UNIVERSITÄT ZU LÜBECK

From the Institute of Mathematics and Image Computing
of the University of Lübeck
Director: Prof. Dr. Jan Modersitzki

Quantum Algorithms for Binary Problems with Applications to Image Processing

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences

Submitted by

Natacha Kuete Meli
from Galim

Lübeck, 2024

First referee:
Second referee:
Date of oral examination:
Approved for printing:

Abstract

In this thesis, we investigate multiple approaches for constructing quantum algorithms with applications to image processing: Hybrid, fully universal, annealing-based, and based on quantum time evolution.

First, we develop a variational quantum algorithm for the Ising Hamiltonian ground-state problem which block-encodes the Ising Hamiltonian in a unitary operator acting on a parameterized quantum state, the parameters of which are optimized in a way to solve the problem using an outer classical optimization routine. Experimentally, the algorithm outperforms the quantum approximate optimization algorithm (QAOA) and challenges dedicated D-Wave annealers in returning approximate solutions.

Second, we propose an iterative, fully universal quantum algorithm which uses quantum amplitude amplification and does not require the classical outer loop. This fully universal method guarantees that, after an optimal number of iterations, a ground state of the Ising Hamiltonian is measured with the highest probability compared to other states.

Turning to a specific application in image processing, we formulate the rigid point set registration problem as a quadratic unconstrained binary optimization (QUBO) problem suitable for quantum annealers. Here, we consider a continuous yet constrained objective function, optimizing over rotation matrices from the special orthogonal group – a Lie group. Moreover, we solve the orthogonality constraint and propose instead to optimize over the Lie algebra. In an iterative process, we construct and solve QUBO problems to adjust the rotation parameters. We successfully deploy our method on D-Wave quantum annealers and demonstrate its viability on several 2D and 3D point sets, improving the state-of-the-art quantum approach for this problem.

Finally, we investigate quantum time evolution for solving parametric non-convex image registration problems. We consider the existing quantum Hamiltonian descent (QHD) algorithm, which is derived from the quantum path integral of dynamical systems and its relation to the continuous time limit of classical gradient descent methods. We classically simulate the time evolution of the QHD Hamiltonian for a non-convex rigid image registration problem and find that the method can effectively find globally optimal solutions.

Zusammenfassung

Das Ziel dieser Arbeit ist die Untersuchung unterschiedlicher Ansätze zur Konstruktion von Quantenalgorithmen mit Anwendungen in der Bildverarbeitung: Hybrid, im universellen Quantencomputingmodell, sowie basierend auf Quantum Annealing und Quantum Time Evolution.

Zunächst wird ein variationeller Quantenalgorithmus für das Grundzustandsproblem des Ising-Hamiltonoperators vorgestellt. Der Hamiltonoperator wird dabei mit Hilfe einer Blockdarstellung umgesetzt und auf einen parametrisierten Quantenzustand angewandt, dessen Parameter in einer äußeren Schleife klassisch optimiert werden. In den Experimenten übertrifft das Verfahren den Quantum Approximate Optimization Algorithm (QAOA) bei der Suche nach Näherungslösungen und liegt gleichauf mit spezialisierten D-Wave-Annealern.

Weiterhin wird ein iterativer Algorithmus basierend auf dem universellen Modell des Quantencomputings vorgestellt, der auf Quantenamplitudenverstärkung aufbaut und keine äußere, klassische Optimierungsschleife benötigt. Dieses vollständig quantenbasierte Verfahren stellt sicher, dass nach einer optimalen Anzahl an Iterationen der Grundzustand des Ising-Hamiltonoperators mit einer größeren Wahrscheinlichkeit als alle anderen Zustände gemessen wird.

Als konkrete Anwendung in der Bildverarbeitung wird das Problem der starren Punktwolkenregistrierung als quadratisches unbeschränktes binäres Optimierungsproblem (QUBO) formuliert. Dabei wird ein beschränktes kontinuierliches Optimierungsproblem aufgestellt, das über der Menge der Rotationsmatrizen der speziellen orthogonalen Gruppe – einer Lie-Gruppe – formuliert ist. Um die Orthogonalitätsnebenbedingung umzusetzen, wird stattdessen über die Lie-Algebra optimiert. Die Parameter der Rotationen werden anschließend angepasst, indem iterativ QUBO-Probleme konstruiert und gelöst werden. Das Verfahren ist real umsetzbar auf D-Wave-Quantenannealern sowohl für 2D- als auch für 3D-Datensätze und in der Praxis eine Verbesserung im Vergleich zu bisherigen Quantenverfahren für dieses Problem.

Schließlich wird ein Ansatz basierend auf Quantum Time Evolution diskutiert, mit dem Ziel, parametrische nicht-konvexe Bildregistrierungsprobleme zu lösen. Der Ausgangspunkt ist das Quantum Hamiltonian Descent-Verfahren (QHD), das durch das Pfadintegral der Quantenmechanik und dessen Verbindung zum zeitkontinuierlichen Limit des klassischen Gradientenabstiegs motiviert ist. Eine klassische Simulation der Zustandsentwicklung des QHD-Hamiltonoperators für ein nichtkonvexes starres Bildregistrierungsproblem zeigt, dass das Verfahren effektiv global optimale Lösungen finden kann.

Acknowledgments

I express my deep gratitude to my advisor, Prof. Jan Lellmann, for introducing me to scientific research. He unveiled me a realm of opportunities, leading me into the captivating and emerging field of quantum computing – a term that I heard for the very first time then, and which completely fascinates me today. Prof. Lellmann instilled in me the sense of curiosity and the joy of discovery, making me take part in my very first scientific conference. Even more, he did me the honor of being his teaching assistant, allowing me to perfect my knowledge in applied mathematics.

I thank Dr. Florian Mannel for his invaluable assistance in writing my manuscripts. His guidance in scientific writing and critical thinking has been transformative.

To my colleagues for being willing to listen to my stammering ideas about science and for all our brainstorming sessions, thank you.

Many thanks to my family, especially to my brother Cesaire, for being a daily support and a listening ear by my side. Special thanks to my mother for being that, my mother.

Warm thanks to Marion, my steadfast friend.

To all who supported me, I extend a huge thank you. This thesis would certainly not have been possible without you.

Symbols & Abbreviations

Linear Algebra

i	complex unit, $i^2 = -1$.
$\mathcal{H}$	Hilbert space.
$\mathbf{M}$	matrix.
$\sigma(\mathbf{M})$	set of eigenvalues of $\mathbf{M}$.
ψ	column vector $\psi = (\psi_1, \ldots, \psi_n)^\top$.
$\square^\top$	transpose of $\square$.
$\overline{\square}$	complex conjugate of $\square$.
$\square^\dagger$	Hermitian conjugate or adjoint of $\square$.
$[\square,\ \triangle]$	commutator of $\square$ and $\triangle$.
$\langle \square, \triangle \rangle$	inner product of $\square$ and $\triangle$.
$\mathrm{Re}(\square)$	real part of complex number $\square$.
$\mathrm{Im}(\square)$	imaginary part of complex number $\square$.

Quantum Computing

$\hbar$	reduced Planck's constant.
$\lvert\psi\rangle$	state-vector of a quantum system ψ, Ket.
$\langle\psi\rvert$	Hermitian conjugate or adjoint of $\lvert\psi\rangle$, Bra.
$\langle\psi\vert\phi\rangle$	inner product of $\lvert\psi\rangle$ and $\lvert\phi\rangle$, Braket.
$\langle\psi\vert\mathbf{M}\vert\phi\rangle$	inner product of $\lvert\psi\rangle$ and $\mathbf{M}\lvert\phi\rangle$, also $\mathbf{M}^\dagger\lvert\psi\rangle$ and $\lvert\phi\rangle$.
$\lvert\psi\rangle \otimes \lvert\phi\rangle$	tensor product of $\lvert\psi\rangle$ and $\lvert\phi\rangle$, also $\lvert\psi\phi\rangle$, $\lvert\psi, \phi\rangle$, or $\lvert\psi\rangle\lvert\phi\rangle$.
$\mathbf{I} \otimes \mathbf{U}^\psi \lvert\psi, \phi\rangle$	controlled-$\mathbf{U}$ – i.e., $\mathbf{I} \otimes \mathbf{U} \lvert\psi, \phi\rangle$ if $\psi = 1$ and $\mathbf{I} \otimes \mathbf{I} \lvert\psi, \phi\rangle$ otherwise.
$\langle\square\rangle$	expectation of observable $\square$.
$\Delta(\square)$	standard deviation of observable $\square$.
$\mathbf{U}(\theta)$	parameterized quantum circuit, or variational circuit.
$[N]$	set $\{0, 1, \ldots, N-1\}$ for computational basis-states.

Numerical Optimization

f	objective function to be optimized.
∇f	gradient of f.
$\nabla^2 f$	hessian of f.
$\frac{\partial}{\partial \triangle}\square$	partial derivative of $\square$ w.r.t. the variable $\triangle$.
$\square^\star$	globally optimal solution of an optimization problem.

Abbreviations

ADAM	Adaptive Moment estimation.
AQC	Adiabatic Quantum Computing.
CNOT	Controlled-NOT.
COBYLA	Constrained Optimization BY Linear Approximation.
CPU	Central Processing Unit.
GD	Gradient Descent.
ICT	Iterative Classical Transformation Estimation.
IQT	Iterative Quantum Transformation Estimation.
NBAA	Non-Boolean Amplitude Amplification.
NISQ	Noisy Intermediate Scale Quantum.
NOT	NOT-quantum gate.
PM-NBAA	Phase Matching-NBAA.
QA	Quantum Approach for transformation estimation.
QAOA	Quantum Approximate Optimization Algorithms.
QHD	Quantum Hamiltonian Descent.
QP	Quadratic Program.
QPU	Quantum Processing Unit.
QUBO	Quadratic Unconstrained Binary Optimization.
SPSA	Simultaneous Perturbation Stochastic Approximation.
TE	Transformation Estimation.
VQE	Variational Quantum Eigensolver.
VQSVD	Variational Quantum Singular Value Decomposition.

Contents

CHAPTER 1

Introduction

In this thesis, we develop quantum computing algorithms applicable to image processing problems. Emphasis is on formulating novel mathematical models and optimization techniques suitable for quantum computation, specifically targeting solutions for binary combinatorial and image registration problems.

1.1 Motivation

The advent of quantum computing offers a completely new computing paradigm. Introduced in the early 1980s by Richard Feynman [1] for simulating nature, quantum computers rely on quantum mechanical elements which obey quantum mechanical laws. When quantum elements are combined to constitute a quantum system, the dimension of the associated computational space expands *exponentially* with the number of elements. Quantum systems can assume superposition states, essentially existing in several pure states simultaneously, which allows for computations to act on all states simultaneously. In addition to *superposition*, quantum states can be *entangled* and can *interfere*, constructively or destructively, with oneself, fundamentally setting quantum computers apart from classical analogues [2].

Quantum computing can be described by two computational models that differ in their functioning: The *adiabatic* quantum model [3] is best suited for optimization problems, typically cast in quadratic unconstrained binary optimization form. Using quantum *tunneling* effects, this model can be used to find near-optimal solutions of optimization problems and is a promising way for escaping local minima that can trap classical gradient-based optimizers. The *universal*

model of quantum computing, also referred to as the gate-based or circuit model, is more flexible for modelling problems and can potentially implement any classical operation, as shown by Bennett [4]. For selected problems such as Shor's factoring [5] and Grover's search [6] algorithms, strong theoretical convergence properties and drastic speed-up of the universal quantum computing model over classical counterparts have been proven.

With the release of commercial devices such as IBM universal quantum computers [7] and D-Wave annealers [8], quantum computing is becoming reality. However, currently available devices, labeled as *noisy intermediate scale quantum* devices [9], are of relatively small size and still prone to noise and imperfect quantum elements. In the era of noisy intermediate-scale devices, it is a challenging task to find real-world applications of quantum computing. Hybrid quantum-classical algorithms are being actively explored for obtaining practical quantum supremacy.

1.2 Applications and Scope

Computer vision, and in particular image processing, abounds with challenging optimization problems, for which quantum computing could provide practical utility. Image processing problems are often non-convex, large-scale, and may have combinatorial components. Some application examples are illustrated in Figure 1.2.1.

Optimization.

In numerical optimization [10, 11], one seeks to numerically find, on a given feasible set, a point that globally minimizes a certain real-valued function; said point is called the *minimizer*. For *differentiable* functions, the minimizer is also a stationary point, i.e., a point where the gradient of the function vanishes. Classical ways to find the minimizer leverage gradient information to search for a direction in which the function value locally decreases. Only in rare cases where the function is convex, this point can be found. In other cases, it is rather difficult to find the global minimum and only local optimal solutions are found, see Figure 1.2.2a. Gradient-free optimizers also exist, but are generally heuristics not necessarily returning a stationary point. For *combinatorial* problems, it is impossible to use gradient information without modifying the problem, as the feasible set is discrete and the function is non-differentiable. Furthermore, the search space dimension of combinatorial problems often grows exponentially with the problem size, making them particularly difficult to solve.

Quantum optimization.

Quantum computation [12, 13] operates on a quantum system that assumes different basis states with certain probabilities once measured. A *quantum probability* is the squared value of a certain complex number called *amplitude*. In general, one

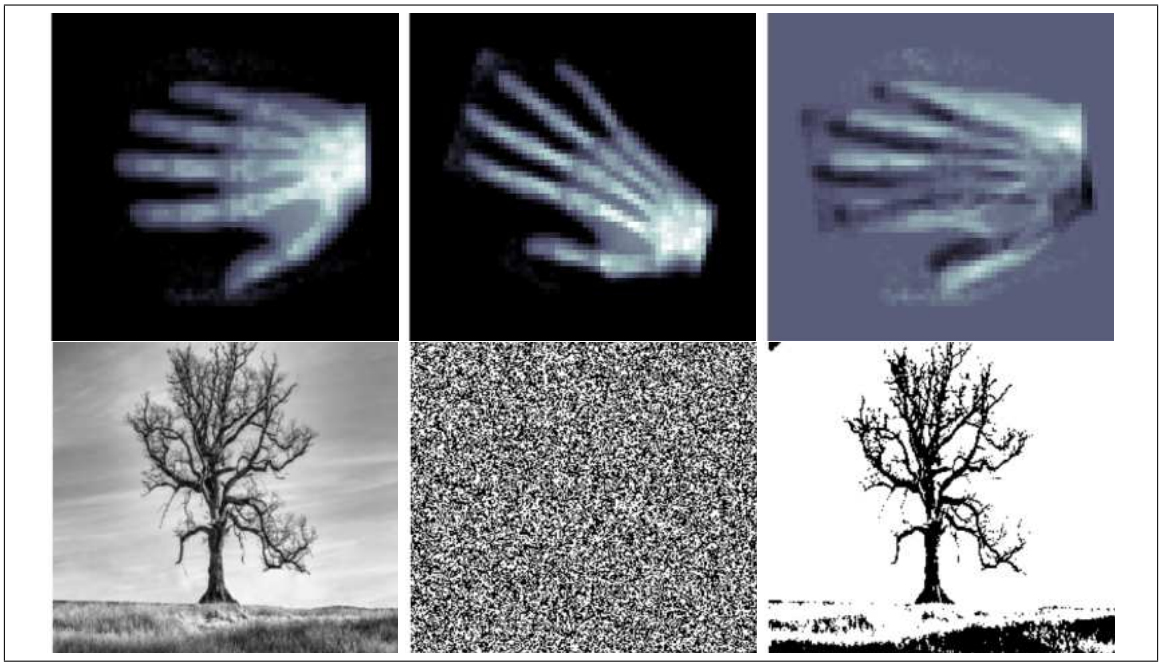
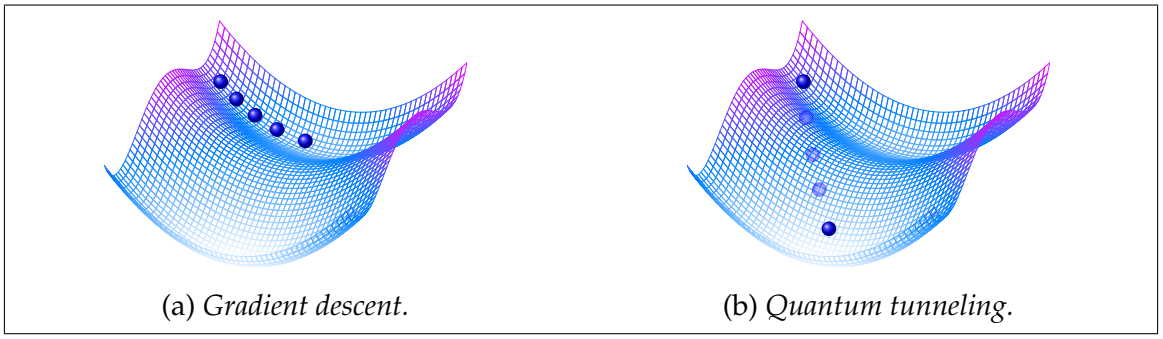


Figure 1.2.1: *Example applications of image processing. (**Top row**) Image registration illustrated as a problem of aligning two images such that they overlap with minimal cost. (**Bottom row**) Image segmentation illustrated as a problem of deciding for each pixel of a random lattice, if the pixel belongs to the image foreground or background.*

can only manipulate the amplitude, thus the probability of finding the system in a particular state. Optimizing with quantum computers typically means to bring the quantum system into a state that is, with high probability, the state encoding the solution of an optimization problem, and into which the quantum system *collapses* when it is observed. Quantum algorithms are developed in this direction [14]:

- *Adiabatic quantum methods* assume a Hamiltonian whose ground state encodes a solution of an optimization problem. Optimization, using quantum tunneling effects, see Figure 1.2.2b, is achieved by performing an adiabatic evolution of the quantum system from the known and easily-prepared ground state of an initial Hamiltonian to that of the problem Hamiltonian.
- *Amplitude amplification methods* assume a quantum oracle capable of evaluating the objective function on a given superposition state of all feasible solutions. Subsequently, they make the amplitudes of non-optimal states destructively interfere, while those of the solutions constructively interfere.
- *Variational methods* are hybrid quantum-classical methods taking as input problem data, a *parameterized* quantum ansatz and a corresponding objective function. Then, the parameterized quantum state is measured in some basis to evaluate the objective function and the parameters of the ansatz are classically updated to optimize over the objective function.



(a) *Gradient descent.* (b) *Quantum tunneling.*

Figure 1.2.2: *Classical gradient descent vs. quantum tunneling.* ***(a)*** *Classical gradient descent steps model a particle navigating the optimization problem's energy landscape, moving towards a local minimum.* ***(b)*** *In quantum mechanics, the optimization landscape represents the energy associated with the state of the quantum system. The quantum element, exhibiting wave-particle duality, can tunnel through energy barriers, enabling exploration of classically prohibited paths to reach a minimal energy state.*

Other recent quantum methods, such as quantum Hamiltonian descent [15], build on Feynman's quantum path integral theory [16] and completely rethink gradient-based optimizers. While in classical gradient-based optimization the gradient dictates the path towards the solution, in quantum computing, quantum elements inherently can take every possible direction. This allows to explore classically prohibited directions to find a near-optimal solution of the optimization problem.

Digital Image Processing.

Digital image processing refers to the manipulation of images by means of a computer [17, 18]. Typical tasks involve processing image data into other image data, such as segmentation, or extracting higher-level information from images, such as classification.

The task of *image segmentation* [19] consists in deciding, for each pixel of the image, to which of a given set of segments it should be assigned. Image segments typically represent semantic objects in the image; these could be foreground and background, in which case the segmentation separates the image into foreground and background. A segment could also designate an object in the image, so that the segmentation classifies each pixel of the image to belong to one object. In any case, image segmentation is a combinatorial optimization problem, which can be regarded as a problem of assigning labels to the nodes of a given graph with minimal cost. Figure 1.2.3 shows an exemplary application of the variational hybrid quantum-classical algorithm developed in this thesis for the binary case.

Another interesting image processing problem is that of *image registration* [20]. In this context, given two images, the goal is to find a reasonable deformation of the pixel coordinates of one image in a way to align this image to the other. When formulated as an optimization problem, the image registration problem

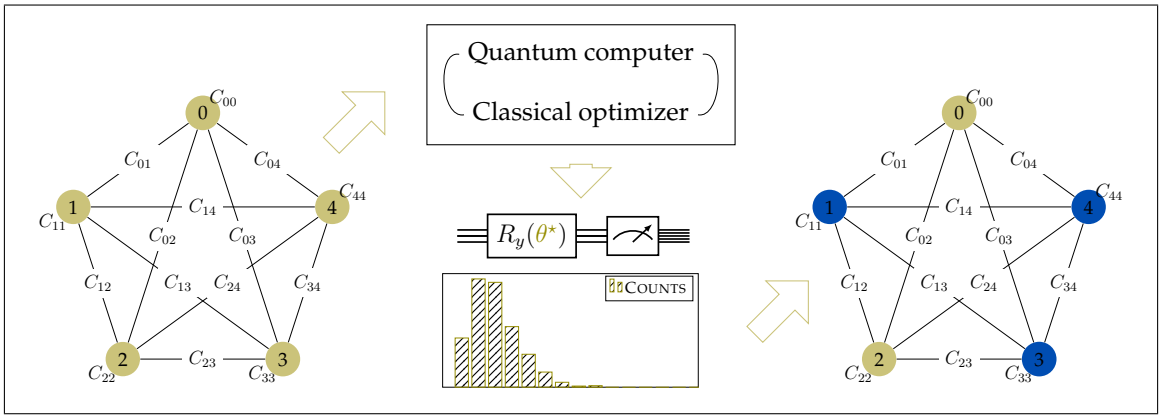


Figure 1.2.3: *Overview of our hybrid quantum-classical method for solving a binary combinatorial problem. The problem is formulated as the problem of classifying the nodes of a weighted graph so that the overall cost is minimal.* ***(Left)*** *The input graph including nodes and edge weights.* ***(Middle)*** *A hybrid quantum-classical loop that optimizes a set of rotation angles θ in order to drive a parameterized ansatz into the optimal state.* ***(Right)*** *Once found, the optimal rotation parameters prepare the quantum system into a state whose measurement delivers a locally optimal node classification. The method is discussed in detail in Chapter 4.*

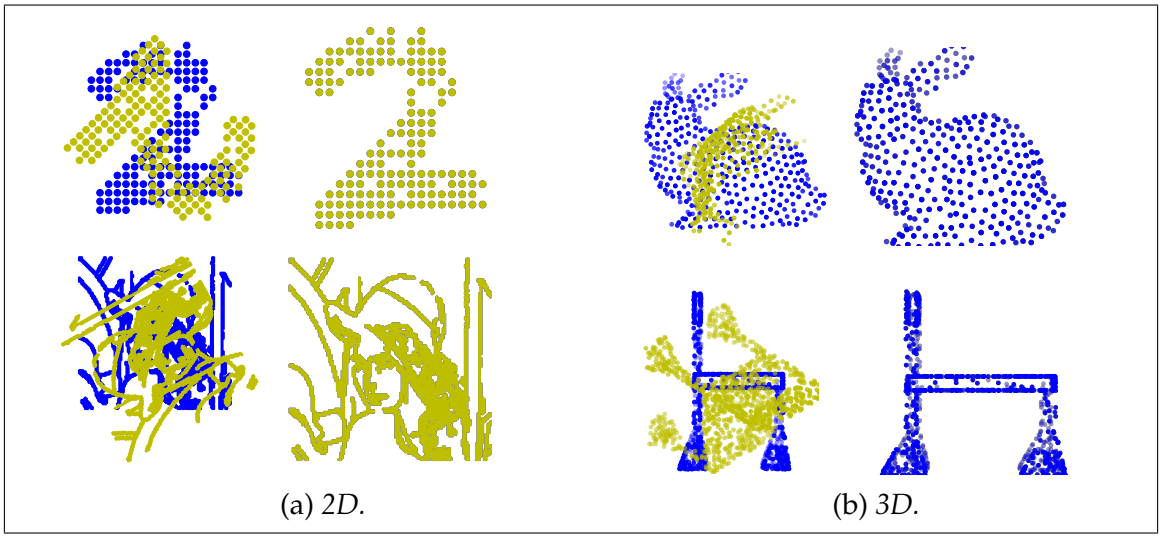
in often a non-convex problem. The quality of registration results obtained, for instance, with gradient-based optimizers is significantly impacted by the initial alignment. To address this, a common practice involves an initial landmark-based pre-registration step, where the registration is achieved by aligning corresponding landmarks or points in the two images. We introduce an iterative quantum approach to register two sets of points. Our results are displayed in Figure 1.2.4.

1.3 Contribution and Outline

This thesis explores several quantum computing algorithms for dealing with image processing problems. We develop adiabatic, universal, as well as hybrid classical-quantum algorithms for solving combinatorial and registration problems in image processing. We also explore quantum-inspired algorithms, by means of quantum path integrals for non-convex image processing problems. Next, we outline and summarize our contribution. Related literature for each topic is discussed within the respective section. [1]

In Chapter 2, we provide a compact review of the fundamental concepts of quantum computations. The quantum universal and adiabatic models are presented

[1] As parts of this thesis have been published in peer-reviewed journals and conferences, we also provide for each section, if applicable, the related author's publication. For each publication, NKM conceived and developed the idea, performed the implementation and wrote the manuscript. FM directed the experiments, and wrote the manuscript. JL supervised the work and proof-read the manuscript.



(a) *2D.* (b) *3D.*

Figure 1.2.4: *Iterative quantum approach for transformation estimation.* ***(a)*** *Registration results on 2D point sets (MNIST [21] and Lena [22]).* ***(b)*** *Registration results on 3D point sets (Stanford bunny [23] and completion3D [24]). Blue points represent reference points and olive points represent template points. The initial alignment is shown on the left and the result of the registration on the right. See Chapter 6 for a full discussion.*

in detail, revisiting gates for the universal model and the adiabatic theorem for the adiabatic model. Additionally, a brief overview on noisy intermediates devices is provided.

In Chapter 3, we review how to optimize parameterized quantum circuits for solving problems. To this end, some known classical optimizers, gradient-based and gradient-free, are presented. We also introduce the Ising problem, a combinatorial problem with multiple applications in image processing [25], as well as the related optimization using adiabatic quantum computing.

In Chapter 4, we propose a new algorithm for solving the Ising problem. The approach hinges on a new, easy to implement and low-depth variational circuit that effectively encodes the Ising problem on quantum hardware. We derive an optimization routine based on gradient descent for the proposed variational circuit in order to drive the quantum system towards the solution of the problem. Experimentally, we validate the novel algorithm and benchmark it against the state-of-the-art gate-based QAOA [26] model, and specialized D-Wave annealers.

Related author's publication:

- Kuete Meli, N., Mannel, F., and Lellmann, J. "A universal quantum algorithm for weighted maximum cut and Ising problems", *Quantum Inf Process* 22, 279 (July 2023). https://doi.org/10.1007/s11128-023-04025-x.

In Chapter 5, we approach the Ising problem from a purely universal quantum perspective. We use iterative quantum amplitude amplification techniques to

boost the amplitude of the quantum state encoding the solution of the problem. To this end, an existing non-Boolean amplitude amplification algorithm [27, 28] is employed and modified to achieve better results, i.e, to measure the optimal state with a higher probability. We mathematically analyze the new algorithm and find that, under some assumptions, it provably boosts the amplitude of the solution more than the original algorithm does.

In Chapter 6, we present a new iterative method for estimating rigid transformations from point sets using adiabatic quantum computation [29]. The method relies on an adaptive scheme to solve the problem to high precision, and does not suffer from inconsistent rotation matrices. Experimentally, the proposed method performs robustly on 2D and 3D data sets and even with high outlier ratio.

Related author's publication:

- Kuete Meli, N., Mannel, F., and Lellmann, J. "An iterative quantum approach for transformation estimation from point sets", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 529-537 (Juli 2022). https://doi.org/10.1109/CVPR52688.2022.00061.

In Chapter 7, we explore further quantum optimization perspectives applicable to image processing problems. Specially, the existing quantum gradient descent algorithm [15] is presented, classically simulated and tested on a rigid image registration problem. While not yet rigorously developed, we find that the algorithm has great potential for finding a globally optimal solution.

In Chapter 8, we conclude the thesis. We summarize the main findings and discuss open questions and promising directions of further research.

CHAPTER 2

Preliminaries

Quantum computing is a way of thinking physically about computation. Quantum computation takes place in a quantum mechanical system made up of quantum particles. These particles are mathematical objects with certain specific properties.

This chapter aims to provide a self-contained introduction to quantum computing and sets up the notation used in the thesis. Furthermore, we will see how quantum computing replaces even the most basic classical operations with ones involving so-called quantum bits. We will see how to encode and manipulate quantum information for problem solving, unlocking new and exciting capabilities for information processing. The definitions and concepts of this chapter mostly follow [12, 13].

2.1 Quantum Bit

To gain some intuition of the mathematical modelling of a quantum system, we consider the simplest possible quantum system known as *quantum bit* or *qubit*. This one-particle system has two potential energy states. Conventionally denoted by $|0\rangle$ and $|1\rangle$, the two basis states are orthogonal unit vectors in $\mathbb{R}^2$:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1}$$

The states $|0\rangle$ and $|1\rangle$ are eigenstates or stationary states of the system. Until it is measured, the qubit is in a complex-valued linear combination of its basis states. Just as the bit in classical computation, the qubit is the fundamental unit of quantum computation.

Definition 2.1.1 (*Quantum bit*)

A quantum bit*, or* qubit*, is a one-particle quantum system whose state is expressed as a* superposition*, i.e., linear combination, of the basis states* $|0\rangle$ *and* $|1\rangle$*,*

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2}$$

with $|\alpha|^2 + |\beta|^2 = 1,\ \alpha, \beta \in \mathbb{C}$.

The numbers α and β are called amplitudes of the basis states. When we observe a qubit, we only acquire much more restricted information about the quantum state: We either get $|0\rangle$, with probability $|\alpha|^2$, or $|1\rangle$, with probability $|\beta|^2$. An example of a superposition state is the uniform superposition state which is in both $|0\rangle$ and $|1\rangle$ with the same probability $|\alpha|^2 = |\beta|^2 = 1/2$. It is denoted as $|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

There are two common ways of geometrically thinking about a qubit $|\psi\rangle$: The first one is to simply represent the qubit as a unit-length complex vector

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \in \mathbb{C}^2. \tag{3}$$

The other way, that will later help to understand operations on qubits more intuitively, is to represent it in the 3D real space: Due to the normalization condition $|\alpha|^2 + |\beta|^2 = 1$, each qubit can be written as

$$|\psi\rangle = e^{i\omega}\left(\cos\frac{\gamma}{2}\,|0\rangle + e^{i\phi}\sin\frac{\gamma}{2}\,|1\rangle\right) \tag{4}$$

with angles $\gamma \in [0, \pi]$ and $\omega, \phi \in [0, 2\pi]$. This representation, called *Bloch-sphere* representation of the qubit $|\psi\rangle$, translates into a unit vector called the *Bloch vector* $\psi_{\text{bloch}} = (\cos\phi\sin\gamma, \sin\phi\sin\gamma, \cos\gamma)^\top \in \mathbb{R}^3$ and can be visualized as a point on the 3D unit-sphere as shown in Figure 2.1.1.

2.2 Postulates of Quantum Mechanics

While quantum mechanics exhibits surprising and sometimes unintuitive effects, its mathematical foundation rests on a relatively small number of basic postulates.

2.2.1 State space

Postulate 2.2.1 (*State space*)

The state of a quantum system at any time $t \in [0, T], T \in \mathbb{N}$*, is described by a unit-norm vector* $|\psi(t)\rangle = \alpha(t)\,|0\rangle + \beta(t)\,|1\rangle$ *called* state vector *and lives in a Hilbert space* $\mathcal{H}$.

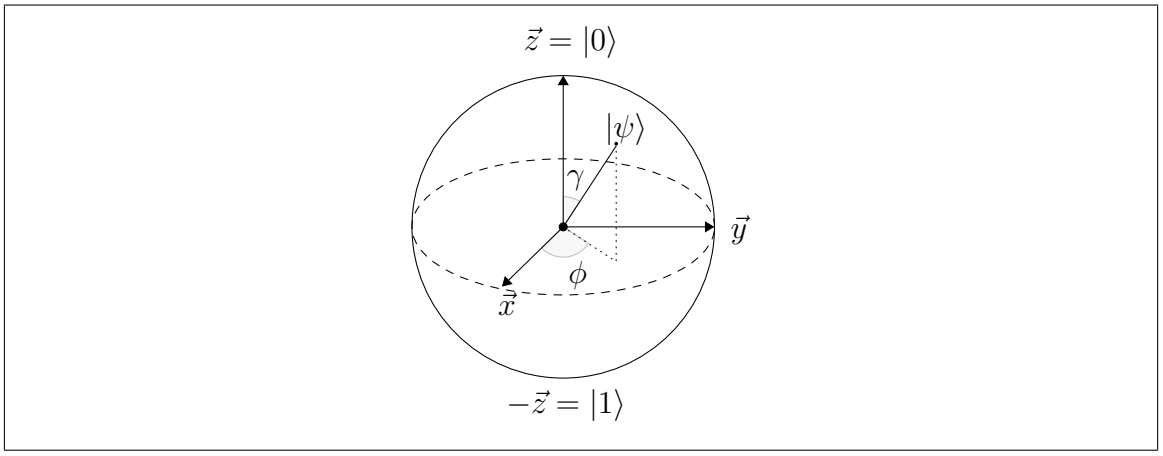


Figure 2.1.1: *Bloch sphere representation of a qubit. The qubit* $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ *can be expressed as a unit vector in* $\mathbb{R}^3$*. Two angles* $\gamma \in [0, \pi]$ *and* $\phi \in [0, 2\pi]$ *fully describe the qubit in the basis spanned by the vectors* $\vec{x}, \vec{y}$ *and* $\vec{z}$*.*

Definition 2.2.1 (*Hilbert space*)

Let $\mathcal{H}$ *be a vector space and* $\mathcal{F}$ *a scalar field. A mapping* $\langle\cdot|\cdot\rangle : \mathcal{H} \times \mathcal{H} \to \mathcal{F}$ *is an* inner product *if*

1. $\forall\ \psi, \phi \in \mathcal{H}$ *it holds* $\langle\psi|\phi\rangle = \overline{\langle\phi|\psi\rangle}$*,*
2. $\forall\ \lambda \in \mathcal{F}$ *and* $\psi, \phi, \phi' \in \mathcal{H}$ *it holds* $\langle\psi|\lambda\phi + \phi'\rangle = \lambda\,\langle\psi|\phi\rangle + \langle\psi|\phi'\rangle$*,*
3. $\forall\ \psi \in \mathcal{H}$ *it holds* $\langle\psi|\psi\rangle \geq 0$*, and* $\langle\psi|\psi\rangle = 0$ *if and only if* $\psi = 0$*.*

A Hilbert space *is a vector space with inner product.*

In practice, for $\mathcal{H} = \mathbb{C}^n$ and $\mathcal{F} = \mathbb{C}$, the mapping $\langle\cdot|\cdot\rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$, defined by $\langle\psi|\phi\rangle := \sum_{i=1}^{n} \overline{\psi_i}\phi_i$, is an inner product, making the complex space $\mathbb{C}^n$ a valid Hilbert space for quantum computation. The norm of ψ is calculated by the canonical norm as $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$. The squared magnitudes of the coefficients $\alpha(t)$ and $\beta(t)$ of the qubit represent the probability of the system of being in the basis states $|0\rangle$ and $|1\rangle$, yielding the normalization $\langle\psi|\psi\rangle = |\alpha(t)|^2 + |\beta(t)|^2 = 1$ if $|\psi(t)\rangle$ is a state vector.

We will reserve the Dirac notation $|\cdot\rangle$ (read Ket $\cdot$) only for state vectors. The notation $\langle\cdot|$ (read Bra $\cdot$) denotes the adjoint – Hermitian conjugate – of $|\cdot\rangle$, and $\langle\cdot|\cdot\rangle$ (read Braket) the inner product. For ease of reading flow and if it is clear from the context, we will sometimes omit the argument and write $|\psi\rangle$ instead of $|\psi(t)\rangle$.

2.2.2 Evolution

Quantum computation is performed by transforming the state vector of the quantum system from an initial state to a desired state, a state which solves a problem. The evolution is framed by following postulate:

Postulate 2.2.2 (*Evolution*)

The evolution *of a closed quantum system at time $t \in [0, T]$ is described by a unitary operator $\mathbf{U}(t)$. The state space $|\psi(t)\rangle$ of the system at time t is related to the state $|\psi(0)\rangle$ at time 0 by*

$$|\psi(t)\rangle = \mathbf{U}(t)\,|\psi(0)\rangle\,. \tag{5}$$

Moreover, at every point in time, the evolution obeys Schrödinger's equation,

$$i\hbar\frac{d}{dt}\,|\psi(t)\rangle = \mathbf{H}(t)\,|\psi(t)\rangle\,, \tag{6}$$

where i is the complex unit, $\hbar$ is a physical constant known as (reduced) Planck's constant*, and $\mathbf{H}$ is a Hermitian operator known as the system-driven* Hamiltonian*. The solution of the Schrödinger equation defines a time-dependent unitary operator $\mathbf{U}(t)$ that transforms the state $|\psi\rangle$ into the state $|\psi(t)\rangle$.*

Definition 2.2.2 (*Hermitian, normal, unitary, and diagonalizable operators*)

Let $\mathbf{A}$ be a linear operator, $\mathbf{A}^\dagger$ the complex adjoint of $\mathbf{A}$, and $\mathbf{I}$ the identity operator on a Hilbert space $\mathcal{H}$. Then,

- $\mathbf{A}$ *is* Hermitian $:\Leftrightarrow \mathbf{A}^\dagger = \mathbf{A}$,
- $\mathbf{A}$ *is* normal $:\Leftrightarrow \mathbf{A}^\dagger\mathbf{A} = \mathbf{A}\mathbf{A}^\dagger$,
- $\mathbf{A}$ *is* unitary $:\Leftrightarrow \mathbf{A}^\dagger\mathbf{A} = \mathbf{A}\mathbf{A}^\dagger = \mathbf{I}$.

The operator $\mathbf{A}$ is diagonalizable *if it admits a decomposition $\mathbf{A} = \sum_x \sigma_x\,|x\rangle\,\langle x|$ called spectral decomposition, where the vectors $|x\rangle \in \mathcal{H}$ form an orthonormal system of eigenvectors to eigenvalues $\sigma_x \in \mathbb{C}$ of $\mathbf{A}$. The set $\sigma(\mathbf{A})$ denotes the set of all eigenvalues of $\mathbf{A}$.*

If we know or provide either the (time-dependent) unitary $\mathbf{U}$ or the system-driven Hamiltonian $\mathbf{H}$, then, we fully understand or can control the system dynamic. In adiabatic quantum computing, to solve optimization problems, one typically chooses $\mathbf{H}(t)$ as an evolution over the time t from an initial Hamiltonian $\mathbf{H}(0)$ to a problem Hamiltonian $\mathbf{H}(T)$. In universal quantum computing, it is more common to think in terms of the unitary $\mathbf{U}(t)$ that transforms the state vector of the system.

It is important for the operator $\mathbf{U}$ acting on the system to be unitary, since then

$$\|\,|\psi(t)\rangle\,\|^2 = \|\mathbf{U}(t)\,|\psi(0)\rangle\,\|^2 = \langle\psi(0)|\mathbf{U}(t)^\dagger\mathbf{U}(t)|\psi(0)\rangle = \langle\psi(0)|\psi(0)\rangle = 1, \tag{7}$$

i.e., the norm of the state vector is preserved, the operation is reversible with no information loss. A unitary operator acting on a state vector is called *quantum gate*.

2.2.3 Measurement

Measurement allows to extract classical information from qubits. It is the only non-reversible operation in quantum computation. Upon measurement, the qubit loses its superposition property and collapses to one of its eigenstates:

Postulate 2.2.3 (*Measurement*)

The measurement *of a quantum system is carried out by a family* $\{\mathbf{M}_m,\ m \in \mathcal{M}\}$ *of measurement operators fulfilling the relation* $\sum_m \mathbf{M}_m^\dagger \mathbf{M}_m = \mathbf{I}$. *The set* $\mathcal{M}$ *is the set of all possible measurement outcomes. For a system prepared in the state* $|\psi\rangle$, *the probability of observing the outcome* m *upon measurement is*

$$p(m) = \langle\psi|\mathbf{M}_m^\dagger \mathbf{M}_m|\psi\rangle\,, \tag{8}$$

and the state of the system immediately after observing outcome m *is*

$$|\psi_m\rangle = \frac{\mathbf{M}_m\,|\psi\rangle}{\sqrt{\langle\psi|\mathbf{M}_m^\dagger \mathbf{M}_m|\psi\rangle}}. \tag{9}$$

If, for some Hermitian operator $\mathbf{M} = \sum_m m\mathbf{P}_m$, *a so-called* observable, *the operators* $\mathbf{M}_m$ *are set to* $\mathbf{M}_m = \mathbf{P}_m$, *where* $\mathbf{P}_m$ *are orthogonal projectors in the eigenspaces to eigenvalues* m *of* $\mathbf{M}$, *and* $\mathcal{M}$ *is set to* $\mathcal{M} = \sigma(\mathbf{M})$, *then the measurement is said to be* projective.

Definition 2.2.3 (*Projector*)

A linear transformation $\mathbf{P}$ *on* $\mathcal{H}$ *is a* projector *if* $\mathbf{PP} = \mathbf{P}$.

Example: We want to perform a measurement on a quantum system prepared in the uniform superposition $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:

1. Using the operators $\mathbf{M}_0 = |0\rangle\langle 0|$ resp. $\mathbf{M}_1 = |1\rangle\langle 1|$, it is easy to verify the identity $\mathbf{M}_0^\dagger\mathbf{M}_0 + \mathbf{M}_1^\dagger\mathbf{M}_1 = \mathbf{I}$. Then, the probability of observing the outcome 0 resp. 1 at the measurement is $p(0) = \langle+|0\rangle\langle 0|0\rangle\langle 0|+\rangle = \|\langle 0|+\rangle\|^2 = 1/2$ resp. $p(1) = 1/2$ and the system immediately after the measurement is in the state $|0\rangle$ resp. $|1\rangle$.
2. In observable notation, suppose that we want to measure the observable

$$\mathbf{Z} := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{10}$$

   The matrix $\mathbf{Z}$ has the spectral decomposition $\mathbf{Z} = |0\rangle\langle 0| - |1\rangle\langle 1|$. We therefore reuse the operators $\mathbf{M}_0$ and $\mathbf{M}_1$ from above as projectors to compute the probabilities of observing the outcomes 1 and -1, the eigenvalues of $\mathbf{Z}$. Unsurprisingly, we get the same result $p(1) = 1/2$ and $p(-1) = 1/2$ as we used the same measurement operators. The only thing that changes is the interpretation of the measurement outcome.

As the measurement outcome is a random variable, we can introduce some statistical definitions on the measurement observable.

