graphs G and G'. Then $\|G\| \leq \|G'\|$.*

Proof. Define $\widetilde{\varphi} : E(G) \to E(G')$ via $\widetilde{\varphi}(vw) \coloneqq \varphi(v)\varphi(w)$. Since φ is a homomorphism, $\widetilde{\varphi}$ is well-defined. Let $v_1w_1, v_2w_2 \in E(G)$ with

$$\varphi(v_1)\varphi(w_1) = \widetilde{\varphi}(v_1w_1) = \widetilde{\varphi}(v_2w_2) = \varphi(v_2)\varphi(w_2),$$

Since φ is injective, it must hold that $v_1 = v_2$ and $w_1 = w_2$ or $v_1 = w_2$ and $w_1 = v_2$. Since $vw = wv \in E(G)$, one obtains that $\widetilde{\varphi}$ is injective, hence $\|G\| \leq \|G'\|$. □

Proposition A.12. *A bijective homomorphism φ between two graphs G and G' is an isomorphism if and only if $\|G\| = \|G'\|$ holds.*

Proof. Assume that $\|G\| = \|G'\|$. Consider again $\widetilde{\varphi} : E(G) \to E(G')$ from Lemma A.11. Since $\widetilde{\varphi}$ is injective and $|E(G)| = \|G\| = \|G'\| = |E(G')|$ holds, $\widetilde{\varphi}$ is already bijective with inverse $\widetilde{\varphi}^{-1} : E(G') \to E(G)$. Let $v'w' \in E(G')$. Since $\varphi : V(G) \to V(G')$ is surjective, there exist $v, w \in V(G)$ such that $\varphi(v) = v'$ and $\varphi(w) = w'$. Furthermore, there exist $x, y \in V(G)$ such that $E(G) \ni \widetilde{\varphi}^{-1}(v'w') = xy$. But since

$$\varphi(v)\varphi(w) = v'w' = \widetilde{\varphi}(xy) = \varphi(x)\varphi(y)$$

holds, one concludes, invoking the injectivity of φ, that (w.l.o.g.) $v = x$ and $w = y$. Therefore, $vw = \varphi^{-1}(v')\varphi^{-1}(w') \in E(G)$. Hence φ^{-1} is an homomorphism, and φ is thus an isomorphism.
Conversely, if φ is an isomorphism, φ is an homomorphism from G to G', and φ^{-1} is an homomorphism from G' to G. Thus, by Lemma A.11, $\|G\| \leq \|G'\|$ and $\|G'\| \leq \|G\|$ hold, i.e. $\|G\| = \|G'\|$. □

In fact, for the bijection $\widetilde{\varphi} : E(G) \to E(G')$ induced by an isomorphism φ between G and G', it holds that

$$\widetilde{\varphi^{-1}}\big(\widetilde{\varphi}(vw)\big) = \widetilde{\varphi^{-1}}\big(\varphi(v)\varphi(w)\big) = vw \qquad \text{and}$$
$$\widetilde{\varphi}\big(\widetilde{\varphi^{-1}}(v'w')\big) = \widetilde{\varphi}\big(\varphi^{-1}(v')\varphi^{-1}(w')\big) = v'w'$$

for all $vw \in E(G)$ and $v'w' \in E(G')$, i.e. $\widetilde{\varphi}^{-1} = \widetilde{\varphi^{-1}}$.

An isomorphism might preserve additional structure. There are two basic notions describing this effect.

Definition A.13. A class of graphs that is closed under isomorphisms is a *graph property*.

Definition A.14. A map taking graphs as arguments is a *graph invariant* if it is constant among isomorphic graphs.

The fact that it necessarily holds that $|G| = |G'|$ and $\|G\| = \|G'\|$ whether G and G' are isomorphic can be expressed in either way: $\{G : |G| = m, \|G\| = n\}$ is a graph property for any $m, n \in \mathbb{N}_0$, or equivalently, $[G \mapsto (|G|, \|G\|)]$ is a graph invariant.

Next, we will discuss obtaining new graphs from given ones. In the following, let $G = (V, E)$ and $G' = (V', E')$ denote two arbitrary graphs.

Definition A.15. The *union* and *intersection* of G and G' are $G \cup G' \coloneqq (V \cup V', E \cup E')$ and $G \cap G' \coloneqq (V \cap V', E \cap E')$, respectively.

Definition A.16. If $V' \subseteq V$ and $E' \subseteq E$, then G' is a *subgraph* of G, $G' \subseteq G$. If, in addition,…

(i) …$G' \neq G$, G' is a *proper* subgraph of G.

(ii) …$v, w \in V'$ with $vw \in E$ implies that $vw \in E'$, G' is an *induced* subgraph of G.

(iii) …$V' = V$, G' is a *spanning* subgraph of G.

Given an arbitrary subset of vertices $U \subseteq V$, we may also consider certain subgraphs of G "built" from U.