Definition 2.2.4 (*Expectation and standard deviation of measurement*)

Let $|\psi\rangle$ be the state vector of a quantum system and $\mathbf{M}$ *a measurement operator. Then,*

- the expectation value *of the measurement is*

$$\langle \mathbf{M} \rangle := \langle \psi | \mathbf{M} | \psi \rangle = \sum_m mp(m), \tag{11}$$

- *and its* standard deviation *is*

$$\Delta(\mathbf{M}) := \sqrt{\langle \mathbf{M}^2 \rangle - \langle \mathbf{M} \rangle^2}. \tag{12}$$

In particular, to refer back to the Bloch sphere in Figure 2.1.1, the expectation value of an observable on a system in the state vector $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ corresponds to the Fourier coefficient of the Bloch vector of $|\psi\rangle$ in the direction of the axis supporting the eigenvectors of the observable. Specially, measuring with the operators $\mathbf{M}_0 = |0\rangle \langle 0|$ and $\mathbf{M}_1 = |1\rangle \langle 1|$ as in the example above is known as measuring in the *computational basis* and the states $|0\rangle$ and $|1\rangle$ are known as *computation basis states*. The expectation $\langle \mathbf{Z} \rangle = |\alpha|^2 - |\beta|^2$ is the z-coordinate of ψ_{bloch}. Measuring in a different basis amounts to projecting the state vector on the corresponding axis.

Most of the commercial devices [7, 8] only measure in the computational basis, i.e., only measure the observable $\mathbf{Z}$. Hence, measuring an arbitrary observable $\mathbf{M}$ involves finding a unitary transformation $\mathbf{U}$ such that $\mathbf{Z} = \mathbf{U}\mathbf{M}\mathbf{U}^\dagger$. Then,

$$\langle \mathbf{M} \rangle = \langle \psi | \mathbf{M} | \psi \rangle = \langle \psi | \mathbf{U}^\dagger \mathbf{U} \mathbf{M} \mathbf{U}^\dagger \mathbf{U} | \psi \rangle = \langle \psi' | \mathbf{Z} | \psi' \rangle \tag{13}$$

and measuring $\mathbf{M}$ on the state $|\psi\rangle$ is equivalent to measuring $\mathbf{M}$ on the state $|\psi'\rangle$.

To approximate the expectation $\langle \mathbf{M} \rangle = \langle \psi | \mathbf{M} | \psi \rangle$, one is experimentally often forced to prepare several copies of the system in the state $|\psi\rangle$, to perform the measurement on each copy, and to aggregate the measurement outcomes. A single measurement on a copy of the system is called *a shot* or *a read*. A pseudocode of the measurement process as done in the practice is summarized in Algorithm 2.2.1.

Algorithm 2.2.1 (*Measurement of operators*)

Input: *Quantum state* $|\psi\rangle$ *and measurement operator* $\mathbf{M} = \mathbf{U}\mathbf{Z}\mathbf{U}^\dagger$.
Output: *Measurement expectation value* $\langle\psi|\mathbf{M}|\psi\rangle = \sum_{m\in\mathcal{M}} m \cdot p(m)$.

Initialize $p = (0, \ldots, 0)$.
for $k = 1, 2, \ldots,$ *num_shots* ***do***
  Prepare the system in the state $|\psi'\rangle = \mathbf{U}|\psi\rangle$.
  $|\psi_m\rangle \leftarrow$ *State after measurement of* $|\psi'\rangle$ *in the computational basis.*
  $p[m] \leftarrow p[m] + \frac{1}{\text{num\_shots}}$.
end for
Return: $\langle\psi|\mathbf{M}|\psi\rangle = \sum_{m\in\mathcal{M}} m \cdot p(m)$.

An important remark on measurement is that, independently of the basis employed, it cannot distinguish between two state vectors that only differ by a global phase. That is, for some $\theta \in \mathbb{R}$ and a state $|\phi\rangle = e^{i\theta}|\psi\rangle$, we have

$$\langle\mathbf{M}\rangle = \langle\psi|\mathbf{M}|\psi\rangle = \langle\psi|e^{-i\theta}\mathbf{M}e^{i\theta}|\psi\rangle = \langle\phi|\mathbf{M}|\phi\rangle\,, \tag{14}$$

so that the global phase information plays no role in the measurement.

2.2.4 Composite system

The above postulates relate to single-particle quantum systems. To make calculations of practical interest, one needs to compose multiple single-particle systems. The last postulate extends the notion of state space from single to composite systems:

Postulate 2.2.4 (*Composite system*)

The state space of an n*-particle quantum system,* $n \in \mathbb{N}$*, is a* tensor product $\mathcal{H} = \otimes_{i=1}^n \mathcal{H}^{(i)}$ *of the state spaces* $\mathcal{H}^{(i)}, i = 1, \ldots, n$*, of the one-particle systems. The state vector*

$$|\psi(t)\rangle = \otimes_{i=1}^n |\psi^{(i)}(t)\rangle = \sum_{q\in\{|0\rangle,|1\rangle\}^{\otimes n}} \alpha_q(t)\,|q\rangle \tag{15}$$

is the tensor product *of the* n *single state vectors.*

Similarly to state vectors, operations on composite quantum systems are often achieved by combining operators on single systems using the tensor product. As an example, considering a two qubit state vector $|\psi\rangle \otimes |\phi\rangle$, one is interested in measuring the second qubit $|\phi\rangle$. The measurement operator family for this reads $\{\mathbf{I} \otimes \mathbf{M}_m, m \in \mathcal{M}\}$, i.e, a tensor product of the identity matrix and measurement operators for single qubits. In matrix representation, the tensor product turns into the Kronecker product:

Definition 2.2.5 (*Kronecker product*)

Let $\mathbf{A}$ *and* $\mathbf{B}$ *be two matrices of dimensions* $k \times l$ *and* $m \times n$*. The* Kronecker product *of* $\mathbf{A}$ *and* $\mathbf{B}$ *is the* $km \times ln$ *block matrix*

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} \mathbf{A}_{11}\mathbf{B} & \cdots & \mathbf{A}_{1l}\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{k1}\mathbf{B} & \cdots & \mathbf{A}_{kl}\mathbf{B} \end{pmatrix}. \tag{16}$$

We will circumstantially adopt the notations $|\psi\phi\rangle$, $|\psi, \phi\rangle$, and $|\psi\rangle\,|\phi\rangle$ to designate the tensor product $|\psi\rangle \otimes |\phi\rangle$ of the two state vectors $|\psi\rangle$ and $|\phi\rangle$. Also, when dealing with computational basis states $\{|0\rangle, |1\rangle\}^{\otimes n}$, it will be convenient to consider, instead of binary (bit) strings, their decimal representations. As an example, the state

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ 1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{17}$$

will simply be abbreviated as $|3\rangle$, as 11_2 in basis 2 is 3_{10} in basis 10.

We see that if $\mathcal{H}^{(i)} = \mathbb{C}^2$ for $i = 1, \ldots, n$, then state space of the composite system $\mathcal{H} = (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$ grows exponentially with n. This capacity of quantum computers to store information with exponentially fewer resources is integral to a potential quantum advantage over classical computing. Moreover, the capability of a qubit of being in superposition is one of the core properties of quantum computation because it allows for a form of high parallelism. In fact, when applied to the system state, a single linear mapping can simultaneously act on the amplitudes of all basis states. This sets quantum computers apart from their classical analogues, which can only perform sequential calculations [2].

There is an exception to Postulate 2.2.4 known as *entanglement*:

Definition 2.2.6 (*Entanglement*)

An entangled *quantum state* $|\psi\rangle$ *is a state vector that* cannot *be written as the tensor product* $|a\rangle \otimes |b\rangle$ *of any two single qubit states* $|a\rangle$ *and* $|b\rangle$*, or more.*

Entanglement is one of the most interesting and intriguing concepts of quantum computing and has no classical analog.

Example : Examples of entangled states include the well-known Bell states

$$|\beta_{ab}\rangle := \frac{|0, b\rangle + (-1)^a\,|1, 1-b\rangle}{\sqrt{2}}, \tag{18}$$

with $a, b \in \{0, 1\}$. As a special case, one cannot find single-qubits states whose tensor product gives the Bell $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

$\mathbf{I} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Identity gate
$\mathbf{X} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Pauli X-gate
$\mathbf{Y} := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Pauli Y-gate
$\mathbf{Z} := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Pauli Z-gate
$\mathbf{H} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Hadamard gate
$\mathbf{S} := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	S-gate

Table 1: *Examples of non-parametric single qubits quantum gates.*

2.3 Quantum Gates and Operations

Quantum gates are impulses applied to a quantum system, taking it from one state into another. Mathematically, they are expressed as unitary operators acting on state vectors. Quantum gates can be classified into single-, composite-, multi- and parameterized gates.

2.3.1 Single-Qubit Quantum Gates

A *single qubit quantum gate* acts on a single qubit, i.e., on the state vector of a one-particle system. Some of the most-used single qubit gates are listed in Table 1. They act on basis state qubits as follows:

$$\mathbf{I}\,|0\rangle = |0\rangle \quad \text{and} \quad \mathbf{I}\,|1\rangle = |1\rangle \tag{19}$$
$$\mathbf{X}\,|0\rangle = |1\rangle \quad \text{and} \quad \mathbf{X}\,|1\rangle = |0\rangle \tag{20}$$
$$\mathbf{Y}\,|0\rangle = -i\,|1\rangle \quad \text{and} \quad \mathbf{Y}\,|1\rangle = i\,|0\rangle \tag{21}$$
$$\mathbf{Z}\,|0\rangle = |0\rangle \quad \text{and} \quad \mathbf{Z}\,|1\rangle = -\,|1\rangle \tag{22}$$
$$\mathbf{H}\,|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{and} \quad \mathbf{H}\,|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{23}$$
$$\mathbf{S}\,|0\rangle = |0\rangle \quad \text{and} \quad \mathbf{S}\,|1\rangle = i\,|1\rangle\,. \tag{24}$$

The identity gate is essentially a "no-operation" and particularly useful when composing operators. The Pauli-gates are named after *Wolfgang Pauli*: The Pauli X-gate is a NOT-gate. The Pauli Y-gate acts as a phase gate in addition to flipping the qubits. The Pauli Z-gate changes the phase of the qubit only if it is in the state $|1\rangle$. Of particular interest is the Hadamard gate. Applied to the basis states, the Hadamard gate creates perfect superposition states. Lastly, the S-gate is a

phase gate that changes the phase of the qubit to i if it is in the state $|1\rangle$, and does nothing otherwise.

2.3.2 Composition of Single Qubit-Gates

The matrix representation of several single qubit gates is, just as composite systems, typically constructed by composing unitary matrices of single gates using the tensor product. A great representative is the Hadamard gate. On two qubits, the Hadamard gate reads

$$\mathbf{H}^{\otimes 2} = \mathbf{H} \otimes \mathbf{H} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \tag{25}$$

On n qubits in the $|0\rangle$ state, the Hadamard gate creates the perfect superposition n-qubits state vector:

$$|+\rangle^{\otimes n} := \mathbf{H}^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{q=0}^{2^n-1} |q\rangle. \tag{26}$$

2.3.3 Multi-qubit quantum gates

Multi-qubit quantum gates are those whose inputs are more than one qubit. A special case of multi-qubits gates are *controlled gates*. These are gates that act on target qubits conditionally on the state of some control qubits. As an example, for a two qubits system in the state $|\psi, \phi\rangle$, we would like to apply the unitary

$$\mathbf{U} := \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix} \tag{27}$$

on the second qubit if and only if the first qubit is the $|1\rangle$ state. The controlled-$\mathbf{U}$ gate for this reads

$$c\mathbf{U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix}. \tag{28}$$

As one can see, controlled gates *cannot* be expressed as a Kronecker product of single independent quantum gates. To simplify the writing, we will adopt the notation $c\mathbf{U}\,|\psi, \phi\rangle = \mathbf{I} \otimes \mathbf{U}^{\psi}\,|\psi, \phi\rangle$ to indicate that we apply $\mathbf{I} \otimes \mathbf{U}$ to $|\psi, \phi\rangle$ if $\psi = 1$ and the identity $\mathbf{I} \otimes \mathbf{I}$ otherwise. Controlled gates are usually used to create entangled states. The case $\mathbf{U} = \mathbf{X}$ is very common and is called the CNOT gate.

2.3.4 Parametric Quantum Gates

The above presented gates are non-parametric gates. Of particular interest for optimization using near-term quantum algorithms are *parametric quantum gates*.

$R_x(\theta) := \begin{pmatrix} \cos\frac{\theta}{2} & -i\cdot\sin\frac{\theta}{2} \\ -i\cdot\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$	R_x rotation gate
$R_Y(\theta) := \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$	R_y rotation gate
$R_z(\theta) := \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$	R_z rotation gate
$Ph(\theta) := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$	Phase gate

Table 2: *Examples of parametric single quantum gates.*

These gates are parameterized by a set of modifiable variables, typically considered as angles. The standard form of 2×2 parametric gate is given by

$$\mathbf{U}(\theta, \phi, \lambda) := \begin{pmatrix} \cos(\frac{\theta}{2}) & e^{-i\lambda}\sin(\frac{\theta}{2}) \\ e^{-i\phi}\sin(\frac{\theta}{2}) & e^{-i(\phi+\lambda)}\cos(\frac{\theta}{2}) \end{pmatrix} \tag{29}$$

with $\theta \in [0, \pi]$ and $\lambda, \phi \in [0, 2\pi]$. Typical representatives for this group are given in Table 2. Up to global phase factors, all non-parametric 2×2 gates are special cases of this parametric $\mathbf{U}$ gate. The Pauli X-gate, for example, can be expressed as $R_x(\pi) = i \cdot \mathbf{U}(\pi, 0, 0)$.

2.3.5 Quantum Gates in the Bloch Sphere

Quantum gates are unitary and norm-preserving operators, just as rotation matrices.

Lemma 2.3.1 (*Quantum gates as 3D rotations, based on [12, Exercise 4.8]*)

Any unitary transformation $\mathbf{U}$ *of a single qubit state vector can be expressed as*

$$\mathbf{U} = e^{i\alpha} \cdot R_{\vec{n}}(\theta), \tag{30}$$

$$R_{\vec{n}}(\theta) : = \cos(\theta/2)\mathbf{I} - i\sin(\theta/2)(n_x \cdot \mathbf{X} + n_y \cdot \mathbf{Y} + n_z \cdot \mathbf{Z}), \tag{31}$$

and can be seen, up to a global phase factor, as a rotation in the $3D$ *real space of the Bloch sphere by the angle* $\theta \in \mathbb{R}$ *and about the axis* $\vec{n} = (n_x, n_y, n_z)^\top \in \mathbb{R}^3$*, with* $\|n\|^2 = 1$.

Proof : The operator $\mathbf{U}$ is unitary and diagonalizable, hence admits the decomposition

$$\mathbf{U} = \Psi\Sigma\Psi^\dagger \tag{32}$$

where $\Psi \in \mathbb{C}^{2\times 2}$ is a unitary matrix comprising column-wise the eigenvectors of $\mathbf{U}$ and $\Sigma = \mathrm{diag}([\sigma_1, \sigma_2]) \in \mathbb{C}^{2\times 2}$ a matrix comprising its eigenvalues. Since $\mathbf{U}$ is unitary, its

eigenvectors are complex numbers with unit modulus, hence there exists a real matrix $\Lambda = \mathrm{diag}([\lambda_1, \lambda_2]) \in \mathbb{R}^{2\times 2}$ such that $\Sigma = e^{i\cdot\lambda}$, yielding $\mathbf{U} = e^{i\cdot\psi\Lambda\psi^\dagger}$.

We express $\Psi\Lambda\Psi^\dagger \in \mathbb{C}^{2\times 2}$ in the basis $\{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$. These matrices, i.e., the identity together with Pauli matrices, form a basis of $\mathbb{C}^{2\times 2}$. Since the matrix Λ is real, there is a real value $v_0 \in \mathbb{R}$ and a real vector $v = (v_x, v_y, v_z)^\top \in \mathbb{R}^3$ such that

$$\Psi\Lambda\Psi^\dagger = v_0 \cdot \mathbf{I} + v_x \cdot \mathbf{X} + v_y \cdot \mathbf{Y} + v_z \cdot \mathbf{Z}, \tag{33}$$

and

$$\mathbf{U} = e^{i\cdot(v_0\cdot\mathbf{I})} e^{i\cdot(v_x\cdot\mathbf{X}+v_y\cdot\mathbf{Y}+v_z\cdot\mathbf{Z})}. \tag{34}$$

By setting $\alpha = v_0$, $\theta = -2\|v\|$, and $n = -\frac{2}{\theta}v$, we arrive at

$$\mathbf{U} = e^{i\cdot\alpha} \underbrace{e^{-i\frac{\theta}{2}\cdot(n_x\cdot\mathbf{X}+n_y\cdot\mathbf{Y}+n_z\cdot\mathbf{Z})}}_{:=R_{\vec{n}}(\theta)}. \tag{35}$$

Lastly, let $\mathbf{A} := n_x \cdot \mathbf{X} + n_y \cdot \mathbf{Y} + n_z \cdot \mathbf{Z}$ be the matrix operator in $R_{\vec{n}}(\theta)$. The matrix $\mathbf{A}$ is Hermitian and unitary, yielding $\mathbf{A}^2 = \mathbf{I}$. By matrix exponentiation, we have

$$\begin{aligned}
\exp(i\theta\mathbf{A}/2) &= \sum_{k=0}^{\infty} \frac{(i\theta\mathbf{A}/2)^k}{k!} && (36)\\
&= \sum_{k=0}^{\infty} \frac{1}{(2k)!} (i\theta\mathbf{A}/2)^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} (i\theta\mathbf{A}/2)^{2k+1} && (37)\\
&= \sum_{k=0}^{\infty} \frac{1}{(2k)!} (i\theta/2)^{2k} \mathbf{A}^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} (i\theta/2)^{2k+1} \mathbf{A}^{2k+1} && (38)\\
&= \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} (\theta/2)^{2k} \mathbf{I} + i \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} (\theta/2)^{2k+1} \mathbf{A} && (39)\\
&= \cos(\theta/2)\mathbf{I} + i\sin(\theta/2)\mathbf{A}. && (40)
\end{aligned}$$

Substituting this back into $R_{\vec{n}}(\theta)$ from Equation (35), we get

$$R_{\vec{n}}(\theta) = \cos(\theta/2)\mathbf{I} - i\sin(\theta/2)(n_x \cdot \mathbf{X} + n_y \cdot \mathbf{Y} + n_z \cdot \mathbf{Z}). \tag{41}$$

The matrix $R_{\vec{n}}(\theta)$ is a special unitary matrix and as such in correspondence with a spacial rotation matrix in $\mathbb{R}^3$, namely the rotation about the angle $\theta \in \mathbb{R}$ and axis $n \in \mathbb{R}^3$, with $\|n\|^2 = 1$. The global phase $e^{i\cdot\alpha}$ is ignored in the measurement. This concludes the proof. ■

Example: For the Hadamard gate, it holds $\theta = -\pi$ and $n = \frac{\sqrt{2}}{2}(1, 0, 1)^\top$.

2.4 Universal Quantum Computing

Universal quantum computing is an abstract representation of a quantum Turing machine [30].

Conceptually, a Turing machine, as proposed by Alan Turing in 1937 [31], is a simple and mathematical abstraction of what can be computed. It models a machine that mechanically operates on a tape. On that tape are symbols from a

finite alphabet that the machine can read and write, one at a time, using a tape head and only a small set of simple universal operations like "If the symbol in state 17 is 0, write 1 and move to state 18". Despite its simplicity, a famous thesis known as Church-Turing [32] thesis states that any function on natural numbers can effectively be computed on the universal Turing machine, and conversely that every function that is un-computable on the Turing machine is considered to be un-computable. This minimalist functioning of Turing machines unfortunately make them limited in practice. Modern classical computers are based on the Random Access Memory (RAM) model that is computationally equivalent to the Turing model, but allows accessing any variable in memory.

Quantum computing started from an article by Feynman in 1982 [1], in which he explained that a classical computer would be incapable of simulating complex quantum physics, and that the latter should better be simulated by computers of the same nature, i.e., quantum computers. In 1985, David Deutsch [33] firstly investigated a quantum version of the Turing machine. In this early representation, a quantum Turing machine is a machine capable of performing quantum operations that can be efficiently described by classical Turing machine computations. In 1989, Deutsch [31, 34] matured the idea and introduced universal quantum circuits as a family of simple operations or gates, together with quantum 'unit wires', and adequate for constructing networks with any possible quantum computational property. Indeed, it is proven that any quantum gate can be decomposed into a finite sequence of gates acting only on a small number of qubits. Those finite sets of basis gates are so-called *universal quantum gates*. Example sets of universal gates for quantum computation, c.f. [35], include

$$\{R_x(\theta), R_y(\theta), R_z(\theta), Ph(\theta), \text{CNOT}\} \tag{42}$$

or

$$\{\mathbf{H}, \mathbf{S}, Ph(\pi/4), \text{CNOT}\}\,. \tag{43}$$

This quantum circuit model is the currently adopted general conception of universal quantum computing. As described in the previous section, quantum computers are powerful with exponentially more states than classical computers and capable, due to superposition and linearity of quantum gates, of accessing all the states at once.

While powerful, quantum computers are still a distance away from being easily applicable to solving practical problems. Often, quantum computers still need to be paired and aided by classical ones in order to be effective in problem solving. In addition to that, a universal quantum computer will be assumed to minimally have the following requirements due to DiVincenzo in 1993 [36]:

- It should operate on a suitable Hilbert space, whose dimension corresponds to the degrees of freedom required to hold data and perform computation. This Hilbert space should be "enumerable", i.e., should be composed of a finite and known number n of qubits.

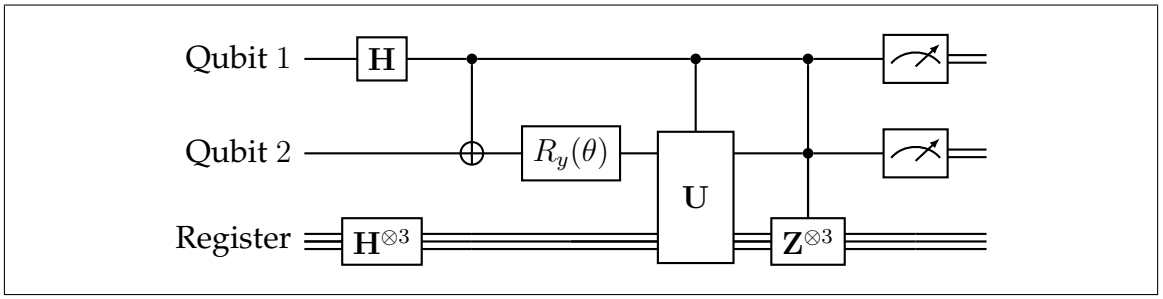


Figure 2.4.1: *A circuit diagram with five qubits and diverse types of quantum gates. Two of the five qubits are modelled by individual wires, the remaining three are grouped together, forming a register. Quantum gates are indicated by boxes on the wires, with letters indicating their names:* $\mathbf{H}$ *and* R_x *are unary gates;* $\mathbf{U}$ *is a multi-qubit gate;* $\oplus$ *is a shortcut for the NOT gate; Controlled and multi-controlled operations are modelled by filled circles on the controlled qubits, followed by lines ending with gates to be performed on target qubits; Measurement is indicated by the meter symbol, as those ending qubits* 1 *and* 2*; A double wire represents a classical bit carrying the result of the measurement.*

- It should be able to prepare the qubits in the computational basis in a finite number of steps. In practice, qubits are often initialized in the state $|0\rangle^{\otimes n}$.
- It should be isolated to a high degree from coupling to its environment. This isolation requirement is linked up with the precision required in quantum computation: the state of the computer at two instants separated by one clock cycle should ideally differ by only a small amount.
- It should physically be able to perform any unitary evolution of the qubits using a set of universal gates. For example, to apply a CNOT gate to any pair of qubits.
- It should be able to measure the qubits in the computational basis.

To fix notation and to provide a short tutorial for readers with a computer science background that are unfamiliar with the particulars of quantum computing, we shortly introduce *circuit diagrams* as abstractions of computation within universal quantum computers. An exemplary diagram with diverse types of gates is displayed in Figure 2.4.1. The diagram is to be read from left to right. Qubits are represented by single wires. Sometimes, when no discrimination on a set of qubits is required, some wires are grouped together, forming a so called *register*. Unless otherwise specified, the qubits are assumed to be initialised in the $|0\rangle^{\otimes n}$ state.

2.5 Adiabatic Quantum Computing

Adiabatic Quantum Computing (AQC) works in a way that is very different to universal quantum computing [3, 26]. Instead of describing the evolution by a sequence of unitary operators, adiabatic quantum computing relies on a smoothly varying Hamiltonian governing the Schrödinger equation.

The adiabatic theorem framing the results of AQC is a well-studied theorem in quantum physics [37]. It originated from the adiabatic hypothesis (appeared originally in German as "Adiabatenhypothese") formulated in 1914 by Einstein in its work on quantum theory. A century ago, Einstein said:

> ❛In the case of an adiabatic –reversible– influence on a system, allowed movements always change into allowed ones.❜

In 1919, Ehrensfest [38] emphasized from the hypothesis that if the system was in a state characterized by certain quantum numbers before the adiabatic change, its state after the change would preserve those quantum numbers:

> ❛The quantized effect variables of an allowed movement remain constant in the transition from that movement to an adiabatically related one.❜

This notion of adiabatic invariance is the key of the adiabatic theorem. It is proven for the first time by Born and Fock in 1928 [39] and modernly formulated as:

> ❛If the system was initially in a state with a certain number, given an adiabatic change, the probability of the system going to a state with a different number is infinitely small, although the energy levels after the change may differ from their initial values by finite amounts.❜

First applications of adiabatic quantum computing for combinatorial optimization were introduced in 1989 by Apolloni et al. [40] and in 2000 by Farhi et al. [26]. The basic idea of the optimization approach is to encode the solution of an optimization problem as a potential energy, often the ground state –the state with the lowest energy– of the varying Hamiltonian $\mathbf{H}(t)$ of the Schrödinger equation. In this optimization regime, the quantum system starts from an easy to prepare ground state of an initial Hamiltonian $\mathbf{H}(0)$ and evolves to that of a final Hamiltonian $\mathbf{H}(T)$ whose ground state is the solution of an optimization problem.

The adiabatic theorem says that if the evolution varies sufficiently slowly, the system will instantaneously track the ground state of the Hamiltonian $\mathbf{H}(t)$ and will end up in the ground state of $H(1)$ with high probability. A generic, non-rigorous formulation of the adiabatic theorem is as follows:

Theorem 2.5.1 (*Adiabatic theorem, adapted from [3]*)

Let $t \in [0,T], T \in \mathbb{N}$ and $s := \frac{t}{T} \in [0,1]$. Let $\mathbf{H}(s)$ be a time-dependent Hamiltonian and $|\epsilon_j(s)\rangle$ denote an eigenstate to the possibly degenerated eigenvalue $\epsilon_j(s)$, i.e., there may be several linearly independent eigenvectors to eigenvalue $\epsilon_j(s)$, of $\mathbf{H}(s)$ with $\epsilon_j(s) < \epsilon_{j+1}(s), \forall s \in [0,1], j \in \{1,2,\cdots\}$. If the system is initialized in the state $|\epsilon_j(s)\rangle$, then the time evolution generated by the Schrödinger equation

$$i\hbar \frac{d}{Tds} |\psi(s)\rangle = \mathbf{H}(s) |\psi(s)\rangle , \tag{44}$$

will instantaneously keep the state $|\psi(s)\rangle$ in the eigenstate $|\epsilon_j(s)\rangle$ provided that the evolution varies sufficiently slowly.

Rigorous proofs of the adiabatic theorem with convergence bounds were first given by Kato in 1950 [41] and adapted several times by, among others, Jansen et al. [37] in 2008, with evidence of the dependence of the success of the optimization on the spectral gap of the Hamiltonian $\mathbf{H}(s)$. They show that the run time T of the adiabatic optimization should be in the order of $\mathcal{O}(1/\Delta^2)$, where

$$\Delta := \min_{s\in[0,1]} \Delta(s) := \min_{s\in[0,1]} \{\epsilon_1(s) - \epsilon_0(0)\} \tag{45}$$

is the so-called *spectral gap*, the smallest difference over the time between the two lowest states energies of the time-dependent Hamiltonian $\mathbf{H}(s)$. In other words, the smaller the spectral gap of $\mathbf{H}(s)$, the longer the required evolution time should be for guaranteeing the success of the optimization. We can now loosely define adiabatic quantum computing as:

Definition 2.5.1 (*AQC, adapted from [3, Definition 1]*)

Let $s \in [0,1]$. An adiabatic computation is specified by two Hamiltonians $\mathbf{H}(0)$ and $\mathbf{H}(1)$ and a time-dependent Hamiltonian

$$\mathbf{H}(s) := \mathcal{A}(s)\mathbf{H}(0) + \mathcal{B}(s)\mathbf{H}(1) \tag{46}$$

acting on n p-state particles, $p \geq 2$, where $\mathcal{A}$ and $\mathcal{B}$ with $\lim_{s\to 1} \mathcal{A}(s) = 0$ and $\lim_{s\to 1} \mathcal{B}(s) = 1$ are scheduling functions. The ground state of $\mathbf{H}(0)$ is unique and is a tensor product state. The adiabatic computation transforms this state into an output state that is "close" in the ℓ_2 norm to a ground state of $\mathbf{H}(1)$.

2.6 Equivalence of AQC and the Universal Model

Despite their apparent differences, several references [3, 12, 26, 42] establish that the universal model of quantum computing is polynomially equivalent, in terms of computational complexity, to the adiabatic model and vice versa:

Theorem 2.6.1 (*Equivalence of AQC and the circuit model [42, 43]*)

The adiabatic model of quantum computation is polynomially equivalent to the universal model of quantum computation.

Proof: The direction indicating capability of the circuit model to simulate adiabatic evolution can be established by solving the Schrödinger equation in Theorem 2.5.1 for the unitary time evolution operator

$$i\hbar\frac{d}{Tds}\mathbf{U}(s) = \mathbf{H}(s)\mathbf{U}(s) \tag{47}$$

and setting $|\psi(1)\rangle = \mathbf{U}(1)\,|0\rangle$. The key is to show that the circuit model can implement the unitary $\mathbf{U}(1)$. We will sketch the proof briefly in order to introduce time discretization, as we will need such techniques in Section 4.1 and chapter 7 to simulate quantum evolution.

If the Hamiltonian $\mathbf{H}$ is time-independent, the Schrödinger equation has the closed-form solution $\mathbf{U}(s) = e^{i\mathbf{H}s}$. Solving the Schrödinger equation for a general time-dependent Hamiltonian $\mathbf{H}(s)$ is one of the greatest challenges of quantum computing. The following lemma adapted from Van Dam et al. tells us that we can deviate from the ideal Hamiltonian $\mathbf{H}(s)$ and approximately solve the equation without introducing a too big error:

Lemma 2.6.1 (*Van Dam, Adapted from [43, Lemma 1]*)

Let $T \in \mathbb{N}, t \in [0,T]$ and $s = t/T \in [0,1]$. Let $\mathbf{H}(s)$ and $\mathbf{H}'(s)$ be two time-dependent Hamiltonians and let $\mathbf{U}(s)$ and $\mathbf{U}(s)$ be the unitary evolutions they induce. If the difference between the Hamiltonians is limited by $\|\mathbf{H}(s) - \mathbf{H}'(s)\|_2 \leq \delta,\ \forall s \in [0,1]$, then the distance between the induced transformations is bounded by $\|\mathbf{U}(1) - \mathbf{U}'(1)\|_2 \leq \sqrt{2T\delta}$.

Proof: We follow the proof of Van Dam et al. [43, Lemma 1] and adapt the results to the change of variable $s = t/T$.

The Schrödinger equation (where for simplicity, we absorbed the term $\hbar$ in the Hamiltonian) implies that

$$\frac{d}{dt}\langle\psi'(s)|\psi(s)\rangle = -iT\,\langle\psi'(s)|(\mathbf{H}(s) - \mathbf{H}'(s))|\psi(s)\rangle\,. \tag{48}$$

Using the fundamental theorem of calculus, it follows that

$$\langle\psi'(s)|\psi(s)\rangle - \langle\psi'(0)|\psi(0)\rangle = \int_0^s -iT\,\langle\psi'(\ell)|(\mathbf{H}(\ell) - \mathbf{H}'(\ell))|\psi(\ell)\rangle\,d\ell. \tag{49}$$

Applying reverse triangle inequality holds

$$\left|\left[\,|\langle\psi'(s)|\psi(s)\rangle| - |1|\,\right]\right| \leq \left|\int_0^s -iT\left[\,\langle\psi'(\ell)|\left(\mathbf{H}(\ell) - \mathbf{H}'(\ell)\right)|\psi(\ell)\rangle\,\right]d\ell\right| \tag{50}$$

$$\leq \int_0^s \left|-iT\left[\,\langle\psi'(\ell)|\left(\mathbf{H}(\ell) - \mathbf{H}'(\ell)\right)|\psi(\ell)\rangle\,\right]\right|d\ell. \tag{51}$$

Next, we use the identity $\langle\square|\triangle\rangle = \|\square\|_2\|\triangle\|_2\cos\angle(\square,\triangle)$, with states $|\square\rangle = |\psi'\rangle$ and

$|\triangle\rangle = (\mathbf{H}(\ell) - \mathbf{H}'(\ell))\,|\psi\rangle$. Since the cosine is always smaller than one, we have

$$\left|\left[\,|\langle\psi'(s)|\psi(s)\rangle| - |1|\,\right]\right| \leq \int_0^s \left|T\left[\,\|\left(\mathbf{H}(\ell) - \mathbf{H}'(\ell)\right)|\psi(\ell)\rangle\,\|_2\right]\right| d\ell \tag{52}$$

$$\leq \int_0^s |T\delta|_2\, d\ell \tag{53}$$

$$= sT\delta, \tag{54}$$

It follows that $|\langle\psi'(s)|\psi(s)\rangle| - |1| \geq -sT\delta$ yielding $|\langle\psi'(1)|\psi(1)\rangle| \geq 1 - T\delta$. The rest of the proof follows Van Dam et al. [43]. ■

Leveraging this result, Van Dam et al. [43] show that the circuit model can effectively implement the approximated evolution

$$\mathbf{U}'(1) := \prod_{j=1}^{r} \mathbf{U}'_j(1) = e^{-i\frac{T}{r}\mathbf{H}'_r} \cdots e^{-i\frac{T}{r}\mathbf{H}'_1}, \tag{55}$$

where $\mathbf{U}'_j = e^{-i\frac{T}{r}\mathbf{H}'_j}$ and $\mathbf{H}'_j = \mathbf{H}(j \cdot \frac{T}{r})$. Indeed, if we regard $\mathbf{H}'$ as a time-dependent Hamiltonian with $\mathbf{H}'(s) := \mathbf{H}_{j(s)}$ where $j(s) = rs$, then $\|\mathbf{H}(s) - \mathbf{H}'(s)\|_2 \leq \mathrm{poly}(n)/r$ and $\|\mathbf{U}(1) - \mathbf{U}'(1)\|_2 \in \mathcal{O}(\sqrt{T \cdot \mathrm{poly}(n)/r})$, where $\mathrm{poly}(n)$ designates a polynomial in n.

The second part of the approximation consists in implementing the unitary transformations

$$\mathbf{U}'_j(1) := e^{-i\frac{T}{r}\mathbf{H}'_j} = e^{-i\frac{T}{r}\left(\mathcal{A}(j\cdot\frac{T}{r})\mathbf{H}(0) + \mathcal{B}(j\cdot\frac{T}{r})\mathbf{H}(1)\right)}. \tag{56}$$

This is usually achieved by using the Trotter formula as stated in next theorem:

Theorem 2.6.2 (*Trotter formula [12, Theorem 4.3]*)

Let $\mathbf{A}$ *and* $\mathbf{B}$ *be Hermitian operators. Then for any real number* $t \in \mathbb{R}$*, it holds*

$$\lim_{\ell\to\infty}\left(e^{it\mathbf{A}/\ell}e^{it\mathbf{B}/\ell}\right) = e^{it(\mathbf{A}+\mathbf{B})}. \tag{57}$$

In practice, one truncates the Trotter formula at $\ell = 2$ and approximates $\mathbf{U}'_j(1)$ by

$$\mathbf{U}''_j(1) := e^{-i\frac{T}{r}\mathcal{A}(j\cdot\frac{T}{r})\mathbf{H}(0)} \cdot e^{-i\frac{T}{r}\mathcal{B}(j\cdot\frac{T}{r})\mathbf{H}(1)}, \tag{58}$$

which incurs the error $\|\mathbf{U}'(1) - \mathbf{U}''(1)\|_2 \in \mathcal{O}(T^2 \cdot \mathrm{poly}(n)/r)$. Hence, the universal model can simulate the adiabatic one with a gate complexity polynomial in the number n of qubits.

The opposite direction, establishing that the adiabatic model can simulate the circuit model, was proven more recently by Aharonov et al. [42]. The challenge is to show that given an arbitrary quantum circuit $\mathbf{U}$, one can design a Hamiltonian $\mathbf{H}$ whose ground state equals the output of the circuit starting from the initial state $|0\rangle^{\otimes n}$, without knowing this output. The proof uses the "circuit-to-Hamiltonian" construction which goes back to Kitaev [44]. In this construction, one supposes that the unitary evolution operator can be decomposed into r unitaries $\mathbf{U}_1, \ldots, \mathbf{U}_r$ acting on one or two qubits each. The quantum

state after the j-th gate is denoted by $|\alpha(j)\rangle$. The goal is to design the Hamiltonian whose ground state is the entire history

$$|\eta\rangle := \frac{1}{\sqrt{r}} \sum_{j=1}^{r} |\alpha(j)\rangle \otimes |1^{\otimes j} 0^{\otimes r-j}\rangle^{c} \tag{59}$$

of the computation. The second r qubit-register is the *clock* register – denoted by the exponent c – and will help verifying the correctness of the quantum propagation at the j-th step. Aharanov [42] introduced the following 4 Hamiltonians, all having as ground state the desired state of computation with energy 0:

- The *clockinit* Hamiltonian
$$\mathbf{H}_{\text{clockinit}} := |1\rangle \langle 1|_1^c \tag{60}$$
checks whether the initial first clock qubit is correctly set. The subscript indicates on which qubit the operator is acting; in this notation, the identity operation on the remaining qubits is implied and omitted for clarity.

- The *clock* Hamiltonian
$$\mathbf{H}_{\text{clock}} := \sum_{j=1}^{r-1} |01\rangle \langle 01|_{j,j+1}^c \tag{61}$$
checks, by knowing the state of the first clock qubit, whether the subsequent clock qubits have the correct form, i.e., a sequence of 1s followed by a sequence of 0s.

- The *input* Hamiltonian
$$\mathbf{H}_{\text{input}} := \sum_{i=1}^{n} |1\rangle \langle 1|^i \otimes |0\rangle \langle 0|_0^c \tag{62}$$
ensures that if the clock qubits are in the state $|0^{\otimes r}\rangle$, then the computational qubits should be in the state $|0^{\otimes n}\rangle$.

- The *circuit* Hamiltonian
$$\mathbf{H}_{\text{circuit}} := \sum_{j=1}^{r} \mathbf{H}_j \tag{63}$$
with
$$\begin{aligned} \mathbf{H}_j :=& \mathbf{I} \otimes |100\rangle \langle 100|_{j-1,j,j+1}^c - \mathbf{U}_j \otimes |110\rangle \langle 100|_{j-1,j,j+1}^c \\ & - \mathbf{U}_j^\dagger \otimes |100\rangle \langle 110|_{j-1,j,j+1}^c + \mathbf{I} \otimes |110\rangle \langle 110|_{j-1,j,j+1}^c \end{aligned} \tag{64}$$
and boundary values
$$\begin{aligned} \mathbf{H}_1 :=& \mathbf{I} \otimes |00\rangle \langle 00|_{1,2}^c - \mathbf{U}_j \otimes |10\rangle \langle 00|_{1,2}^c \\ & - \mathbf{U}_j^\dagger \otimes |00\rangle \langle 10|_{1,2}^c + \mathbf{I} \otimes |10\rangle \langle 10|_{1,2}^c \end{aligned} \tag{65}$$
$$\begin{aligned} \mathbf{H}_r :=& \mathbf{I} \otimes |10\rangle \langle 10|_{r-1,r}^c - \mathbf{U}_j \otimes |11\rangle \langle 10|_{r-1,r}^c \\ & - \mathbf{U}_j^\dagger \otimes |10\rangle \langle 11|_{r-1,r}^c + \mathbf{I} \otimes |11\rangle \langle 11|_{r-1,r}^c \end{aligned} \tag{66}$$
describes the time evolution with operators $\mathbf{U}_j$ and $\mathbf{U}_j^\dagger$, while the clock register

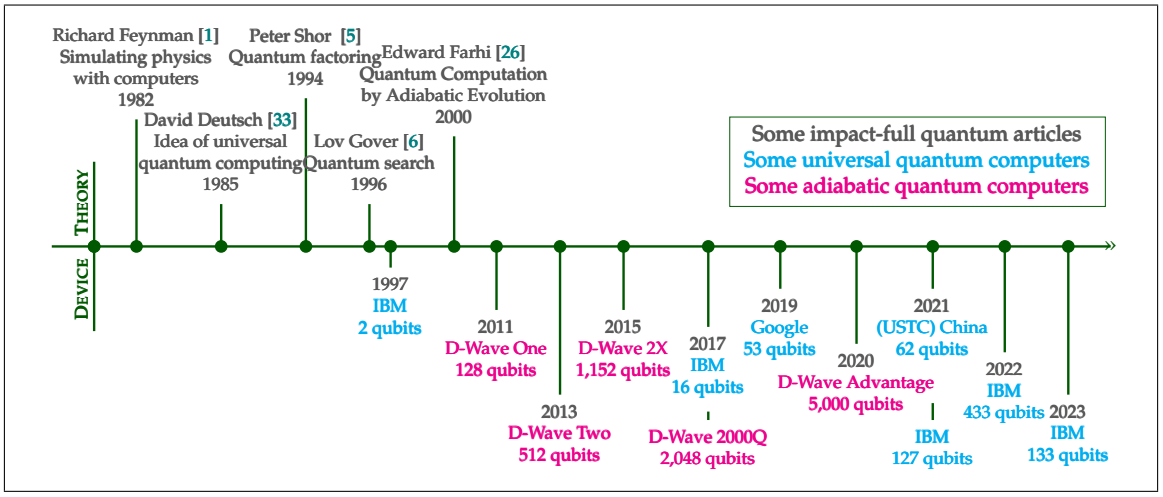


Figure 2.7.1: *Brief overview of the evolution of quantum computing over the years.* ***(Upper timeline)*** *Development of the theory.* ***(Below timeline)*** *Development of quantum devices. Enormous efforts have been made on both sides to build the foundation of quantum computing.*

state moves forwards or backwards by one step, or stays unchanged.

By defining the Hamiltonians

$$\mathbf{H}(0) := \mathbf{H}_{\text{clockinit}} + \mathbf{H}_{\text{input}} + \mathbf{H}_{\text{clock}} \tag{67}$$
$$\mathbf{H}(1) := \mathbf{H}_{\text{circuit}} + \mathbf{H}_{\text{input}} + \mathbf{H}_{\text{clock}}, \tag{68}$$

one verifies that the time-variying Hamiltonian $\mathbf{H}(s)$ for the Schrödinger equation has ground states $|0\rangle^{\otimes n+r}$ and $|\eta\rangle$ at $s = 0$ and $s = 1$ respectively, both with eigenvalues 0. ■

In summary, the adiabatic and universal models of quantum computing are equivalent. However, the translation from one model to the other often brings additional overhead, so that it is prudent to select the model that is most suitable for the practical problem to be solved.

2.7 Noisy Intermediate Scale Quantum Devices

Quantum computing is still in an early stage, and tremendous efforts are made in both research and hardware construction to enable full practically useful quantum computing.

Figure 2.7.1 provides an overview timeline on the evolution of quantum computing over the years with selected particularly impactful events.

On the research side, after the speculations on a potential quantum computer, the first practically relevant demonstration of the supremacy of universal quantum over classical computing is arguably the factoring algorithm proposed by Shor in

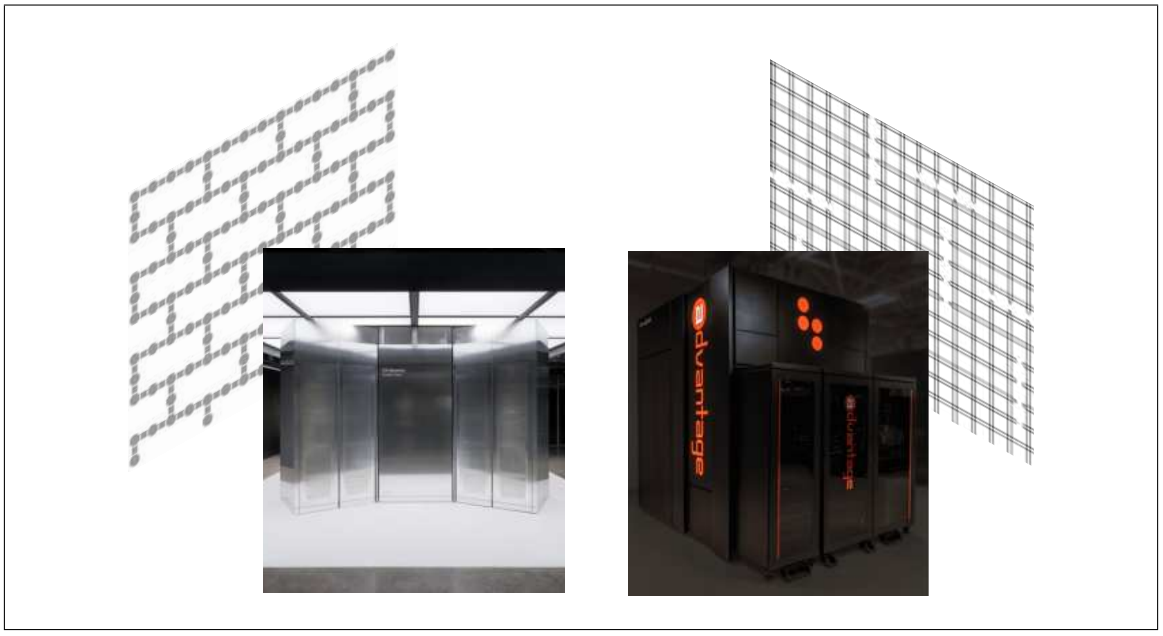


Figure 2.7.2: *Noisy intermediate scale devices as state-of-the-art devices in year* 2023. ***(Left)*** *The IBM quantum system two (foreground) running with three* 133*-qubit heron processors (node topology in the background) [7].* ***(Right)*** *The D-Wave quantum annealer (foreground) with its advantage processor having more than* 5000 *qubits (bars in the background) [8]. Noisy intermediate scale devices are still physically large, mainly because of the cooling required to keep the qubits in a coherent state. Note the sparse connectivity of the qubits for both machines.*

1994 [5], which could factorize any integer in polynomial time and thus exponentially faster than any known classical algorithm to that date. A second celebrated universal quantum algorithm was proposed by Grover in 1996 [6] for searching an element in an unsorted database quadratically more faster than any classical algorithm. In the early 2000, Farhi et al. [26] proposed an adiabatic algorithm for solving a combinatorial constraint satisfaction problem.

On the hardware development side, several quantum computer manufacturing companies have emerged. Notable examples at the time of writing this thesis were IBM [7] and Google [45] for universal and D-Wave [8] for adiabatic quantum computers. A first working two-qubit quantum computer was proposed in 1997 following the requirements of Divincenzo [36]. D-Wave released the first adiabatic quantum computer of 128 qubits in 2011. In 2023, the last releases of quantum computers with the most qubits included the 133-qubit computers from IBM and 5000-qubit quantum annealer from D-Wave.

Currently available quantum computers are composed of only a few hundreds of non-error corrected qubits [9]. These early computers are called *noisy intermediate-scale quantum (NISQ)* computers. It is practically difficult to completely isolate a quantum system, which makes it difficult to preserve quantum coherence, i.e., the time period in which the qubits still have their quantum properties to per-

form calculation. In consequence, the qubits are noisy, not corrected, and perform imperfect operations.

Another limitation of NISQ devices is their typically non-fully connected qubit topology. An example of such NISQ devices with their qubit topology is provided in Figure 2.7.2. Embedding logical qubits from the computation onto the hardware topology with limited qubit connectivity increases the computation overhead. Designing quantum algorithms that can achieve quantum advantage with NISQ devices should take into account all their limitations and should be hardware friendly. Preferably, NISQ algorithms should not require long coherence time. They should, for example, consist in very short depth quantum circuits and respect the connectivity limitations.

2.8 Conclusion

While the universal and adiabatic models of quantum computing are equivalent, current hardware does not allow to seamlessly switch between both. Therefore, from an image processing point of view, it is interesting to investigate which prototypical sub-problems are particularly amenable to one of the two models, and to compare the solution strategies.

In Chapter 4, we will investigate variational methods to solve the Ising problem and in Chapter 5 we will apply fully universal quantum methods to the same problem. Adiabatic quantum computing will be used to point sets registration in Chapter 6 and quantum evolution will serve to build the theory of quantum Hamiltonian descent in Chapter 7.

CHAPTER 3

An Overview of Variational Methods and Quantum Optimization

This chapter provides a brief overview of optimization in and with quantum computing. It builds on the basic notions provided in Chapter 2 and serves as foundation for Chapters 4, 6 and 7.

The first part of the chapter presents optimization from a general and classical point of view. In the second part we move to variational quantum methods. These are methods seeking to optimize a set of gate parameters to produce a specific output. We review the parameterization schemes and optimization strategies in variational quantum computing. The third part of the chapter is dedicated to optimization with adiabatic quantum computing. The focus here is put on the Ising model and its variants, as well as on quantum annealing on D-Wave machines.

3.1 Optimization

In finite-dimensional optimization [10, 11], the goal is to extremize, commonly to minimize, a certain real-valued function $f : \mathcal{X} \to \mathbb{R}$ on a feasible set $\mathcal{X} \subseteq \mathbb{R}^n$:

$$\inf_{x \in \mathcal{X}} f(x). \tag{69}$$

More important than the *minimum* value of f is the point $x^\star \in \mathcal{X}$ that extremizes the function. The function f, called the *objective function*, encodes a practically useful problem and the point $x^\star$, called the *minimizer*, the solution of the problem. The numerical routine used for optimizing the objective is called *optimizer* or *solver*.

Only in rare cases where the objective function is convex we can hope to robustly find a global minimizer. For non-convex problems, as is generally the case in variational quantum computing, minimizing the objective function can be extremely challenging. Finding the minimum of the objective function can classically be achieved by different ways, which can be classified in two categories: On the one hand are the gradient-based solvers, and on the other-hand the gradient-free solvers. In the following, we comprehensively review some popular classical solvers and discuss their applicability to variational quantum objectives.

3.1.1 Gradient-based Optimizers

We assume the objective function f in Equation (69) to be sufficiently smooth, i.e., continuously differentiable as often as required. Gradient-based solvers, as their name suggests, rely on optimality conditions and gradient information to find the minimizer $x^\star$ of f. A first-order necessary conditions for $x^\star$ being the minimizer of f is that the gradient ∇f vanishes at $x^\star$. This makes $-\nabla f$ one of the most-obvious, "steepest-descent" direction in which to search for the minimizer.

Intuitively, as presented in Figure 1.2.2 in the introduction, the iterates of gradient-based methods can be understood as a ball rolling downhills on a landscape, with direction and speed according to the local slope of the landscape. Given that the gradient is a local property of f, the ball has no information about the rest of the landscape. In the non-convex setting, the best we can generally expect is for the ball to roll into a local minimum; there is no guarantee of reaching the global minimum.

There exist a large variety of gradient-based solvers, ranging from p-order methods – using p-th order local approximations of f – to accelerated methods – using, in addition to the derivatives of f, also information from previous optimization steps. Below we give few common examples of gradient-based methods:

Gradient Descent.

Perhaps the most simple gradient-based method is the *gradient-descent* algorithm which typically operates in the non-constrained setting where $\mathcal{X} = \mathbb{R}^n, n \in \mathbb{N}$. It uses first-order optimality conditions to generate a series of iterates

$$x_{k+1} := x_k + \alpha_k p_k, \quad p_k := -\nabla f(x_k), \tag{70}$$

where, for $k = 0, 1, \ldots$, the term p_k is the *search direction* and α_k the so-called *step size*. One can think of the iterate x_{k+1} as the minimizer of a regularized, convex Taylor approximation of f. Indeed, one recovers the iterate x_{k+1} by simply writing down the necessary optimality condition of the problem:

$$x_{k+1} := \arg\min_{x \in \mathcal{X}} \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}. \tag{71}$$

The first two terms linearly and locally approximate f at x_k and ensure a descent direction, while the last term regularizes the local search by ensuring that the next iterate x_{k+1} will not be too far, in the Euclidean norm, from the current iterate x_k.

It is worth mentioning the crucial task of choosing of the step-size α_k. While the search direction provides a direction in which the objective function decreases, the steps size provides how long a step should be taken in that direction. Choosing the step size is a trade-off: If the step size is too small, the optimization can be too slow. On the contrary, if the step size is too large, the method can diverge. Only in rare cases, such as optimizing a convex quadratic function, a closed formula for the optimal step size can be found. In general, determining the optimal step size is more difficult and can require more evaluations of the objective function. Backtracking line-search methods [46], as an example, evaluate the objective for a sequence of step sizes and return one from the sequence satisfying certain conditions that guarantee existence of a converging sub-sequence.

Normalized Gradient Descent.

Gradient descent is inherently designed to converge towards stationary points where the gradient of the objective function vanishes. In the non-convex setting, escaping local minima or saddle points can be time-consuming. Normalized gradient descent [47] presents a natural modification of gradient descent, emphasizing the negative direction of the gradient while disregarding its amplitude. The update rule is given by

$$x_{k+1} := x_k + \alpha_k p_k, \quad p_k := -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|_2}. \tag{72}$$

Recent work by Suzuki et al. [48] identifies normalized gradient descent as a potent optimization strategy for variational quantum algorithms. Specifically, in [48], experimental evidence demonstrates that normalized gradient steps are more effective in escaping non-global minima than traditional gradient steps.

Newton's Method.

In *Newton's* method, the iterate is obtained by minimizing a second-order approximation of the function:

$$x_{k+1} := \arg\min_{x \in \mathcal{X}} \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|^2_{\nabla^2 f(x_k)} \right\}, \tag{73}$$

with the special norm in the last term defined as $\|\cdot\|_\square := \sqrt{\langle \cdot | \square | \cdot \rangle}$. In particular, Newton's method adjusts the regularization by modifying the metric on the feasible set. If the Hessian $\nabla^2 f(x_k)$ of f at x_k is positive definite, then the Newton search direction is a descent direction and reads

$$p_k := (\nabla^2 f(x_k))^{-1}(-\nabla f(x_k)). \tag{74}$$

This adjustment can be proven to locally achieve quadratic convergence, which – counting iterations – is much faster than the vanilla gradient descent method, which converges only linearly in the best case and sub-linearly in general. Of course, computing the Hessian and solving linear equation systems at every step is the cost to pay for obtaining this speed-up.

Discussion.

Optimizing quantum objectives comes with another serious challenge, that of noisy qubits and probabilistic shots. In fact, a more realistic model of quantum objective functions in the NISQ area could read

$$\inf_{x\in\mathcal{X}} \hat{f}(x), \quad \hat{f}(x) = \eta_1(f(x)) + \eta_2(x), \tag{75}$$

where f is the true function encoding the problem, η_1 is a function modelling the quantum noise and η_2 that modelling the shot noise. The noise functions differ at each iteration.

Gradient descent in the shot noise-only, i.e., measurement noise-only, setting has similarities to stochastic gradient descent, a well-known optimization technique used in machine learning [49, 50]. A classical update rule of stochastic gradient descent reads

$$x_{k+1} := x_k + \alpha_k p_k, \quad p_k := -\nabla f(x_k) + \mathcal{N}(\mu, \sigma), \tag{76}$$

where $\mathcal{N}(\mu, \sigma)$ is Gaussian noise with mean μ and variance σ. Although shot noise can severely affect function and gradient evaluations and affect descent directions, it can sometimes be useful, as it can cause the rolling ball to escape local minima and possibly find its way to better minima or even the global minimum.

More severe is quantum noise such as Pauli noise or depolarizing noise which transforms quantum states into mixed states [12, Section 8.3]: While pure states can be understood as 3-dimensional units vectors on the Bloch sphere, c.f. Figure 2.1.1, mixed states are vectors with length smaller than one and lying within the sphere. A recent work [51] has shown that quantum noise may cause vanishing gradients, a phenomenon known as noise-induced barren plateaus. Roughly, a barren plateaus tends to flatten the objective landscape. On the flat areas, the gradient vanishes exponentially with the number of qubits, so that more measurement shots is required to gain any accuracy in the gradient evaluation.

3.1.2 Gradient-free Optimizers

Gradient-free optimizers have attracted great interest in the quantum computing community. They are borrowed from classical computing where either the gradient of the objective is not available or the objective function is too noisy for optimization. An overview of gradient-free optimizers in provided in [52], see also [10]. Below, we present some representative ones that typically build on the

notion of *simplex*, which in general consists of all convex combinations of $n+1$ affinely independent points in $\mathbb{R}^n$: A line segment in one dimension, a triangle in two dimensions, a tetrahedron in three dimensions, and so on.

The Nelder-Mead Simplex Method.

Nelder and Mead [53] proposed a simplex method for function minimization. The method found great applications in quantum computing. It was, for instance, used to optimize the VQE objective function in the original publication [54]. The method iteratively compares the objective values at the vertices $\{x^0, x^1, \ldots, x^n\}$ defining a simplex, replacing from the set the point x^h with the highest objective value, that is, the vertex such that $f(x^h) = \max\{f(x^j), j = 1, \ldots, n\}$. The simplex adapts itself to the local landscape at each iteration and contracts on a final local minimum. The next x^h_{new} candidate for replacing x^h is computed as

$$x^h_{\text{new}} := -\alpha x^h + (1+\alpha)\bar{x}, \quad \text{with} \quad \bar{x} := \frac{1}{n}\sum_{j=0, j\neq h}^{n} x^j, \tag{77}$$

where α is a "reflection coefficient", a parameter chosen in the range $(0, 1)$. We see that if f is a non-constant linear function, we have $f(x^h_{\text{new}}) < f(\bar{x})$ and we expect the iteration to be successful in the sense that the new function value is better than the function value of the average point.

In the general case where f is non-linear, three situations may occur while trying to replace x^h by x^h_{new}:

1. If
$$\min_{j\neq h, j=0,\ldots,n} \left\{f(x^j)\right\} \leq f(x^h_{\text{new}}) \leq \max_{j\neq h, j=0,\ldots,n} \left\{f(x^j)\right\}, \tag{78}$$
the reflection is accepted and x^h is replaced by x^h_{new}.

2. If
$$f(x^h_{\text{new}}) < \min_{j\neq h, j=0,\ldots,n} \left\{f(x^j)\right\}, \tag{79}$$
the new vertex x^h_{new} is better than all other vertices in the simplex, and the method tries to expand this decreasing direction by computing
$$x^h_{\text{new}^\star} := \gamma x^h_{\text{new}} + (1-\gamma)\bar{x}, \tag{80}$$
where γ is the "expansion coefficient", a constant chosen greater than 1. If $f(x^h_{\text{new}^\star}) < f(x^h_{\text{new}})$, the expansion is accepted and x^h is replaced by $x^h_{\text{new}^\star}$. If this fails to happen, the expansion is rejected in favor of the reflection and x^h is replaced by x^h_{new}.

3. If
$$f(x^h_{\text{new}}) > \max_{j\neq h, j=0,\ldots,n} \left\{f(x^j)\right\}, \tag{81}$$

the reflection has failed and the method tries to contract the search direction by computing

$$x^h_{\text{new}^\star} := \beta x^h + (1-\beta)\bar{x}, \tag{82}$$

where β is the "contraction coefficient", a constant chosen in the range $(0,1)$. If $f(x^h_{\text{new}^\star}) < \min\left\{f(x^h), f(x^h_{\text{new}})\right\}$, the contraction is accepted and x^h is replaced by $x^h_{\text{new}^\star}$. If this fails to happen, the method "shrinks" the simplex and replaces each vertex x^j by $x^j_{\text{new}} := (x^j + x^\ell)/2$, where x^ℓ is the vertex with lowest objective value, i.e., the vertex for which it holds $f(x^\ell) = \min\{f(x^j), j = 0, \ldots, n\}$.

The parameters α, β, γ serve to control the factor by which the volume of the simplex changes by the reflection, contraction and expansion respectively. They should be chosen in general to avoid that the simplex collapses into a degenerated one with volume equal to zero. Typical values are $\alpha = 1$, $\beta = 1/2$ and $\gamma = 2$. Illustrative examples for these typical parameters in two dimensions are presented in Figure 3.1.1.

The COBYLA Method.

Constrained Optimization by Linear Approximation (COBYLA) is a method proposed by Powell [55], initially to extend the Nelder-Mead method to constrained optimization. It has recently found multiple applications in quantum computing because of its relatively low run-time [56, 57] compared to gradient-based solvers for the same problem. The algorithm takes into consideration a lot of factors and is only roughly sketched here.

The COBYLA method is concerned with the constrained optimization problem

$$\inf_{x\in\mathcal{X}} \quad f(x) \tag{83}$$

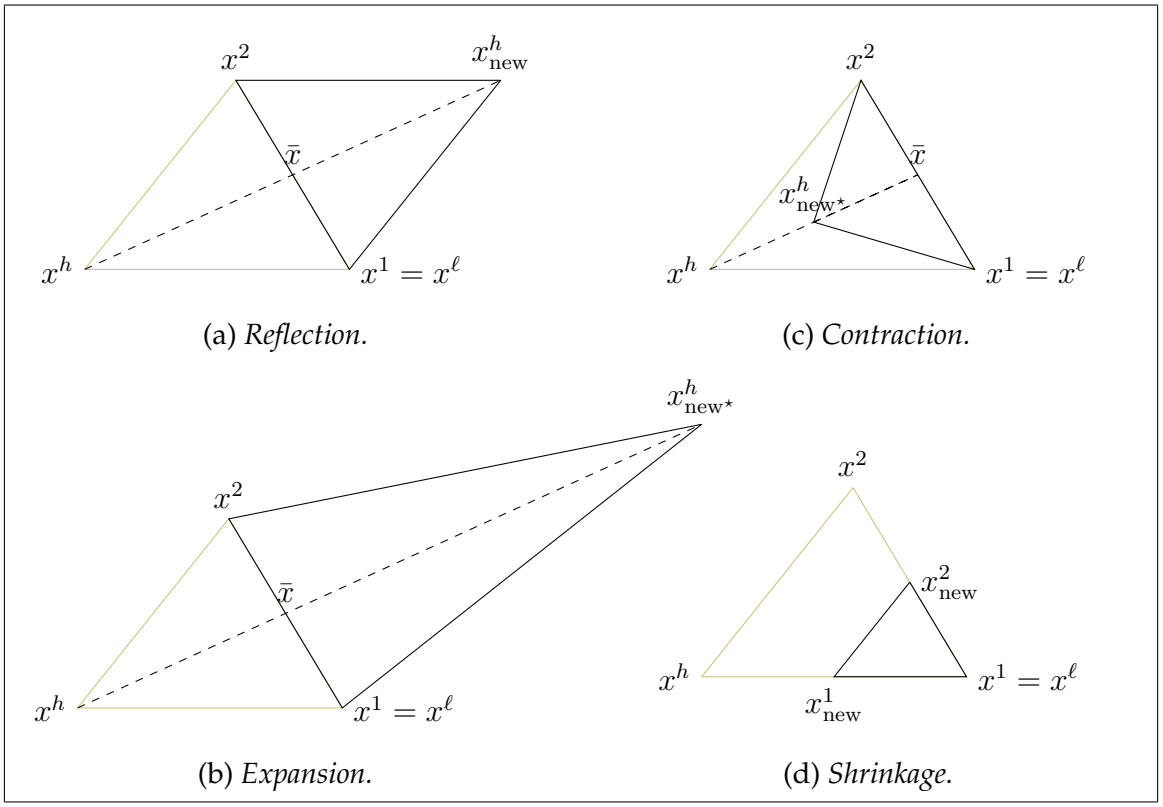
$$\text{s.t.} \quad h(x) \geq 0, \tag{84}$$

for some functions $h : \mathcal{X} \to \mathbb{R}^m, m \in \mathbb{N}$ where $h(x) \geq 0$ acts element-wise on the argument. The feasible set here is $\mathcal{X} = \mathbb{R}^n$, so that it does not encode more constraints. The method also tracks a simplex $\{x^0, x^1, \ldots, x^n\}$ and changes one vertex per iteration, but selects both the vertex to be modified and the new vertex candidate differently than Nelder-Mead.

A *merit function*

$$\psi(x) := f(x) + \mu \cdot \max\{0, \max\{-h_i(x), i = 1, 2, \ldots, m\}\} \tag{85}$$

is introduced to evaluate the quality of a point, where the parameter μ is adjusted automatically. The vertex x^h to be changed is guided by two factors: Either the current simplex is "unacceptable" and replacing x^h by a new point x_{new^Δ} will make it acceptable, or we seek to significantly decrease the value of the merit function and x^h is replaced by a new point $x_{\text{new}^\star}$:



(a) *Reflection.* (c) *Contraction.*

(b) *Expansion.* (d) *Shrinkage.*

Figure 3.1.1: *An iteration of the Nelder-Mead simplex method for function minimization. The initial simplex is presented in olive and the update in solid black. The objective function is evaluated at the vertices of the simplex, and the vertex x^h with the highest objective value is selected for replacement.* ***(a)*** *The method tries to reflect the vertex x^h.* ***(b)*** *If the new point x^h_{new} produces a new minimum, the simplex tries to expand itself in the direction of x^h_{new}.* ***(c)*** *If x^h_{new} still has the highest objective value, the simplex is contracted by a small step in the direction of x^h_{new}.* ***(d)*** *If the resulting point $x^h_{\text{new}^\star}$ is still bad, the simplex is shrunk.*

1. The current simplex is unacceptable.

   Let σ^j be the Euclidian distance of the vertex x^j to the opposite face of the current simplex and η^j be the length of the edge between x^j and the current optimal vertex x^ℓ. The current simplex is "acceptable" if and only if

   $$\sigma^j \geq \alpha\rho \quad \text{and} \quad \eta^j \leq \beta\rho, \tag{86}$$

   for $j = 1, 2, \ldots, n$. The parameter ρ controls the size of the trust region, and the parameters α, β are chosen such that $0 < \alpha < 1 < \beta$ with typical values $\alpha = 1/4$ and $\beta = 2.1$, control that the simplex does not completely shrink. If the acceptability conditions in Equation (86) are not fulfilled, the vertex x^h that violates the most either one or the other condition is replaced by

   $$x^h_{\text{new}^\Delta} := x^\ell \pm \gamma\rho n^h, \tag{87}$$

where $\gamma \in (\alpha, 1)$ is a constant with typical value $\gamma = 1/2$ and n^h is the unit vector that is perpendicular to the simplex face opposite to x^h. The sign $\pm$ is chosen according to which sign most reduces the merit function.

2. The merit function significantly decreases.

   If the current simplex is acceptable, the method still considers the vertex x^h with the worst condition value σ^j or η^j, c.f. Equation (86), as the vertex to be replaced. The method computes the new vertex $x^h_{\text{new}^\star}$ as the solution of the problem

$$\inf_{x \in \mathcal{X}} \quad \hat{f}(x) \tag{88}$$

$$\text{s.t.} \quad \hat{h}(x) \geq 0 \tag{89}$$

$$\|x - x^\ell\|_2 \leq \rho, \tag{90}$$

   where $\hat{f}$ and $\hat{h}$ are functions that linearly interpolate f and h, respectively, on the simplex vertices. If the value of the merit function at $x^h_{\text{new}^\star}$ satisfies $\psi(x^h_{\text{new}^\star}) < \psi(x^h)$, the vertex x^h is replaced. If not the method reduces the trust region ρ and moves to the next iteration.

In the original publication [55], Powell investigated some convergence properties of the method, emphasizing its suitability for optimizing functions with a small number of variables such as $n \leq 9$, as linear approximations may be inefficient.

Discussion

The Nelder-Mead simplex and the COBYLA methods pose fewer restrictions – such as differentiability – on f and individual steps are relatively efficient as no derivatives have to be computed. However, they remain heuristic algorithms that provably [58] not necessarily converge to stationary points, i.e., points with vanishing gradients.

3.2 Variational Quantum Algorithms

In this section, we discuss how to encode practical problems in quantum computing. The general way is to encode the problem into the *energy* of a quantum system [59], which serves as objective function. Given that observables are Hermitian matrices, they inherently possess real eigenvalues, thereby enabling their effective deployment for encoding optimization problems. In scenarios where the measurement operators lack Hermitian characteristics, the optimization procedure typically involves working with a real-valued function derived from the expectation.

A typical workflow of variational quantum algorithms in provided in Figure 3.2.1. It mostly consists of an hybrid quantum-classical loop which uses a quantum computer to estimate the objective function and a classical numerical solver to update the variational parameters. A detailed explanation of each figure component is provided next.

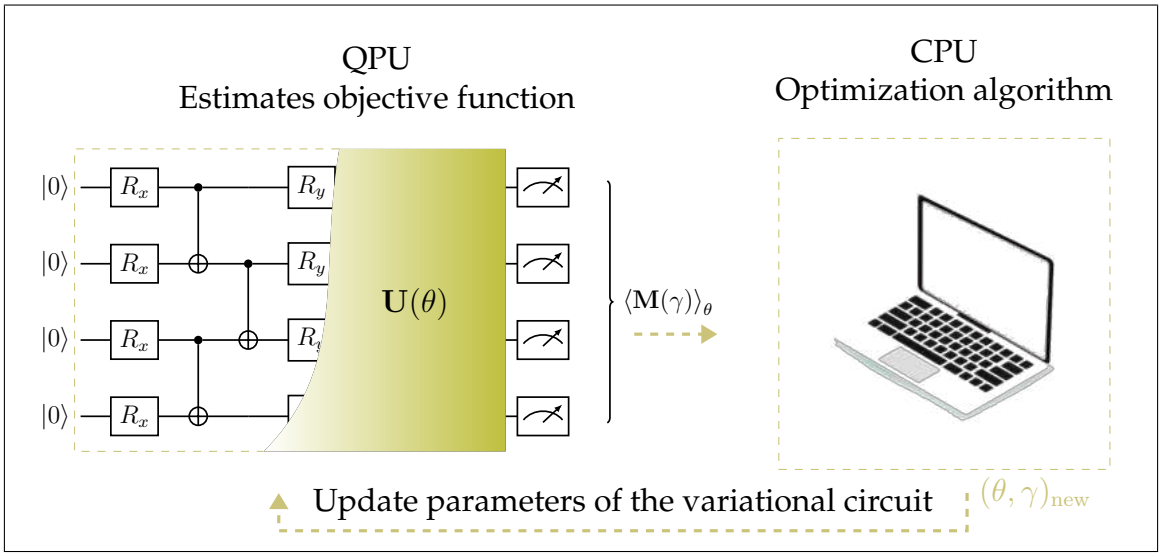


Figure 3.2.1: *Schematic representation of variational quantum algorithms, adapted from [60, Figure 2.3]. The quantum machine uses a set of variational angles to prepare and measure an ansatz* $|\psi(\theta)\rangle = \mathbf{U}(\theta)\,|0\rangle$. *The outcomes serve to estimate the objective function* $f(\theta, \gamma) = \sum_k f_k(\langle \mathbf{M}(\gamma) \rangle_\theta)$. *The resulting information is forwarded to a classical machine, which uses a numerical solver to estimate the new parameter* $(\theta, \gamma)_{\text{new}}$. *The loop is repeated until a stop criterion is met.*

3.2.1 Variational Quantum Energies

The term *variational* assumes a dual interpretation in our context: The state of the quantum system may be parameterized by a set of angles, or the observable is constructed to depend on some modifiable parameters, or both. The objective function generally takes the form

$$f(\theta, \gamma) = \sum_k f_k \left(\langle \psi(\theta) | \mathbf{M}(\gamma) | \psi(\theta) \rangle \right), \tag{91}$$

for θ resp. γ from some parameter spaces Θ resp. Γ, and some set of real valued functions $\{f_k : \Theta \times \Gamma \to \mathbb{R}\}$ [59]. The variational state $|\psi(\theta)\rangle := \mathbf{U}(\theta)\,|0\rangle$, with $\mathbf{U}$ being a quantum circuit that depends on rotation parameters θ, is called *ansatz*.

Example : The following examples illustrate how practical problems can be encoded into variational quantum energies:

1. Variational Quantum Eigensolver.

   The variational quantum eigensolver (VQE) introduced by Peruzzo et al. [54] belongs to the earliest proposals for encoding practical problems using quantum energies. It uses the variational principle of quantum mechanics to find the ground state $|q^\star\rangle$ of a quantum Hamiltonian $\mathbf{H}$. The variational principle states that the expectation $\langle \psi | \mathbf{H} | \psi \rangle$ is lower-bounded by the ground state energy $\mathcal{C}_{\text{min}}$ of $\mathbf{H}$. Indeed,

for an arbitrary superposition state $|\psi\rangle = \sum_{q=0}^{2^n-1} \alpha_q |q\rangle$, we have

$$\langle\psi|\mathbf{H}|\psi\rangle = \sum_{q=0}^{2^n-1} |\alpha_q|^2 \langle q|\mathbf{H}|q\rangle \tag{92}$$

$$\geq \sum_{q=0}^{2^n-1} |\alpha_q|^2 \langle q^\star|\mathbf{H}|q^\star\rangle =: \mathcal{C}_{\min}, \tag{93}$$

where $\mathcal{C}_{\min} := \min\{\langle q|\mathbf{H}|q\rangle, q = 0, \ldots, 2^n - 1\}$ it the minimum function value. The variational eigensolver now uses a set of angles θ to prepare the ansatz $|\psi(\theta)\rangle$. The goal is ultimately to minimize the function

$$f(\theta) = \langle\psi(\theta)|\mathbf{H}|\psi(\theta)\rangle . \tag{94}$$

The variational ansatz $|\psi(\theta)\rangle$ is prepared and measured on a quantum machine, delivering outcomes used to estimate the function value $f(\theta)$, and eventually its derivatives. A classical solver uses those estimates to compute the new parameters minimizing the objective function.

2. Variational Quantum Singular Value Decomposition.

   The variational quantum singular value decomposition (VQSVD) algorithm proposed by Wang et al. [61] seeks to find the singular value decomposition of a matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ which is assumed to be decomposable into a linear combination of unitaries as $\mathbf{M} = \sum_k c_k \mathbf{A}_k$ with unitary matrices $\mathbf{A}_k$ and real values c_k. The goal is to find the singular value decomposition $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\dagger$, where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices in $\mathbb{R}^{n,n}$ and Σ is a diagonal real matrix with r positive real entries $d_1, \ldots, d_r$, where r is the rank of $\mathbf{M}$. The idea of the variational quantum eigensolver is to search for parameters $\alpha^\star$ and $\beta^\star$ of two parameterized circuits $\mathbf{U}(\alpha)$ and $\mathbf{U}(\beta)$ that best approximate the matrix $\mathbf{M}$. The goal is achieved by minimizing the objective function

$$f(\alpha, \beta) = -\sum_{j=1}^{T} q_j \cdot \mathrm{Re}(\langle\psi_j|\mathbf{U}(\alpha)^\dagger \mathbf{M}\, \mathbf{V}(\beta)|\psi_j\rangle) \tag{95}$$

   with real positive, empirically chosen numbers $q_1 > \ldots > q_T$ and computational basis states $|\psi_j\rangle$, $j = 1, \ldots, T$ for some $T \in \mathbb{N}_{>0}$.

   The function $\Re(\cdot)$, denoting the real part of the argument, is estimated on a quantum computer and classically used to minimize the objective function f. The authors show that the objective f is minimized if and only if $\langle\psi_j|\mathbf{U}(\alpha^\star)^\dagger \mathbf{M}\, \mathbf{V}(\beta^\star)|\psi_j\rangle = d_j$, where $d_1 > \ldots > d_T$ are the largest singular values of $\mathbf{M}$. Moreover, $\langle\psi_j|\,\mathbf{U}(\alpha^\star)^\dagger$ and $\mathbf{V}(\beta^\star)\,|\psi_j\rangle$ are the left and right singular vectors, respectively.

3.2.2 Computing Quantum Gradients

Gradient-based algorithms as discussed in the previous section require to efficiently compute gradients. The question of how to compute gradients is addressed now. We start from the familiar definition of the gradient:

Definition 3.2.1 (*Gradient*)

Let $f : \mathcal{X} \to \mathbb{R}$ be a function on $\mathcal{X} \subseteq \mathbb{R}^n$. The gradient $\nabla f(x)$ *of f at $x \in \mathcal{X}$ is the vector*

$$\nabla f(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{pmatrix}, \tag{96}$$

where

$$\frac{\partial}{\partial x_j} f(x) = \lim_{\epsilon \to 0} \frac{f(x + \epsilon \cdot e_j) - f(x)}{\epsilon} \tag{97}$$

is the partial derivative of f with respect to x_j, with e_j being the the j-th canonical unit-vector.

In classical optimization, there are several strategies for computing gradients: Analytic derivation, which is error-prone and inflexible, approximation by finite differences, which is imprecise and requires careful handling in order to not affect convergence, and automatic differentiation, which is convenient but still requires additional computations. In contrast, one of the peculiarities of quantum computing is that it allows to evaluate analytical gradients for a large class of quantum objectives.

Finite Differences.

The *finite differences* method is a numerical scheme for approximating the gradient $\nabla f(x)$. It relies on the first order Taylor-approximation

$$f(x + h) = f(x) + h^\top \nabla f(x) + \mathcal{O}(\|h\|_2^2), \tag{98}$$

for an arbitrary vector $h \in \mathcal{X}$. Given a candidate implementation $\tilde{\nabla} f(x)$ of the gradient, this approximation allows, by choosing arbitrarily a vector h with a small norm, to verify the correctness of the implementation, as the j-th gradient component $(\tilde{\nabla} f(x))_j$ should be approximated by the weighted difference $\frac{1}{h_j}(f(x + h) - f(x))$ with an error in the order of $O(\|h\|_2)$. This immediately leads to the *forward differences* approximation

$$\frac{\partial}{\partial x_j} f(x_j) \approx \frac{1}{\epsilon_j}(f(x + \epsilon_j \cdot e_j) - f(x)) \tag{99}$$

for the gradient components. Similarly, the *backward differences* approximation is

$$\frac{\partial}{\partial x_j} f(x_j) \approx \frac{1}{\epsilon_j}(f(x) - f(x - \epsilon_j \cdot e_j)), \tag{100}$$

and by summing up the forward and backward formulae with $\frac{\epsilon_j}{2}$ instead of ϵ_j, one obtains the *central differences* formula

$$\frac{\partial}{\partial x_j} f(x_j) \approx \frac{1}{\epsilon_j} \left(f\left(x + \frac{\epsilon_j}{2} \cdot e_j\right) - f\left(x - \frac{\epsilon_j}{2} \cdot e_j\right)\right). \tag{101}$$

Finite differences are simple to implement and require only function evaluations. However, finite-differences methods are limited in precision, in particular with finite-precision computations. Also, the required number of function evaluations can still be very large for high-dimensional problems: In general, $2n$ evaluations are required for approximating the gradient of an n-dimensional objective f.

Parameter Shift Rule.

The parameter shift rule [62] is a simple but *exact* method for evaluating the analytical gradient of quantum objective functions with respect to ansatz parameters.

Lemma 3.2.1 (*Parameter shift rule, adapted from [62, Section E]*)

Let f be an objective function of the form

$$f(\theta) = \langle \psi(\theta) | \mathbf{M} | \psi(\theta) \rangle, \quad \textit{with} \quad |\psi(\theta)\rangle = \mathbf{U}(\theta)\,|0\rangle\,. \tag{102}$$

Furthermore, let the parameterized circuit $\mathbf{U}(\theta)$ be a chain $\mathbf{U}(\theta) := \prod_{j=1}^{n} \mathbf{U}_j(\theta_j)$ of single unitary operators $\mathbf{U}_j$ that are one-parameter dependent and in turn generated by unitary and Hermitian operators (Section 2.3.5). Then the exact partial derivative of f with respect to θ_j can be computed by

$$\frac{\partial}{\partial \theta_j} f(\theta) = \frac{1}{2}\left(f\left(\theta + \frac{\pi}{2} \cdot e_j\right) - f\left(\theta - \frac{\pi}{2} \cdot e_j\right)\right), \tag{103}$$

i.e., by evaluating the objective function twice with parameter θ_j shifted by $\pm\pi/2$.

Proof: We follow the ideas of [62]. As discussed in Section 2.3.5, we can write $\mathbf{U}_j(\theta_j) = \exp(-i\theta_j \mathbf{A}_j/2)$ with unitary and Hermitian operators A_j. The proof of the parameter shift rule uses partial derivatives of $\mathbf{U}_j$ with respect to θ_j. First, note that

$$\frac{\partial}{\partial \theta_j} \mathbf{U}_j(\theta_j) = \left(-\frac{i\mathbf{A}_j}{2}\right) \cdot \mathbf{U}_j(\theta_j). \tag{104}$$

For convenience, we will write $\mathbf{U} := \mathbf{U}(\theta)$, $\mathbf{U}_j := \mathbf{U}_j(\theta_j)$ and $\mathbf{U}_{j:k} = \mathbf{U}_j \cdots \mathbf{U}_k$. Using the chain rule for computing derivatives, we obtain

$$\frac{\partial}{\partial \theta_j} f(\theta) = \left\langle 0 \middle| \mathbf{U}_{j+1:n}^{\dagger} \left(\frac{\partial}{\partial \theta_j} \mathbf{U}_j(\theta_j)\right)^{\dagger} \mathbf{U}_{1:j-1}^{\dagger} \mathbf{M} \mathbf{U} \middle| 0 \right\rangle \tag{105}$$

$$+ \left\langle 0 \middle| \mathbf{U}^{\dagger} \mathbf{M} \mathbf{U}_{1:j-1} \left(\frac{\partial}{\partial \theta_j} \mathbf{U}_j(\theta_j)\right) \mathbf{U}_{j+1:n} \middle| 0 \right\rangle, \tag{106}$$

in which we plug in the expression in Equation (104) to get

$$\frac{\partial}{\partial\theta_j}f(\theta) = \frac{1}{2}\left\langle 0 \middle| \mathbf{U}_{j:n}^\dagger(-i\mathbf{A}_j)^\dagger\mathbf{U}_{1:j-1}^\dagger\mathbf{M}\mathbf{U} \middle| 0\right\rangle + \frac{1}{2}\left\langle 0 \middle| \mathbf{U}^\dagger\mathbf{M}\mathbf{U}_{1:j-1}(-i\mathbf{A}_j)\mathbf{U}_{j:n} \middle| 0\right\rangle, \quad (107)$$

which collapses to

$$\frac{\partial}{\partial\theta_j}f(\theta) = -\frac{i}{2}\left\langle 0 \middle| \mathbf{U}_{j:n}^\dagger\left[\mathbf{U}_{1:j-1}^\dagger\mathbf{M}\mathbf{U}_{1:j-1},\ \mathbf{A}_j\right]\mathbf{U}_{j:n} \middle| 0\right\rangle, \quad (108)$$

where $[\square,\ \triangle] = \square^\dagger\triangle - \triangle^\dagger\square$ is the commutator of $\square$ and $\triangle$.