Definition A.17. The *closure* of $U \subseteq V$ (in G) is $\overline{U} \coloneqq \overline{U}^G \coloneqq U \cup \mathrm{nbhd}_G(U)$. If $\overline{U} = U$, U is *closed* or *neighborless* in G. If $\overline{U} = V$, U is *dense* in G.

Definition A.18. For $U \subseteq V$, the graph *spanned* by U in G is

$$G[U] \coloneqq \left(U, \bigcup\{vw \in E : v, w \in U\}\right),$$

and $E_G[U] \coloneqq E(G[U])$ is the edges *spanned* by U in G.

We introduce the following concepts for directed networks to cover the characterization of completely non-diagonal matrices. Therefore, let $G = (V, E, f)$ denote an arbitrary directed network.

Definition A.19. For a prespecified ordering $V = \{v_1, \dots, v_n\}$, the *adjacency matrix* of G is defined as $A = (a_{ij})_{n\times n}$ with

$$a_{ij} \coloneqq \begin{cases} f(e), & \text{if } (v_i, v_j) = e \in E \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

Note that the adjacency matrix A of G depends on the ordering of the vertices. Therefore, it is merely unique up to permutations of the vertices. Conversely, A contains all structural information about G. Thus, we say $G_A \coloneqq G$ is the directed network with adjacency matrix A. An immediate consequence of the definition is that G_A is duplex if and only if $(a_{ij} \neq 0 \;\Leftrightarrow\; a_{ji} \neq 0)$. Furthermore, G_A is symmetric if and only if A is symmetric. Furthermore, the adjacency matrix is the door opener into the field of linear algebraic graph theory, which we do not pursue here.

Next, we examine the connectivity properties of graphs and networks and their relation to the adjacency matrix. This requires first the notion of (directed) paths.

Definition A.20. A *path* is a non-empty graph $P = (V, E)$ of the form

$$V = \{x_1, \dots, x_{k+1}\}, \quad E = \{x_1x_2,\, x_2x_3, \dots,\, x_kx_{k+1}\}.$$

The vertices x_1 and x_{k+1} are the *ends* of the path P, $x_2, \dots, x_k$ are its *inner vertices*. The *length* of P is given by $\|E\|$, and the path of length $k \in \mathbb{N}_0$ is denoted by P^k.

Definition A.21. A *directed path* is a non-empty directed graph $P = (V, E)$ of the form

$$V = \{x_1, \dots, x_{k+1}\}, \quad E = \{(x_1, x_2), (x_2, x_3), \dots, (x_k, x_{k+1})\}.$$

x_1 is the *tail* of P, while x_{k+1} is its *head*, and P is a path *from* x_1 *to* x_{k+1}.

Definition A.22. A graph G is connected if it is non-empty and for any pair of vertices $v, w \in V(G)$, there exists a path in G with ends v and w.

We can further characterize connectedness of a graph in terms of its closed vertex subsets, which fits more naturally in our discussion in Section 3.3.

Proposition A.23. *A non-empty graph G is connected if and only if it has no non-trivial closed vertex subsets.*

Proof. Let G be connected and let $U \subset V(G)$ be a non-trivial subset of vertices. Since U is non-trivial, there exist $v \in U$ and $w \in V(G) \setminus U$, and connectivity of G implies the existence of a path P with ends v and w. Then there necessarily exists an edge $e \in E(P) \setminus E_G[U]$. Espescially, $\mathrm{nbhd}_G(U) \neq \emptyset$ holds, i.e. U is not closed in G.
Conversely, let G have no non-trivial closed vertex subsets, and let $v, w \in V(G)$, be arbitrary. A path between v and w is built inductively: set $x_1 \coloneqq v$. For $i \in \mathbb{N}$, assume a path $P_i = (U_i, F_i)$ with $w \notin U_i = \{x_1, \dots, x_i\}$ and $F_i = \{x_1x_2, \dots, x_{i-1}x_i\}$ has already been constructed. If possible, select a vertex $x_{i+1} \in \mathrm{nbhd}_G(U_i) \setminus \mathrm{nbhd}_G(U_{i-1})$ and set $U_{i+1} = U_i \cup \{x_i\}$, $F_{i+1} = F_i \cup \{x_ix_{i+1}\}$, and $P_{i+1} = (U_{i+1}, F_{i+1})$. Then x_{i+1} necessarily is a neighbor of x_i, hence P_{i+1} is again a path. If $x_{i+1} = w$, P_{i+1} is thus a path between v and w. If, on the other hand,

$$\mathrm{nbhd}_G(U_i) \setminus \mathrm{nbhd}_G(U_{i-1}) = \emptyset, \tag{A.3}$$