The nice property that the generators $\mathbf{A}_j$ are Hermitian and unitary yields, for any operator $\mathbf{O}$ of compatible dimensions, the identities

$$\begin{aligned}\exp\left(-\frac{i\pi}{4}\mathbf{A}_j\right)\mathbf{O}\exp\left(\frac{i\pi}{4}\mathbf{A}_j\right) &= \frac{2}{4}\left(\mathbf{I} - i\mathbf{A}_j\right)\mathbf{O}(\mathbf{I} + i\mathbf{A}_j) \\ &= \frac{2}{4}\left(\mathbf{O} + i\mathbf{O}\mathbf{A}_j - i\mathbf{A}_j\mathbf{O} + \mathbf{A}_j\mathbf{O}\mathbf{A}_j\right), \quad (109)\\ \exp\left(\frac{i\pi}{4}\mathbf{A}_j\right)\mathbf{O}\exp\left(-\frac{i\pi}{4}\mathbf{A}_j\right) &= \frac{2}{4}\left(\mathbf{I} + i\mathbf{A}_j\right)\mathbf{O}(\mathbf{I} - i\mathbf{A}_j) \\ &= \frac{2}{4}\left(\mathbf{O} - i\mathbf{O}\mathbf{A}_j + i\mathbf{A}_j\mathbf{O} + \mathbf{A}_j\mathbf{O}\mathbf{A}_j\right). \quad (110)\end{aligned}$$

Hence,

$$\begin{aligned} i\left[\mathbf{O},\ \mathbf{A}_j\right] &= \exp\left(-\frac{i\pi}{4}\mathbf{A}_j\right)\mathbf{O}\exp\left(\frac{i\pi}{4}\mathbf{A}_j\right) - \exp\left(\frac{i\pi}{4}\mathbf{A}_j\right)\mathbf{O}\exp\left(-\frac{i\pi}{4}\mathbf{A}_j\right) \quad (111)\\ &= \mathbf{U}_j\left(\frac{i\pi}{2}\right)^\dagger\mathbf{O}\mathbf{U}_j\left(\frac{i\pi}{2}\right) - \mathbf{U}_j\left(-\frac{i\pi}{2}\right)^\dagger\mathbf{O}\mathbf{U}_j\left(-\frac{i\pi}{2}\right), \quad (112)\end{aligned}$$

where now, and according to the construction in Section 2.3.5, the parentheses indicate the argument of $\mathbf{U}_j$ and *not* a multiplication. Inserting this into Equation (108) yields

$$\begin{aligned}\frac{\partial}{\partial\theta_j}f(\theta) = \frac{1}{2}&\left\langle 0 \middle| \mathbf{U}_{j:n}^\dagger\mathbf{U}_j\left(\frac{i\pi}{2}\right)^\dagger U_{1:j-1}^\dagger\mathbf{M}\mathbf{U}_{1:j-1}\mathbf{U}_j\left(\frac{i\pi}{2}\right)\mathbf{U}_{j:n} \middle| 0\right\rangle \quad (113)\\ -\frac{1}{2}&\left\langle 0 \middle| \mathbf{U}_{j:n}^\dagger\mathbf{U}_j\left(-\frac{i\pi}{2}\right)^\dagger U_{1:j-1}^\dagger\mathbf{M}\mathbf{U}_{1:j-1}\mathbf{U}_j\left(-\frac{i\pi}{2}\right)\mathbf{U}_{j:n} \middle| 0\right\rangle, \quad (114)\end{aligned}$$

so that

$$\frac{\partial}{\partial\theta_j}f(\theta) = \frac{1}{2}\Big(f\left(\theta + \frac{\pi}{2}\cdot e_j\right) - f\left(\theta - \frac{\pi}{2}\cdot e_j\right)\Big), \quad (115)$$

This concludes the proof. ■

The parameter shift rule uses the *exact* same circuit as the function evaluation and, up to shot noise, allows to compute the *exact* gradient. Its drawback, just as that of finite differences, is that it requires $2n$ function evaluations to compute the gradient of the n-dimensional objective function f.

Simultaneous Perturbation Stochastic Approximation.

The simultaneous perturbation stochastic approximation (SPSA) method allows to approximate the gradient with only two function calls, irrespective of the number n of parameters [63, 64]. It consists of choosing a random perturbation direction $\delta = (\delta_1, \ldots, \delta_n)$ and evaluating the change in function value when simultaneously perturbating all the variables. The partial derivative with respect to x_j is proportional to this change in the function value:

$$\frac{\partial}{\partial x_j} f(x) = \frac{1}{2c\delta_j} \Big(f\left(x + c \cdot \delta\right) - f\left(x - c \cdot \delta\right) \Big), \tag{116}$$

where $c \in \mathbb{R}_{\geq 0}$ is a positive real constant. Typically, $c = \text{const}\,/(1 + \text{iter})^{\gamma}, \gamma \in \mathbb{R}$ and δ is a zero-mean vector with each entry generated from a Bernoulli ± 1 distribution with probability of $1/2$ for each ± 1 outcome. SPSA can provably find, if they exist, local and global minima of the optimization problem in approximately the same number of iterations as finite-differences-like methods, but with significantly fewer function evaluations. The basic idea of the proof [65] is to show that the difference between the expectation value of the estimated gradients up to iteration k and the true gradient of the function at iterate x_k vanishes.

The SPSA method is in some settings recommended for problems with noisy measurements of the objective function, as it is the case with quantum measurement [66, 67, 68]. Although yet not used in our experiments, it could be applied to optimize the objective function of Chapter 4.

3.3 Adiabatic Quantum Optimization

An important class of optimization problems particularly suited for adiabatic quantum computers, such as D-Wave machines, are *binary optimization problems*. Typical representatives are the Ising problem and its variants.

3.3.1 The Ising Problem

The Ising model [69] describes a quantum mechanical system with $n \in \mathbb{N}$ particles or spin systems, each of which can be in two possible states: Each spin s_i for $i = 1, \ldots, n$, can be in the states $+1$ or -1, written

$$s_i = (-1)^{q_i}, \quad q_i \in \{0, 1\}\,. \tag{117}$$

Each spin system i can interact with some external energy of strength $\mathcal{C}_{ii}$ or with an adjacent system j by a mutual interaction energy $\mathcal{C}_{ij}$. The complete system can be modelled by a general undirected n-vertices graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$ with node set $\mathcal{S} := \{s_1, \ldots, s_n\}$, $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ and a cost function $\mathcal{C} : \mathcal{E} \to \mathbb{R}$ with $\mathcal{C}(s_i, s_j) := \mathcal{C}_{ij}$ on $\mathcal{E}$. The total energy assigned by the Ising model to a system S is given by

$$\mathcal{J}(\mathcal{S}) := \sum_{i=1}^{n} \mathcal{C}_{ii} s_i + \sum_{1 \leq i < j \leq n} \mathcal{C}_{ij} s_i s_j. \tag{118}$$

The Ising problem aims to solve

$$\arg \min_{\mathcal{S}\in\{-1,1\}^n} \mathcal{J}(\mathcal{S}), \tag{119}$$

i.e., to search for a spin configuration of the system with minimal energy.

Hamiltonian Formulation.

Noting that the spin s_i is the eigenvalue of the Pauli-Z operator to the eigenvector $|q_i\rangle$, i.e., $\mathbf{Z}\,|q_i\rangle = s_i\,|q_i\rangle\,, \quad q_i \in \{0,1\}$, it follows that, on a quantum system in the state $|q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle\,, q_i \in \{0,1\}$, the energy $\mathcal{J}(S)$ is the eigenvalue to the eigenvector $|q\rangle$, or the expectation $\langle q|\mathbf{C}|q\rangle$, of the $(2^n \times 2^n)$ Hamiltonian

$$\mathbf{C} = \sum_{i=1}^{n} \mathcal{C}_{ii}\mathbf{Z}_i + \sum_{1\le i<j\le n} \mathcal{C}_{ij}\mathbf{Z}_i\mathbf{Z}_j, \tag{120}$$

where

$$\mathbf{Z}_i := \mathbf{I}\otimes\cdots\otimes\mathbf{I}\otimes \overbrace{\mathbf{Z}}^{i-\text{th position}} \otimes\mathbf{I}\otimes\cdots\otimes\mathbf{I} \tag{121}$$

denotes the Pauli-Z operator acting on the i-th particle of the system. Problem (119) can equivalently be phrased as

$$\begin{aligned} \underset{|q\rangle}{\text{minimize}} \quad & \langle q|\mathbf{C}|q\rangle \\ \text{subject to} \quad & |q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle\,,\ q_i \in \{0,1\}\,. \end{aligned} \tag{122}$$

As the Pauli-Z operator are diagonal, the observable $\mathbf{C}$ in Equation (120) is a diagonal matrix. From the variation principle – see the VQE-example from Section 3.2 – we recall that the expectation $\langle\mathbf{C}\rangle$ is lower-bounded by the smallest eigenvalue $\mathcal{C}_{\text{min}}$ of $\mathbf{C}$: For an arbitrary superposition state $|\psi\rangle = \sum_{q=0}^{2^n-1} \alpha_q\,|q\rangle$, we have

$$\langle\psi|\mathbf{C}|\psi\rangle = \sum_{q=0}^{2^n-1} |\alpha_q|^2\,\langle q|\mathbf{C}|q\rangle \tag{123}$$

$$\geq \sum_{q=0}^{2^n-1} |\alpha_q|^2\,\langle q^\star|\mathbf{C}|q^\star\rangle = \mathcal{C}_{\text{min}}. \tag{124}$$

Solving the Ising problem is practically hard. Brute-force solutions for searching for the ground state, which amounts to computing the diagonal elements of $\mathbf{C}$, require 2^n function evaluations. The problem is discrete, so that classical methods [70, 71, 72] typically solve only approximated or relaxed version of it.

3.3.2 Formulation as QUBO Problem

An equivalent and very frequently encountered formulation of the Ising problem is the formulation as Quadratic Unconstrained Optimization (QUBO) problem. In this formulation, instead of spin variables, one solves over binary variables $\mathcal{X} := \{x_1, \ldots, x_n\}$ with $x_i \in \{0, 1\}$ a problem of the form

$$\arg\min_{\mathcal{X}\in\{0,1\}^n} \quad \sum_{i=1}^{n} \mathcal{D}_{ii} x_i + \sum_{1\leq i<j\leq n} \mathcal{D}_{ij} x_i x_j + C, \tag{125}$$

with

$$\mathcal{D}_{ii} := 2\left(\mathcal{C}_{ii} - \sum_{j=i+1}^{n} \mathcal{C}_{ij} - \sum_{j=0}^{i} \mathcal{C}_{ii}\right), \quad \mathcal{D}_{ij} := 4\mathcal{C}_{ij}, \quad C := -\sum_{i=1}^{n} \mathcal{C}_{ii} + \sum_{1\leq i<j\leq n} \mathcal{C}_{ij}.$$

This is obtained by the change of variables $s_i = 2x_i - 1$.

The more concise form

$$\arg\min_{x\in\{0,1\}^n} x^\top \mathcal{D} x, \tag{126}$$

which assumes that x is a binary vector, is also very frequently used for QUBOs.

The minimizers of the two formulations Equation (119) and Equation (125) have the same binary configuration, however, their objective function values differ by the offset C. In the QUBO formulation, the binary terms $\mathcal{D}_{ij}$ and unary terms $\mathcal{D}_{ii}$ are commonly called *couplers* and *biases*.

3.3.3 Maximum Cut Problem

With the special choice $\mathcal{C}_{ii} = 0$ for all i, problem Equation (119) reduces to the so-called weighted maximum cut (MaxCut) problem [70]. In the MaxCut problem, we are interested in partitioning the set $\mathcal{S}$ of nodes of the graph into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ such that the weighted sum of edges between the two subsets is maximal.

Assigning to each partition a binary vector $x \in \{0, 1\}^n$ where $x_i = 1$ if $s_i \in \mathcal{S}_1$ and $x_i = 0$ if $s_i \in \mathcal{S}_2$, the *weighted maximum cut* is the partition that solves the problem

$$\arg\max_{x\in\{0,1\}^n} \mathcal{J}(x), \quad \mathcal{J}(x) := \sum_{i,j=1}^{n} \mathcal{C}_{ij} x_i (1 - x_j). \tag{127}$$

By setting $s_i = 2x_i - 1$, we can reformulate problem (127) as

$$\arg\max_{\mathcal{S}\in\{-1,1\}^n} \quad \sum_{i,j=1}^{n} \frac{1}{2} \mathcal{C}_{ij} (-s_i s_j + s_i - s_j + 1), \tag{128}$$

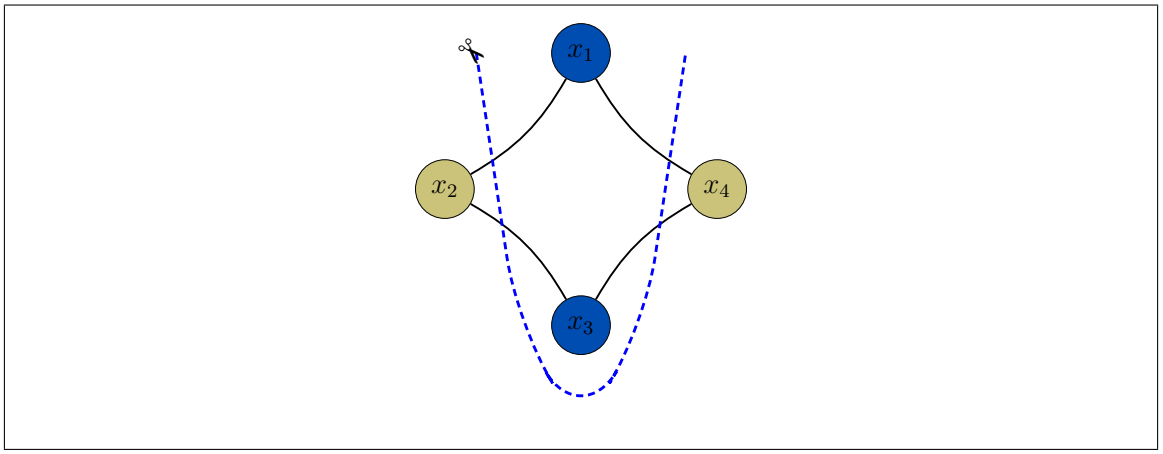


Figure 3.3.1: *Illustration of the maximum cut problem. The problem is to partition the nodes of the graph into two subsets such that the weighted sum of edges between the two subsets is maximal.*

which is, up to a constant offset and a scaling factor, equivalent to the problem

$$\arg \min_{\mathcal{S}\in\{-1,1\}^n} \quad \sum_{i,j=1}^{n} \mathcal{C}_{ij} s_i s_j. \tag{129}$$

The symmetry $\mathcal{C}_{ij} = \mathcal{C}_{ji}$ allows to write problem (129) as

$$\arg \min_{\mathcal{S}\in\{-1,1\}^n} \quad \sum_{1\leq i<j\leq n} \mathcal{C}_{ij} s_i s_j, \tag{130}$$

which is nothing else than a special form of problem (119). Thus, solving the Ising problem with $\mathcal{C}_{ii} = 0$ for all i is equivalent to finding the maximum cut of a given graph. By flipping the sign of all the couplers $\mathcal{C}_{ij}$, the problem turns into what is known as *weighted minimum cut* problem.

Example: As an example, we consider the graph in Figure 3.3.1 with cost values given by $\mathcal{C}_{12} = \mathcal{C}_{23} = \mathcal{C}_{34} = \mathcal{C}_{41} = 1$ and $\mathcal{C}_{ij} = 0$ for all $i, j = 1, \ldots, 4$ and $j \neq i + 1 \mod 4$. We can evaluate the objective function according to Equation (127) as

$$\mathcal{J}(0000) = \mathcal{J}(1111) = 0 \tag{131}$$
$$\mathcal{J}(1000) = \mathcal{J}(0100) = \mathcal{J}(0010) = \mathcal{J}(0001) = 2 \tag{132}$$
$$\mathcal{J}(0011) = \mathcal{J}(1100) = \mathcal{J}(0110) = \mathcal{J}(1001) = 2 \tag{133}$$
$$\mathcal{J}(0111) = \mathcal{J}(1011) = \mathcal{J}(1101) = \mathcal{J}(1110) = 2 \tag{134}$$
$$\mathcal{J}(0101) = \mathcal{J}(1010) = 4, \tag{135}$$

and we read out that maximizers for this problem instance are $x^\star = 1010$ or $x^\star = 0101$ and $\mathcal{J}(x^\star) = 4$.

3.3.4 Application Examples

The Ising model finds practical applications in varying fields such as computer vision [73], data clustering [74], and communications network design [75]. Reference [76] also provides more examples of how to apply the Ising model for modelling many practically relevant NP-problems. In the following, we provide some applications of the Ising model for modelling ferromagnetism in statistical mechanics, as well as combinatorial problems in image processing.

Ferromagetism.

The Ising model is a powerful tool describing a simplified 2D model of ferromagnetism in statistical mechanics [69].

The couplers $\mathcal{C}_{ij}$ describe the interaction field between spin systems i and j, it is non-zero only if i and j are neighboring particles. The biases $\mathcal{C}_i$ indicate the external magnet field applied to each particle i. Spins can be up ($\uparrow$), or down ($\downarrow$). Once excited and then left to rest, ferromagnetic objects tend towards a low-energy configuration of the spins:

- If $\mathcal{C}_{ij} > 0$, the system prefers anti-aligned neighboring spins i and j (e.g., $\uparrow\downarrow$ or $\downarrow\uparrow$). The neighbors i and j are said to be anti-ferromagnetic. If $\mathcal{C}_{ij} < 0$, aligned neighbors i and j are preferable, and they are said to be ferromagnetic.
- If $\mathcal{C}_i > 0$, the system favors a down-spin s_i; conversely an up-spin if $\mathcal{C}_i < 0$.

In practice, a major factor in the evolution of the spins is also the temperature of the experimental environment.

Chan-Vese Segmentation.

The Ising model is a one-to-one translation of a binary image segmentation problem. This segmentation approach was first introduced by Greig et. al. [77] in the image processing context and later developed into what is now known as the Chan-Vese-model [78].

Considering a discretized image represented as a vector $g \in \mathbb{R}^n$, where each component represents an intensity value for one of n discrete points in the image domain.

It is assumed that the conditional densities of the intensities $p(g_i|x_i)$ are known, i.i.d., and normally distributed with mean $c_i \in \mathbb{R}$ and variance $\sigma_i^2 \in \mathbb{R}$:

$$p(g_i|x_i = 0) = \mathcal{N}(g_i; c_0, \sigma_0^2), \quad \text{and} \quad p(g_i|x_i = 1) = \mathcal{N}(g_i; c_1, \sigma_1^2). \tag{136}$$

In order to encourage smoothness of the region boundaries, an a-priori distribution of the variable x is incorporated. A typical simple model is

$$p(x) := \exp\left(-\sum_{i\sim j} \frac{1}{2}|x_i - x_j|\right), \tag{137}$$

where the relation $\sim$ denotes neighboring pixels.

With these assumptions, a maximum-a-posteriori (MAP) estimation can be performed for x, maximizing the conditional probability $p(x|g)$:

$$x^\star := \arg \max_{x \in \{0,1\}^n} p(x|g). \tag{138}$$

Taking the monotonicity of the logarithm into account and using the Bayes rule, one can write

$$x^\star = \arg \min_{x \in \{0,1\}^n} -\log \frac{p(g|x) \cdot p(x)}{p(g)}, \tag{139}$$

which leads to

$$x^\star = \arg \min_{x \in \{0,1\}^n} \sum_{i=1}^{n} \frac{1}{2} \left((g_i - c_1)^2 - (g_i - c_0)^2 \right) x_i + \sum_{1 \le i < j \le n} \frac{1}{2} \lambda_{ij} x_i x_j, \tag{140}$$

where λ_{ij} is strictly positive and constant if i and j are neighboring points, and zero if they are not. Equation (140) is nothing else than a QUBO or equivalently an Ising problem.

Discussion.

In the ferromagnetism and segmentation examples, the couplers are typically sparse, as they are non-zero only for neighboring particles. For such examples, see Figure 3.3.2, a relatively simple method that is close to the physical evolution is the Metropolis-Hastings algorithm [79]. It works as follows: The system starts in a random configuration of the spin lattice. At each time step, a spin s_i is randomly chosen from the lattice an its contribution e_i to the cost is evaluated. If $e_i > 0$, the spin s_i is flipped, otherwise it is flipped with a probability $\exp(e_i \cdot t)$, where t is a user controllable parameter. The parameter t models the room temperature, as some physical system change their magnetic phase with the temperature.

It is worth pointing out that other more efficient classical methods for solving the Ising, under some specific assumptions, problem also exist: If the couplers $\mathcal{C}_{ij}$ are positive, the problem can be turned into a sub-modular optimization problem and solve with graph cut techniques [80, 73]. References [81, 82] also provide extensive review of combinatorial optimization algorithms.

3.3.5 Quantum Annealing on D-Wave Machines

D-Wave [8] quantum annealers are intermediate-scale devices conceived for solving Ising problems based on the AQC optimization principle. Recall from Definition 2.5.1 that the AQC model relies on an evolving Hamiltonian

$$\mathbf{H}(s) = \mathcal{A}(s)\mathbf{H}(0) + \mathcal{B}(s)\mathbf{H}(1) \tag{141}$$

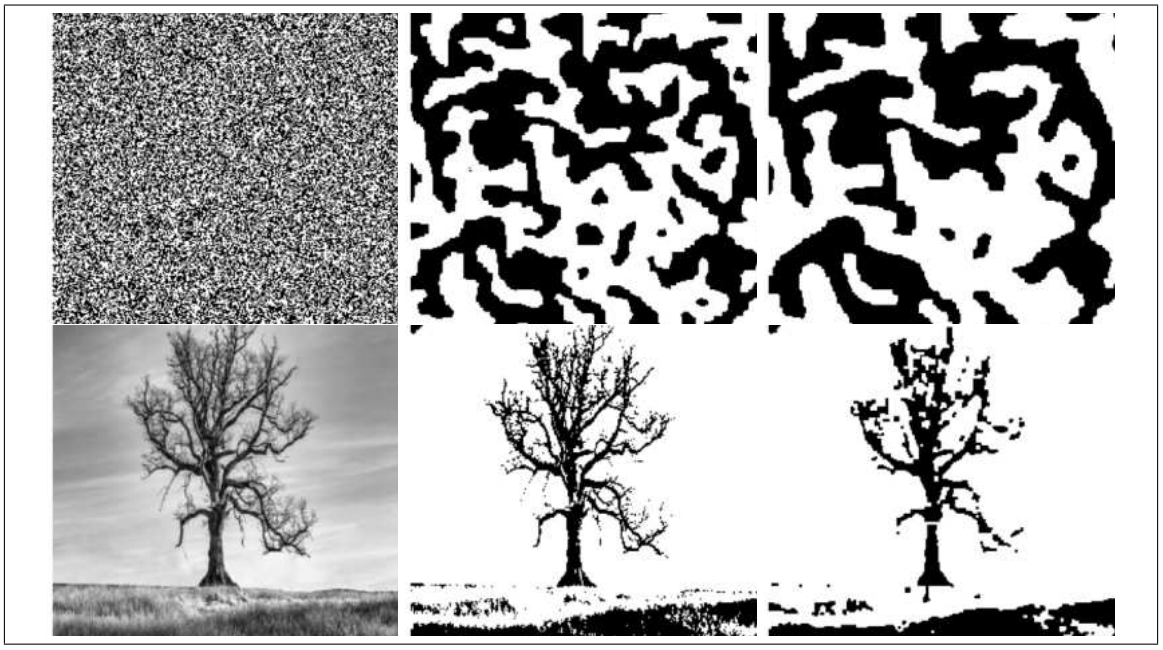


Figure 3.3.2: *Example applications of the Ising model, with approximate solutions computed using the Metropolis–Hastings algorithm.* ***(Top row)*** *Illustration of a 2-dimensional ferromagnetic lattice. Each qubit corresponds to one particle in the lattice and the spin configuration is obtained by solving the Ising problem with $C_{ii} = 0$ for all i and $C_{ij} = 1$ for all neighboring pixels i, j, so that neighboring particles prefer to align. The lattice (left) is randomly initialized and, under a certain critical temperature, converges (left to right) towards a state where each pixel takes the state of its neighbors – the material self-magnetizes.* ***(Bottom row)*** *When modifying the biases C_{ii} and couplers C_{ij} to match the Chan-Vese model in Equation (140) for a given image (left), the randomly initialized Ising lattice converges to a segmented image (middle and right). The presented segmentation results correspond to different values of the regularization terms λ_{ij}. The larger the regularization term (middle to right), the more neighboring pixels tend to align.*

where it is assumed that the ground state of the initial Hamiltonian $\mathbf{H}(0)$ is easy to prepare, and, according to the adiabatic theorem, Theorem 2.5.1, should evolve to the ground state of the problem Hamiltonian $\mathbf{H}(1)$.

On D-Wave machines, the initial Hamiltonian is $\mathbf{H}(0) = \mathbf{B}$ with

$$\mathbf{B} := \sum_{i=1}^{n} \mathbf{X}_i, \tag{142}$$

where, as above, $\mathbf{X}_i$ denotes the Pauli-X operator acting on the i-th particle of the system. The ground state of $\mathbf{B}$ is the equal superposition state $|+\rangle^{\otimes n}$.

The problem Hamiltonian $\mathbf{H}(1)$ is the Ising Hamiltonian $\mathbf{C}$ from Equation (120),

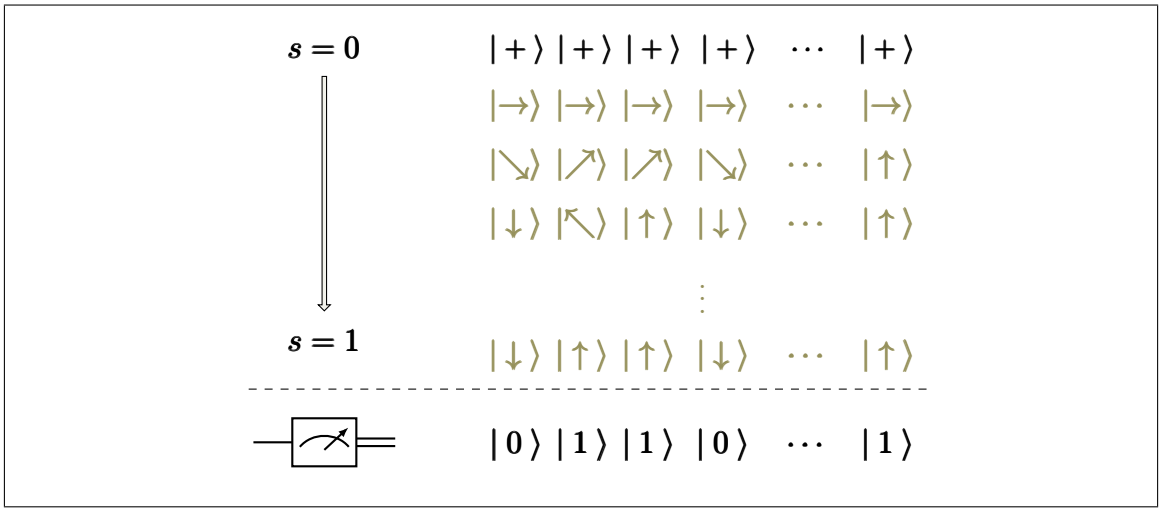


Figure 3.3.3: *An illustration of quantum annealing; image adapted from [83].* ***(Top to bottom)*** *All qubits in the quantum system start in a perfect superposition state $|+\rangle$, illustrated as $|\rightarrow\rangle$. In the annealing process, as the time s evolves from 0 to 1, the qubits are subjected to magnetic fields and tend to adopt a state configuration that globally minimizes the system's energy.* ***(Bottom)*** *At the end of the evolution, their final state is read out and returned as the solution of an optimization problem.*

i.e.,

$$\mathbf{C} := \sum_{i=1}^{n} \mathcal{C}_{ii}\mathbf{Z}_i + \sum_{1\leq i<j\leq n} \mathcal{C}_{ij}\mathbf{Z}_i\mathbf{Z}_j, \tag{143}$$

where, as the name suggests, the couplers and biases $\mathcal{C}_{ij}$ and $\mathcal{C}_{ii}$ are problem specific.

Programming on D-Wave machines consists in defining couplers and biases of a problem and sending them to a quantum processing system using a web-based API. The quantum processor creates a network of logical qubits according to the problem size, which is embedded in the quantum hardware. An intuitive understanding is provided in Figure 3.3.3. The network starts in a global superposition of all possible basis states. In a process called annealing, as the time s evolves from 0 to 1, the provided couplers and biases are changed into magnetic fields that deform the state landscape, emphasizing the state that is most likely the solution of the underlying optimization problem.

Discussion.

D-Wave is a programmable machine which allows the user to define problem specific couplers and biases. Not only problem parameters, but also solver parameters are programmable on D-Wave machines. A non-exhaustive list of programmable solver parameters include:

- Number of shots.

As mentioned earlier in Section 2.2.3, quantum measurement is a probabilistic experiment. A single shot provides a single basis-state. In order to approximate the probability on the system for being in a specific state, one needs to repeat the experiment several time and to aggregate the results. The state with the lowest energy is returned as the solution of the submitted problem. D-Wave allows the user to define the number of shots.

- Chain strength.

  The embedding of the logical problem onto the quantum hardware often faces the sparse qubit connectivity problem. In order to create non-existing connections, the QPU *chains* a set of physical qubits by setting the strength of their connecting couplers high enough to strongly correlate them. Physical qubits in a chain correspond to the same logical qubit. In case of a chain breakage, the state of the logical qubit is determined by a majority vote of the qubits in the chain. The strength of the chain on D-Wave can be specified by the user.

- Total annealing time.

  The default annealing time of one shot on D-Wave is $20\mu S$. However, as discussed and known from Theorem 2.5.1, this should be set depending on the spectral gap. Though the spectral gap is difficult to compute in general, D-Wave allows to change the annealing time.

- Annealing functions.

  The default annealing functions $\mathcal{A}(s)$ and $\mathcal{B}(s)$ presented in Figure 3.3.4 are available on D-Wave, but could be customize as well. [2]

While being an efficient scheme for combinatorial optimization that has the potential to ultimately super-cede classical computers, AQC has a deficiency. As discussed in Section 2.5, the smaller the energy gap between the ground state and the first excited state of the adiabatic Hamiltonian $\mathbf{H}(s)$, the longer the required annealing time for guaranteeing the success of the optimization [3, 37, 84].

3.4 Conclusion

On take-away of this chapter is that many classical optimization routines can be effectively employed in a hybrid quantum-classical regime to train the parameters of parameterized circuits for specific goals. Moreover, we observed that a significant class of quantum objective functions allows for the evaluation of exact analytical gradients on quantum platforms, facilitating the use of classical gradient-based solvers. Furthermore, we reviewed adiabatic quantum computing for solving combinatorial problems, as implemented by D-Wave machines.

[2] The D-Wave functions $A(s)$ and $B(s)$ correspond to the $\mathcal{A}(s)$ and $\mathcal{B}(s)$ functions of this thesis.

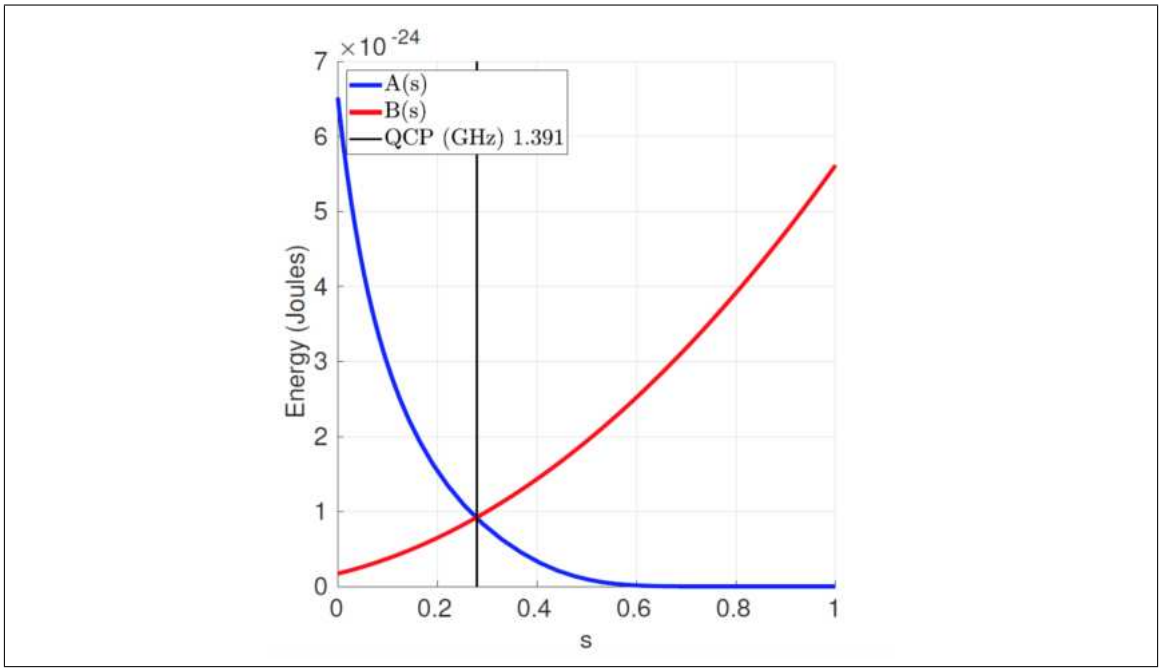


Figure 3.3.4: *Default annealing functions, see Equation* (141)*, on D-Wave machines, image taken from [8]. The function* $\mathcal{A}(s)$ *monotonically decreases, taking the system away from the eigenstates of the initial Hamiltonian* $\mathbf{H}(0)$*, whilst* $\mathcal{B}(s)$ *increases to take the system into an eigenstate of the problem Hamiltonian* $\mathbf{H}(1)$*.*

In the realm of image processing, particularly as discussed in Chapters 4 and 6, this integration of classical and quantum approaches opens up promising avenues to enhance optimization tasks, providing innovative solutions and improved performance.

CHAPTER 4

A Variational Quantum Algorithm for Ising problems

We consider an undirected n-vertices graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$ with $\mathcal{S} = \{s_1, \ldots, s_n\}$, $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ and a cost function $\mathcal{C} : \mathcal{E} \to \mathbb{R}$ with $\mathcal{C}(s_i, s_j) := \mathcal{C}_{ij}$ on $\mathcal{E}$. The purpose of this chapter is, using variational quantum methods (Chapter 3), to

$$\begin{aligned} &\underset{|q\rangle}{\text{minimize}} \quad \langle q|\mathbf{C}|q\rangle \\ &\text{subject to} \quad |q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle\,,\ q_i \in \{0,1\}\,, \end{aligned} \tag{144}$$

where

$$\mathbf{C} = \sum_{i=1}^{n} \mathcal{C}_{ii}\mathbf{Z}_i + \sum_{1\leq i<j\leq n} \mathcal{C}_{ij}\mathbf{Z}_i\mathbf{Z}_j \tag{145}$$

is the Ising Hamiltonian. In essence, the goal is to prepare a ground state $|q^\star\rangle$ of $\mathbf{C}$.

The chapter begins with a review of relevant literature. Then, the new approach is presented, which includes the construction and optimization of a parameterized quantum circuit encoding the problem. Experiments are conducted at the end of the chapter to validate and benchmark the method against the state-of-the-art quantum approximate optimization algorithm and D-Wave quantum annealers. The approach discussed in this chapter is a quantum-classical hybrid. In contrast, in Chapter 5, we will approach the problem with fully universal quantum amplitude amplification techniques.

The chapter builds on the following author's publications:

- Kuete Meli, N., Mannel, F., and Lellmann, J. "A universal quantum algorithm for weighted maximum cut and Ising problems", *Quantum Inf Process* **22**, 279 (July 2023). https://doi.org/10.1007/s11128-023-04025-x.

4.1 Related Work

The Quantum approximate optimization algorithm (QAOA), firstly introduced by Farhi et al. [85] and widely discussed in the literature [86, 87, 88, 89] is a universal algorithm mimicking the AQC computational model. Problem (144) is *approximated* by finding a state $|\psi\rangle = \sum_q \alpha_q |q\rangle$ such that $\langle\psi|\mathbf{C}|\psi\rangle$ is minimized. QAOA uses techniques presented in Section 2.6 and solves the Ising problem by *trotterizing* the evolution generated by Equation (141). Its pseudo-code is recapitulated in Algorithm 4.1.1 and its circuit in Figure 4.1.1:

Algorithm 4.1.1 (*Quantum Approximate Optimization Algorithm [85]*)

Input: *Weighted graph* $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$*, layer depth* $p \in \mathbb{N}$*,*
and initial parameter vector $(\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)_{init}$*.*
Output: *Minimizer* $|\psi\rangle = |\gamma, \beta\rangle$ *of problem* (144)*.*

Initialize parameter $(\gamma, \beta) \leftarrow (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)_{init}$
while *stopping criteria not met* ***do***
Initialize quantum system in state $|+\rangle = \sqrt{2^{-n}} \sum_{q=0}^{2^n-1} |q\rangle$*.*
Prepare $|\gamma, \beta\rangle := \mathbf{U}(\mathbf{B}, \beta_p)\mathbf{U}(\mathbf{C}, \gamma_p) \cdots \mathbf{U}(\mathbf{B}, \beta_1)\mathbf{U}(\mathbf{C}, \gamma_1) |+\rangle^{\otimes n}$*.*
Measure $|\gamma, \beta\rangle$ *in the computational basis.*
Classically compute $\langle\gamma, \beta|\mathbf{C}|\gamma, \beta\rangle$*.*
Update (γ, β) *using a classical optimizer to obtain* $(\gamma, \beta)_{new}$*.*
end while
Return: $|\psi\rangle = |\gamma, \beta\rangle$*.*

First, the system is prepared in the perfect superposition state $|+\rangle^{\otimes n}$. Then, trotterization consists in choosing a *depth* p and repeatedly – p times – applying to the state the unitaries $\mathbf{U}(\mathbf{C}, \gamma_k) := \exp(-i\gamma_k \mathbf{C})$ and $\mathbf{U}(\mathbf{B}, \beta_k) := \exp(-i\beta_k \mathbf{B})$ generated by the *problem Hamiltonian* $\mathbf{C}$ and a so-called *mixing Hamiltonian* $\mathbf{B}$ as defined in Equation (142). The (small) time steps $\gamma_k, \beta_k \in \mathbb{R}$, $1 \leq k \leq p$, are optimization parameters in QAOA. The resulting state is called $|\gamma, \beta\rangle$:

$$|\gamma, \beta\rangle := \mathbf{U}(\mathbf{B}, \beta_p)\mathbf{U}(\mathbf{C}, \gamma_p) \cdots \mathbf{U}(\mathbf{B}, \beta_1)\mathbf{U}(\mathbf{C}, \gamma_1) |+\rangle^{\otimes n} . \tag{146}$$

Repeated measurements in the computational basis are performed to estimate the state $|\gamma, \beta\rangle = \sum_q \alpha_q |q\rangle$, which subsequently classically serves to evaluate the objective function

$$\langle\gamma, \beta|\mathbf{C}|\gamma, \beta\rangle = \sum_q \alpha_q \langle q|\mathbf{C}|q\rangle . \tag{147}$$

If required by the classical solver, the derivatives of the objective function may be calculated as well. The solver subsequently updates the parameter vector

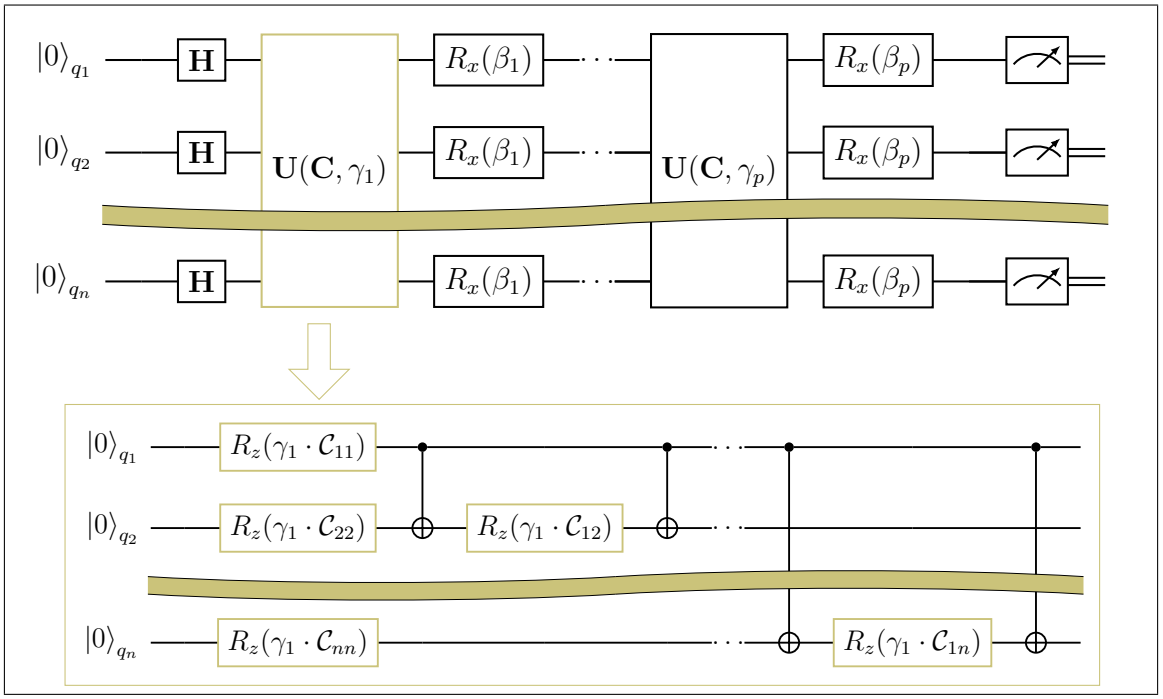


Figure 4.1.1: *A depth-p QAOA ansatz circuit.* ***(Top row)*** *The complete circuit. First, the uniform superposition state $|+\rangle^{\otimes n}$ is prepared by applying a Hadamard gate to each qubit. Next, the state $|\gamma, \beta\rangle$ is prepared by transforming $|+\rangle^{\otimes n}$ by p layers of parameterized unitary operators $\mathbf{U}(\mathbf{C}, \gamma_k)$ and $\mathbf{U}(\mathbf{B}, \beta_k)$.* ***(Bottom row)*** *A zoom-in on the first unitary $\mathbf{U}(\mathbf{C}, \gamma_1)$. Finally, $|\gamma, \beta\rangle$ is measured in the computational basis and used in a classical optimization routine to output the ground state $|q^\star\rangle$ of the in-puted Hamiltonian with high probability.*

$(\gamma, \beta) := (\gamma_1, \ldots, \gamma_p, \beta_1, \ldots, \beta_p)$. The process is repeated until stopping criteria are met.

There have been several discussions on the improvement of the performances of QAOA, ranging from the choice of the mixing Hamiltonian $\mathbf{B}$, to that of the initial parameters. Bärtschi and Eidenbenz [90] proposed Grover mixers to relate the initial state to the mixig Hamiltonian. Wang et al. [91] proposed XY mixers that allows, if applicable, to encode some hard constraints into the simulated quantum evolution. A more recent work by [92] shows that best performances of the algorithm are obtained when the initial state of QAOA is set to be the ground state of the mixing Hamiltonian. Other recent work [93] proposed to use solutions of continuous-valued classical relaxations of the problem to initialize the parameters, a technique known as warm-start.

Discussion.

For $p \to \infty$ the results of [85] guarantee that there exist parameters (γ, β) for which measuring $|\gamma, \beta\rangle$ gives the desired ground state $|q^\star\rangle$ with high probability. However, the QAOA objective is difficult to optimize [86, 88, 89]. We believe

that this is partially due to the fact that the QAOA ansatz encodes problem information in the argument of $\exp$ as phases of the qubits, which is partially lost at the measurement. Another issue of QAOA is that its repetitive layers are still too expensive for running on current and near-term devices. In this work, we propose a quantum circuit that encodes the problem more effectively and does not require the repetitive layers of QAOA. Adhering to the promising concept of designing hybrid quantum algorithms, we embed the circuit in a classical optimization method to output the desired ground state $|q^\star\rangle$ with high probability.

4.2 Proposed Variational Quantum Optimization

In this section, we aim, in a hybrid quantum-classical optimization regime, for an efficient way to search for parameters $\theta \in \Theta$ that solve the variational problem

$$\underset{\theta\in\Theta}{\text{minimize}} \quad \mathcal{L}(\theta), \quad \mathcal{L}(\theta) := \langle\psi(\theta)|\mathbf{C}|\psi(\theta)\rangle\,, \tag{148}$$

where Θ is a suitable parameterization of the search space. VQE-like methods (VQE[54], QAOA[85]) minimize the same energy functional by only preparing the ansatz $|\psi(\theta)\rangle$ on a quantum computer and evaluating the objective function classically. In contrast, our contribution enables to evaluate even the objective function on the quantum machine, reducing the dependence on classical computation – only the optimization update will be carried out classically.

4.2.1 A Block-Encoding Framework for the Ising Hamiltonian

Our method builds on the notion of *block-encoding* introduced in [94, 95], which allows to embed non-unitary matrices as the principal block of a unitary operator acting on the quantum system. Block-encoding is typically achieved by enlarging the Hilbert space of the quantum system.

Definition 4.2.1 (*Block-encoding [94, 95]*)

Let $a, n \in \mathbb{N}$ and $m := a+n$. The $m \times m$ unitary $\mathbf{U}$ is said to be a block-encoding of the $n \times n$ matrix $\mathbf{C}$ if there is $\kappa \in (0,\infty)$ such that

$$\kappa\mathbf{C} = \left[\langle 0|^{\otimes a} \otimes \mathbf{I}_n\right] \mathbf{U} \left[|0\rangle^{\otimes a} \otimes \mathbf{I}_n\right]. \tag{149}$$

In other words, $\kappa\mathbf{C}$ is the top left block of the unitary $\mathbf{U}$:

$$\mathbf{U} = \begin{pmatrix} \kappa\mathbf{C} & \star \\ \star & \star \end{pmatrix}. \tag{150}$$

By adding a single qubit to the system, our goal is to implement the $(2^{1+n})\times(2^{1+n})$ *unitary* operator

$$\mathbf{U} := \mathbf{U}(\mathbf{C}, a, b) := \begin{pmatrix} \cos(\hat{\mathbf{C}}) & -\sin(\hat{\mathbf{C}}) \\ \sin(\hat{\mathbf{C}}) & \cos(\hat{\mathbf{C}}) \end{pmatrix}, \quad \hat{\mathbf{C}} := a\mathbf{C} + b\mathbf{I}, \tag{151}$$

for the Ising Hamiltonian $\mathbf{C}$ from Equation (145) and suitably chosen *constants* $a, b \in \mathbb{R}$. We use a, b to re-scale and shift all entries of $\mathbf{C}$ into an interval $I \subset [0, \pi]$, where the cosine function is strictly monotone and invertible. As $\mathbf{C}$ is a diagonal matrix, the sine and cosine functions directly apply to the diagonal elements [12, Section 2.1.8]. Specially, Equation (151) block-encodes a bijective transformation of $\mathbf{C}$. In Section 4.2.3, we provide suitable choices for a and b.

The purpose of introducing the unitary $\mathbf{U}$ in Equation (151) is to encode information about the problem directly in probability amplitudes of the qubits. The unitary $\mathbf{U}$ operates on a $(1+n)$-qubit system prepared in the state $|\hat{\psi}, \psi\rangle := |\hat{\psi}\rangle_c \otimes |\psi\rangle$, where $|\hat{\psi}\rangle_c = \alpha\,|0\rangle_c + \beta\,|1\rangle_c$ is a 1-qubit register which we call the *cost qubit* and $|\psi\rangle$ is the n-qubit *working register*. As required by the block-encoding concept in Definition 4.2.1, as long as the cost qubit $|\hat{\psi}\rangle_c$ is kept in the $|0\rangle_c$ state, it holds

$$\langle 0, \psi \,|\, \mathbf{U} \,|\, 0, \psi\rangle = \left\langle \psi \,\middle|\, \cos(\hat{\mathbf{C}}) \,\middle|\, \psi \right\rangle . \tag{152}$$

Therefore, if a, b are chosen such that the entries of the modified matrix $\hat{\mathbf{C}}$ fit inside the monotone region of the $\cos$-function, then, solving Equation (144) is equivalent to

$$\begin{aligned} \underset{|q\rangle}{\text{minimize}} \quad & \langle 0, q|\mathbf{U}|0, q\rangle \\ \text{subject to} \quad & |q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle\,,\ q_i \in \{0, 1\}\,, \end{aligned} \tag{153}$$

where now $\mathbf{U}$ is a unitary matrix that directly serves as quantum gate.

Implementing the Block-Encoding Circuit.
Our main contribution lies in a circuit-compatible implementation of the operator $\mathbf{U}$:

Lemma 4.2.1 (*Block-encoding of the Ising Hamiltonian*)

Let $a, b \in \mathbb{R}$. Given the Ising Hamiltonian $\mathbf{C}$ in Equation (145), the block encoding matrix $\mathbf{U}(\mathbf{C}, a, b)$ in Equation (151) acting on a cost qubit $|0\rangle_c$ and a basis state $|q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle$ can be implemented as a product of controlled unitary operations

$$\begin{aligned} \mathbf{U}(\mathbf{C}, a, b) = & \prod_{i=1}^{n} \mathbf{X}^{q_i} \cdot R_y(2a\mathcal{C}_{ii}) \cdot \mathbf{X}^{q_i} \cdot \\ & \prod_{1 \le i < j \le n} \mathbf{X}^{q_i + q_j} \cdot R_y(2a\mathcal{C}_{ij}) \cdot \mathbf{X}^{q_i + q_j} \cdot R_y(2b) \otimes \mathbf{I}^{\otimes n}. \end{aligned} \tag{154}$$

Proof: Observe that $\mathbf{U}$ in Equation (151) is a block matrix of diagonal matrices. Applying it to the basis state $|0, q\rangle$ gives the same result as applying to the cost qubit $|0\rangle_c$ the controlled (2×2)-operator

$$\mathbf{U}_{2\times 2}(q) := \begin{pmatrix} \cos(\langle q|\hat{\mathbf{C}}|q\rangle) & -\sin(\langle q|\hat{\mathbf{C}}|q\rangle) \\ \sin(\langle q|\hat{\mathbf{C}}|q\rangle) & \cos(\langle q|\hat{\mathbf{C}}|q\rangle) \end{pmatrix}, \tag{155}$$

where

$$\langle q|\hat{\mathbf{C}}|q\rangle = a\,\langle q|\mathbf{C}|q\rangle + b \tag{156}$$

is the scaled and shifted version of the objective function value

$$\langle q|\mathbf{C}|q\rangle = \sum_{i=1}^{n}(-1)^{q_i}\mathcal{C}_{ii} + \sum_{1\le i<j\le n}(-1)^{q_i+q_j}\mathcal{C}_{ij} \tag{157}$$

of the state $|q\rangle$ according to the Ising model in Equation (145).

Operator $\mathbf{U}_{2\times 2}(q)$ performs a rotation of the cost qubit about the angle $\langle q|\hat{\mathbf{C}}|q\rangle$ around the y-axis in the Bloch sphere. Therefore, the unitary $\mathbf{U}(\mathbf{C}, a, b)$ can be written as

$$\begin{aligned}\mathbf{U}(\mathbf{C}, a, b) &= R_y\left(2\,\langle q|\hat{\mathbf{C}}|q\rangle\right) &&\otimes \mathbf{I}^{\otimes n} &(158)\\ &= R_y\left(2a\,\langle q|\hat{\mathbf{C}}|q\rangle + 2b\right) &&\otimes \mathbf{I}^{\otimes n} &(159)\\ &= R_y\left(2a\,\langle q|\hat{\mathbf{C}}|q\rangle\right)\cdot R_y(2b) &&\otimes \mathbf{I}^{\otimes n}. &(160)\end{aligned}$$

The last equation is obtained by applying the identity $R_y(\theta_1+\theta_2) = R_y(\theta_2)\cdot R_y(\theta_1)$ for rotations in two dimensions, where $\theta_1, \theta_2 \in \mathbb{R}$.

By recursively applying the above mentioned rotation identity to Equation (157), to decompose the $R_y(2a\,\langle q|\hat{\mathbf{C}}|q\rangle)$ term, we get

$$\begin{aligned}\mathbf{U}(\mathbf{C}, a, b) = &\prod_{i=1}^{n} R_y\left((-1)^{q_i}2a\mathcal{C}_{ii}\right)\cdot\\ &\prod_{1\le i<j\le n} R_y\left((-1)^{q_i+q_j}2a\mathcal{C}_{ij}\right)\cdot R_y(2b)\otimes\mathbf{I}^{\otimes n}.\end{aligned} \tag{161}$$

Further, by matrix multiplication, one can verify that for all $\theta\in\mathbb{R}$ and $q_i\in\{0,1\}$, it holds $R_y\left((-1)^{q_i}\theta\right) = \mathbf{X}^{q_i}\cdot R_y\left(\theta\right)\cdot\mathbf{X}^{q_i}$. Applying this to Equation (161) yields

$$\begin{aligned}\mathbf{U}(\mathbf{C}, a, b) = &\prod_{i=1}^{n} \mathbf{X}^{q_i}\cdot R_y(2a\mathcal{C}_{ii})\cdot\mathbf{X}^{q_i}\cdot\\ &\prod_{1\le i<j\le n}\mathbf{X}^{q_i+q_j}\cdot R_y(2a\mathcal{C}_{ij})\cdot\mathbf{X}^{q_i+q_j}\cdot R_y(2b)\otimes\mathbf{I}^{\otimes n},\end{aligned} \tag{162}$$

which concludes the proof. ■

As a result, the weighted sum of Pauli-Z operators of the Ising Hamiltonian naturally translates into a product of unitary transformations, which is very compatible with the gate-based model of quantum computing. For any given superposition state $|\psi\rangle = \sum_q \alpha_q\,|q\rangle$, we have $\langle 0,\psi\,|\,\mathbf{U}\,|\,0,\psi\rangle = \sum_q |\alpha_q|^2\cos(a\,\langle q|\mathbf{C}|q\rangle + b)$. For a basis state $|\hat{\psi},\psi\rangle = |0,q\rangle$, we can even recover the exact objective function value by $\langle q|\mathbf{C}|q\rangle = \frac{1}{a}\left(\arccos\langle 0,q|\mathbf{U}|0,q\rangle - b\right)$.

The operator $\mathbf{U}$ is implemented using the circuit given in Figure 4.2.1:

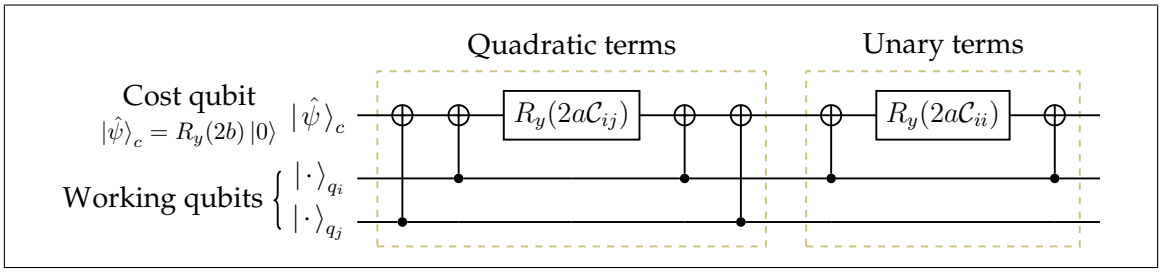


Figure 4.2.1: *Implementation of the* $(2^{1+n}) \times (2^{1+n})$ *operator* $\mathbf{U}$ *in Equation* (154). *The cost qubit is initialized in the state* $|\hat{\psi}\rangle_c = R_y(2b)\,|0\rangle$ *and is the target of all operations. Note that this does not contradict the idea of keeping it in the* $|0\rangle$*-state, as the operation* $R_y(2b)$ *implements the last part of* $\mathbf{U}$. ***(Left*** *For each coupling edge of weight* $\mathcal{C}_{ij}$ *between nodes* q_i *and* q_j*, rotate the cost qubit by* $R_y(2a\mathcal{C}_{ij})$ *if their corresponding working qubits are in the same state, else by* $R_y(-2a\hat{\mathcal{C}}_{ij}) = \mathbf{X} \cdot R_y(2a\mathcal{C}_{ij}) \cdot \mathbf{X}$. ***(Right*** *For each unary edge of weight* $\mathcal{C}_{ii}$ *involving node* q_i*, rotate the cost qubit by* $R_y(2a\mathcal{C}_{ii})$ *if its corresponding working qubit is in the* $|0\rangle$*-state, else by* $R_y(-2a\hat{\mathcal{C}}_{ii}) = \mathbf{X} \cdot R_y(2a\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X}$.

- We initialize the cost qubit in the state $|\hat{\psi}\rangle_c = R_y(2b)\,|0\rangle$.
- For each weight $\mathcal{C}_{ij}$ between two nodes q_i and q_j, we rotate the cost qubit by $R_y(2a\mathcal{C}_{ij})$ if the corresponding working qubits are in the same state, and rotate by $R_y(-2a\hat{\mathcal{C}}_{ij}) = \mathbf{X} \cdot R_y(2a\mathcal{C}_{ij}) \cdot \mathbf{X}$ if they are not.
- For each unary weight $\mathcal{C}_{ii}$ involving node q_i, we rotate the cost qubit by $R_y(2a\mathcal{C}_{ii})$ if the corresponding working qubit is in the $|0\rangle$-state, and rotate by $R_y(-2a\hat{\mathcal{C}}_{ii}) = \mathbf{X} \cdot R_y(2a\mathcal{C}_{ii}) \cdot \mathbf{X}$ if it is not.

Whenever the unary costs satisfy $\mathcal{C}_{ii} = 0$ for all i, we refer to Equation (154) as the *Universal Quantum Maximum Cut (UQMaxCut)* model, else as the *Universal Quantum Ising (UQIsing)* model.

4.2.2 The Parameterized Quantum Circuit

The overall architecture of our algorithm is outlined in Figure 4.2.2. The complete circuit consists of 3 registers: a 1-qubit register containing an ancilla qubit, another 1-qubit register for the cost qubit, and an n-qubit register for the working qubits encoding the variables of the problem.

First, the working qubits are rotated by a set of angles $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$, constructing the ansatz

$$|\psi(\theta)\rangle := R_y(\theta_1) \otimes \cdots \otimes R_y(\theta_n)\,|\psi\rangle\,, \tag{163}$$

from a system previously prepared in the state $|\psi\rangle$. Each qubit q_i, representing the i-th node of the graph, is rotated by θ_i around the y-axis. Next, a Hadamard sandwich involving a controlled version of the $\mathbf{U}$-operator is applied to the ancilla qubit. Finally, the ancilla qubit is measured, leaving the cost and working qubits in the state $\mathbf{U}\,|0, \psi(\theta)\rangle = |0\rangle_c \otimes \cos(\hat{\mathbf{C}})\,|\psi(\theta)\rangle + |1\rangle_c \otimes \sin(\hat{\mathbf{C}})\,|\psi(\theta)\rangle$.

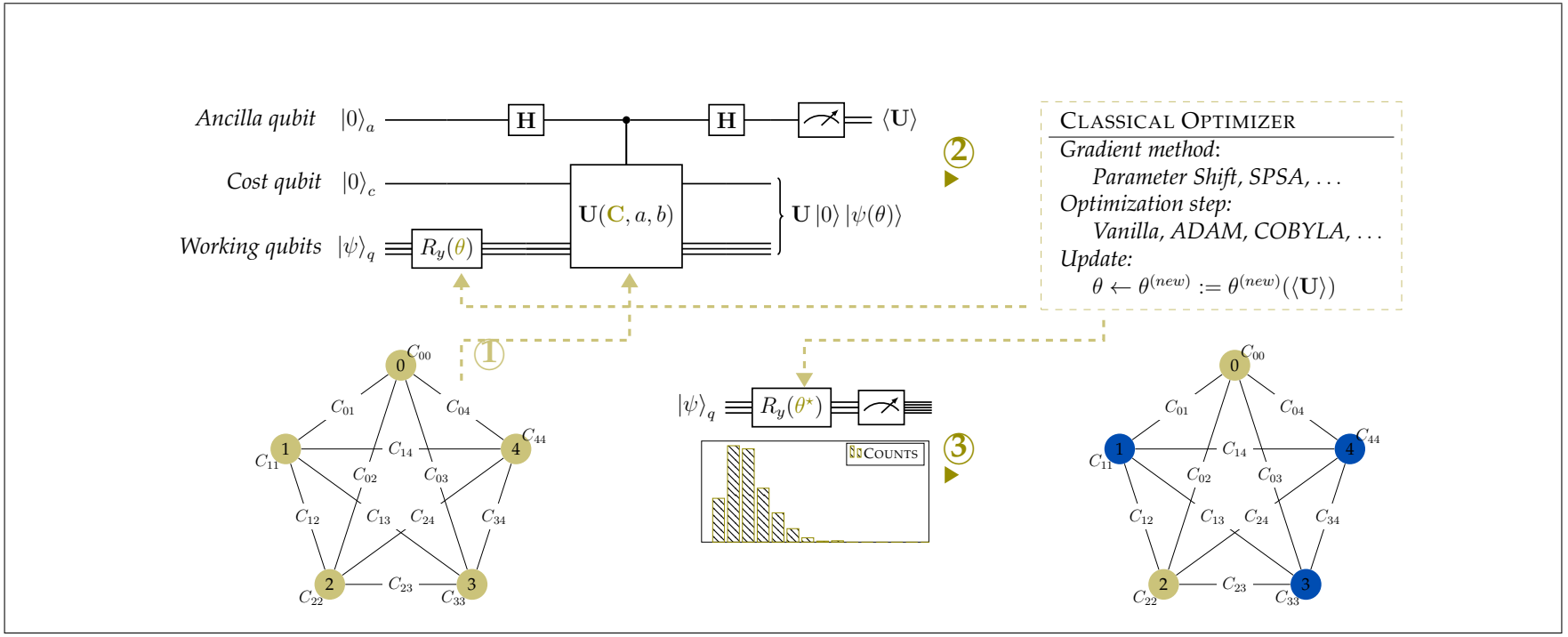


Figure 4.2.2: *Overview of the proposed algorithm for the Universal Quantum MaxCut and Ising Model (UQMaxCut and UQIsing).* ***First*** ①*, the cost (node and edge weight) information for the given graph are used to implement the operator* $\mathbf{U}(\mathbf{C}, a, b)$ *that block-encodes the problem Hamiltonians. This operator is applied to a trial state made up of the variational ansatz* $|\psi(\theta)\rangle$ *and the cost qubit.* ***Second*** ②*, using the principle of implicit measurement, the expectation* $\langle \mathbf{U} \rangle = \langle 0, \psi(\theta)|\mathbf{U}|0, \psi(\theta)\rangle$*, which equals the objective* $\mathcal{L}(\theta)$*, is approximated by measuring several copies of the circuit. This computed expectation and eventually its gradient are iteratively used in a classical optimization routine which drives the parameterized state* $|\psi(\theta)\rangle$ *towards the state* $|\psi(\theta^\star)\rangle$ *that potentially gives the global minimal cost value.* ***Finally*** ③*, the optimal state* $|\psi(\theta^\star)\rangle$ *is measured in the computational basis, and the most frequently measured state is returned.*

Our goal is to solve the variational problem

$$\underset{\theta\in\mathbb{R}^n}{\text{minimize}} \quad \mathcal{L}(\theta), \quad \mathcal{L}(\theta) = \langle 0, \psi(\theta)|\mathbf{U}|0, \psi(\theta)\rangle\,, \tag{164}$$

i.e., to find a set of angles $\theta^\star$ such that $|\psi(\theta^\star)\rangle = |q^\star\rangle$. As derived in the next section, the objective function $\mathcal{L}(\theta)$ can be calculated by $\mathcal{L}(\theta) = p(0) - p(1)$, where $p(0)$ and $p(1)$ are the probabilities of measuring the ancilla qubit in the $|0\rangle$ and $|1\rangle$ state. This allows us to compute the expectation $\langle \mathbf{U}\rangle$ without having to measure $\langle 0, \psi(\theta)|$ needed for the inner product.

Optimizing the Circuit.

The circuits (UQMaxCut, UQIsing) are optimized by normalized gradient descent [47, 48] with decreasing step size, with gradients computed by the parameter shift rule [62]. At each iteration k, our update rule reads

$$\theta^{(k)} \leftarrow \theta^{(k-1)} - \left[\frac{\pi n}{2}\right]^{1/2} \exp\left[-\frac{4k^2}{k_{max}^2}\right] \cdot \frac{\nabla_\theta \mathcal{L}(\theta^{(k-1)})}{\|\nabla_\theta \mathcal{L}(\theta^{(k-1)})\|_2}. \tag{165}$$

The design of the update rule Equation (165) is motivated by the following consideration: We know that we produce bit-strings by either flipping $|q_i\rangle$ or not, thus $\theta_i^\star = \ell\pi, \ell \in \mathbb{Z}$. At iteration $k = 0$, the update rule Equation (165) allows each θ_i to get updated to $\theta_i^1 \leftarrow \theta_i^0 \pm \pi \cdot g_i/2$, where $g_i \in [-1, 1]$ is the normalized contribution of θ_i^0 in the objective function value $\mathcal{L}(\theta^{(k-1)})$. Subsequently, we let the step size decay exponentially to zero when approaching the maximum number of iterations k_{max}. Given the noisy nature of quantum measurement, it is helpful that this frees us from the difficult task of determining a stopping condition for the algorithm.

Evaluating the Cost Function.

The objective function is evaluated using the so-called *Hadamard test* [96]: Let $\mathbf{U}$ be a unitary operator. The measurement of the first qubit of the circuit in Figure 4.2.3 in the computational basis performs a measurement of the real part $\mathrm{Re}\left(\langle\psi_{in}\,|\,\mathbf{U}\,|\,\psi_{in}\rangle\right)$ of the expectation of $\mathbf{U}$ on the second register.

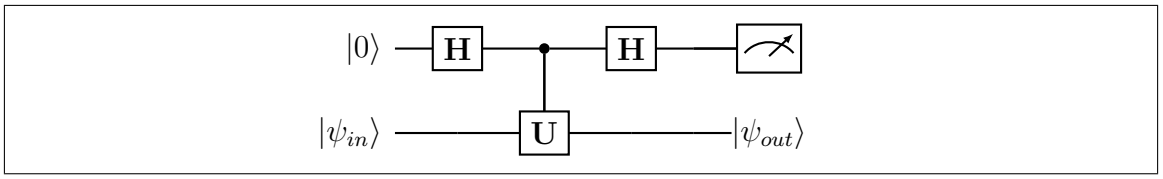


Figure 4.2.3: *Circuit illustrating the Hadamard test.*

To see this, we define operators

$$\mathbf{P}_\pm := \frac{1}{2}(\mathbf{I} \pm \mathbf{U}). \tag{166}$$

Evidently, it holds

$$\mathbf{P}_{\pm} = \frac{1}{4}\left(\mathbf{I} \pm \mathbf{U} \pm \mathbf{U}^{\dagger} + \mathbf{U}^{\dagger}\mathbf{U}\right) \qquad \text{and} \qquad \mathbf{P}_{+}^{\dagger}\mathbf{P}_{+} - \mathbf{P}_{-}^{\dagger}\mathbf{P}_{-} = \frac{1}{2}\left(\mathbf{U} + \mathbf{U}^{\dagger}\right). \quad (167)$$

Before measurement, the system is in the state

$$\frac{1}{2}\Big[(|0\rangle + |1\rangle) \otimes \mathbf{I}\,|\psi_{in}\rangle + (|0\rangle - |1\rangle) \otimes \mathbf{U}\,|\psi_{in}\rangle\Big] \quad (168)$$

$$= |0\rangle \otimes \mathbf{P}_{+}\,|\psi_{in}\rangle + |1\rangle \otimes \mathbf{P}_{-}\,|\psi_{in}\rangle\,. \quad (169)$$

A measurement of the first qubit in the computational basis, with the measurement operators $\mathbf{P}_0 = |0\rangle\langle 0| \otimes \mathbf{I}$ and $\mathbf{P}_1 = |1\rangle\langle 1| \otimes \mathbf{I}$, yields the probabilities

$$p(0) = \left\langle \psi_{in} \middle| \mathbf{P}_{+}^{\dagger}\mathbf{P}_{+} \middle| \psi_{in} \right\rangle \qquad \text{and} \qquad p(1) = \left\langle \psi_{in} \middle| \mathbf{P}_{-}^{\dagger}\mathbf{P}_{-} \middle| \psi_{in} \right\rangle. \quad (170)$$

Then, we compute the difference

$$1 \cdot p(0) + (-1) \cdot p(1) = \left\langle \psi_{in} \middle| \left(\mathbf{P}_{+}^{\dagger}\mathbf{P}_{+} - \mathbf{P}_{-}^{\dagger}\mathbf{P}_{-}\right) \middle| \psi_{in} \right\rangle \quad (171)$$

$$= \frac{1}{2}\left\langle \psi_{in} \middle| \left(\mathbf{U} + \mathbf{U}^{\dagger}\right) \middle| \psi_{in} \right\rangle \quad (172)$$

$$= \frac{1}{2}\left(\langle \psi_{in} \,|\, \mathbf{U} \,|\, \psi_{in}\rangle + \left\langle \psi_{in} \middle| \mathbf{U}^{\dagger} \middle| \psi_{in} \right\rangle\right) \quad (173)$$

$$= \mathrm{Re}\left(\langle \psi_{in} \,|\, \mathbf{U} \,|\, \psi_{in}\rangle\right). \quad (174)$$

Since in our application, c.f. Equation (152), the state $|\psi_{in}\rangle$ always has the form $|\psi_{in}\rangle = |0, \psi\rangle$, it holds $\langle 0, \psi \,|\, \mathbf{U} \,|\, 0, \psi\rangle = \langle \psi | \cos(\hat{\mathbf{C}}) | \psi \rangle \in \mathbb{R}$. Hence

$$\mathrm{Re}\left(\langle \psi_{in} \,|\, \mathbf{U} \,|\, \psi_{in}\rangle\right) = \langle \psi_{in} \,|\, \mathbf{U} \,|\, \psi_{in}\rangle\,. \quad (175)$$

Thus, we can approximate the objective function by measuring a single qubit. This is a crucial feature of our algorithm, as it makes it possible, having $\mathbf{U}\,|\psi_{in}\rangle$ to approximate the scalar product $\langle \psi_{in} \,|\, \mathbf{U} \,|\, \psi_{in}\rangle$ without having to sample $|\psi_{in}\rangle$.

A Note on the Accuracy of Objective Function Evaluation.

The Hadamard test differs from projective measurements and often requires more measurement shots to accurately approximate the objective function value.

Each measurement outcome yields ± 1. Thus, while performing M shots on a single qubit and computing the expectation, as done in the Hadamard test, we can only hope to represent at most M different objective values. However in Equation (152), we saw that for a basis state $|q\rangle$ it holds $\langle 0, q|\mathbf{U}|0, q\rangle = \langle q| \cos(\hat{\mathbf{C}})|q\rangle$. Hence, measuring on a basis state produces an eigenvalue of $\cos(\hat{\mathbf{C}})$. Accurately approximating these eigenvalues would require to perform a number of measurement shots that equals at least the cardinality of the spectrum of $\cos(\hat{\mathbf{C}})$.

Indeed, due to the Hoeffding inequality [97], the measurement error, say ϵ, scales inversely quadratically with M. It holds

$$p\left(|\text{estimated } \langle \mathbf{U} \rangle - \text{effective } \langle \mathbf{U} \rangle| < \epsilon\right) \geq 1 - 2e^{-M\epsilon^2/2}. \tag{176}$$

Consequently, if one wants the probability $1 - 2e^{-M\epsilon^2/2}$ to equal $1 - \eta$, one needs to set $M = \frac{2}{\epsilon^2} \log(2/\eta)$. See also [98] for a method for improving this bound based on quantum phase estimation.

4.2.3 Impact of Re-Scaling and Shifting the Costs

When applying the method, it is important to appropriately choose the constants $a, b \in \mathbb{R}$ for re-scaling and shifting the costs $\hat{\mathbf{C}} = a\mathbf{C} + b\mathbf{I}$. The goal is to fix a, b such that $\mathrm{diag}(\hat{\mathbf{C}}) \subset [0, \pi]^{2^n}$. Furthermore, in this interval the costs are reversed to guarantee that $\cos(\mathrm{diag}(\hat{\mathbf{C}}))$ preserves the relative ordering of the original costs in $\mathrm{diag}(\mathbf{C})$. We set $a = -\frac{1}{K} \cdot \lambda$ and $b = \lambda$ with $\lambda \in [0, \pi/2]$ and some sufficiently large $K \in \mathbb{R}_+$.

The choice is motivated by the following, also illustrated in Figure 4.2.4:

1. The constant K should be large enough to scale the costs into $I \subset [-1, 1]^{2^n}$. The multiplication by $\lambda \in [0, \pi/2]$ brings them into the interval $I \subset [-\frac{\pi}{2}, \frac{\pi}{2}]^{2^n}$.
2. The negative sign of constant a keeps the costs into the interval $I \subset [-\frac{\pi}{2}, \frac{\pi}{2}]^{2^n}$, but reverses the order of the costs such that the minimizer receives the highest objective function value and the maximizes the lowest.
3. The shift by the constant $b = \lambda$ brings the costs into the interval $I \subset [0, \pi]^{2^n}$ where the cosine is monotonically decreasing. However, the reverse order of the costs turns the cosine transformed cost in the right order back, preserving the original ordering.

In order to ensure that we remain in the monotonic region of the cosine, it is tempting to choose a very large $K \gg \mathcal{C}_{\max} := \max_k \mathbf{C}_{kk}$, where $\mathbf{C}_{kk}$ denotes the k-th diagonal element, i.e., considerably larger than the largest possible objective function value. However, as K approaches $+\infty$, the operator $\mathbf{U}(\mathbf{C}, a, b)$ in Equation (151) approaches $\mathbf{I}^{\otimes(1+n)}$. This in turn means that the eigenvalues are distributed over a smaller interval around $\cos(\lambda)$, so more shots are expected in order to accurately separate and distinguish them from each other. Fortunately, in many problems, a tighter upper bound to the maximal cost $\mathcal{C}_{\max}$ can be computed from the original weights without knowing $\mathbf{C}$. For example, choosing

$$K = \mathcal{C}_{\max} := \sum_{i=1}^{n} |\mathcal{C}_{ii}| + \sum_{1 \leq i < j \leq n} |\mathcal{C}_{ij}| \tag{177}$$

suffices to guarantee an equivalent transformation of the initial problem (148) into the form (164).

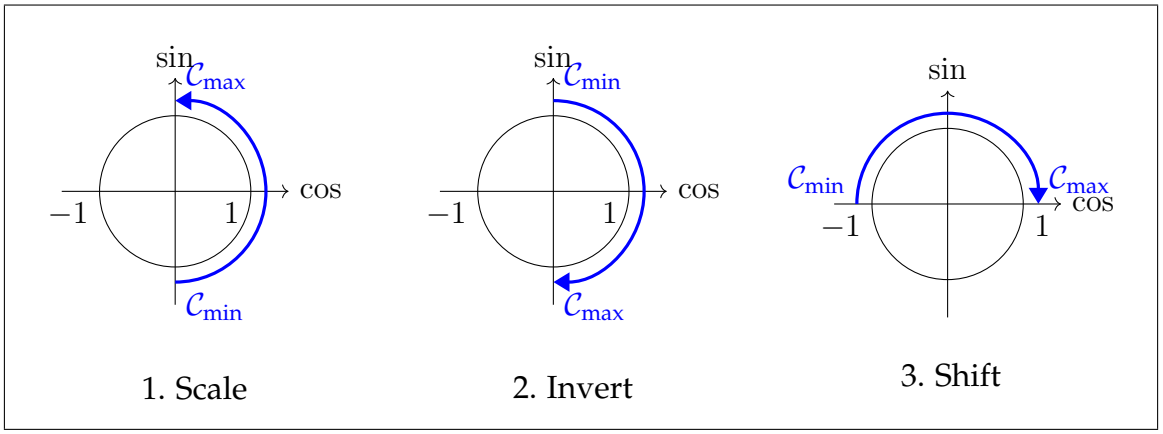


Figure 4.2.4: *Scaling and shifting the original cost of the Ising Hamiltonian.* ***(Left)*** *A first operation brings the cost into the interval* $I \subset [-\frac{\pi}{2}, \frac{\pi}{2}]^{2^n}$. ***(Middle)*** *A second operation reverses the order of the costs.* ***(Right)*** *A third operation brings the costs into the interval* $I \subset [0, \pi]^{2^n}$ *where the cosine is monotonically decreasing. However, the reverse order of the costs turn the cosine transformed cost in the right order back, preserving the original ordering.*

4.2.4 Scalability and Computational Complexity

For a given graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$, the circuit construction presented in Figure 4.2.1 requires at most one NOT-gate, $|\mathcal{E}|$ single-qubit rotation gates and $4|\mathcal{E}|-2|\mathcal{S}|$ CNOT gates. Note that since the edges can be treated in arbitrary order, two consecutive CNOT gates having the same control qubit cancel each other out as their product is the identity, further reducing the number of required CNOT gates.

The controlled-$\mathbf{U}(\mathbf{C}, a, b)$ gate appearing in Figure 4.2.2 and conditioned by the ancilla qubit $|\cdot\rangle_\alpha$ can be fully decomposed into single qubit rotations and CNOT gates without using any Toffoli gates. To see this, note that it can be expressed as

$$\text{controlled-}\mathbf{U}(\mathbf{C}, a, b) = \mathbf{I} \otimes \mathbf{U}(\mathbf{C}, a, b)^\alpha \tag{178}$$

$$\equiv \mathbf{I} \otimes \prod_{i=1}^{n} \mathbf{X}^{q_i} \cdot [R_y(2a\hat{\mathcal{C}}_{ii})]^\alpha \cdot \mathbf{X}^{q_i} \cdot \prod_{1 \le i < j \le n} \mathbf{X}^{q_i+q_j} \cdot [R_y(2a\hat{\mathcal{C}}_{ij})]^\alpha \cdot \mathbf{X}^{q_i+q_j} \cdot \mathbf{X} \otimes \mathbf{I}^{\otimes n}. \tag{179}$$

In particular, in order to control the complete $\mathbf{U}(\mathbf{C}, a, b)$ gate, it suffices to control only the rotation gates R_y. This controlled rotation can be decomposed into two CNOT gates and two single qubit rotations as

$$[R_y(\theta)]^\alpha = \mathbf{X}^\alpha \cdot R_y(-\theta/2) \cdot \mathbf{X}^\alpha \cdot R_y(\theta/2). \tag{180}$$

Table 3 summarizes the main differences between UQIsing and a conventional QAOA of depth p. For $p \geq 3$, QAOA requires more quantum resources than our

	QAOA	UQIsing [Ours]
# qubits	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}\rvert + 2$
# CNOT gates	$p \cdot (2\lvert\mathcal{E}\rvert - 2\lvert\mathcal{S}\rvert)$	$1 + 6\lvert\mathcal{E}\rvert - 2\lvert\mathcal{S}\rvert$
# single qubits rotations	$p \cdot (\lvert\mathcal{E}\rvert + \lvert\mathcal{S}\rvert)$	$2\lvert\mathcal{E}\rvert$
# Hadamard gates	$\lvert\mathcal{S}\rvert$	2
qubit connectivity	graph-dependent: $\lvert\mathcal{E}\rvert - \lvert\mathcal{S}\rvert$	one-to-all: $\lvert\mathcal{S}\rvert + 1$

Table 3: *Resources and complexity comparison of a depth-p conventional QAOA and our UQIsing model on a graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$. The set $\mathcal{S}$ is the set of vertices. Note that the set $\mathcal{E}$ comprises all the edges of the graph and also includes the "self-loops" for the unary costs. Our construction requires fewer quantum and classical resources than QAOA for $p \geqslant 3$.*

UQIsing model. Further, physically mapping the QAOA ansatz onto the quantum hardware has to take into account a graph-dependent qubit connectivity. For our method, connectivity is independent of the input graph, and only requires that one qubit (the cost qubit) is connected to all other qubits (ancilla and working qubits).

4.3 Experimental Results

In order to validate the practical usefulness of UQMaxCut and UQIsing, we benchmark against two state-of-the-art approaches for solving binary combinatorial optimization with quantum computing: QAOA for the gate-based model and D-Wave solvers for the adiabatic model. Random graphs in the experiments are generated using the PYTHON language package NETWORKX [99]. The unary and quadratic edge weights are all randomly and uniformly chosen in the range $[1, 10]$ and the graphs are all fully connected. The gate-based circuits in the experiments (UQMaxCut, UQIsing, QAOA) are implemented in PYTHON and simulated in a noise-free framework using the QISKIT library and the IBM-QASM simulator [7]. For the adiabatic model, D-Wave solvers that run on the actual quantum hardware are used. D-Wave quantum annealers are made available through the Leap quantum cloud service [100], and the D-Wave quantum algorithms can be implemented in Python using the Ocean software [101]. We perform 1024 measurement shots for all gate-based algorithms. On D-Wave [8], the experiment is run with the default annealing time of $20\mu s$ and 50 sample reads on the Advantage topology.

4.3.1 Benchmark Metrics

We denote by $|q^\star\rangle$ a ground-truth global minimizer and by $|\psi^\star\rangle$ the proposed (basis) ground state returned by each method (UQMaxCut/UQIsing, QAOA and D-Wave). In the experiments, we adopt the following two metrics to evaluate the performance of the methods:

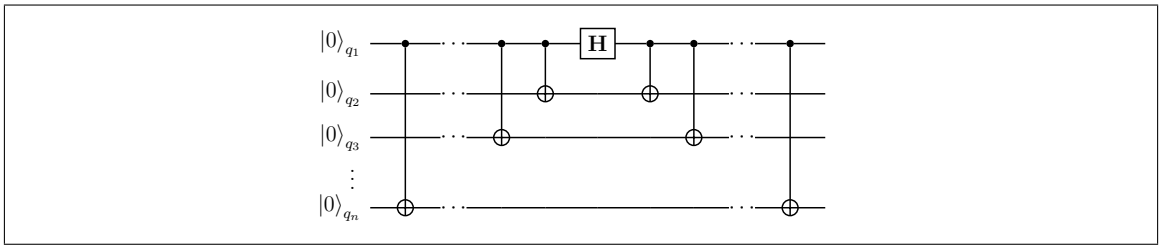


Figure 4.3.1: *Entanglement circuit to disambiguate symmetric solutions for MaxCut. Using this circuit, the optimization is performed only on the last $n-1$ variables and the qubit for the node q_1 is kept constant in the state $|0\rangle$ or $|1\rangle$.*

- The approximation ratio

$$r(\psi^\star) := \frac{\langle\psi^\star|\mathbf{C}|\psi^\star\rangle - \mathcal{C}_{\text{max}}}{\mathcal{C}_{\text{min}} - \mathcal{C}_{\text{max}}} = 1 - \frac{\langle\psi^\star|\mathbf{C}|\psi^\star\rangle - \mathcal{C}_{\text{min}}}{\mathcal{C}_{\text{max}} - \mathcal{C}_{\text{min}}} \tag{181}$$

  informs about the quality of the result, i.e., how confident the method is with its solution proposal and how far the cost of this proposal is from the global minimum $\mathcal{C}_{\text{min}} := \min_k \mathbf{C}_{kk}$ of the cost function, cf. [102, Section E]. All the terms appearing in r are classically evaluated. It holds $0 \leq r(\psi^\star) \leq 1$ and $r(\psi^\star) = 1$ if and only if $\psi^\star$ is a global minimizer.

- The approximation index

$$i(\psi^\star) := \mathbf{1}_{\langle q^\star|\mathbf{C}|q^\star\rangle = \langle q_{max}|\mathbf{C}|q_{max}\rangle} \tag{182}$$

  is a Boolean variable that indicates whether the state that is most likely to be returned by an algorithm is actually a global solution. Here $|q_{max}\rangle$ denotes the state that has the largest probability $|\alpha_{max}|^2$ to be returned by an algorithms. It holds $i(\psi^\star) = 1$ if $\langle q^\star|\mathbf{C}|q^\star\rangle = \langle q_{max}|\mathbf{C}|q_{max}\rangle$ and 0 otherwise. Note that this differs from the usual approach of D-Wave, where the sampled state with the minimal energy is regarded as the best solution proposal.

4.3.2 Benchmark Results

Symmetric Solutions and Entanglement.

Before discussing the results on the MaxCut problem, it is important to notice that for MaxCut, solutions always exist in symmetric pairs. Specifically, the basis states $|q\rangle = \bigotimes_{i=1}^n |q_i\rangle, q_i \in \{0,1\}$ and $|\bar{q}\rangle := \bigotimes_{i=1}^n |1-q_i\rangle$ possess the same objective function values.

In theory, it is possible to reduce the search space and to get rid of this ambiguity by introducing the entanglement circuit given in Figure 4.3.1 after the rotation layer in Figure 4.2.2.

The circuit has the matrix representation

$$E = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & & & & & & 1 \\ & \ddots & & & & \iddots & \\ & & 1 & 1 & & & \\ & & 1 & -1 & & & \\ & \iddots & & & & \ddots & \\ 1 & & & & & & -1 \end{pmatrix}, \tag{183}$$

and the particularity to map the basis state $|q\rangle$ to the prefect superposition state $\frac{1}{\sqrt{2}}(|q\rangle + |\bar{q}\rangle)$ if the first qubit of $|q\rangle$ is 0, or to $\frac{1}{\sqrt{2}}(|q\rangle - |\bar{q}\rangle)$ if the first qubit is 1. Thus, the entanglement allows to optimize over $n-1$ angles instead of n. Without entanglement our method just outputs one of the two possible solutions. However, it is also easy to rectify this ambiguity without entanglement, for example, by classically flipping all qubits of the obtained basis state. Therefore, in the experiments, we use UQMaxCut without entanglement; if the algorithm outputs either one of the two solutions, it is accepted as a global solution and we set the approximation index to 1.

UQMaxCut vs. QAOA and D-Wave.

For the outer optimization algorithm of the UQMaxCut circuit we use normalized gradient descent as described in Section 4.2.2. Other optimization algorithms such as vanilla gradient descent or adaptive moment estimation (ADAM [103]) could be used as well. However, it proved difficult to find a suitable step size control for these methods, so we leave them for future research.

The QAOA layers depth in the experiments is set to $p = \lceil n/2 \rceil$ to allow for a fair comparison, as then both QAOA and UQMaxCut optimize over approximately n real variables. We also attempted to optimize QAOA using the same optimizer as for UQMaxCut, but the results were not competitive. Hence, we also show the results of QAOA when using a gradient-free optimizer; we used the COBYLA solver [55] as implemented in the SCIPY library [104].

The results are presented in Figure 4.3.2 for fully-connected graphs of $n = 3$, 5, and 10 nodes with strictly positive edge weights. For each n, the results are averaged over 20 graph instances and all algorithms are tested on the same instances. The angles for UQMaxCut and QAOA are all initialized to zero.