reduce i until $|\mathrm{nbhd}_G(U_i) \setminus \mathrm{nbhd}_G(U_{i-1})| > 1$ holds and select a different vertex x'_{i+1} to form U_{i+1} and F_{i+1}. This is always possible because $w \notin U_i$, i.e. U_i is non-trivial and therefore $\mathrm{nbhd}_G(U_i) \neq \emptyset$ by assumption; thus (A.3) implies that there exists an $i' < i$ with

$$\mathrm{nbhd}_G(U_{i'}) \cap \mathrm{nbhd}_G(U_i) \neq \emptyset. \tag{A.4}$$

But since $\mathrm{nbhd}_G(U_i) \not\ni x_{i'+1} \in \mathrm{nbhd}_G(U_{i'})$, one has that $|\mathrm{nbhd}_G(U_{i'})| > 1$. Now select the smallest $i' \in \mathbb{N}$ so that (A.4) is fulfilled, then $|\mathrm{nbhd}_G(U_{i'}) \setminus \mathrm{nbhd}_G(U_{i'-1})| > 1$. Therefore, the iteration can be repeated until w is reached. □

Definition A.22 can be applied to directed graphs as well: for a directed graph $H = (W, F)$, we define its *underlying* graph $\mathrm{u}H = (V, E)$ via

$$V \coloneqq W, \quad E \coloneqq \{\{v, w\} : (v, w) \in F \text{ or } (w, v) \in F\}. \tag{A.5}$$

Now, connectivity properties of H are attributed to the underlying graph.

Definition A.24. A directed network (H, f) is (*weakly*) *connected* if $\mathrm{u}H$ is connected.

Definition A.25. A directed network (H, f) is *strongly connected* if it is non-empty and it contains a directed path from v to w and a directed path from w to v for every pair of vertices $v, w \in V(H)$.

Every strongly connected directed network is weakly connected but not necessarily vice versa. However, for duplex directed networks, both notions are equivalent.

B. Convergence Proofs

Here, we provide a proof of the convergence of the QAA with initial Hamiltonian (3.1.4) and an arbitrary objective Hamiltonian C.

Proof of Theorem 3.6. Denote by $\lambda_{\max}(t)$ the largest eigenvalue of $H_{\mathrm{lin}(B,C)}(t)$, for $0 \leq t \leq 1$, respectively. Let $0 \leq t_0 < 1$. Since B is completely non-diagonal, so is $(1-t_0)B$. As C is diagonal, also $H_{\mathrm{lin}(B,C)}(t_0) = (1-t_0)B + t_0 C$ is completely non-diagonal by Corollary 3.5. In addition, B as well as C are non-negative. In summary, $H_{\mathrm{lin}(B,C)}(t_0)$ is non-negative and completely non-diagonal. According to the Perron-Frobenius theorem (cf. [Fro12]), $\lambda_{\max}(t_0)$ is non-degenerate.
Furthermore, the mapping

$$H_{\mathrm{lin}(B,C)}(\,\cdot\,) : \mathbb{R} \to \mathcal{L}(\mathbb{q}^{\otimes N}), \quad t \mapsto H_{\mathrm{lin}(B,C)}(t) = (1-t)B + tC \tag{B.1}$$

is analytic and $H_{\mathrm{lin}(B,C)}(t)$ is symmetric for all $t \in \mathbb{R}$. Let L denote the discrete set of level crossings/eigenvalue splittings of $H(\,\cdot\,)$. According to [Kat95, Theorem 6.1], the instantaneous eigenvalues of $H_{\mathrm{lin}(B,C)}(t)$, $t \in \mathbb{R}$, can be sorted as $\{\lambda_m(t) : 1 \leq m \leq M\}$, $M \leq 2^N$, such that $[t \mapsto \lambda_m(t)] \in C^\omega(\mathbb{R}, \mathbb{R})$ and for the corresponding spectral projections $\mathbb{P}_m(t)$, it holds that $[t \mapsto \mathbb{P}_m(t)] \in \mathbb{C}^\omega\Big(\mathbb{R} \setminus L, \mathcal{L}(\mathbb{q}^{\otimes N})\Big)$, for every $1 \leq m \leq M$. Furthermore, the spectral projections have removable singularities in L, i.e. there exist analytic continuations P_m, defined on whole $\mathbb{R}$, such that $P_m(t) = \mathbb{P}_m(t)$ for $t \in \mathbb{R} \setminus L$, for all $1 \leq m \leq M$. By continuity, these continuations are themselves orthogonal projections with constant rank and fulfill

$$H_{\mathrm{lin}(B,C)}(t) P_m(t) = \lambda_m(t) P_m(t)$$

for all $t \in \mathbb{R}$. W.l.o.g. assume $\lambda_1(0) = \lambda_{\max}(0)$. Since $\lambda_{\max}(t_0)$ remains non-degenerate for $0 \leq t_0 < 1$, it follows that $\lambda_1 \equiv \lambda_{\max}$ on $[0,1)$ and by continuity of λ_1 that $\lambda_1 \equiv \lambda_{\max}$ on $[0,1]$. In addition, the corresponding spectral projection $\mathbb{P}_1$ is well-defined on $[0,1)$. Therefore, its continuation P_1 fulfills all properties necessary to apply Theorem 3.2, i.e. (3.1.8) holds. Since $P_1(0) = \mathbb{P}_1(0)$, one especially obtains that

$$\begin{aligned} & 0 = \lim_{T\to\infty} (\mathbb{1} - P_1(1)) U_T(1) P_1(0) \,|+\rangle = \lim_{T\to\infty} (\mathbb{1} - P_1(1)) U_T(1) \,|+\rangle \\ \Leftrightarrow \quad & \lim_{T\to\infty} U_T(1) \,|+\rangle = P_1(1) \lim_{T\to\infty} U_T(1) \,|+\rangle \,. \end{aligned}$$

Since $P_1(1)$ is a projection with $H_{\mathrm{lin}(B,C)}(1) P_1(1) = \lambda_{\max}(1) P_1(1)$, one concludes (3.1.9). □

In the following, we prove that complete non-diagonality of the initial Hamiltonian is indeed a necessary property for the convergence in the generic case.

Proof of Proposition 3.7. Let $\mathcal{R}$ denote a proper invariant coordinate subspace of H_{I}. The mutual orthogonality of the CB states yields the decomposition

$$\mathbb{q}^{\otimes N} = \mathcal{R} \oplus \mathcal{R}^{\perp},$$

where $\mathcal{R}^{\perp}$ is again a coordinate subspace. Accordingly, write $|\iota\rangle = |\iota_1\rangle + |\iota_2\rangle \in \mathcal{R} \oplus \mathcal{R}^{\perp}$. Consider an objective Hamiltonian C that also leaves $\mathcal{R}$ invariant. Since H_{I} and C are self-adjoint, they also have $\mathcal{R}^{\perp}$ as an invariant subspace. Consequently, $\mathcal{R}$ as well as $\mathcal{R}^{\perp}$ are invariant subspaces of $H_{\mathrm{lin}(H_{\mathrm{I}},C)}(t)$ for all $0 \leq t \leq 1$. Solving the time evolution equation

$$\frac{\mathrm{d}}{\mathrm{d}s}\tilde{U}_T(s) = -iH_{\mathrm{lin}(H_{\mathrm{I}},C)}(s/T)\tilde{U}_T(s), \quad 0 \leq s \leq T; \quad \tilde{U}_T(0) = \mathbb{1}$$

restricted to $\mathcal{R}$ and $\mathcal{R}^{\perp}$, respectively, further implies that both $\mathcal{R}$ and $\mathcal{R}^{\perp}$ are invariant subspace of $\tilde{U}_T(s) = \tilde{U}_T(s)|_{\mathcal{R}} + \tilde{U}_T(s)|_{\mathcal{R}^{\perp}}$ for all $T > 0$ and $0 \leq s \leq T$. This eventually yields that

$$\lim_{T\to\infty} U_T(1)\,|\iota_1\rangle \in \mathcal{R} \quad \text{and} \quad \lim_{T\to\infty} U_T(1)\,|\iota_2\rangle \in \mathcal{R}^{\perp}.$$

If $|\iota_1\rangle = 0$, choose C such that $S_{\mathrm{opt}} \subseteq \mathcal{R}$, then

$$\lim_{T\to\infty} U_T(1)\,|\iota\rangle = \lim_{T\to\infty} U_T(1)\,|\iota_2\rangle \notin \mathcal{R} \supseteq S_{\mathrm{opt}}.$$

If instead $|\iota_1\rangle \neq 0$, choose C such that $S_{\mathrm{opt}} \subseteq \mathcal{R}^{\perp}$, then

$$\lim_{T\to\infty} U_T(1)\,|\iota_1\rangle \in \mathcal{R} \setminus \{0\} \;\Rightarrow\; \lim_{T\to\infty} U_T(1)\,|\iota_1\rangle + \lim_{T\to\infty} U_T(1)\,|\iota_2\rangle \notin \mathcal{R}^{\perp} \supseteq S_{\mathrm{opt}}.$$

□

Built on the convergence of QAA, we show in the following that the QAOA also converges. This requires first a rather technical result.