The approximation ratio for the three methods (QAOA, D-Wave, UQMaxCut) is not adversely affected by the number of variables n, but the approximation index drops sharply as the size of the problem increases. The gradient-free Cobyla-optimized version of QAOA performs much better than QAOA with gradient descent. We conjecture that the gradient-based optimization of QAOA often gets trapped by saddle points of the QAOA loss function landscape. In contrast, UQMaxCut clearly outperforms the two QAOA variants and challenges D-Wave in producing good approximate solutions. Furthermore, the approximation index

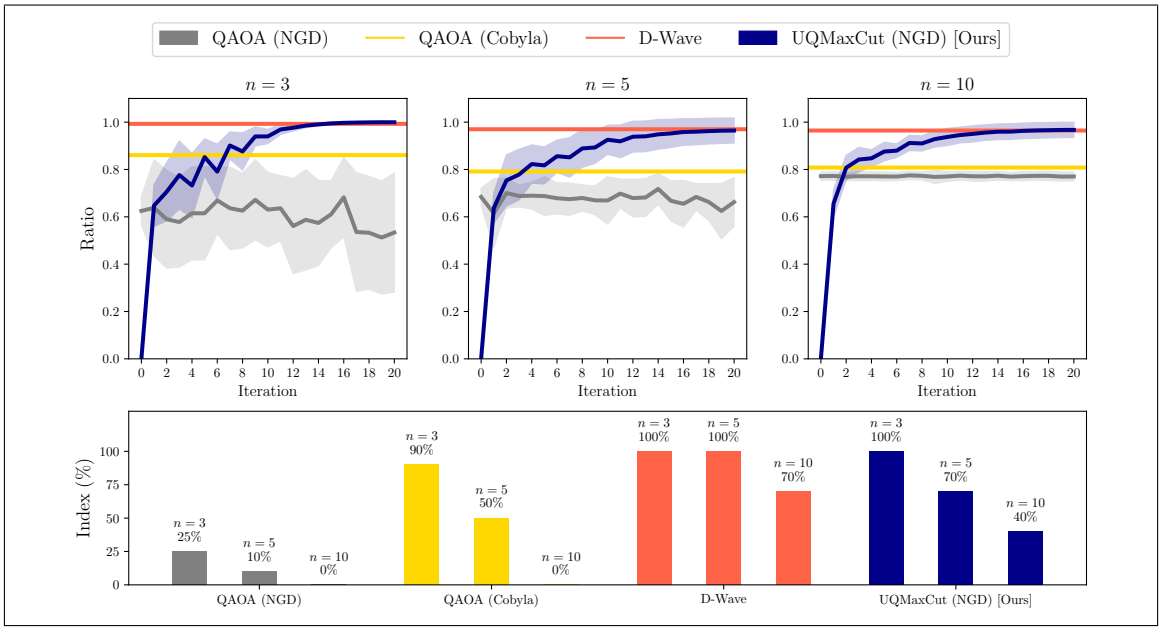


Figure 4.3.2: *Experimental comparison of the proposed Universal Quantum MaxCut (UQMaxCut) algorithm with QAOA and D-Wave. Results are shown for fully-connected graphs of $n = 3$, 5, and 10 nodes with strictly positive edge weights, and are averaged over 20 random graph instances for each n.* ***(Top row)*** *Approximation ratio (larger is better). The lines represent the averaged ratios over the 20 instances and the shaded areas indicate the standard deviations.* ***(Bottom row)*** *The approximation index, i.e., the percentage of instances where a global solution is found. All gate-based methods (QAOA and UQMaxCut) less frequently return a global solution for larger n. Yet, in contrast to QAOA, D-Wave and the proposed UQMaxCut also consistently return very good approximate solutions, i.e., points whose function values are very close to the global minimal function value.*

demonstrates that it returns a global minimizer significantly more often than QAOA and less often than D-Wave whose architecture is specifically designed to solve such problems.

UQIsing vs. D-Wave.

For the Ising model, we benchmark the proposed UQIsing algorithm against the D-Wave annealer, the adiabatic quantum computer specialized in solving this type of problems. The variational circuit for UQIsing is optimized in the same way as for UQMaxCut, see Section 4.2.2, with the exception that the initial angles are set to $\pi/2$ instead of 0. This change is done in order to start in a perfect superposition state so that the initial state should not be too distant from the global solution. In contrary to MaxCut, the solutions are no more symmetric. The results are depicted in Figure 4.3.3.

Notable differences in performance between UQIsing and D-Wave are consistent with the MaxCut experiments. Specifically, its high approximation ratio, similar

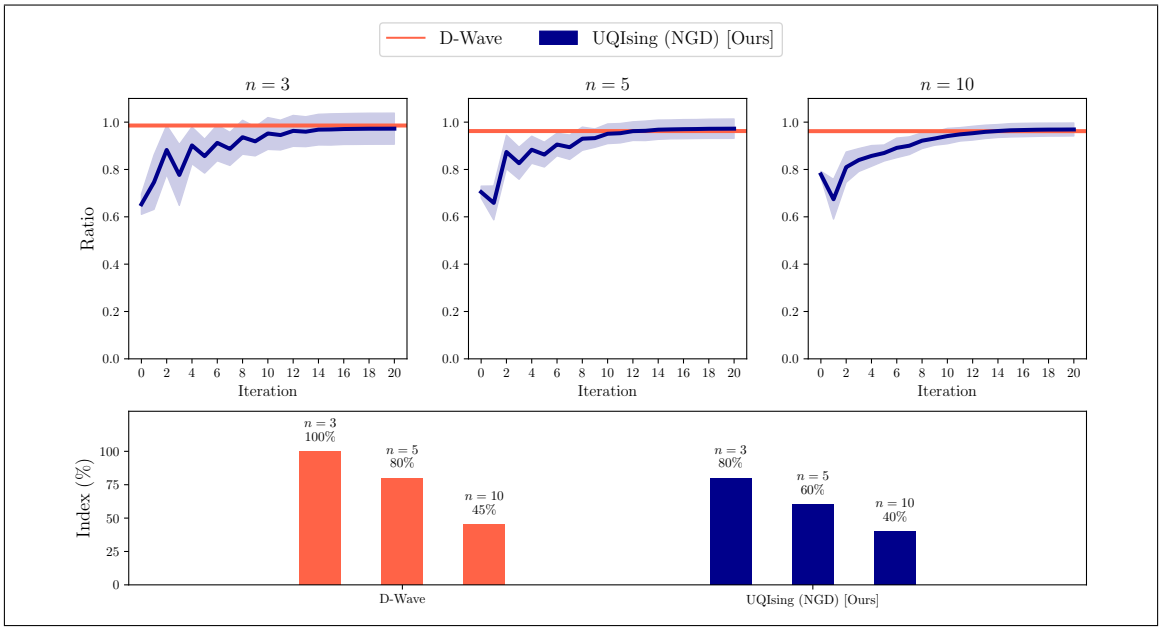


Figure 4.3.3: *Experimental comparison of the proposed Universal Quantum Ising Model (UQIsing) algorithm with D-Wave. Results are shown for fully-connected graphs of $n = 3, 5,$ and 10 nodes with strictly positive edge weights. They are averaged over 20 random graph instances for each n.* ***(Top row)*** *The approximation ratio. The lines represent the averaged ratios over the 20 instances and the shaded areas indicate the standard deviations.* ***(Bottom row)*** *The approximation index, i.e., the percentage of instances where a global solution is found. Our method can compete with D-Wave solvers in predicting approximate solutions and finding the global minimum for moderate n.*

to that of D-Wave, indicates that UQIsing always produces either globally optimal solutions or extremely good approximations thereof. On the other hand, the approximation index shows that D-Wave identifies a globally optimal solution more often than UQIsing.

4.4 Conclusion

We have presented a new low-depth quantum circuit for the preparation on a universal quantum machine of the ground state of the Ising Hamiltonian, which relates to the MaxCut problem. The resulting universal quantum MaxCut (UQ-MaxCut) approach outperforms the state-of-the-art quantum approximate optimization algorithms (QAOA) by the lower depth, by the computed approximation ratios and by a higher probability of outputting optimal solutions. It also challenges the D-Wave quantum annealers that are specifically designed to solve such combinatorial problems. On the MaxCut as well as on the Ising model, UQ-MaxCut and UQIsing achieve better approximation ratios and can compete with D-Wave in producing globally optimal solutions.

We believe that the proposed approach could enable the design of new hybrid methods for solving practically-sized problems on universal quantum machines.

The next chapter however rely on the novel operator **U** to design a fully universal algorithm without the classical outer optimization loop, replacing the latter with fully universal methods.

CHAPTER 5

Solving the Ising Problem by Quantum Search

While hybrid quantum-classical method are currently the most practical choice, they are somewhat unsatisfactory from a theoretical viewpoint and difficult to analyze due to the mixed computational paradigms. Therefore, in this chapter, we approach the Ising problem again from a purely quantum viewpoint.

We consider once again an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{C})$ with $\mathcal{S} = \{s_1, \ldots, s_n\}$, $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$, and a cost function $\mathcal{C} : \mathcal{E} \to \mathbb{R}$ with $\mathcal{C}(s_i, s_j) := \mathcal{C}_{ij}$ on $\mathcal{E}$. Our purpose is to employ quantum amplitude amplification methods in order to solve the problem

$$\begin{aligned} \underset{|q\rangle}{\text{minimize}} \quad & \langle q|\mathbf{C}|q\rangle \\ \text{subject to} \quad & |q\rangle = \bigotimes_{i=1}^{n} |q_i\rangle \,,\ q_i \in \{0,1\} \,, \end{aligned} \tag{184}$$

where

$$\mathbf{C} = \sum_{i=1}^{n} \mathcal{C}_{ii} \mathbf{Z}_i + \sum_{1 \le i < j \le n} \mathcal{C}_{ij} \mathbf{Z}_i \mathbf{Z}_j \tag{185}$$

is the Ising Hamiltonian. In essence, the goal is to prepare – with high probability – a ground state $|q^\star\rangle$ of $\mathbf{C}$.

We begin this chapter by reviewing some relevant literature, namely, the Grover algorithm, the related phase matching condition, and the non-Boolean quantum amplitude amplification (NBAA) algorithm. Building on this, we present a modified NBAA algorithm and analyze conditions in which it is applicable to the Ising problem. In the experimental section, the proposed method is benchmarked

against the original NBAA algorithm and the hybrid quantum-classical method from Chapter 4.

This chapter was developed during supervision of the following Bachelor thesis, on which it is partly based:

- Oettinger J. E. "Non-Binary quantum amplitude amplification for discrete iotimization", *University of Lübeck*, Bachelor thesis, supervised by J. Lellmann and N. Kuete Meli (October 2023).

5.1 Related Work

5.1.1 Grover's Quantum Search

The notation and approach in this section follows Brassard et al. [27].

Let $n \in \mathbb{N}$, $N = 2^n$ and $[N] := \{0, 1, \ldots, N-1\}$. Let $f : [N] \to \{0, 1\}$ be a function partitioning $[N]$ into solutions and non-solutions. We will call an element $q \in [N]$ *a solution* of f if $f(q) = 1$ and *a non-solution* otherwise. The set $[N]$ will be referred to as *database*.

The purpose of the quantum search algorithm, as proposed by Grover [6] and generalized by Brassard et al. [27], is to boost the amplitude of solutions of f in the expense of that of non-solutions.

Assuming that we start from a known state $|0\rangle := |0\rangle^{\otimes n}$, any purely quantum algorithm can be represented as an operator $\mathbf{A}$, which turns the initial state into a superposition of all the basis states corresponding to possible inputs of f:

$$|\psi_0\rangle := \mathbf{A}\,|0\rangle = \sum_{q \in [N]} \alpha_0(q)\,|q\rangle \tag{186}$$

Any such operator can be thought of as splitting the set of basis states into $|\psi_0\rangle = |\psi_+\rangle + |\psi_-\rangle$, where $|\psi_+\rangle$ is a superposition of solutions and $|\psi_-\rangle$ that of non-solutions. Now, let a be the probability that the measurement of $|\psi_0\rangle$ yields a solution, i.e.,

$$\langle\psi_+|\psi_0\rangle = \langle\psi_+|\psi_+\rangle =: a \tag{187}$$
$$\langle\psi_-|\psi_0\rangle = \langle\psi_-|\psi_-\rangle =: 1 - a. \tag{188}$$

Suppose that f is a black box and we repeat the process of running $\mathbf{A}$, performing a measurement and requesting f to check whether the measured outcome is a solution. On average, it will take $\mathcal{O}(1/a)$ evaluations of f to find a solution.

Grover's algorithm achieves this same goal by applying on average $\mathcal{O}(1/\sqrt{a})$ times the operator $\mathbf{A}$, its inverse $\mathbf{A}^\dagger$, and a so-called *quantum oracle*, a gate capable to evaluate f on basis states $|0\rangle, \ldots, |N-1\rangle$, provided that one has access to such an oracle.

The algorithm works as summarized in Algorithm 5.1.1 and uses two key operations, that we now analyse:

Algorithm 5.1.1 (*Grover's Quantum Search [6]*)

Input: $\mathbf{A}$*: Initialization operator,* $\mathbf{O}_\varphi$*: Oracle operator,* K*: Number of iterations.*
Output: *Basis state* $|q^\star\rangle$*, ideally such that* $\varphi(q^\star) = \pi$*, i.e.,* $f(q^\star) = 1$*.*

Let $\mathbf{D} := -\mathbf{A}(\mathbf{I} - 2\,|0\rangle\langle 0|)\mathbf{A}^\dagger$. ▷ *Diffusion operator*
Initialize the system in the state $|\psi_0\rangle = \mathbf{A}\,|0\rangle$. ▷ *Initialization*
for $k = 0, \ldots, K$ ***do***
  $|\psi_{k+1}\rangle \leftarrow \mathbf{D}\mathbf{O}_\varphi\,|\psi_k\rangle$. ▷ *Oracle & diffusion call*
end for
return *Measure* $|\psi_{k+1}\rangle$ *in computational basis and return outcome* $|q^\star\rangle$.

The Oracle.

The oracle evaluates the function f on basis states. For convenience, we introduce the *oracle function* $\varphi : [N] \to \{0, \pi\}$ such that $\varphi(q) = \pi \cdot f(q)$. In the computational basis, the oracle is the diagonal matrix

$$\mathbf{O}_\varphi := \begin{pmatrix} e^{i\varphi(0)} & & \\ & \ddots & \\ & & e^{i\varphi(N-1)} \end{pmatrix}. \tag{189}$$

Its effect on a basis state $|q\rangle$ is

$$\mathbf{O}_\varphi\,|q\rangle = e^{i\varphi(q)}\,|q\rangle = \begin{cases} -\,|q\rangle\,, & \text{if } f(q) = 1, \\ \;\;\,|q\rangle\,, & \text{if } f(q) = 0. \end{cases} \tag{190}$$

As a result, the oracle rotates the phase of the solution by π radians and leaves the system unchanged otherwise.

The Diffusion Operator.

Let $\mathbf{S}_0 = \mathbf{I} - 2\,|0\rangle\langle 0|$. The diffusion operator, defined as

$$\mathbf{D} := -\mathbf{A}\mathbf{S}_0\mathbf{A}^\dagger = -(\mathbf{I} - 2\,|\psi_0\rangle\langle\psi_0|), \tag{191}$$

performs a reflection operation of the state vector about the initial state $|\psi_0\rangle$.

It turns out that Grover's algorithm is a repetition of a composition of two reflections: The oracle $\mathbf{O}_\varphi$ reflects the state vector about the superposition-of-solutions state $|\psi_+\rangle$ and the diffusion operator $\mathbf{D}$ reflects the state vector about the initial state $|\psi_0\rangle$. The *Grover operator* is defined as follows:

Definition 5.1.1 (*Grover Operator [6, 27]*)

The Grover operator *is defined as*

$$\mathbf{G} := \mathbf{D}\mathbf{O}_\varphi = -\mathbf{A}\mathbf{S}_0\mathbf{A}^\dagger\mathbf{O}_\varphi. \tag{192}$$

Let us normalize the states $|\psi_+\rangle$ and $|\psi_-\rangle$ as

$$|\eta_+\rangle := \frac{1}{\sqrt{a}}|\psi_+\rangle \quad \text{and} \quad |\eta_-\rangle := \frac{1}{\sqrt{1-a}}|\psi_-\rangle\,. \tag{193}$$

Interestingly, the 2-dimensional subspace $\mathcal{H}$ spanned by $|\eta_+\rangle$ and $|\eta_-\rangle$ is stable under the action of the Grover's operator. It holds:

Lemma 5.1.1 (*Stability under Grover's operator, follows [27, Lemma 1]*)

The space spanned by $\{|\eta_+\rangle, |\eta_-\rangle\}$ *is invariant under the Grover operator:*

$$\mathbf{G}|\eta_+\rangle = \quad (1-2a)|\eta_+\rangle - 2\sqrt{a(1-a)}\,|\eta_-\rangle\,, \tag{194}$$
$$\mathbf{G}|\eta_-\rangle = 2\sqrt{a(1-a)}\,|\eta_+\rangle + \quad (1-2a)|\eta_-\rangle\,. \tag{195}$$

Proof: From [27, Lemma 1], it holds

$$\mathbf{G}|\psi_+\rangle = (1-2a)|\psi_+\rangle - \quad 2a|\psi_-\rangle\,, \tag{196}$$
$$\mathbf{G}|\psi_-\rangle = 2(1-a)|\psi_+\rangle + (1-2a)|\psi_-\rangle\,, \tag{197}$$

such that

$$\frac{1}{\sqrt{a}}\mathbf{G}|\psi_+\rangle = \quad (1-2a)\cdot\frac{1}{\sqrt{a}}|\psi_+\rangle - \frac{2a\sqrt{1-a}}{\sqrt{a}}\cdot\frac{1}{\sqrt{1-a}}|\psi_-\rangle\,, \tag{198}$$
$$\frac{1}{\sqrt{1-a}}\mathbf{G}|\psi_-\rangle = \frac{2\sqrt{a}(1-a)}{\sqrt{1-a}}\cdot\frac{1}{\sqrt{a}}|\psi_+\rangle + \quad (1-2a)\cdot\frac{1}{\sqrt{1-a}}|\psi_-\rangle\,, \tag{199}$$

yielding the desired result. ■

The normalization in Equation (193) allows to rewrite the initial state $|\psi_0\rangle$ as $|\psi_0\rangle = \sqrt{a}\,|\eta_+\rangle + \sqrt{1-a}\,|\eta_-\rangle$, and we can find an angle $\theta \in [0, \pi/2]$ such that $a = \sin^2(\theta)$, yielding for $\mathbf{G}$ the matrix representation

$$\mathbf{G} = \begin{pmatrix} \cos(2\theta) & \sin(2\theta) \\ -\sin(2\theta) & \cos(2\theta) \end{pmatrix} \tag{200}$$

in the subspace $\mathcal{H}$ with respect to its two basis vectors. This matrix representation as rotation is a very comfortable representation of the Grover iteration: Multiple applications of the same rotation can be collapsed into a single rotation, so that

applying k Grover iterations on $|\psi_0\rangle$ results in

$$|\psi_k\rangle = \mathbf{G}^k |\psi_0\rangle = \sin((2k+1)\theta)\,|\eta_+\rangle + \cos((2k+1)\theta)\,|\eta_-\rangle\,. \tag{201}$$

Consequently, as long as k is not too large, one step in Grover's algorithm rotates the state vector towards the superposition-of-solutions state.

Optimal Number of Iterations.

To maximize the probability of measuring a solution, we want to have the coefficient $\sin((2k+1)\theta)$ as close as possible to one. Therefore we need

$$K = \frac{\pi}{4\theta} - \frac{1}{2} \tag{202}$$

Grover iterations, which is in $\mathcal{O}(1/\sqrt{a})$. The quadratic speedup is formally framed in the following theorem:

Theorem 5.1.1 (*Quadratic speedup of Grover's search [27, Theorem 2]*)

Let $\mathbf{A}$ *be any quantum operator that makes no measurement and the function* $\varphi : [N] \to \{0, \pi\}$ *an oracle function. Let* $a > 0$ *be the initial success probability of* $\mathbf{A}$ *and set* $K = \lfloor \frac{\pi}{4\theta} \rfloor$*, where* θ *is defined such that* $a = \sin^2(\theta)$ *and* $0 < \theta < \frac{\pi}{2}$*. Then, if we compute* $\mathbf{G}^K \mathbf{A}\,|0\rangle$ *where* $\mathbf{G}$ *is the Grover operator from Equation (192) and measure the system, the outcome yields a solution with probability at least* $\max(1-a, a)$.

Consequently, the success rate after the measurement is always at least $\frac{1}{2}$, so that we retain the average computational complexity of $O(1/\sqrt{a})$ even when taking into account that the process might have to be repeated if a non-solution is measured.

Note that K depends on a, which depends on the overlap of the initial state $|\psi_0\rangle$ with the state of solutions $|\psi_+\rangle$ before running the algorithm. The quantum counting algorithm proposed by Brassard et al. [105] allows to approximate this quantity. The basic idea is to embed the Grover's matrix from Equation (200) into the quantum phase estimation algorithm [12, 44], which allows to compute, given a unitary matrix $\mathbf{O}$ and a one of its eigenvectors $|\psi\rangle$, the phase φ such that $\mathbf{O}\,|\psi\rangle = e^{i2\pi\varphi}\,|\psi\rangle$. For the Grover operator $\mathbf{G}$, this phase is effectively $\pm\theta/\pi$.

5.1.2 Phase Matching Condition in Grover's Search

Suppose that in the above binary search problem, instead of $f : [N] \to \{0, 1\}$, the Boolean function is given by $f : [N] \to \{0, C\}$, where $C \in \mathbb{R}$ is a constant that is not necessarily one. In this setting, the quantum oracle takes the form:

$$\mathbf{O}_\varphi^\alpha\,|q\rangle := \begin{cases} e^{i\alpha} & |q\rangle\,, \quad \text{if } f(q) = \text{const}, \\ & |q\rangle\,, \quad \text{if } f(q) = 0, \end{cases} \tag{203}$$

for some $\alpha \in [0, 2\pi]$. Will Grover's search still find a solution?

More generally, let us replace the Grover reflections by arbitrary phase rotations

$$\mathbf{O}_{\varphi}^{\alpha} := -\left(\mathbf{I} - \left(1 - e^{i\alpha}\right) |\eta_+\rangle \langle\eta_+|\right) \tag{204}$$

$$\text{and} \quad \mathbf{D}^{\beta} := -\left(\mathbf{I} - \left(1 - e^{i\beta}\right) |\psi_0\rangle \langle\psi_0|\right), \tag{205}$$

where $\beta \in [0, 2\pi]$ is another phase factor. Long et al. [106] observed that the response to the above question is affirmative if and only if the *phase matching condition* is fulfilled, a condition imposing that $\alpha = \beta$. This phenomenon was analyzed multiple times afterwards, and in particular adapted to find the solution with certainty, meaning that probability for finding a solution is one[107, 108]. It will play a crucial role for the algorithm that we will construct later for solving the Ising problem, deserving a more profound understanding.

Definition 5.1.2 (*Phase matching condition [106]*)

Let $\mathbf{O}_{\varphi}^{\alpha}$ and $\mathbf{D}^{\beta}$ as defined in Equations (204) and (205) be two phase rotation operators and the modified Grover operator be defined as

$$\mathbf{G}^{\alpha,\beta} = \mathbf{D}^{\beta} O_{\varphi}^{\alpha}. \tag{206}$$

We will say that the Grover phase matching condition *is fulfilled if $\alpha = \beta = \pi$.*

For $\mathbf{O}_{\varphi}$ and $\mathbf{D}$ from Equations (204) and (205), Lemma 5.1.1 becomes:

Lemma 5.1.2 (*Stability under the modified Grover operator*)

The space spanned by $\{|\eta_+\rangle, |\eta_-\rangle\}$, is invariant under the modified Grover operator in Equation (206):

$$\mathbf{G}^{\alpha,\beta} |\eta_+\rangle = (-e^{i\alpha} - ae^{i\alpha}(1 - e^{i\beta})) |\eta_+\rangle + e^{i\alpha}(1 - e^{i\beta})\sqrt{a(1-a)} |\eta_-\rangle, \tag{207}$$

$$\mathbf{G}^{\alpha,\beta} |\eta_-\rangle = (1 - e^{i\beta})\sqrt{a(1-a)} |\eta_+\rangle + (-e^{i\beta} - a(1 - e^{i\beta})) |\eta_-\rangle. \tag{208}$$

Proof: It holds

$$\mathbf{G}^{\alpha,\beta} |\eta_+\rangle = \left(\mathbf{I} - \left(1 - e^{i\beta}\right) |\psi_0\rangle \langle\psi_0|\right) \left(\mathbf{I} - \left(1 - e^{i\alpha}\right) |\eta_+\rangle \langle\eta_+|\right) |\eta_+\rangle \tag{209}$$

$$= \left(\mathbf{I} - \left(1 - e^{i\beta}\right) |\psi_0\rangle \langle\psi_0|\right) \left(|\eta_+\rangle - \left(1 - e^{i\alpha}\right) |\eta_+\rangle\right) \tag{210}$$

and

$$\mathbf{G}^{\alpha,\beta} |\eta_-\rangle = \left(\mathbf{I} - \left(1 - e^{i\beta}\right) |\psi_0\rangle \langle\psi_0|\right) \left(\mathbf{I} - \left(1 - e^{i\alpha}\right) |\eta_+\rangle \langle\eta_+|\right) |\eta_-\rangle \tag{211}$$

$$= \left(\mathbf{I} - \left(1 - e^{i\beta}\right) |\psi_0\rangle \langle\psi_0|\right) |\eta_-\rangle. \tag{212}$$

Writing $|\psi_0\rangle$ as $|\psi_0\rangle = \sqrt{a} |\eta_+\rangle + \sqrt{1-a} |\eta_-\rangle$ and rearranging the terms delivers the claim. ■

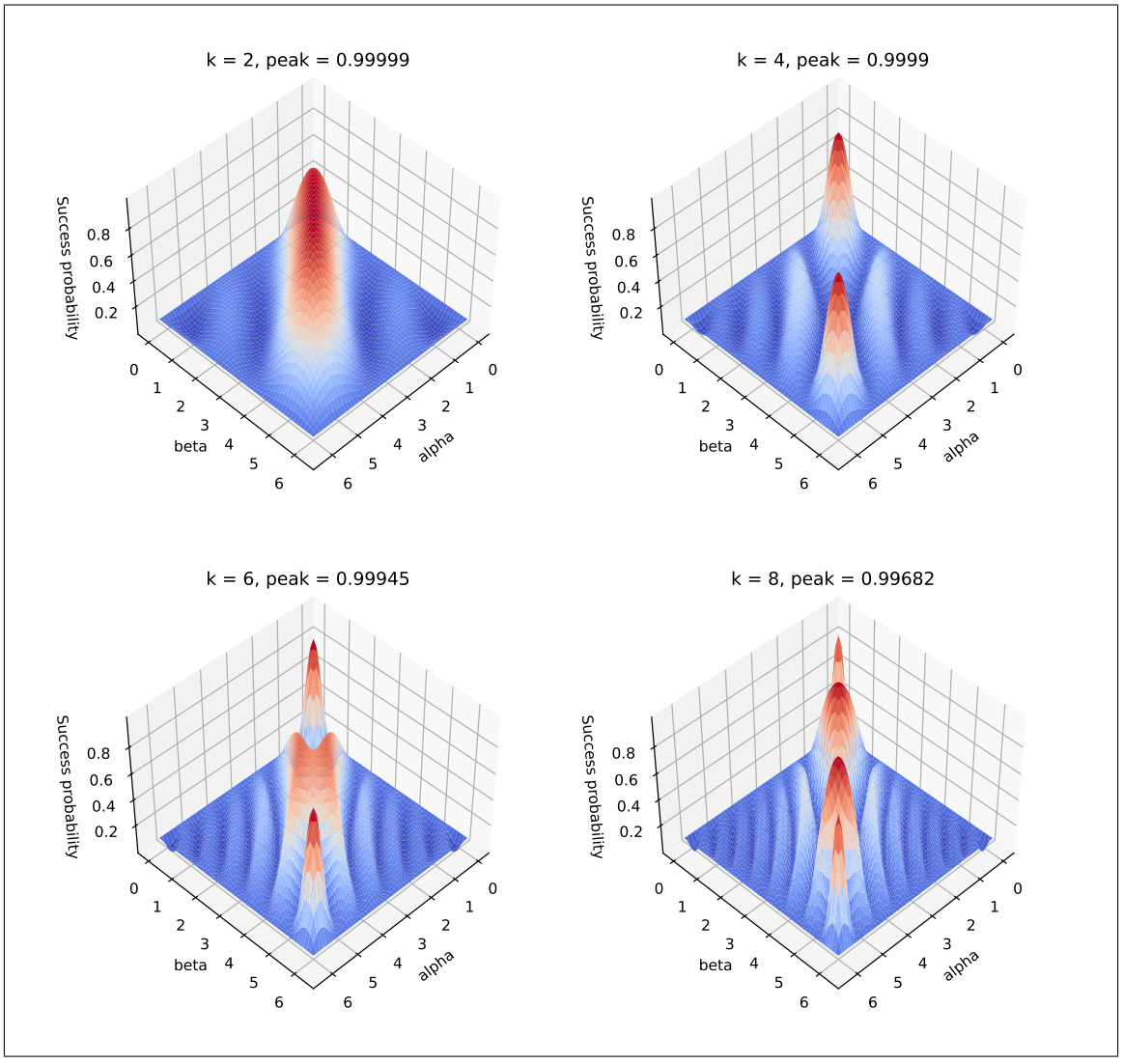


Figure 5.1.1: *Phase matching in quantum search. The initial vector of the algorithm is a state vector* $|\psi_0\rangle = \sqrt{a}\,|\eta_+\rangle + \sqrt{1-a}\,|\eta_-\rangle$ *with* $a = 1/10$*. At each iteration, the oracle marks the solution with phase* α*, while the diffusion operator marks the initial state with phase* β*.* ***(From left to right, top to bottom)*** *On the vertical axis is the probability of measuring a solution,* $|\eta_+\rangle$*, after* k *iterations. We see that the success probability peaks on the diagonal, when* α *and* β *satisfy the phase matching condition.*

Using again that $a = \sin^2(\theta)$, we can re-write $\mathbf{G}^{\alpha,\beta}$ in matrix form as

$$\mathbf{G}^{\alpha,\beta} = \begin{pmatrix} -e^{i\alpha} - \frac{1}{2}e^{i\alpha}(1-e^{i\beta})(1-\cos(2\theta)) & \frac{1}{2}(1-e^{i\beta})\sin(2\theta) \\ \frac{1}{2}e^{i\alpha}(1-e^{i\beta})\sin(2\theta) & -e^{i\beta} - \frac{1}{2}(1-e^{i\beta})(1-\cos(2\theta)) \end{pmatrix}. \tag{213}$$

Figure 5.1.1 showcases the effect of the operator $\mathbf{G}$ for different values of α and β on a state vector $|\psi_0\rangle = \sqrt{a}\,|\eta_+\rangle + \sqrt{1-a}\,|\eta_-\rangle$ with $a = 1/10$. On the z-axis is the probability of measuring $|\eta_+\rangle$ after a total number of k iterations. Indeed, we

see that the success probability boils and peaks on the diagonal, where $\alpha \approx \beta$. The highest peak is obtained for $k = 2$, which is the optimal number of iterations in the ideal Grover's setting, in which $\alpha = \beta = \pi$. In the regions where $\alpha \neq \beta$, the success probability wave never crosses the 0.5 threshold, suggesting that such configurations are inaccurate for an efficient search.

To analytically understand this phenomenon, Long et al. [106] considered that a is in general very small, in the order of $\frac{1}{N}$. This allows to write the approximation $\cos(2\theta) \approx 1$ and $\sin(2\theta) \approx 2\theta$, reducing $\mathbf{G}$ to

$$\mathbf{G}^{\alpha,\beta} = \begin{pmatrix} -e^{i\alpha} & (1-e^{i\beta})\theta \\ e^{i\alpha}(1-e^{i\beta})\theta & -e^{i\beta} \end{pmatrix} \tag{214}$$

Up to the global phase factor $e^{i(\alpha+\beta)/2}$, we can rewrite $\mathbf{G}$ as

$$\mathbf{G}^{\alpha,\beta} = \begin{pmatrix} -e^{i(\alpha-\beta)/2} & \frac{1}{2}e^{-i\alpha/2}\left(e^{-i\beta/2} - e^{i\beta/2}\right)(2\theta) \\ \frac{1}{2}e^{i\alpha/2}\left(e^{-i\beta/2} - e^{i\beta/2}\right)(2\theta) & -e^{i(-\alpha+\beta)/2} \end{pmatrix}. \tag{215}$$

If the phase matching condition $\alpha = \beta$ is satisfied, we have that

$$\mathbf{G}^{\alpha,\beta} = \mathbf{I} + 2\theta \sin(\beta/2)\left(\mathbf{G}^{\alpha,\beta}\right)', \tag{216}$$

where

$$\left(\mathbf{G}^{\alpha,\beta}\right)' = \begin{pmatrix} 0 & -ie^{-i\alpha/2} \\ -ie^{i\alpha/2} & 0 \end{pmatrix} \tag{217}$$

is a skew-Hermitian matrix: $\left(\mathbf{G}^{\alpha,\beta}\right)'^2 = -\mathbf{I}$, $\left(\mathbf{G}^{\alpha,\beta}\right)'^3 = -\left(\mathbf{G}^{\alpha,\beta}\right)'$ and $\left(\mathbf{G}^{\alpha,\beta}\right)'^4 = \mathbf{I}$. Nothing that $\mathbf{G}^{\alpha,\beta}$ in Equation (216) is the first order Taylor approximation of $\exp\left(\mathbf{G}^{\alpha,\beta}\right)'$ and and writing the exact expression of this exponentiation, we get

$$\mathbf{G}^{\alpha,\beta} \approx \exp(i2\theta\sin(\beta/2)\left(\mathbf{G}^{\alpha,\beta}\right)' = \cos(i2\theta\sin(\beta/2))\mathbf{I} + \sin(i2\theta\sin(\beta/2))\left(\mathbf{G}^{\alpha,\beta}\right)' \tag{218}$$

yielding:

$$\left(\mathbf{G}^{\alpha,\beta}\right)^k \approx \begin{pmatrix} \cos\left(2k\theta\sin(\beta/2)\right) & -ie^{-i\alpha/2}\sin\left(2k\theta\sin(\beta/2)\right) \\ -ie^{i\alpha/2}\sin\left(2k\theta\sin(\beta/2)\right) & \cos\left(2k\theta\sin(\beta/2)\right) \end{pmatrix}. \tag{219}$$

By changing the basis $\{|\eta_+\rangle, |\eta_-\rangle\}$ to $\{|\eta_+\rangle, |\tilde{\eta}_-\rangle\} := \left\{|\eta_+\rangle, -ie^{-i\alpha/2}|\eta_-\rangle\right\}$, one finds that $\left(\mathbf{G}^{\alpha,\beta}\right)^k$ is similar to the matrix

$$\left(\tilde{\mathbf{G}}^{\alpha,\beta}\right)^k = \begin{pmatrix} \cos\left(2k\theta\sin(\beta/2)\right) & \sin\left(2k\theta\sin(\beta/2)\right) \\ -\sin\left(2k\theta\sin(\beta/2)\right) & \cos\left(2k\theta\sin(\beta/2)\right) \end{pmatrix}. \tag{220}$$

Starting from the initial vector $|\psi_0\rangle$ in the new basis, the quantum state at the k-th

iteration is given by

$$|\psi_k\rangle \approx \left(\tilde{\mathbf{G}}^{\alpha,\beta}\right)^k |\psi_0\rangle \tag{221}$$

$$= \sin\left((2k\sin(\beta/2)+1)\theta\right)|\eta_+\rangle + \cos\left((2k\sin(\beta/2)+1)\theta\right)|\tilde{\eta}_-\rangle\,. \tag{222}$$

Optimal Number of Iterations.

Again, to maximize the amplitude of solution basis states, we want k such that $(2k\sin(\beta/2)+1)\theta \approx 1$, requiring, for $\beta \neq 0 \mod 2\pi$, to set

$$K = \left\lfloor \frac{1}{\sin(\beta/2)}\left(\frac{\pi}{4\theta} - \frac{1}{2}\right)\right\rfloor. \tag{223}$$

If $\alpha = \beta = \pi$, we are in the Grover setting. Otherwise we have $\sin(\beta/2) < 1$ and we see that more iterations are necessary to reach the peak success probability.

For the case where the phase matching condition is not fulfilled, i.e., $\alpha \neq \beta$, we refer to the analysis of Long et al. [106], where it is shown that the operator $\mathbf{G}$ does not increase the probability of measuring solutions and the quantum search fails.

5.1.3 Non-Boolean Quantum Amplitude Amplification

The above section introduced quantum search for Boolean functions. In this section, we shortly review the *non-Boolean amplitude amplification (NBAA)* algorithm proposed by Shyamsundar [28] to extend the search algorithm to non-Boolean functions. In contrast to Boolean functions that rigorously separate solutions from non-solutions, non-Boolean functions allow for more degrees of freedom to characterize the "goodness" of a solution, akin to objective functions in classical optimization problems such as the Ising problem.

Given a function $f : [N] \to [0,1]$, resp., an oracle function $\varphi : [N] \to [0,\pi]$, the NBAA algorithm aims to find the solution $|q^\star\rangle$ that solves

$$\underset{q\in[N]}{\text{maximize}}\ f(q), \quad \text{resp.,} \quad \underset{q\in[N]}{\text{maximize}}\ \varphi(q). \tag{224}$$

NBAA turns this maximization problem into an equivalent minimization problem

$$\underset{q\in[N]}{\text{minimize}}\ \cos(\varphi(q)). \tag{225}$$

To solve the problem, the domain of f resp. φ is relaxed by introducing an ancilla qubit. The basis of the Hilbert space of computation becomes

$$\{|0,q\rangle\,,\ q\in[N]\}\ \cup\ \{|1,q\rangle\,,\ q\in[N]\}\,. \tag{226}$$

In the above expression, the first register is called *ancilla register* and the second

register the *working register*. The utility of the ancilla qubit will be made clear later in this section.

The NBAA algorithm is summarized in Algorithm 5.1.2. It makes use of oracle calls and diffusion operations to amplify the amplitudes of basis states $|q^\star\rangle$ for the solution $q^\star$ maximizing f, resp. φ.

Algorithm 5.1.2 (*Non-Boolean Amplitude Amplification [28]*)

Input: $\mathbf{A}$*: Initialization operator,* $\mathbf{O}_\varphi$*: Oracle operator,* K*: Number of iterations.*
Output: *Basis state* $|q^\star\rangle$*, ideally such that* $\varphi(q^\star) = \max\{\varphi(q), \quad q \in [N]\}$.

Let $\mathbf{D} = -\mathbf{H} \otimes \mathbf{A}(\mathbf{I} - 2\,|0,0\rangle\langle 0,0|)\mathbf{H}^\dagger \otimes \mathbf{A}^\dagger$. ▷ *Diffusion operator*
Initialize the system in the state $|\psi_0\rangle = \mathbf{H} \otimes \mathbf{A}\,|0,0\rangle$. ▷ *Initialization*
for $k = 0, \ldots, K$ ***do***
  if k *is odd* ***then***
    $|\psi_{k+1}\rangle \leftarrow \mathbf{D}\mathbf{O}_\varphi\,|\psi_k\rangle$. ▷ *Oracle & diffusion call*
  else
    $|\psi_{k+1}\rangle \leftarrow \mathbf{D}\mathbf{O}_\varphi^\dagger\,|\psi_k\rangle$. ▷ *Oracle & diffusion call*
  end if
end for
return *Measure* $|\psi_{k+1}\rangle$ *in computational basis and return outcome* $|q^\star\rangle$.

At the beginning of the algorithm, a Hadamard gate is applied to the ancilla qubit. Here also, the operator $\mathbf{A}$ prepares from the working register initially in the state $|0\rangle$ a supposition $\sum_{q\in[N]} \alpha_0(q)\,|q\rangle$ of elements on $[N]$. We define the initial state $|\psi_0\rangle$ as $|\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_{q\in[N]} \alpha_0(q)(|0,q\rangle + |1,q\rangle)$.

Next, we present the oracle and diffusion operations of NBAA:

The Oracle.

The NBAA-oracle is a controlled operator acting as on basis states according to

$$\mathbf{O}_\varphi\,|0,q\rangle := e^{i\varphi(q)}\quad |0,q\rangle\,, \tag{227}$$
$$\mathbf{O}_\varphi\,|1,q\rangle := e^{-i\varphi(q)}\,|1,q\rangle\,. \tag{228}$$

The algorithm also makes use of the inverse oracle defined as

$$\mathbf{O}_\varphi^\dagger\,|0,q\rangle := e^{-i\varphi(q)}\,|0,q\rangle\,, \tag{229}$$
$$\mathbf{O}_\varphi^\dagger\,|1,q\rangle := e^{i\varphi(q)}\quad |1,q\rangle\,. \tag{230}$$

The Diffusion Operator.

Let $\mathbf{S}_0 = \mathbf{I} - 2\,|0,0\rangle\langle 0,0|$. The NBAA diffusion operator is defined as

$$\mathbf{D} := -(\mathbf{H} \otimes \mathbf{A})\mathbf{S}_0(\mathbf{H}^\dagger \otimes \mathbf{A}^\dagger). \tag{231}$$

Remark a Grover-like oracle: While $\mathbf{H}$ initializes the ancilla register, $\mathbf{A}$ initializes the working register. Hence, $\mathbf{H} \otimes \mathbf{A}$ is the initialization operator for the 2-register quantum system.