Lemma B.1. *For $\varepsilon > 0$ and $m \in \mathbb{N}$, let $\{V_j\}_{j=1}^m, \{W_j\}_{j=1}^m, \subset \mathcal{L}(\Xi)$ be families of unitary operators so that*

$$\|V_j - W_j\| < \varepsilon \tag{B.2}$$

holds for all $j \in [m]$. Then the following estimate is valid:

$$\left\| \prod_{j=1}^{m} V_j - \prod_{j=1}^{m} W_j \right\| < (1+\varepsilon)^m - 1. \tag{B.3}$$

Proof. Since (B.2) holds, one can find linear operators $R_j \in \mathcal{L}(\Xi)$ with $\|R_j\| \leq 1$ and $V_j = W_j + \varepsilon R_j$ for each $j \in [m]$, respectively. (B.3) clearly holds for $m = 1$. Therefore, it remains to show that if (B.3) holds for an $m \in \mathbb{N}$, then it also holds for $m+1$:

$$\begin{aligned}
\left\| \prod_{j=1}^{m+1} V_j - \prod_{j=1}^{m+1} W_j \right\| &= \left\| \left(\prod_{j=1}^{m} V_j \right) V_{m+1} - \left(\prod_{j=1}^{m} W_j \right) W_{m+1} \right\| \\
&= \left\| \left(\prod_{j=1}^{m} V_j \right) (W_{m+1} + \varepsilon R_{m+1}) - \left(\prod_{j=1}^{m} W_j \right) W_{m+1} \right\| \\
&= \left\| \left(\prod_{j=1}^{m} V_j \right) (\mathbb{1} - \varepsilon R_{m+1} W_{m+1}^*) - \left(\prod_{j=1}^{m} W_j \right) \right\| \\
&\leq \left\| \prod_{j=1}^{m} V_j - \prod_{j=1}^{m} W_j \right\| + \left\| \left(\prod_{j=1}^{m} V_j \right) \varepsilon R_{m+1} W_{m+1}^* \right\| \\
&< (1+\varepsilon)^m - 1 + \varepsilon \\
&\leq (1+\varepsilon)^m - 1 + \varepsilon(1+\varepsilon)^m = (1+\varepsilon)^{m+1} - 1.
\end{aligned}$$

□

The next lemma shows that the quasi-adiabatic evolution can be approximated to arbitrary precision (in matrix norm) by QAOA gates.

Lemma B.2. *For all $T > 0$, $t \in [0,1]$, and $\delta > 0$, there exist finitely many angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ such that*

$$\|V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(t)\| < \delta.$$

Proof. Fix $T > 0$, $t \in [0,1]$, and $\delta > 0$. Then there exists an $m \in \mathbb{N}$ so that

$$\left\| \prod_{j=1}^{m} e^{-iH_{\mathrm{lin}(B,C)}\left(j\frac{tT}{m}\right) j\frac{tT}{m}} - U_T(t) \right\| < \frac{\delta}{2}. \tag{B.4}$$

In the following, set

$$W_j \coloneqq e^{-iH_{\mathrm{lin}(B,C)}\left(j\frac{tT}{m}\right) j\frac{tT}{m}}.$$

The Lie product formula implies that for all $j \in [m]$, there exist $n_j \in \mathbb{N}$ such that for all $\tilde{n} \geq n_j$ it holds that

$$\left\| \left(e^{-i\frac{1-j\frac{tT}{m}}{\tilde{n}} j\frac{tT}{m} B} e^{-i\frac{\left(j\frac{tT}{m}\right)^2}{\tilde{n}} C} \right)^{\tilde{n}} - W_j \right\| < \sqrt[m]{\frac{\delta}{2} + 1} - 1, \tag{B.5}$$

respectively. Taking $n \coloneqq \max\{n_j : j \in [m]\}$, this estimate holds for all $j \in [m]$ and (especially) $\tilde{n} = n$. Now, choose $q = n^m$ angles $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m)$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_m)$ as

$$\begin{aligned}\left(\boldsymbol{\beta}_j\right)_k &\equiv \frac{1 - j\frac{tT}{m}}{n} j\frac{tT}{m} \mod \pi \\ \left(\boldsymbol{\gamma}_j\right)_k &\equiv \frac{\left(j\frac{tT}{m}\right)^2}{n} \mod 2\pi\end{aligned}$$

for all $k \in [n]$ and all $j \in [m]$. Then, by construction, (B.5) translates into

$$\left\|V(\boldsymbol{\beta}_j, \boldsymbol{\gamma}_j) - W_j\right\| < \sqrt[m]{\frac{\delta}{2} + 1} - 1.$$

Thus, by (B.4) and Lemma B.1, it follows that

$$\begin{aligned}\|V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(t)\| &= \left\|\prod_{j=1}^{m} V(\boldsymbol{\beta}_j, \boldsymbol{\gamma}_j) - U_T(t)\right\| \\ &\leq \left\|\prod_{j=1}^{m} V(\boldsymbol{\beta}_j, \boldsymbol{\gamma}_j) - \prod_{j=1}^{m} W_j\right\| + \left\|\prod_{j=1}^{m} W_j - U_T(t)\right\| \\ &< \frac{\delta}{2} + \frac{\delta}{2} = \delta.\end{aligned}$$

□

This suffices to prove the convergence of QAOA.

Proof of Theorem 3.8. Let $\varepsilon > 0$. According to the proof of Theorem 3.6, there exists a $T > 0$ such that

$$\|(\mathbb{1} - P_1(1))U_T(1)\,|+\rangle\| < \frac{\varepsilon}{2}.$$

Since $\dim(\mathbb{q}^{\otimes N}) > 1$ and $P_1(1)$ has rank one by continuity, $\alpha \coloneqq \|\mathbb{1} - P_1(1)\| > 0$. Furthermore, by Lemma B.2, one can choose (for $t = 1$ and $\delta = \varepsilon/(2\alpha)$) finitely many angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ such that

$$\|V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(1)\| < \frac{\varepsilon}{2\alpha}.$$

Then it follows that

$$\begin{aligned}\|(\mathbb{1} - P_1(1))\,|\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle\| &= \|(\mathbb{1} - P_1(1))V(\boldsymbol{\beta}, \boldsymbol{\gamma})\,|+\rangle\| \\ &\leq \|(\mathbb{1} - P_1(1))(V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(1))\,|+\rangle\| + \|(\mathbb{1} - P_1(1))U_T(1)\,|+\rangle\| \\ &< \|\mathbb{1} - P_1(1)\|\frac{\varepsilon}{2\alpha} + \frac{\varepsilon}{2} = \varepsilon.\end{aligned}$$

Since $\mathrm{im}(P_1(1)) \subseteq S_{\mathrm{opt}}$, one obtains the assertion. □

The convergence proof for the QAOA can now be extended to QAO instances.

Proof of Theorem 3.17. Since $\{H_i\}_{i\in I}$ is COP-mixing, $H_\mathrm{I} \in \mathcal{L}(S)$ is non-negative and completely non-diagonal. In addition, $H_\mathrm{F} \coloneqq H_\mathrm{P}|_S \in \mathcal{L}(S)$ is diagonal and non-negative. Therefore, the analytic mapping

$$H_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}(\,\cdot\,) : \mathbb{R} \to \mathcal{L}(S), \quad t \mapsto H_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}(t) \coloneqq (1-t)H_\mathrm{I} + tH_\mathrm{F} \tag{B.6}$$

provides symmetric matrices $H_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}(t)$, $t \in \mathbb{R}$, with $H_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}(t_0)$ also being non-negative and completely non-diagonal for $0 \leq t_0 < 1$. For $T > 0$, let U_T denote the quasi-adiabatic evolution w.r.t. to

$$H_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}(t), \quad 0 \leq t \leq 1.$$

Let $\varepsilon > 0$. An analgous reasoning as in the proof of Theorem 3.6 then shows that there exists a $T > 0$ so that

$$\|(\mathbb{1} - P_1(1))U_T(1)\,|\iota\rangle\| < \frac{\varepsilon}{2}$$

holds. As in the proof of Theorem 3.6, $P_1(\,\cdot\,)$ denotes the analytic continuation of the spectral projection corresponding to the largest eigenvalue curve (on $[0,1]$). W.l.o.g. assume that $\dim(S) > 1$. Otherwise the statement would be trivial as $\dim(S_\mathrm{opt}) \neq 0$ then implies that $|\iota\rangle \in S = S_\mathrm{opt}$. Thus, w.l.o.g. $\alpha \coloneqq \|\mathbb{1} - P_1(1)\|_{\mathcal{L}(S)} > 0$ since $P_1(1)$ has rank one be continuity.

Simultaneous mixer: Assume that $\sigma = 0$. Invoking Lemma B.1, one can utilize the definition of the time evolution operator (compare (B.4)) and the Lie product formula (compare (B.5)) to conclude the existence of fininitely many angles $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ such that

$$\|V(\boldsymbol{\beta},\boldsymbol{\gamma}) - U_T(1)\|_{\mathcal{L}(S)} < \frac{\varepsilon}{2\alpha}$$

holds.

Sequential mixer: W.l.o.g. assume that $\sigma = \mathrm{id}_I$. By the definition of the time evolution operator, there exists an $m \in \mathbb{N}$ so that

$$\left\| \prod_{j=1}^{m} e^{-iH_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}\left(j\frac{T}{m}\right)j\frac{T}{m}} - U_T(1) \right\|_{\mathcal{L}(S)} < \frac{\varepsilon}{4\alpha} \tag{B.7}$$

holds. In the following, set

$$W_j \coloneqq e^{-iH_{\mathrm{lin}(H_\mathrm{I},H_\mathrm{F})}\left(j\frac{T}{m}\right)j\frac{T}{m}}.$$

The multivariate Lie product formula [Bha97, Problem IX.8.5] imples that for all $j \in [m]$, there exist $n_j \in \mathbb{N}$ so that for all $\tilde{n} \geq n_j$ it holds that

$$\left\| \left(\left(\prod_{i \in I} e^{-i \frac{\left(1 - j\frac{T}{m}\right)}{\tilde{n}} j \frac{T}{m} H_i|_S} \right) e^{-i \frac{\left(j\frac{T}{m}\right)^2}{\tilde{n}} H_{\mathrm{F}}} \right)^{\tilde{n}} - W_j \right\|_{\mathcal{L}(S)} < \sqrt[m]{\frac{\varepsilon}{4\alpha} + 1} - 1, \tag{B.8}$$

respectively. Choosing the same angles as in the proof of Lemma B.2, one eventually obtains that

$$\|V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(t)\|_{\mathcal{L}(S)} < \frac{\varepsilon}{2\alpha}.$$