To analyze the iterations of the algorithm, we introduce the states

$$|\boldsymbol{\alpha}\rangle := \mathbf{O}_\varphi |\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_{q\in[N]} \alpha_0(q) \left(e^{i\varphi(q)} |0,q\rangle + e^{-i\varphi(q)} |1,q\rangle\right), \tag{232}$$

$$|\boldsymbol{\beta}\rangle := \mathbf{O}_\varphi^\dagger |\psi_0\rangle = \frac{1}{\sqrt{2}} \sum_{q\in[N]} \alpha_0(q) \left(e^{-i\varphi(q)} |0,q\rangle + e^{i\varphi(q)} |1,q\rangle\right), \tag{233}$$

and the angle $\theta \in [0,\pi]$ such that

$$\cos(\theta) := \langle\psi_0|\boldsymbol{\alpha}\rangle = \sum_{q\in[N]} |\alpha_0(q)|^2 \cos(\varphi(q)). \tag{234}$$

The term $\cos(\theta)$ is a weighted average of the cosines of the oracle function values. The ancilla qubit precisely helps to form the cosine from the exponential function, resulting from Euler's formula: $\cos(x) = \left(e^{ix} + e^{-ix}\right)/2$ for $x \in \mathbb{R}$. The following lemma is immediate:

Lemma 5.1.3 (*Stability under the NBAA iterations [28, Equations 51-54]*)

The space spanned by $\{|\psi_0\rangle, |\boldsymbol{\alpha}\rangle, |\boldsymbol{\beta}\rangle\}$ is invariant under the NBAA iterations:

$$\mathbf{D}\mathbf{O}_\varphi |\psi_0\rangle = 2\cos(\theta) |\psi_0\rangle - |\boldsymbol{\alpha}\rangle, \tag{235}$$

$$\mathbf{D}\mathbf{O}_\varphi^\dagger |\boldsymbol{\alpha}\rangle = |\psi_0\rangle, \tag{236}$$

$$\mathbf{D}\mathbf{O}_\varphi^\dagger |\psi_0\rangle = 2\cos(\theta) |\psi_0\rangle - |\boldsymbol{\beta}\rangle, \tag{237}$$

$$\mathbf{D}\mathbf{O}_\varphi |\boldsymbol{\beta}\rangle = |\psi_0\rangle. \tag{238}$$

The lemma confirms that the NBAA iterations alternatingly bring the state vector into the space spanned by $\{|\psi_0\rangle, |\boldsymbol{\alpha}\rangle\}$, then into the space spanned by $\{|\psi_0\rangle, |\boldsymbol{\beta}\rangle\}$, and back. In matrix form, at iteration k we can write, see [28],

$$|\psi_k\rangle = \begin{cases} (|\psi_0\rangle \quad |\boldsymbol{\alpha}\rangle) \begin{pmatrix} 2\cos(\theta) & 1 \\ -1 & 0 \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & \text{if } k \text{ is odd,} \\ (|\psi_0\rangle \quad |\boldsymbol{\beta}\rangle) \begin{pmatrix} 2\cos(\theta) & 1 \\ -1 & 0 \end{pmatrix}^k \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & \text{if } k \text{ is even,} \end{cases} \tag{239}$$

which results in

$$|\psi_k\rangle = \begin{cases} \frac{1}{\sin(\theta)} \left(\sin((k+1)\theta) |\psi_0\rangle - \sin(k\theta) |\boldsymbol{\alpha}\rangle\right), & \text{if } k \text{ is odd,} \\ \frac{1}{\sin(\theta)} \left(\sin((k+1)\theta) |\psi_0\rangle - \sin(k\theta) |\boldsymbol{\beta}\rangle\right), & \text{if } k \text{ is even.} \end{cases} \tag{240}$$

We read out the amplitudes of the basis states at this iteration as

$$\alpha_k(\ell, q) = \begin{cases} \frac{\alpha_0(q)}{\sin(\theta)} \left(\sin((k+1)\theta)\,|\psi_0\rangle - \sin(k\theta)e^{i\varphi(q)}\right), & \text{if } k+\ell \text{ is odd,} \\ \frac{\alpha_0(q)}{\sin(\theta)} \left(\sin((k+1)\theta)\,|\psi_0\rangle - \sin(k\theta)e^{-i\varphi(q)}\right), & \text{if } k+\ell \text{ is even.} \end{cases} \tag{241}$$

Let $p_k(\ell, q) := \frac{1}{2}\,|\alpha_k(\ell, q)|^2$ denote the probabilities of measuring the states $|\ell, q\rangle$ for $\ell = 0, 1$. It follows that we measure each basis state $|q\rangle$ at iteration k with probability

$$p_k(q) = p_k(0, q) + p_k(1, q) \tag{242}$$

$$= |\alpha_0(q)|^2 \left(1 - \lambda_k \left(\cos(\varphi(q)) - \cos(\theta)\right)\right), \tag{243}$$

where, assuming $\sin(\theta) \neq 0$,

$$\lambda_k := \frac{2\sin(k\theta)\sin((k+1)\theta)}{\sin^2(\theta)}. \tag{244}$$

It is important to mention that $p(0, q) = p(1, q)$, meaning that the ancilla and working register are not entangled. The consequence is that measuring the ancilla qubit or not has no influence on the outcome of the algorithm.

Optimal Number of Iterations.

Now, we want to compute the number of NBAA iterations needed to maximally amplify the amplitude of the solution.

From Equation (243), we see that $p_k(q)$ is linear in $\cos(\varphi(q))$. If the slope λ_k is positive, the states with $\cos(\varphi(q)) < \cos(\theta)$ are amplified; states satisfying the converse inequality are attenuated. The amplification factor depends on the magnitude of the slope λ_k and on the difference $\cos(\varphi(q)) - \cos(\theta)$. For our target solution, this difference is maximal compared to all other possible q. Hence, if we regard λ_k in Equation (244) as a function of k, as long as $\lambda_k > 0$ and increases, the probability of measuring the $q^\star$ maximizing f gets amplified the most in each iteration.

The derivative of λ_k at k is zero if $\sin((2k+1)\theta) = 0$. Hence, for λ_k to monotonically increase, it is sufficient that $0 \leq (2k+1)\theta \leq \pi$. The algorithm stops right before the iteration that will cause λ_k to decrease, yielding for NBAA the optimal number of iteration

$$K = \left\lfloor \frac{\pi}{2\theta} \right\rfloor. \tag{245}$$

Example: [28, Section 5] As an example, we consider the non-binary optimization problem with $n = 8$ and $N = 2^n = 256$ basis states. The oracle function is set to the linear function

$$\varphi(q) = \frac{q}{255}\frac{\pi}{4}, \quad \text{for } q = 0, 1, \ldots, 255. \tag{246}$$

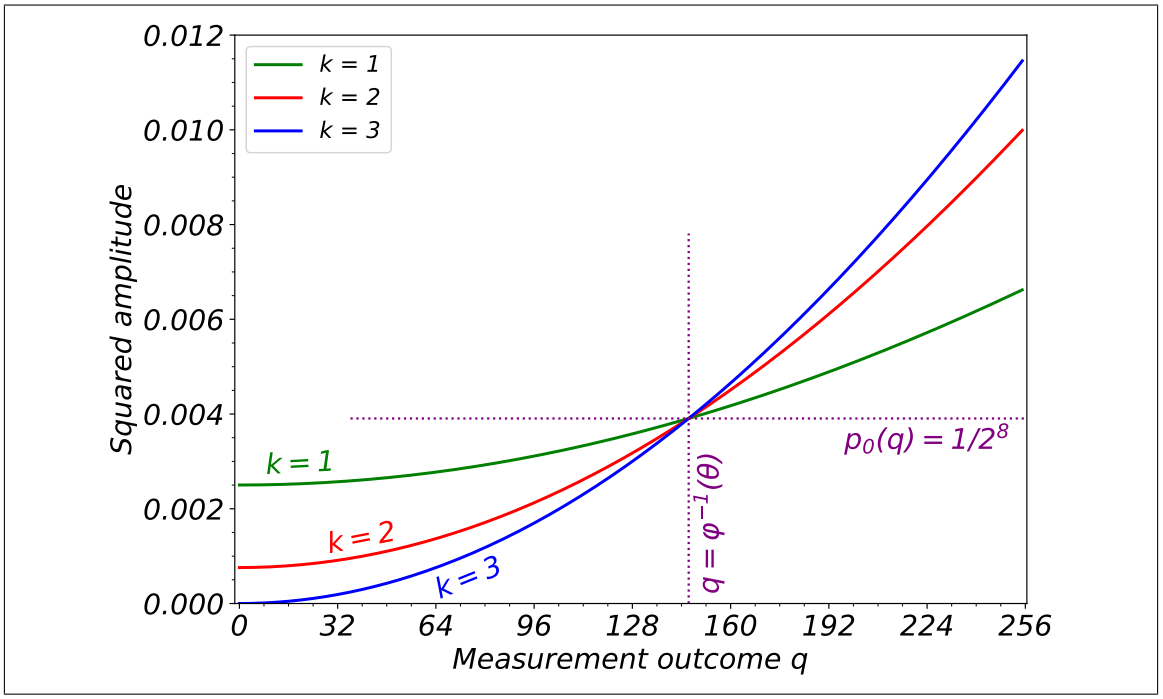


Figure 5.1.2: *NBAA amplification histograms on the example oracle function in Equation (246), image adapted from [28, Figure 9]. The optimal solution is $q^\star = 255$ and the optimal number of iterations is $K = 3$. The average of the cosines of the oracle function values, c.f. Equation (234), is obtained as $q = \varphi^{-1}(\theta)$. The amplitudes of all basis states with value less than the value of q are gradually decreased while those for basis states above q are amplified for $k = 1, 2, 3$.*

Let the initial state be

$$|\psi_0\rangle = \frac{1}{\sqrt{256}} \sum_{q=0}^{255} |q\rangle \,. \tag{247}$$

Then, we have

$$\cos(\theta) = \frac{1}{256} \sum_{q=0}^{255} \cos\left(\frac{q}{255}\frac{\pi}{4}\right) \approx 0.9001 \tag{248}$$

and thus $\theta \approx 0.4507$. From Equation (245), we obtain that the optimal number of iterations is $K = 3$.

Figure 5.1.2 visualizes the amplification histograms for this example for $k = 1, 2, 3$. For each basis state, the squared amplitude, i.e., probability of measurement in Equation (243), is calculated analytically. We see that amplitudes of all basis states with oracle function value above $\cos(\theta)$ are gradually decreased while those with oracle function value below $\cos(\theta)$ are amplified for $k = 1, 2, 3$. The histograms clearly peaks at the solution, thus agreeing with the expectations in Section 5.1.3.

Discussion.

Grover's Boolean search guarantees to find the solution with a probability at least 0.5. In NBAA after the optimal number of iterations, the solution certainly has

the highest amplitude, but, as illustrated in the NBAA example, not necessarily greater than 0.5. A natural question emanating from NBAA is whether we can amplify the amplitude of the solution further.

5.2 Proposed PM-NBAA for non-Boolean Functions

In this section, we propose and analyze a modification of NBAA termed *PM-NBAA*, for *Phase Matching-NBAA*. It is based on the observation that, under some assumptions, repeatedly applying the oracle instead of alternating between the oracle and its inverse as in NBAA can amplify the amplitude of the solution further, possibly at the cost of more iterations than required for NBAA.

Assumptions.

The phase matching condition from section 5.1.2 suggests that the phase α of the solution must match with the phase β of the diffusion in order to solve the search problem efficiently. Furthermore, it has been shown that the closer the phases are to π, the faster the search.

However, we know from Section 5.1.3 that an important requirement for the applicability of NBAA is that $\cos(\theta)$ in Equation (234) is positive. Scaling and shifting $\cos(\theta)$ to satify this requirement requires to scale the oracle function φ in an appropriate interval, such that the cosine values are positive. Considering now an example configuration in which we restrict NBAA to a binary search problem with the oracle function scaled such that $\varphi(q) \in \{0, \alpha\}$. Since the diffusion operation of NBAA is $\mathbf{D} = -(\mathbf{I} + 2\left|\psi_0\right\rangle\left\langle\psi_0\right|)$, we have $\beta = \pi$ and since the oracle marks the solution as $\mathbf{O}_\varphi \left|q^\star\right\rangle = e^{i\alpha}\left|q^\star\right\rangle$, having $\alpha \neq \pi$ causes the phases to no longer match, which in general greatly reduces the probability of measuring the state of the solution. This problem is addressed by using different scaling intervals in a two-step optimization process.

We assume the following:

1. Initial Good Overlap with the Solution.

   In a first step, the initial state

   $$\left|\psi_0\right\rangle := \mathbf{D}\mathbf{O}_{\varphi_{\text{NBAA}}} \left|0\right\rangle^{\otimes n} := \frac{1}{\sqrt{2}} \sum_{q\in[N]} \alpha_0(q)(\left|0, q\right\rangle + \left|1, q\right\rangle) \tag{249}$$

   is prepared to have a good overlap with the solution $q^\star$, meaning that the amplitude $|\alpha_0(q^\star)|$ of the solution is greater than those of all non-solutions. Additionally, we will require the phases of the complex amplitudes $\alpha_0(q)$ to have the same ordering as $\varphi(q)$. This condition can be satisfied by scaling φ to map into $[0, \frac{\pi}{2}]$ and running one NBAA Iteration. We will call this first oracle function φ_{NBAA}.

2. Phase Matching Condition.

   In the second step of the algorithm, the oracle function φ is re-scaled to map into $[0, \pi]$ in a way to mark the solution by the phase $\varphi(q^\star) = \pm\pi$, satisfying the phase matching condition. This condition justifies the name of the algorithm, PM-NBAA for *Phase Matching-NBAA*. We will call the second oracle function φ_{PM}. In this second step of the algorithm, the state vector of the system is computed as

$$|\psi_{k+1}\rangle = \mathbf{DO}_{\varphi_{\text{PM}}} |\psi_k\rangle \,. \tag{250}$$

The Modified Algorithm.

The full PM-NBAA algorithm is shown in Algorithm 5.2.1. Different to plain NBAA is the preparation of the initial state, and that the algorithm does not alternate between the oracle and its inverse. PM-NBAA can be seen as a generalization of Grover's quantum search algorithm to non-Boolean function for an arbitrary initial amplitude distribution [109].

Algorithm 5.2.1 (*Phase Matching-NBAA (PM-NBAA)*)

Input: $\mathbf{A}$*: Initialization operator,* $\mathbf{O}_\varphi$*: Oracle operator,* K*: Number of iterations.*
Output: *Basis state* $|q^\star\rangle$*, ideally such that* $\varphi(q^\star) = \max\{\varphi(q), \quad q \in [N]\}$ *.*

Let $\mathbf{D} := -(\mathbf{H} \otimes \mathbf{A})(\mathbf{I} - 2\,|0,0\rangle\langle 0,0|)(\mathbf{H}^\dagger \otimes \mathbf{A}^\dagger)$. ▷ *Diffusion operator*
Scale φ *to map into* $[0, \frac{\pi}{2}]$*, get* φ_{NBAA}. ▷ *Satisfy the initial good overlap condition*
Compute $|\psi_0\rangle \leftarrow \mathbf{DO}_{\varphi_{\text{NBAA}}} |0\rangle^{\otimes n}$ ▷ *Initialization*
Scale φ *to map into* $[0, \pi]$*, get* φ_{PM}. ▷ *Satisfy the phase matching condition*
for $k = 0, \ldots, K$ ***do***
  $|\psi_{k+1}\rangle \leftarrow \mathbf{DO}_{\varphi_{\text{PM}}} |\psi_k\rangle$. ▷ *Oracle & diffusion call*
end for
return *Measure* $|\psi_{k+1}\rangle$ *in computational basis and return outcome* $|q^\star\rangle$.

Analysis.

In the following, we track the amplitudes of the basis states and inductively show that the amplitude of the solution is amplified.

Let

$$|\psi_k\rangle = \frac{1}{\sqrt{2}} \sum_{q \in [N]} \left(\alpha_k(0,q)\,|0,q\rangle + \alpha_k(1,q)\,|1,q\rangle\right) \tag{251}$$

be the state of the quantum system after iteration $k > 0$. After one iteration of

PM-NBAA, see Equation (250), we have

$$|\psi_{k+1}\rangle = \mathbf{DO}_{\varphi_{\text{PM}}} |\psi_k\rangle \tag{252}$$

$$= \frac{1}{\sqrt{2}} \sum_{q\in[N]} \left(2\alpha_0(q)\cos(\theta_k) - \alpha_k(0,q)e^{i\varphi_{\text{PM}}(q)}\right)|0,q\rangle + \frac{1}{\sqrt{2}} \sum_{q\in[N]} \left(2\alpha_0(q)\cos(\theta_k) - \alpha_k(1,q)e^{-i\varphi_{\text{PM}}(q)}\right)|1,q\rangle\,, \tag{253}$$

with

$$\cos(\theta_k) = \frac{1}{2}\sum_{q\in[N]} |\alpha_0(q)| \left(\alpha_k(0,q)e^{i\varphi_{\text{PM}}(q)} + \alpha_k(1,q)e^{-i\varphi_{\text{PM}}(q)}\right). \tag{254}$$

The amplitudes of the basis states at this iteration are obtained from Equation (253) as

$$\begin{cases} \alpha_{k+1}(0,q) &= 2\alpha_0(q)\cos(\theta_k) - \alpha_k(0,q)e^{i\varphi_{\text{PM}}(q)}, \\ \alpha_{k+1}(1,q) &= 2\alpha_0(q)\cos(\theta_k) - \alpha_k(1,q)e^{-i\varphi_{\text{PM}}(q)}. \end{cases} \tag{255}$$

To understand the behavior of the algorithm, it is interesting to write down the complex amplitude $\alpha_k(\ell,q) = x + iy$ in the form $|\alpha_k(\ell,q)|\, e^{i\epsilon_k(\ell,q)}$ for $\ell = 0,1$, where $|\alpha_k(\ell,q)| = \sqrt{x^2+y^2}$ and $\epsilon_k(\ell,q) = \text{atan2}(y,x)$ are the magnitude and the phase of $\alpha_k(\ell,q)$. From Equation (255), we see that if $|\alpha_k(0,q)| = |\alpha_k(1,q)|$ and $\epsilon_k(0,q) = -\epsilon_k(1,q)$, then $|\alpha_{k+1}(0,q)| = |\alpha_{k+1}(1,q)|$ and $\epsilon_{k+1}(0,q) = -\epsilon_{k+1}(1,q)$ for all k. This evidently holds, as the initial state has been prepared by NBAA, see Equation (241).

Again, we denote by $p(\ell,q) = \frac{1}{2}|\alpha_k(\ell,q)|^2$ the probability of measuring $|\ell,q\rangle$ for $\ell = 0,1$. It holds from Equation (255) that the probability of measuring each basis state is

$$p_{k+1}(q) = p_{k+1}(0,q) + p_{k+1}(1,q) \tag{256}$$

$$= |\alpha_k(0,q)|^2 + 4\,|\alpha_0(q)|^2\cos^2(\theta_k) - 4\,|\alpha_0(q)|\,|\alpha_k(0,q)|\cos\left(\varphi(q) + \epsilon_k(0,q)\right)\cos(\theta_k). \tag{257}$$

Furthermore, we can rewrite $\cos(\theta_k)$ from Equation (254) as

$$\cos(\theta_k) = \sum_{q\in[N]} |\alpha_0(q)|\,|\alpha_k(0,q)|\cos\left(\varphi(q) + \epsilon_k(0,q)\right), \tag{258}$$

which again has the form of a weighted average of the $\cos\left(\varphi(q) + \epsilon_k(0,q)\right)$ terms.

Comparing Equation (257) and Equation (258) allows to make the following statement: At each iteration, the cosine the oracle function values are modified according to

$$\cos(\varphi(q)) \mapsto |\alpha_k(0,q)|\cos\left(\varphi(q) + \epsilon_k(0,q)\right) \tag{259}$$

and compared to their average value $|\alpha_0(q)|\cos(\theta_k)$. As long as $\cos(\theta_k)$ is positive, the amplitudes of basis states with cost below the (iteration-dependent) average are amplified and those of basis states with oracle function values above the average are attenuated. The basis states with oracle function values equal to the average are always left unchanged. This statement is very similar, but not equal to that of NBAA. However, while NBAA compares the same oracle function values to the same average value, PM-NBAA iteratively adjusts the oracle function values and the average.

To prove the effectiveness of PM-NBAA, the only thing left to guarantee is that, according to the sign of the average and within a certain number of iterations, the solution $q^\star$ always has its oracle function value on the right side of the average to get amplified.

We now take a closer look on the phases $\epsilon_k(0,q)$. After running the single NBAA iteration in the first step of the algorithm – which supposes that $\cos(\theta)$ is positive – we obtain, according to the $\mathrm{atan2}$ formulae, the following expressions for the phases, c.f. Equation (241):

- If q is a basis state such that $(2\cos(\theta) - \cos(\varphi_{\mathrm{NBAA}}(q))) > 0$ – which is a condition that is fulfilled by the solution $q^\star$, we have:

$$\epsilon_0(0,q) = \arctan\left(\frac{\mathrm{Im}(\alpha_{K_{\mathrm{NBAA}}}(0,q))}{\mathrm{Re}(\alpha_{K_{\mathrm{NBAA}}}(0,q))}\right) \tag{260}$$

$$= \arctan\left(\frac{-\sin(\varphi_{\mathrm{NBAA}}(q))}{2\cos(\theta) - \cos(\varphi_{\mathrm{NBAA}}(q))}\right). \tag{261}$$

- If q is a basis state such that $(2\cos(\theta) - \cos(\varphi_{\mathrm{NBAA}}(q))) < 0$, we have:

$$\epsilon_0(0,q) = \arctan\left(\frac{\mathrm{Im}(\alpha_{K_{\mathrm{NBAA}}}(0,q))}{\mathrm{Re}(\alpha_{K_{\mathrm{NBAA}}}(0,q))}\right) - \pi \tag{262}$$

$$= \arctan\left(\frac{-\sin(\varphi_{\mathrm{NBAA}}(q))}{2\cos(\theta) - \cos(\varphi_{\mathrm{NBAA}}(q))}\right) - \pi. \tag{263}$$

Note that $\epsilon_0(0,q)$ is well-defined as the numerator is always negative for φ_{NBAA} mapping into $[0, \frac{\pi}{2}]$. In both cases, ϵ_0 is negative and $q^\star = \arg\max_{q\in[N]} \epsilon_0(0,q)$. This is the reason for requiring the phases to have the same ordering as the oracle function values, as such the angle $\epsilon_k(0,q^\star)$ for subsequent iterations remains near to zero and ensures the phase matching condition. Indeed, for the subsequent PM-NBAA iterations the oracle function φ is re-scaled to satisfy $\varphi_{\mathrm{PM}}(q^\star) = \pm\pi$, we have from Equation (255)

$$\epsilon_{k+1}(0,q^\star) = \arctan\left(-\frac{|\alpha_k(0,q^\star)|\sin\left(\varphi_{\mathrm{PM}}(q^\star) + \epsilon_k(0,q^\star)\right)}{2|\alpha_0(q)|\cos(\theta_k) - |\alpha_k(0,q^\star)|\cos\left(\varphi_{\mathrm{PM}}(q^\star) + \epsilon_k(0,q^\star)\right)}\right) \tag{264}$$

goes to zero for k going to ∞.

Assuming now that the phase matching and ideal scaling conditions are fulfilled, we have $\varphi_{\text{PM}}(q^\star) = \pm\pi$, and $|\alpha_k(\ell, q^\star)|^2 > |\alpha_k(\ell, q)|^2$ for all $q \neq q^\star$. Hence, for positive $\cos(\theta_k)$, that

$$\cos(\varphi_{\text{PM}}(q^\star) + \epsilon_k(0, q^\star)) \leq \cos\left(\varphi_{\text{PM}}(q) + \epsilon_k(0, q)\right), \tag{265}$$

$$-4\,|\alpha_0(q)|\,|\alpha_k(0, q^\star)|\cos(\theta_k) \leq -4\,|\alpha_0(q)|\,|\alpha_k(0, q)|\cos(\theta_k). \tag{266}$$

It follows from Equation (257) that $p_{k+1}(q^\star) \geq p_k(q^\star)$ and $p_{k+1}(q^\star) \geq p_{k+1}(q)$ for all $q \neq q^\star$.

This means for the amplitude of the solution state $q^\star$ at iteration $k+1$ not only that it is amplified compared to its value at iteration k, but also that it is greater than the amplitudes of all other non-solution states.

Optimal Number of Iterations.

In the PM-NBAA algorithm, the amplitude of the solution is boosted as long as $\cos(\theta_k)$ is positive. Unfortunately, we cannot currently analytically say when this happens, which also makes it impossible to find a closed expression for the optimal number of iterations K. In our experiments, we set $K = \sqrt{N}$.

Example: To illustrate the functioning of PM-NBAA, we reconsider the NBAA example with $N = 2^8 = 255$ and

$$\varphi(q) = \frac{q}{255}\frac{\pi}{4}, \quad \text{for } q = 0, 1, \ldots, 255. \tag{267}$$

The first step of the algorithm consists in running NBAA to prepare the quantum system in a state that fulfils the initial good overlap condition, since $\varphi(q) \in [0, \frac{\pi}{4}] \subset [0, \frac{\pi}{2}]$ already satisfies our requirement, we have $\varphi_{\text{NBAA}}(q) \mapsto \varphi(q)$. After that, the oracle function is re-scaled as $\varphi_{\text{PM}}(q) \mapsto 4 \cdot \varphi(q)$, thus marking the solution with the phase $\alpha = \pi$ and fulfilling the phase matching condition. The PM-NBAA algorithm is executed for $K = \sqrt{N} = 16$ iterations.

In Figure 5.3.1, we show the amplification histograms for $k = 1, 8, 16$. The amplification histogram are calculated analytically according to Equation (257). We see that PM-NBAA certainly needs more iterations than NBAA, but clearly amplifies the amplitude of the solution more over NBAA.

5.3 Solving the Ising Problem

We now turn towards applying the NBAA and proposed PM-NBAA methods for solving the Ising problem, which comes down to defining the initialization and oracle operators for the problem. For the oracle, we use the UQIsing circuit from the previous chapter. The quantum system now consist of three registers: A 1-qubit *ancilla register* from the (PM-)NBAA algorithm, a 1-qubit *cost register* from UQIsing, and, finally, the n-qubits *working register*.

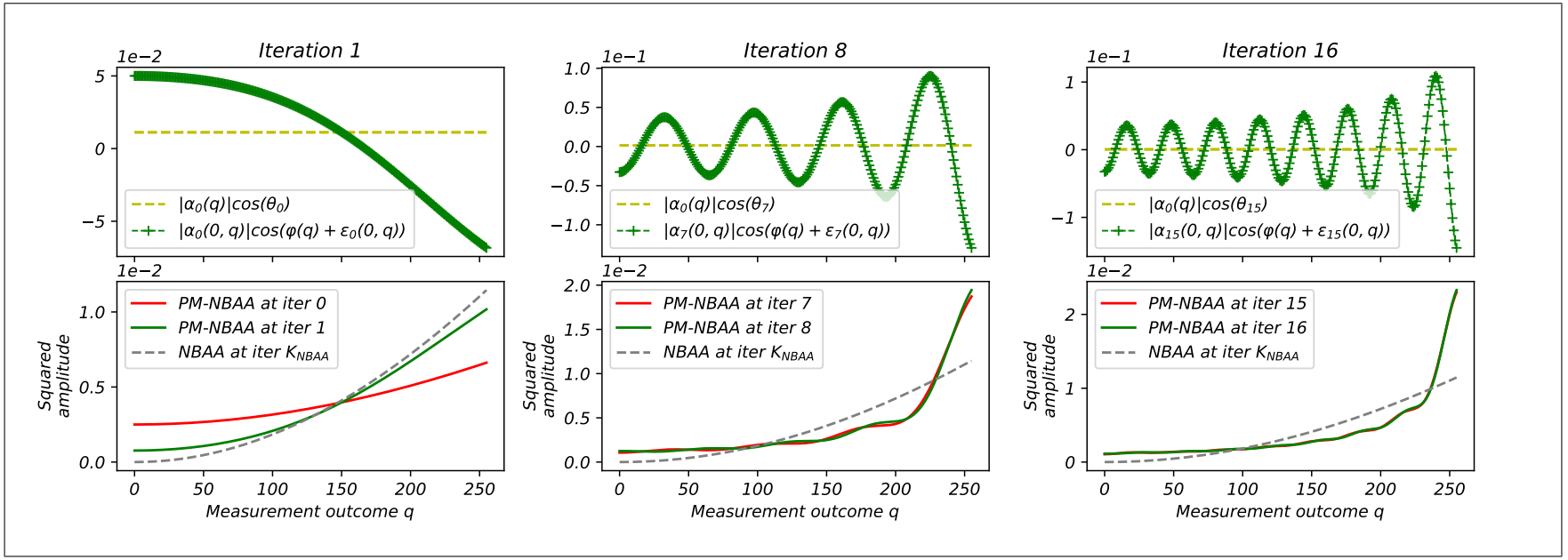


Figure 5.3.1: *An illustrative example of how the proposed PM-NBAA works.* ***(Top row)*** *Iteration dependent oracle function values and their average, see Equations (257) and (258)* ***(Bottom row)*** *Squares of the amplitudes computed by NBAA after the optimal K_{NBAA} iteration and PM-NBAA after iteration $k = 1, 8, 16$, compared to the situation at iteration $k = 0, 7, 15$. The squared amplitudes were computed analytically according to Equations (243) and (257). One iteration of NBAA is run first to prepare the system in a state that fulfills the initial good overlap condition, according to the first step of the algorithm Equation (249). The PM-NBAA quantum state at iteration $k = 0$ is the output of NBAA after one iteration. From $k = 1$, the oracle is re-scaled to fulfill the phase matching condition. We clearly see that PM-NBAA continues to amplify the amplitude of the solution, continuing the work of NBAA.*

In Chapter 4, we introduced for the Ising Hamiltonian the unitary operator

$$\mathbf{U}(\mathbf{C}, a, b) := \begin{pmatrix} \cos(\hat{\mathbf{C}}) & -\sin(\hat{\mathbf{C}}) \\ \sin(\hat{\mathbf{C}}) & \cos(\hat{\mathbf{C}}) \end{pmatrix}, \quad \hat{\mathbf{C}} := a\mathbf{C} + b\mathbf{I}, \tag{268}$$

where a and b were chosen to fit the oracle function values in an interval where the cosine is monotone, c.f. Equation (151). The eigenvectors of $\mathbf{U}(\mathbf{C}, a, b)$ have the form $|\psi_\pm\rangle := |\pm_i\rangle \otimes |q\rangle$, where $q \in [N]$ and $|\pm_i\rangle := \frac{1}{\sqrt{2}}(|0\rangle \pm i\,|1\rangle)$, with i being the complex unit. The associated eigenvalues are given by $\sigma_q := \exp(i\,\langle q|\hat{\mathbf{C}}|q\rangle)$.

Initialization.

At the beginning of the algorithm, the ancilla and cost qubits are prepared in the entangled state $\frac{1}{2}(|0, -_i\rangle + |1, +_i\rangle)$. Let

$$\tilde{\mathbf{H}} = \frac{1}{2}\Big((|0\rangle\langle 0| \otimes \mathbf{HSZ}) + (|1\rangle\langle 1| \otimes \mathbf{HS}) \Big) \tag{269}$$

be the two-qubits operator preparing this entanglement. The operator $\tilde{\mathbf{A}}$ prepares from the working register, initially in the state $|0\rangle$, a superposition $\sum_{q\in[N]} \alpha_0(q)\,|q\rangle$ of all possible basis states. The complete initial state is

$$|\psi_0\rangle = \frac{1}{2}\sum_{q\in[N]} \alpha_0(q)(|0, -_i, q\rangle + |1, +_i, q\rangle). \tag{270}$$

The initialization operator is given by $\mathbf{A} = \tilde{\mathbf{H}} \otimes \tilde{\mathbf{A}}$.

Oracle.

The complete three-register oracle for solving the Ising problem is given by the operator $\mathbf{O}_\varphi = \mathbf{I} \otimes \mathbf{U}(\mathbf{C}, a, b)$, whose effect on the basis states is

$$\mathbf{O}_\varphi\,|0, -_i, q\rangle = e^{i\varphi(q)}\ |0, -_i, q\rangle\,, \tag{271}$$

$$\mathbf{O}_\varphi\,|1, +_i, q\rangle = e^{-i\varphi(q)}\,|1, +_i, q\rangle\,, \tag{272}$$

where $\varphi(q) = \langle q|\hat{\mathbf{C}}|q\rangle$ is the oracle function.

In Equations (271) and (272) the first qubit is the (PM-)NBAA ancilla qubit, while the two last register form together the (PM-)NBAA working register, hence preserving the register structure of the algorithm and also the analytical analysis.

Computing the Average Cost and Number of Iterations for NBAA.

Our principal reason for using the UQIsing unitary $\mathbf{U}(\mathbf{C}, a, b)$ as oracle is that it allows for computing an estimate of $\cos(\theta)$, the average oracle function values of all basis states. This is achieved by preparing the working register in the perfect

superposition state $|\psi_0\rangle = \sum_{q\in[N]} \alpha_0(q)\,|q\rangle$, the cost qubit in the $|0\rangle$ state, and running the UQIsing circuit:

$$\langle \mathbf{U}\rangle = \langle\psi_0|\mathbf{U}|\psi_0\rangle = \sum_{q\in[N]} |\alpha_0(q)|^2 \cos(\varphi(q)) = \cos(\theta) \tag{273}$$

Having $\cos(\theta)$ allows to estimate the optimal number of interactions required to run NBAA via Equation (245).

We note that the QAOA unitary operator $\mathbf{U}(\mathbf{C}, 1)$, c.f., Section 4.1, could be used a oracle circuit as well, but it does not allow to estimate the value $\cos(\theta)$ needed to compute the optimal number of iterations. Estimating $\cos(\theta)$ would require, for example, to use the quantum mean estimation algorithm described in [28], which incurs a considerable additional computation cost.

5.4 Experimental Results

In this section, we apply the NBAA and proposed PM-NBAA algorithms to the Ising problem and compare the results to those of the variational UQIsing method developed from the previous chapter. The algorithms are again tested on random graphs generated using the PYTHON language package NETWORKX [99]. The unary and quadratic edge weights are all randomly and uniformly chosen in the range $[-10, 10]$ and the graphs are all fully connected. The quantum circuits are implemented in PYTHON and simulated in a noise-free framework using the QISKIT library and the IBM-QASM simulator [7]. We perform 1024 measurement shots for evaluating each circuit.

5.4.1 Benchmark Metrics

As in the previous chapter, we denote by $|\psi^\star\rangle$ the proposed (basis) ground state returned by each method (NBAA, PM-NBAA, and UQIsing). The approximation ratio and index as in the previous chapter will serve as benchmark metrics:

- As a remainder, the approximation ratio is defined as

$$r(\psi^\star) := \frac{\langle\psi^\star|\mathbf{C}|\psi^\star\rangle - \mathcal{C}_{max}}{\mathcal{C}_{min} - \mathcal{C}_{max}} = 1 - \frac{\langle\psi^\star|\mathbf{C}|\psi^\star\rangle - \mathcal{C}_{min}}{\mathcal{C}_{max} - \mathcal{C}_{min}} \tag{274}$$

  where $\mathcal{C}_{min}$ and $\mathcal{C}_{max}$ are the lowest and highest energy values.

- We re-define the approximation index i as

$$i(\psi^\star) := |\alpha_K(q^\star)|^2\,, \tag{275}$$

  i.e., the approximation index now ranges from 0 to 1 and gives the probability of measuring the optimal solution $q^\star$.

Each method is run for K maximum iterations. K_{NBAA} is computed according to Equations (245) and (273), K_{PM} is set to $\sqrt{N}$ and K_{UQIsing} to 30.

5.4.2 Benchmark Results

Our benchmark results are presented in Figure 5.4.1. The approximation ratio and index are averaged over 20 random graph instances. NBAA is run for two intervals, as this is crucial for scaling the average cost $\cos(\theta)$: The oracle function φ is scaled to fit into the interval $[0, b]$ for NBAA. and the experiments are run once for $b = \frac{\pi}{2}$ and once for $b = \frac{\pi}{4}$. The different values for b do not induce a notable difference in the approximation ratio and index.

On the other hand, see that PM-NBAA amplifies the amplitude of the solution more than NBAA, but not necessarily enhances the approximation ratio. We attribute this to the fact that PM-NBAA does no ensure that non-solutions are suppressed, but merely that the probability of the solution increases at each iteration. Hence, in PM-NBAA, non-solutions could contribute more to the approximation ratio. In NBAA on the contrary, for both values of b, the amplitude of each basis state is proportional to it "goodness", yielding the good approximation ratios despite the low approximation indices. UQIsing performs better, in terms of approximation ratio and indices, than both NBAA and PM-NBAA.

5.5 Conclusion

To conclude, this chapter has investigated how to solve the Ising problem using a fully universal quantum algorithm. We applied the existing NBAA algorithm [28] to the problem by constructing an oracle function for the algorithm. In order to further amplify the probability of measuring the solution , we introduced a modified version of the algorithm that we called PM-NBAA, which aims to approximately satisfy the phase matching condition of Grover-like quantum search methods. The resulting PM-NBAA certainly enhance the amplitude of the solution, but not necessarily improve the approximation ratio. Nonetheless, NBAA and PM-NBAA are fully quantum methods that do not require any outer classical optimization support as in the variational UQIsing method [25]. PM-NBAA, compared to UQIsing, is not trapped into local minima, and compared to NBAA, delivers a higher probability of measuring a solution.

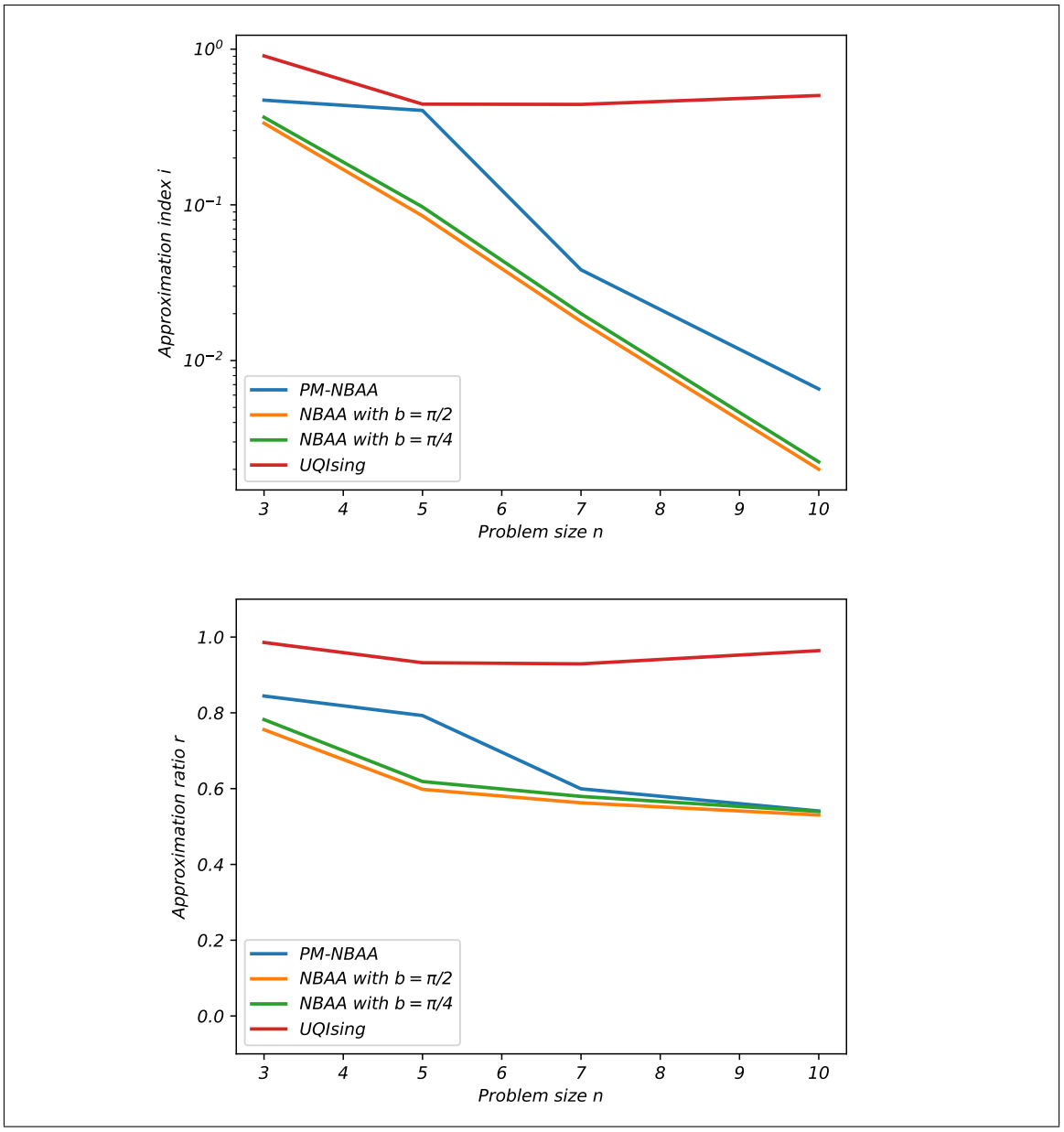


Figure 5.4.1: *Benchmarking the fully universal NBAA and PM-NBAA algorithm against the variational, hybrid quantum-classical UQIsing algorithm on the Ising problem.* ***(Top row)*** *Approximation index, note the logarithmic scale.* ***(Bottom row)*** *Approximation ratio. The approximation ratios and indices are averaged over* 20 *random graph instances. We see on the approximation index that PM-NBAA amplifies the amplitude of the solution more than NBAA, approximately about a factor of* 10*, but only substantially increases the approximation ratio for smaller problem instances. This is due to the fact that PM-NBAA does no ensure that non-solutions are suppressed, only that the probability of measuring the solution increases at each iteration. Hence, in PM-NBAA, "very" non-solutions can occur more often and contribute more to the lower approximation ratio. Overall, both NBAA-based purely quantum methods are still outperformed in practice by the hybrid quantum-classical UQIsing method as proposed in Chapter 4.*

CHAPTER 6

An Iterative Quantum Approach to Point Set Registration

In this chapter, we turn towards a specific application from the field of computer vision: Determining the best rigid transformation that aligns two coordinate systems based on corresponding landmarks, also known as *transformation estimation (TE)*. We propose an iterative method for estimating rigid transformations from point sets. The chapter builds on the basic notions of adiabatic quantum computing from Section 2.5 and quantum annealing from Section 3.3.5.

We begin the chapter by modelling the problem. After that, our iterative method is presented. The algorithm allows to solve the problem to high precision, and does not suffer from inconsistent rotation matrices. Experiments are performed at the end of the chapter, demonstrating the potential of the method in aligning several datasets in two and three dimensions, and robustly even in settings with high outlier ratios.

The chapter builds on the following author's publications:

- Kuete Meli, N., Mannel, F., and Lellmann, J. "An iterative quantum approach for transformation estimation from point sets", *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 529-537 (Juli 2022). https://doi.org/10.1109/CVPR52688.2022.00061.