Conclusion: Therefore, both cases yield

$$\begin{aligned} \|(\mathbb{1} - P_1(1)) \, |\boldsymbol{\beta}, \boldsymbol{\gamma}\rangle\| &= \|(\mathbb{1} - P_1(1)) V(\boldsymbol{\beta}, \boldsymbol{\gamma}) \, |\iota\rangle\| \\ &\leq \|(\mathbb{1} - P_1(1))(V(\boldsymbol{\beta}, \boldsymbol{\gamma}) - U_T(1)) \, |\iota\rangle\| + \|(\mathbb{1} - P_1(1)) U_T(1) \, |\iota\rangle\| \\ &< \|\mathbb{1} - P_1(1)\|_{\mathcal{L}(S)} \frac{\varepsilon}{2\alpha} + \frac{\varepsilon}{2} = \varepsilon. \end{aligned}$$

Then $\mathrm{im}(P_1(1)) \subseteq S_{\mathrm{opt}}$ proves the assertion. □

Bibliography

[AE99] J. E. Avron and A. Elgart. "Adiabatic Theorem without a Gap Condition". In: *Comm. Math. Phys.* 203 (1999), pp. 445–463.

[Ama+21a] D. Amaro et al. *A case study of variational quantum algorithms for a job shop scheduling problem*. 2021. arXiv: 2109.03745.

[Ama+21b] D. Amaro et al. *Filtering variational quantum algorithms for combinatorial optimization*. 2021. arXiv: 2106.10055.

[Ben80] P. Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *J. Stat. Phys.* 22 (1980), pp. 563–591.

[Bha+22] K. Bharti et al. "Noisy intermediate-scale quantum algorithms". In: *Rev. Mod. Phys.* 94.1 (2022).

[Bha97] R. Bhatia. *Matrix Analysis*. Springer, New York, NY, 1997.

[BR22] J. S. Baker and S. K. Radha. *Wasserstein Solution Quality and the Quantum Approximate Optimization Algorithm: A Portfolio Optimization Case Study*. 2022. arXiv: 2202.06782.

[Dam+21] W. van Dam et al. "Quantum Optimization Heuristics with an Application to Knapsack Problems". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 160–170.

[Deu85] D. Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". In: *Proc. R. Soc. Lond. A* 400 (1985), pp. 97–117.

[Dia22] O. Dial. *Eagle's quantum performance*. https://research.ibm.com/blog/eagle-quantum-processor-performance. 2022.

[Die17] R. Diestel. *Graph Theory*. Springer Berlin, Heidelberg, 2017.

[Dua20] R. Duan. *Quantum Adiabatic Theorem Revisited*. 2020. arXiv: 2003.03063.

[Eke+02] A. K. Ekert et al. "Direct Estimations of Linear and Nonlinear Functionals of a Quantum State". In: *Phys. Rev. Lett.* 88 (2002).

[Far+00] E. Farhi et al. *Quantum Computation by Adiabatic Evolution*. 2000. arXiv: quant-ph/0001106.

[Fey82] R. P. Feynman. “Simulating physics with computers”. In: *Int. J. Theor. Phys.* 21 (1982), pp. 467–488.

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028.

[FQ85] E. C. Freuder and M. J. Quinn. “Taking advantage of stable sets of variables in constraint satisfaction problems”. In: *In IJCAI'85*. 1985, pp. 1076–1078.

[Fro12] G. Frobenius. “Über Matrizen aus nicht negativen Elementen”. In: *Berl. Ber.* (1912), pp. 456–477.

[GE21] C. Gidney and M. Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. In: *Quantum* 5 (2021), p. 433.

[GH19] P. D. de la Grand'rive and J. Hullo. *Knapsack Problem variants of QAOA for battery revenue optimisation*. 2019. arXiv: 1908.02210.

[GM19] G. G. Guerreschi and A. Y. Matsuura. “QAOA for Max-Cut requires hundreds of qubits for quantum speed-up”. In: *Sci. Rep.* 9 (2019), p. 6903.

[Gro96] L. K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *Proc. Annu. ACM Symp. Theory Comput.* 1996, pp. 212–219.

[Had+19] S. Hadfield et al. “From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz”. In: *Algorithms* 12.2 (2019), p. 34.

[Had18] S. Hadfield. *Quantum Algorithms for Scientific Computing and Approximate Optimization*. 2018. arXiv: 1805.03265.

[Har+21] M. P. Harrigan et al. “Quantum approximate optimization of non-planar graph problems on a planar superconducting processor”. In: *Nat. Phys.* 17 (2021), pp. 332–336.

[HJ12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

[Kar72] R. M. Karp. “Reducibility among Combinatorial Problems”. In: ed. by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger. Springer US, 1972, pp. 85–103.

[Kat95] T. Kato. *Pertubation Theory for Linear Operators*. Springer, Berlin, Heidelberg, 1995.

[Kay20] A. Kay. *Tutorial on the Quantikz Package*. 2020. arXiv: quant-ph/1809.03842.

[Kit95] A. Yu. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem*. 1995. arXiv: quant-ph/9511026.

[LB13] D. A. Lidar and T. A. Brun. *Quantum Error Correction*. Cambridge University Press, 2013.

[LB17] Y. Li and S. C. Benjamin. "Efficient Variational Quantum Simulator Incorporating Active Error Minimization". In: *Phys. Rev. X* 7.2 (2017).

[Lei77] F. T. Leighton. "A Graph Coloring Algorithm for Large Scheduling Problems". In: *J. Res. Natl. Bur. Stand.* 84.6 (1977), pp. 489–506.

[Llo18] S. Lloyd. *Quantum approximate optimization is computationally universal*. 2018. arXiv: `1812.11075`.

[Llo96] S. Lloyd. "Universal Quantum Simulators". In: *Science* 273 (1996), pp. 1073–1078.

[Mac34] J. K. L. MacDonald. "On the Modified Ritz Variation Method". In: *Phys. Rev.* 46 (1934), pp. 828–828.

[MBZ20] M. E. S. Morales, J. D. Biamonte, and Z. Zimborás. "On the universality of the quantum approximate optimization algorithm". In: *Quantum Inf. Process.* 19 (2020).

[McA+19] S. McArdle et al. "Variational ansatz-based quantum simulation of imaginary time evolution". In: *npj Quantum Inf.* 5.1 (2019), p. 75.

[McL64] A. D. McLachlan. "A variational solution of the time-dependent Schrodinger equation". In: *Mol. Phys.* 8 (1964), pp. 39–44.

[Mon16] A. Montanaro. "Quantum algorithms: an overview". In: *npj Quantum Inf.* 2 (2016), p. 15023.

[Ort+01] G. Ortiz et al. "Quantum algorithms for fermionic simulations". In: *Phys. Rev. A* 64 (2001).

[Pag+20] G. Pagano et al. "Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator". In: *Proc. Natl. Acad. Sci. U.S.A.* 117 (2020), pp. 25396–25401.

[Pan56] S. Pancharatnam. "Generalized theory of interference, and its applications". In: *Proc. Indian Acad. Sci* 44.5 (1956), pp. 247–262.

[Per+14] A. Peruzzo et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nat. Commun.* 5.1 (2014).

[Pre18] J. Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[Qia+18] Xiaogang Qiang et al. "Large-scale silicon quantum photonics implementing arbitrary two-qubit processing". In: *Nat. Photonics* 12 (2018), pp. 534–539.

[RBW06] F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming*. Elsevier Sicence Inc., 2006.

[Rel37] F. Rellich. “Störungstheorie der Spektralzerlegung. I. Mitteilung. Analytische Störung der isolierten Punkteigenwerte eines beschränkten Operators”. In: *Math. Ann.* 113 (1937), pp. 600–619.

[Rit09] W. Ritz. “Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik.” In: 1909.135 (1909), pp. 1–61.

[RMS19] M. Radzihovsky, J. Murphy, and M. Swofford. *A QAOA solution to the traveling salesman problem using pyQuil*. 2019.

[Sag01] B. E. Sagan. *The Symmetric Group*. Springer, New York, NY, 2001.

[Sch+19] M. Schuld et al. “Evaluating analytic gradients on quantum hardware”. In: *Phys. Rev. A* 99 (2019), p. 032331.

[Sho94] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134.

[Sim17] B. Simon. *Tosio Kato’s Work on Non-Relativistic Quantum Mechanics*. 2017. arXiv: 1711.00528v1.

[Sjö+00] E. Sjöqvist et al. “Geometric Phases for Mixed States in Interferometry”. In: *Phys. Rev. Lett.* 85.14 (2000), pp. 2845–2849.

[Str11] J. W. Strutt. *The Theory of Sound*. Vol. 1. Cambridge Library Collection - Physical Sciences. Cambridge University Press, 2011.

[Teu01] S. Teufel. “A Note on the Adiabatic Theorem Without Gap Condition”. In: *Lett. Math. Phys.* 58.3 (2001), pp. 261–266.

[WHB19] D. Wang, O. Higgott, and S. Brierley. “Accelerated Variational Quantum Eigensolver”. In: *Phys. Rev. Lett.* 122 (2019).

[Wic54] G. C. Wick. “Properties of Bethe-Salpeter Wave Functions”. In: *Phys. Rev.* 96 (1954), pp. 1124–1134.

List of Figures