6.1 The Transformation Estimation Problem

Let $n, d \in \mathbb{N}$. Given a *reference point set* $\tilde{\mathbf{X}} := (\tilde{x}_i)_{i=1}^n \in \mathbb{R}^{d,n}$ and a *template point set* $\tilde{\mathbf{Y}} := (\tilde{y}_i)_{i=1}^n \in \mathbb{R}^{d,n}$, the purpose of transformation estimation (TE) [110, 111, 112]

is to determine a rigid transformation $f^\star : \mathbb{R}^d \to \mathbb{R}^d$ that solves

$$\min_{f \in \mathrm{SE}(d)} \sum_{i=1}^{n} \mathcal{S}(\tilde{x}_i, f(\tilde{y}_i)), \tag{276}$$

where $\mathcal{S}$ is an appropriate similarity measure and $\mathrm{SE}(d)$ is the special Euclidean group on $\mathbb{R}^d$. The latter is formed by compositions of rotations and translations; its degrees of freedom is given by $p = d(d+1)/2$. Any transformation $f \in \mathrm{SE}(d)$ can be represented in the form $f(\tilde{y}_i) = \mathbf{R}\tilde{y}_i + t$, where $\mathbf{R} \in \mathbb{R}^{d \times d}$ is a rotation matrix and $t \in \mathbb{R}^d$ is a translation vector. Choosing for $\mathcal{S}$ the squared Euclidean distance yields minimizing the sum-of-squares objective function

$$\sum_{i=1}^{n} \mathcal{S}(\tilde{x}_i, f(\tilde{y}_i)) := \sum_{i=1}^{n} \|\tilde{x}_i - f(\tilde{y}_i)\|_2^2 = \sum_{i=1}^{n} \|\tilde{x}_i - \mathbf{R}\tilde{y}_i - t\|_2^2. \tag{277}$$

It is well-known [110, 111] that by centering the points around their centroids, i.e., defining $\mathbf{X} := \tilde{\mathbf{X}} - \bar{\tilde{\mathbf{X}}}$ and $\mathbf{Y} := \tilde{\mathbf{Y}} - \bar{\tilde{\mathbf{Y}}}$, with $\bar{\tilde{\mathbf{X}}}$ and $\bar{\tilde{\mathbf{Y}}}$ denoting the centroids of the original point sets, the rigid registration problem reduces to finding the best *rotation matrix*

$$\mathbf{R}^\star := \arg \min_{\mathbf{R} \in \mathrm{SO}(d)} \sum_{i=1}^{n} \|x_i - \mathbf{R}y_i\|_2^2. \tag{278}$$

This reduces the transformation space of the problem to the d-dimensional special orthogonal group $\mathrm{SO}(d)$ with *exactly* $p = d(d-1)/2$ degrees of freedom. The optimal translation $t^\star$ is then deducted from $\mathbf{R}^\star$ by $t^\star = \bar{\mathbf{X}} - \mathbf{R}^\star\bar{\mathbf{Y}}$.

Transformation estimation finds practical application in computer vision fields such as robotics and image processing. For instance, in image registration, an important preparatory step consists in roughly aligning the images based on corresponding landmarks [20]. Certain properties are desirable from a method for solving the TE problem. They include, among others, the ability of the method to accurately model the transformation that aligns the point sets, the ability to handle high-dimensional data as well as data of increasing size, and the robustness against noise and outliers [110].

6.2 Related Work

AQC-based algorithms have already been developed for several computer vision and image processing problems, much of them naturally dealing with binary variables. Examples include graph matching [113], and permutation related problems [114, 115]. In this work, we will focus the one of aligning point sets.

While computer vision abounds with powerful classical algorithms to solve the TE problem, e.g., ICP [116, 117] and CPD [118], there are still very few methods available for quantum computers. Our work is inspired by Golyanik and

Theobalt [83], who pioneered a quantum approach *(QA)* for solving the transformation estimation problem based on approximating rotation matrices by linear combinations of basis matrices, where the linear coefficients are binary variables computed by AQC.

In their approach, the authors approximate $\mathbf{R}$ using binary coefficients $q_k \in \{0, 1\}$ and basis matrices $\mathbf{Q}_k$ as

$$\mathbf{R} \approx \tilde{\mathbf{R}} = \sum_{k=1}^{K} q_k \mathbf{Q}_k, \tag{279}$$

where

- in $d = 2$ dimensions

$$\begin{aligned} \mathbf{Q}_k \in Q := \{ w\mathbf{C} \in \mathbb{R}^{2,2} \quad | \quad & w \in \{0.5, 0.2, 0.1, 0.1, 0.05\}, \\ & \mathbf{C} \in \{\mathbf{I}, -\mathbf{I}, \mathbf{M}, -\mathbf{M}\}\} \end{aligned} \tag{280}$$

  with

$$\mathbf{I} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{M} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{281}$$

- and in $d = 3$ dimensions

$$\begin{aligned} \mathbf{Q}_k \in Q := \{ w\mathbf{C} \in \mathbb{R}^{3,3} \quad | \quad & w \in \{0.5, 0.2, 0.1, 0.1, 0.05\}, \\ & \mathbf{C} \in \{\mathbf{I}, -\mathbf{I}, \mathbf{M}_a, -\mathbf{M}_a, \mathbf{M}_b, -\mathbf{M}_b, \mathbf{M}_c, -\mathbf{M}_c, \\ & \mathbf{M}_d, -\mathbf{M}_d, \mathbf{M}_e, -\mathbf{M}_e, \mathbf{M}_f, -\mathbf{M}_f\}\} \end{aligned} \tag{282}$$

  with

$$\begin{aligned} \mathbf{I} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_a := \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_d := \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\ \mathbf{M}_b := \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_e := \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \\ \mathbf{M}_c := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{M}_f := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \end{aligned} \tag{283}$$

The basis matrices $\mathbf{I}$, $\mathbf{M}$ and $\mathbf{M}_a, \ldots, \mathbf{M}_f$ are obtained from Rodrigues' formulae [119] for construction rotation matrices in SO(d) from elements of the Lie algebra $\mathfrak{so}$(d) by exponential mapping.

Recalculating the objective functional with the above approximation and omitting constant terms yields the QUBO problem

$$\min_{\mathbf{R} \in SO(d)} \sum_{i=1}^{n} \|x_i - \mathbf{R}y_i\|^2 \approx \min_{q \in \{0,1\}^K} \sum_{i=1}^{n} \|x_i - \tilde{\mathbf{R}}y_i\|^2 = \min_{q \in \{0,1\}^K} q^\top \mathcal{D} q, \tag{284}$$

with couplers $\mathcal{D}_{kl} = \sum_{i=1}^{n} \langle \mathbf{Q}_k y_i, \mathbf{Q}_l y_i \rangle$ and biases $\mathcal{D}_{kk} = -2 \sum_{i=1}^{n} \langle x_i, \mathbf{Q}_k y_i \rangle$.

A quick first performance demonstration of QA on D-Wave machines is shown in Figure 6.2.1 on a selected 2D point set registration problem from the MNIST point cloud data set [21]. One can visualize how dense the matrix $\mathcal{D}$ can be for practical problems and how accurate the quantum annealer can compute the deformation matrices.

Discussion

QA [83] has certain limitations:

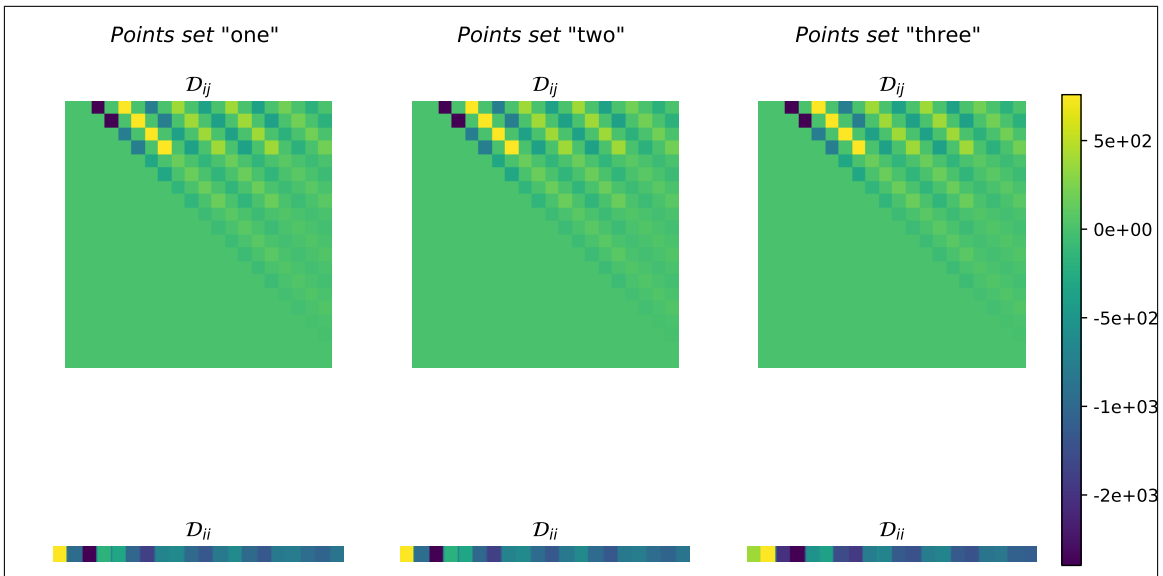
- In QA, the authors treat $\cos\theta$ and $\sin\theta$ as independent variables, say ω_1, ω_2 in 2D (3D similar), and then optimize with respect to both. This has the disadvantage that the resulting matrix $\mathbf{R} = \omega_1 \mathbf{I} + \omega_2 \mathbf{M}$ may not be a rotation matrix, as this requires $\omega_1^2 + \omega_2^2 = 1$. Solving for the closest orthogonal approximation of $\mathbf{R}$ induces a considerable additional computational cost.
- The non-orthogonality issue of QA generates another issue, namely that QA hardly handle noise, as it allows the computed matrices to scale and to capture noise instead of aligning non-noisy points.
- The number K of logical qubits used in QA is fixed and results from the cardinality of the basis matrix set Q; QA needs $K = 20$ qubits in 2D and $K = 70$ qubits in 3D, a relatively high number compared to the original degrees of freedom of the problem. Reducing this number is only possible trough a carefully construction of the set Q with a reduced number of basis matrices, a task that QA does not handle and leaves to the user.

In this study, our goal is to address the limitations of QA and propose a method that scales more efficiently, produces orthogonal rotation matrices, and robustly handles noise.

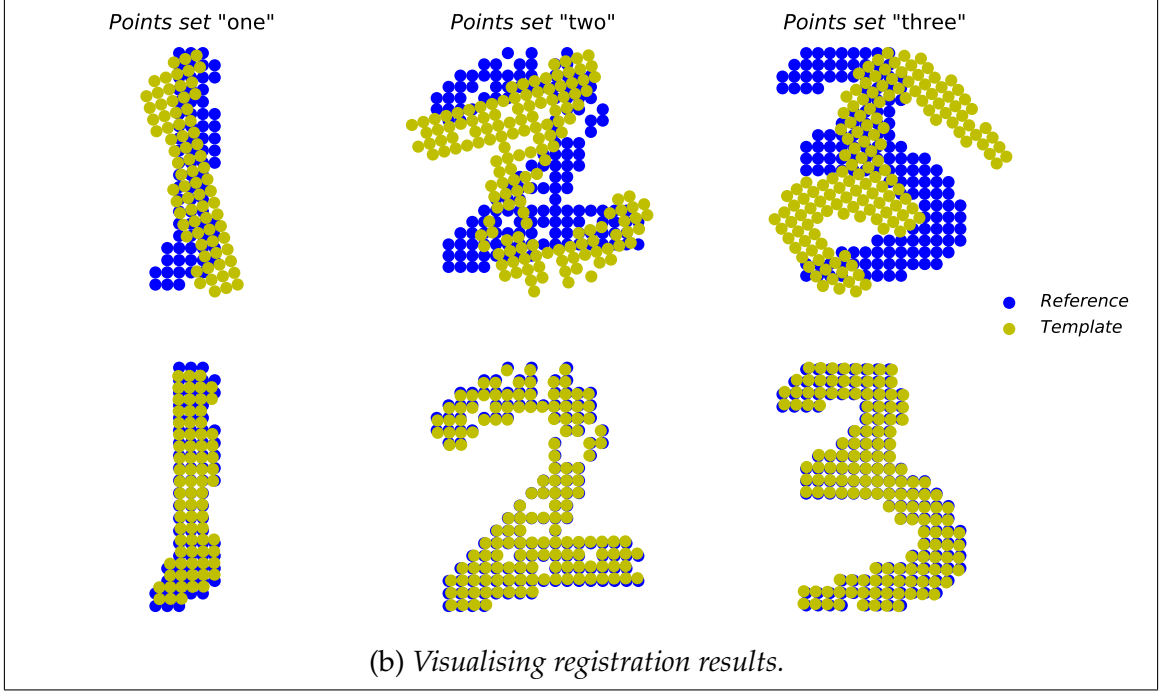
6.3 Proposed Quantum Registration Approach

Like many computer vision problems, the TE problem in Equation (278) is intrinsically not a binary optimization problem and as such not directly solvable on current quantum annealers. We show that linearizing the rotation matrix and introducing binary coefficients allows to approximate the least-squares registration problem by a QUBO problem. Since the 2D and 3D cases are very similar, we present our method from a general point of view and only switch to a case-by-case discussion when necessary.

Unlike QA, our approach contains the number of qubits K as a free parameter that can be chosen by the user. This is a critical step towards an algorithmic scheme that can be carried out on current quantum hardware. Our algorithm successively increases the approximation precision of the rotation parameters, thereby enabling its computation to arbitrary precision.



(a) *Visualising couplers and biases.*



(b) *Visualising registration results.*

Figure 6.2.1: *Inspection of QA [83] on a selected transformation estimation problem.* ***(a)*** *Cost matrix $\mathcal{D}$ containing couplers and biases of the QUBO. Computing $\mathcal{D}$ is the most costly part of the algorithm, as this matrix is dense. Using $\mathcal{D}$, the D-Wave quantum annealer accurately computes a bit-string encoding of the solution of the problem.* ***(b)*** *Registration results of QA. Blue and olive are reference and template sets. Top and bottom is before and after the registration. QA performs well, but sometimes returns non orthogonal matrices as visible in the "one" points set where the registered template points are scaled compared to the reference ones.*

Another issue is that optimizing over the rotation group $\mathrm{SO}(d)$ calls for an orthogonality constraint on the rotation matrix $\mathbf{R}$. One way to avoid this constraint, as already introduced in [120, 121], is to leverage the one-to-one correspondence between the Lie group $\mathrm{SO}(d)$ and its algebra $\mathfrak{so}(d)$. This allows to optimize Equation (278) over the *linear* space $\mathfrak{so}(d)$, which is the set of skew-symmetric matrices $\mathbf{M}$ of dimension d. Specifically, we will use the exponential map $\mathbf{R} = \exp(\mathbf{M})$ to associate to $\mathbf{M} \in \mathfrak{so}(d)$ the rotation matrix $\mathbf{R} \in \mathrm{SO}(d)$ [119].

6.3.1 Approximation of Rotation Matrices

In order to represent rotation matrices with increasing accuracy in our quantum approach, we use *K-bit binary representations* and quantize the interval $[l, u]$ according to

$$x_q := l + \frac{u-l}{s} \sum_{k=0}^{K-1} q_k 2^k, \tag{285}$$

with binary variables $q_k \in \{0, 1\}$ and a scaling factor $s = 2^K - 1$. We also refer the reader to [122, 123] for alternative approaches for fixed- and floating-point representations on quantum annealers.

Rotation Matrices in two Dimensions.

In 2D, all possible skew-symmetric matrices in $\mathfrak{so}(2)$ are of the form

$$\mathbf{S}(\theta) = \theta \mathbf{M} := \theta \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{286}$$

where $\theta \in \mathbb{R}$ represents the rotation angle. The associated rotation matrix is

$$\mathbf{R}(\theta) := e^{\mathbf{S}(\theta)} = (\cos\theta)\mathbf{I} + (\sin\theta)\mathbf{M}, \tag{287}$$

and any 2D rotation matrix can be written in this way [119]. To obtain a QUBO problem that avoids the non-orthogonality issue of QA, we regard $\mathbf{R}(\theta)$ as a nonlinear function of θ and determine the optimal θ in a multi-step process. Given a current angle θ_c, we linearize $\mathbf{R}(\theta)$ around θ_c using first-order Taylor expansion:

$$\mathbf{R}(\theta) \approx \mathbf{R}(\theta_c) + (\theta - \theta_c)\mathbf{R}'(\theta_c) = \mathbf{R}(\theta_c) - \theta_c \mathbf{R}'(\theta_c) + \theta \mathbf{R}'(\theta_c), \tag{288}$$

where $\mathbf{R}'(\theta_c) = (-\sin\theta_c)\mathbf{I} + (\cos\theta_c)\mathbf{M}$ is the first derivative of $\mathbf{R}$ with respect to θ at θ_c.

To obtain a QUBO problem, we quantize θ according to the K-bit binary representation introduced in Equation (285): For a fixed search window size $\delta > 0$, we express $\theta \in [\theta_c - \delta, \theta_c + \delta]$ by

$$\theta \approx \theta_c - \delta + \hat{v}, \quad \text{with} \quad \hat{v} := \frac{2\delta}{s} \sum_{k=0}^{K-1} q_k 2^k. \tag{289}$$

Expressing Equation (288) in terms of the new variable $\hat{v}$, we obtain the approximation

$$\mathbf{R}(\hat{v}) \approx \underbrace{\mathbf{R}(\theta_c) - \delta\, \mathbf{R}'(\theta_c)}_{=:\mathbf{R}_c} + \hat{v}\, \underbrace{\mathbf{R}'(\theta_c)}_{=:\mathbf{R}'}, \tag{290}$$

where the term $\mathbf{R}_c$ depends on θ_c and δ, but is independent of the unknown $\hat{v}$ and hence of the optimization variables $q_k \in \{0, 1\}$, $k = 0, \ldots, K-1$.

For any of the mass-centered points y_i, $i = 1, \ldots, n$, as in Equation (278), we thus find

$$\mathbf{R}(\hat{v}) y_i \approx \mathbf{R}_c y_i + (\mathbf{R}' y_i)\hat{v}, \tag{291}$$

The approximation Equation (291) is at the heart of the QUBO problem that we solve. The number K of qubits used for quantizing $\hat{v}$ can be chosen by the user, for instance based on the available hardware. In particular, as quantum computers become more powerful, the precision can be scaled to larger values for K than possible on current-day hardware.

Rotation Matrices in three Dimensions.

In the 3D case, the complete rotation parameters can be parameterized using a vector $v = (v_1, v_2, v_3)^\top \in \mathbb{R}^3$ encoding the rotation angle $\theta(v) := \|v\|_2$ and the rotation axis $x(v) := v/\|v\|_2 = (x_1(v), x_2(v), x_3(v))^\top$. Any skew-symmetric matrix in $\mathfrak{so}(3)$ can be parameterized using v as

$$\mathbf{S}(v) = \mathbf{M}(v) := \theta \cdot \begin{pmatrix} 0 & -x_3(v) & x_2(v) \\ x_3(v) & 0 & -x_1(v) \\ -x_2(v) & x_1(v) & 0 \end{pmatrix}, \tag{292}$$

and the exponential map, see [119], turns each such matrix $\mathbf{S}(v)$ into a rotation matrix

$$\mathbf{R}(v) = e^{\mathbf{S}(v)} = \mathbf{I} + g(v)\mathbf{M}(v) + h(v)\mathbf{M}^2(v), \tag{293}$$

with functions

$$g(v) := \frac{\sin \|v\|_2}{\|v\|_2} \quad \text{and} \quad h(v) := \frac{1 - \cos \|v\|_2}{\|v\|_2^2} \tag{294}$$

for $v \neq 0$ and $g(0) := 1$, $h(0) := 1/2$, $g'(0) = h'(0) = 0$. Analogously to the two-dimensional case, given a current vector v_c, we linearize $\mathbf{R}$ around v_c using first-order Taylor expansion as

$$\mathbf{R}(v) \approx \mathbf{R}(v_c) + \nabla\mathbf{R}(v_c)(v - v_c) = \mathbf{R}(v_c) - \nabla\mathbf{R}(v_c)(v_c) + \nabla\mathbf{R}(v_c)(v), \tag{295}$$

where

$$\begin{aligned}\nabla\mathbf{R}(v_c) \approx\ & \nabla g(v_c)\mathbf{M}(v_c) + \\ & g(v_c)\nabla\mathbf{M}(v_c) + \\ & \nabla h(v_c)\mathbf{M}^2(v_c) + \\ & h(v_c)[\nabla\mathbf{M}(v_c)\mathbf{M}(v_c) + \mathbf{M}(v_c)\nabla\mathbf{M}(v_c)]\end{aligned} \tag{296}$$

with

$$\nabla g(v_c) = v_c \cdot \left(\frac{\|v_c\|_2 \cos\|2\|_2 - \sin\|v_c\|_2}{\|v_c\|_2^3}\right) \quad \text{and} \tag{297}$$

$$\nabla h(v_c) = v_c \cdot \left(\frac{\|v_c\|_2 \sin\|2\|_2 - 2(1-\cos\|v_c\|_2)}{\|v_c\|_2^4}\right). \tag{298}$$

$$\tag{299}$$

Each of the terms in the sum in Equation (296) is a linear operator from $\mathbb{R}^3$ to $\mathbb{R}^{3\times 3}$.

Assuming that the components v_j for $j = 1, 2, 3$ are each in the search window $[(v_c)_j - \delta, (v_c)_j + \delta]$ for some $\delta > 0$, we use the discretization $v_j = (v_c)_j - \delta + \hat{v}_j$ from Equation (285) with $\hat{v}_j := \frac{2\Delta}{s}\sum_{k=0}^{K-1} q_k^{(j)} 2^k$. Introducing $q_k := (q_k^{(1)}, q_k^{(2)}, q_k^{(3)})^\top$, we can rewrite v as $v \approx v_c - \delta + \hat{v}$. Thus, we arrive at the approximation

$$\mathbf{R}(\hat{v}) \approx \underbrace{\mathbf{R}(v_c) - \delta\ \nabla\mathbf{R}(v_c)}_{=:\mathbf{R}_c} + \nabla\mathbf{R}(v_c)(\hat{v}) \tag{300}$$

$$\begin{aligned}&= \mathbf{R}_c + \\ &\quad \mathbf{M}(v_c)\nabla g(v_c)^\top \hat{v} + \\ &\quad g(v_c)\nabla\mathbf{M}(v_c)(\hat{v}) + \\ &\quad \mathbf{M}^2(v_c)\nabla h(v_c)^\top \hat{v} + \\ &\quad h(v_c)[\nabla\mathbf{M}(v_c)\mathbf{M}(v_c) + \mathbf{M}(v_c)\nabla\mathbf{M}(v_c)](\hat{v}),\end{aligned} \tag{301}$$

where the term R_c depends on v_c and Δ, but not on the optimization variables $q_k \in \{0,1\}^3$, $k = 0, \ldots, K-1$. Interestingly, as $\mathbf{M}(v)$ is linear in v, we have $\nabla\mathbf{M}(v_c)(w) = \mathbf{M}(w)$ for any $v_c, w \in \mathbb{R}^3$, so that we can rewrite the linearization as

$$\begin{aligned}\mathbf{R}(\hat{v}) \approx\ & \mathbf{R}_c + \\ & \mathbf{M}(v_c)\nabla g(v_c)^\top \hat{v} + \\ & g(v_c)\mathbf{M}(\hat{v}) + \\ & \mathbf{M}^2(v_c)\nabla h(v_c)^\top \hat{v} + \\ & h(v_c)[\mathbf{M}(\hat{v})\mathbf{M}(v_c) + \mathbf{M}(v_c)\mathbf{M}(\hat{v})].\end{aligned} \tag{302}$$

The matrix multiplication $\mathbf{M}(\hat{v})w$ represents the cross product of $\hat{v}$ and w, so that

it holds $\mathbf{M}(\hat{v})w = -\mathbf{M}(w)\hat{v}$, yielding, in the slight abuse of notation $\mathbf{R}'y_i$,

$$\begin{aligned}\mathbf{R}(\hat{v})y_i &\approx \mathbf{R}_c y_i + (\mathbf{R}'y_i)\hat{v} && (303)\\ &= \mathbf{R}_c y_i + \\ &\quad \underbrace{\Big[\mathbf{M}(v_c)y_i\nabla g(v_c)^\top - \\ &\quad g(v_c)\mathbf{M}(y_i) + \\ &\quad \mathbf{M}^2(v_c)y_i\nabla h(v_c)^\top - \\ &\quad h(v_c)[\mathbf{M}(\mathbf{M}(v_c)y_i) + \mathbf{M}(v_c)\mathbf{M}(y_i)]\Big]}_{=:\mathbf{R}'y_i}\hat{v}. && (304)\end{aligned}$$

As before, the number K of qubits in this approximation can be chosen by the user, permitting the use of larger values if computationally feasible.

Final Step of the QUBO Preparation.

As a final step, we introduce

$$q = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{K-1}\end{pmatrix} \in \{0,1\}^{pK} \qquad \text{and} \tag{305}$$

$$\mathbf{U} = \frac{2\delta}{s}\begin{pmatrix}\mathbf{D}_0 & \mathbf{D}_1 & \dots & \mathbf{D}_{K-1}\end{pmatrix} \in \mathbb{R}^{p\times pK}, \tag{306}$$

where for $k = 0, \dots, K-1$ the diagonal matrix $\mathbf{D}_k \in \mathbb{R}^{p\times p}$ has the entries 2^k. This notation allow us to rewrite $\hat{v} = \mathbf{U}q$, so $\mathbf{R}'y_i\hat{v} = \mathbf{R}'y_i\mathbf{U}q$, both in 2D and in 3D.

6.3.2 QUBO Formulation of the Problem

In the previous subsection, we derived an approximation of the products $\mathbf{R}y_i$ based on a bit string $q \in \{0,1\}^{pK}$, with $p = d(d-1)/2$ being the degrees of freedom of $\mathfrak{so}(d)$. Now, we provide a QUBO formulation of the transformation estimation problem.

By design, we have $\mathbf{R}y_i \approx \mathbf{R}_c y_i + (\mathbf{R}'y_i)\hat{v}$ by Equation (291) in the 2D case and a corresponding (slightly more complicated) expression in the 3D case by Equation (303). Crucially, $\hat{v}$ and thus q appear linearly in these approximations, therefore $\|x_i - \mathbf{R}y_i\|_2^2$ is quadratic in q. By ignoring the q-independent terms, we find that Equation (278) can be approximated in terms of $\hat{v}$ as

$$\min_{q\in\{0,1\}^{pK}} \sum_{i=1}^{n} \left(\|\mathbf{R}'y_i\hat{v}\|_2^2 + 2\left\langle \mathbf{R}_c y_i - x_i, \mathbf{R}'y_i\hat{v}\right\rangle\right). \tag{307}$$

As derived in Equation (305), we have $\hat{v} = \mathbf{U}q$. Therefore, we propose the following QUBO to approximate Equation (278):

$$\min_{q \in \{0,1\}^{pK}} q^\top \mathcal{D} q, \tag{308}$$

where

$$\begin{aligned} \mathcal{D}_{kl} &= \mathbf{U}_{k\cdot}^\top \sum_{i=1}^{n} \langle \mathbf{R}' y_i, \mathbf{R}' y_i \rangle \, \mathbf{U}_{\cdot l} \\ \text{and} \quad \mathcal{D}_{kk} &= \mathbf{U}_{k\cdot}^\top \sum_{i=1}^{n} \langle \mathbf{R}' y_i, \mathbf{R}' y_i \rangle \, \mathbf{U}_{\cdot k} + 2\mathbf{U}_{k\cdot}^\top \sum_{i=1}^{n} \langle \mathbf{R}' y_i, \mathbf{R}_c y_i - x_i \rangle \, . \end{aligned} \tag{309}$$

The terms $U_{k\cdot}^\top \in \mathbb{R}^{1,p}$ and $\mathbf{U}_{\cdot l} \in \mathbb{R}^{p,1}$ designate the transpose of the k-th column and l-th column of $\mathbf{U}$.

To solve Equation (308), we only need to provide the QUBO matrix $\mathcal{D} \in \mathbb{R}^{pK \times pK}$ containing the couplers and biases to the quantum annealer. Note that the dimensions of $\mathcal{D}$ do not depend on the number of points n. At the end of the annealing procedure, the measured bit-string q is used to compute the rotation parameter according to

$$\theta = \theta_c - \delta + \mathbf{U}q, \quad \text{and} \quad v = v_c - \delta + \mathbf{U}q, \tag{310}$$

which completes one rotation estimate based on linearization around θ_c.

6.3.3 Iterative Quantum Transformation Estimation Algorithm

Our full iterative quantum transformation estimation strategy (IQT), described for the 3D case (2D similar) in Algorithm 6.3.1, consists in solving Equation (308) for different Taylor approximation points θ_c (2D case) or v_c (3D case). We start by making an initial guess for the rotation parameter such as $\theta_c = 0 \in \mathbb{R}$ in 2D, respectively, $v_c = 0 \in \mathbb{R}^3$ in 3D. For the search interval, we use an initial radius of $\delta = \pi$. After assembling the matrix $\mathcal{D}$, Equation (308) is solved. The solution q provides the next iterate θ or v via Equation (310).

An important part of our algorithm is that we treat δ as an adaptive parameter; we aim to decrease δ whenever possible, which results in an increasingly better approximation of the solution and the associated rotation matrix. We decrease δ whenever the difference between the current and the previous rotation parameter becomes smaller than some threshold function of the binning size $2\delta/s$, which we view as an indicator that the best solution within the current K-bits interval discretization of $[v_c - \delta, v_c + \delta]$ has been reached.

Algorithm 6.3.1 (*Iterative quantum transformation estimation (IQT)*)

Input: $\mathbf{X}$, $\mathbf{Y}$*: Point sets to register, maxit: maximal number of iterations,* v_c*: initial 3D rotation parameter (2D similar),* δ*: radius of the search interval,* κ*: threshold for reduction of* δ*, and* K*: number of qubits for the discretization.*
Output: *Rotation matrix* $\mathbf{R}^\star$*, ideally such that* $\|\mathbf{X} - \mathbf{R}^\star\mathbf{Y}\|_F^2 = 0$

$\theta_c^0 \leftarrow \theta_c$. ▷ *Initialization*
Let $\tau := \frac{\kappa\delta}{2^K - 1}$. ▷ *Initialization*
for $j = 0, 1, 2, \ldots,$ *maxit* ***do***
  Compute $\mathbf{R}_c \leftarrow \mathbf{R}_c(v_c^j)$ *via Equation* (300).
  Construct $\mathbf{R}_c y_i$, $\mathbf{R}' y_i$ *for* $i = 1, \ldots, n$ *via Equation* (303).
  Construct $\mathbf{U}$ *via Equation* (305).
  Compute $\mathcal{D}$ *with all* $\mathbf{R}_c y_i$ *and* $\mathbf{R}' y_i$ *according to Equation* (309).
  Obtain $q \in \{0,1\}^{pK}$ *by solving Equation* (308) *using Quantum Annealing.*
  Compute v^j *according to Equation* (310).
  if $\|v^j - v_c^j\|_2 < \tau$ ***then***
    $\delta \leftarrow \delta/4, \quad \tau \leftarrow \tau/4$. ▷ *Shrink search interval*
  end if
  Update $v_c^{j+1} \leftarrow v^j$. ▷ *Update parameters*
end for
Compute $\mathbf{R}(v^j)$ *via Equation* (300).
return *Current rotation parameter* v^j*, and associated rotation matrix* $\mathbf{R}^\star = \mathbf{R}(v^j)$.

6.3.4 Classical Transformation Estimation

In Section 6.4 we compare the IQT algorithm with its classical relaxed version named *iterative classical transformation estimation (ICT)* that results from replacing the QUBO problem by a classical unconstrained quadratic programming (QP) problem. Specifically, with $R\tilde{y}_i = R_c\tilde{y}_i + R_i v$ from Equation (291), resp., Equation (303), and continuous variables $v \in \mathbb{R}^P$, we construct the QP problem

$$\min_{v\in\mathbb{R}^P} v^\top \underbrace{\left(\sum_{i=1}^n \langle \mathbf{R}' y_i, \mathbf{R}' y_i\rangle\right)}_{=:\mathbf{W}} v + 2\underbrace{\left(\sum_{i=1}^n \langle \mathbf{R}' y_i, \mathbf{R}_c y_i - x_i\rangle\right)^\top}_{=:c^\top} v, \tag{311}$$

whose solution is equivalent to that of the system of linear equations $2\mathbf{W}v = -c$.

6.4 Numerical Experiments

In this section, we experimentally evaluate accuracy, robustness and run times of a hybrid quantum-classical implementation of the proposed IQT algorithm. The QUBO matrix $\mathcal{D}$ is prepared in Python 3.9.2 on an Intel i7-7700K CPU machine

	dataset name	Number of points
2D	Synthetic points	150 (variable)
	MNIST [21]	150
	Lena edges [22]	4845
3D	Synthetic points	150 (variable)
	Stanford bunny [23]	500
	completion3D [24]	2048

Table 4: *Datasets used in the experiments.*

with 16 GB RAM and the QUBO problem is solved on a D-Wave annealer with the default annealing time of $20\mu s$ and 100 reads per optimization iteration. After the reads, the obtained state with the lowest energy is selected as approximate solution of the QUBO.

The datasets used in the experiments are listed in Table 4. They are synthetic point sets, the MNIST [21] and Lena edges [22] point sets in 2D; and the Stanford bunny [23] and the completion3D [24] point sets in 3D. The number of points varies from 150 for the synthetic point sets to 4845 in Lena edges. Synthetic data was generated by sampling parameterized 2D (ellipses) and 3D (cylinder) base shapes at different spacing and applying a rotation with random rotation parameters: In 2D we select θ uniformly in the range $[0, 360]$, and in 3D we select θ uniformly in the range $[0, 360]$, that we multiply with a normally distributed rotation axis x to form the rotation parameter v. This allows for a fair investigation of scaling effects without changing the difficulty of the underlying problem.

Choice of K.

Design choices of our method include the number of qubits K used for the numerical representation of the unknown variables and the D-Wave graph topology. We evaluate the precision of the reconstructed rotation parameter and matrix for $K \in \{3, 5, 10\}$ on synthetic 2D and 3D point sets for 15 iterations. Recall that in total pK qubits are necessary to encode the QUBO problem, where $p = 1$ in 2D and $p = 3$ in 3D. We run the experiment on the D-Wave 2000Q QPU which uses the Chimera topology [124] and on the D-Wave Advantage 1.1 QPU which uses the Pegasus topology [125]. We compare the computed rotation parameters and matrices with the ground truth.

The results are displayed in Table 5. We observe that a greater number of qubits does not necessarily increase the precision. As an example, the error for $K = 10$ qubits in the 3D case is much larger than the errors for $K = 3$ and $K = 5$. This discrepancy is related to the average chain break fraction, implying that more qubits translate into a noisier system, thereby increasing the amount of miscalculations. In the subsequent experiments, we use $K = 10$ in 2D and $K = 5$ in 3D, and execute the algorithm on the D-Wave 2000Q QPU, which seems to be the best configuration for the task at hand.

	QPU Topology	2000Q			Advantage 1.1		
	# Qubits K	3	5	10	3	5	10
2D	$\lvert\theta - \theta^\star\rvert$	$2.42 \cdot 10^{-5}$	$1.06 \cdot 10^{-10}$	$\mathbf{1.66 \cdot 10^{-14}}$	$2.28 \cdot 10^{-4}$	$1.13 \cdot 10^{-8}$	$\mathbf{2.58 \cdot 10^{-12}}$
	$\lVert\mathbf{R} - \mathbf{R}^\star\rVert_F$	$3.42 \cdot 10^{-5}$	$2.26 \cdot 10^{-10}$	$\mathbf{2.24 \cdot 10^{-14}}$	$4.0 \cdot 10^{-4}$	$1.06 \cdot 10^{-8}$	$\mathbf{3.65 \cdot 10^{-12}}$
	Chain break	**0**	**0**	**0**	**0**	**0**	**0**
3D	$\lVert v - v^\star\rVert_2$	$4.2 \cdot 10^{-4}$	$\mathbf{4.00 \cdot 10^{-6}}$	$8.76 \cdot 10^{-1}$	$9.7 \cdot 10^{-5}$	$\mathbf{6.71 \cdot 10^{-7}}$	$0.8 \cdot 10^{-2}$
	$\lVert\mathbf{R} - \mathbf{R}^\star\rVert_F$	$3.8 \cdot 10^{-4}$	$\mathbf{1.63 \cdot 10^{-6}}$	$6.39 \cdot 10^{-1}$	$1.73 \cdot 10^{-4}$	$\mathbf{1.45 \cdot 10^{-6}}$	$1.96 \cdot 10^{-1}$
	Chain break	**0**	**0**	0.01	**0**	**0**	0.002

Table 5: *Distance of the rotation parameters θ and matrices $\mathbf{R}$ reconstructed by IQT to the ground truth $(\theta^\star, \mathbf{R}^\star)$ for different numbers of qubits K and on two D-Wave graph topologies (2000Q and Advantage 1.1). The results are reported after 15 optimization steps of the same problem in both the 2D and the 3D case. In addition, the chain break fractions averaged over the 15 iterations are provided. Smallest errors are observed using $K = 10$ qubits in the 2D case and $K = 5$ in 3D. For 3D problems, larger number of qubits lead to chain breakage, which degrades the optimization process.*

6.4.1 Benchmark Metrics

In order to verify the robustness of our method against outliers, we randomly add noise to a fraction of the template points; said fraction is called the *outlier ratio*. To evaluate the accuracy of the proposed method, we adopt two metrics from [83]:

$$e_R := \|\mathbf{I} - \mathbf{R}^\top \mathbf{R}\|_F \quad \textit{(consistency error)} \tag{312}$$

$$\text{and} \quad e_A := \frac{\|\mathbf{X} - \mathbf{R}\mathbf{Y}\|_F}{\|\mathbf{X}\|_F} \quad \textit{(alignment error)}, \tag{313}$$

where $\|\cdot\|_F$ is the Frobenius norm. While the consistency error measures the orthonormality of the reconstructed rotation matrix, the alignment error expresses how accurately the template points are mapped to the reference points. We remark that if $\mathbf{R}^\star$ is the optimal solution of (278) obtained for $v^\star$ and if $v_c^j \to v^\star$ for v_c generated by Algorithm 6.3.1, then the approximation of the rotation matrix given by the right-hand side of the Taylor approximations in Equation (290), respectively, Equation (300) converges to $\mathbf{R}^\star$ (more precisely, with an error of $\mathcal{O}(\|\theta_c^j - \theta^\star\|_2^2)$, resp., $\mathcal{O}(\|v_c^j - v^\star\|_2^2)$). Thus, the computed rotation matrices converge towards truly orthogonal matrices.

6.4.2 Benchmark Results

In Figure 6.4.1 we visualize the registration errors of IQT and its classical counterpart ICA as a function of the outlier ratio. We plot the errors for the 5-th, 10-th, and 15-th iterations of IQT on 2D and 3D synthetic point sets. The errors in 2D are compared with those of the QA approach from [83] with the complete datasets as interaction points, i.e., $K = n$ in the sense of QA.

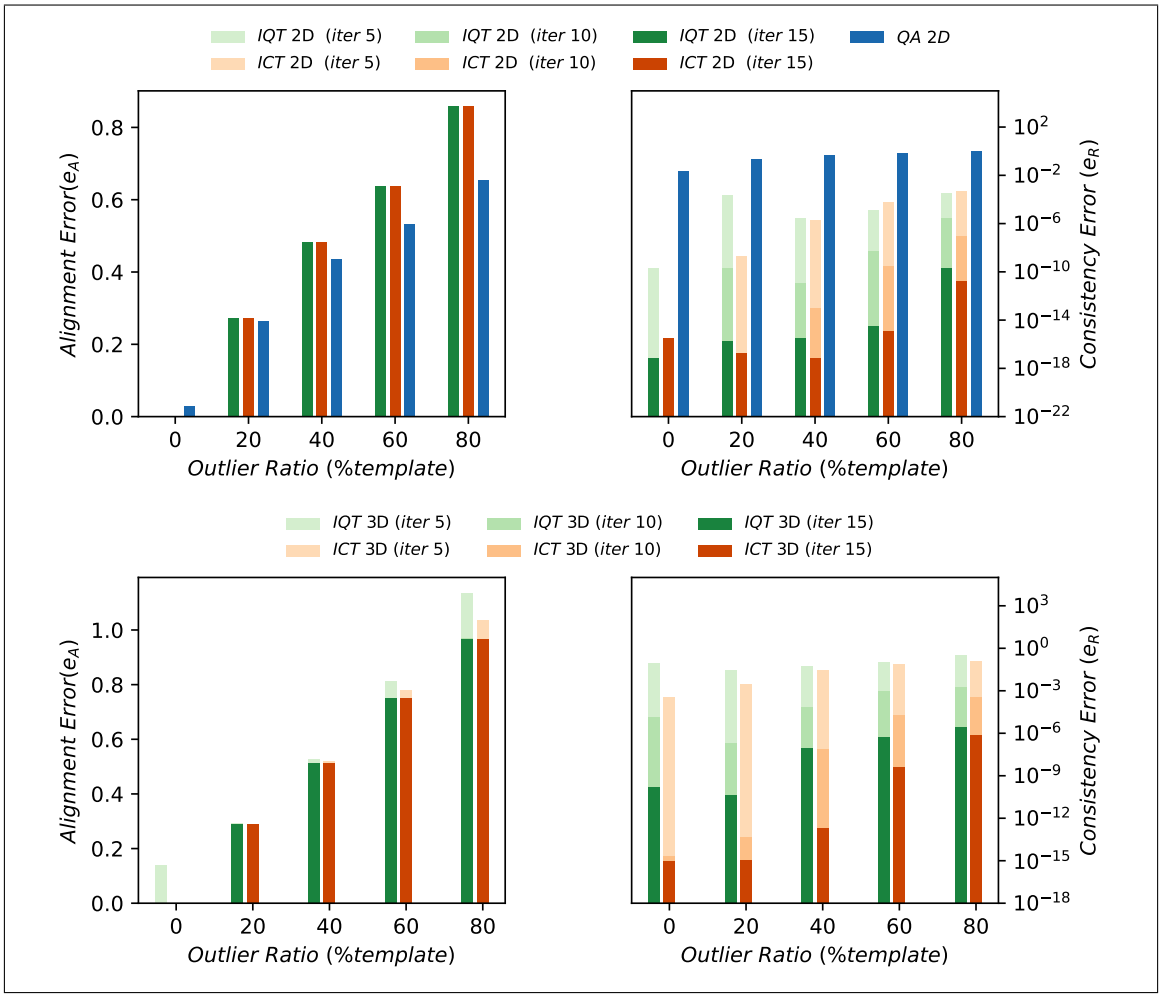


Figure 6.4.1: *Benchmarking our IQT method against it classical ICA counterpart and against the QA method from [83]. (**Top row**) Results on the synthetic 2D data set. (**Bottom row**) Results on the synthetic 3D data set. Shown on the left is the alignment error e_A and on the right the consistency error e_R (note logarithmic scale) as function of the outlier ratio for the 5-th, 10-th, 15-th iteration. The proposed IQT method consistently achieves the same accuracy as the classical ICA method while avoiding consistency errors introduced by the QA method. The alignment error of QA in the 2D case seems better than that if IQT. This is namely because, as shown by the consistency error, QA produces non-orthogonal matrices and enlarge the feasible set of the initial optimization problem.*

As expected, both IQT and ICA reduce the alignment error and converge to orthonormal matrices, with the consistency error being virtually independent of the outlier ratio. In contrast, QA [83] is more susceptible to produce non-orthogonal matrices as the outlier ratio increases. We highlight the convergence and the robustness similarities between IQT and ICT, justifying the usefulness of the K-bits discretization and the ability of the quantum computer to approximate real numbers by such a discretization.

	# Qubits K	IQT Matrix preparation QPU access time (ms)			ICT Matrix preparation CPU access time (ms)		
N		150	1500	20000	150	1500	20000
2D	$K = 10$	2.5	23.2	311.25			
		34.67	34.67	34.69			
	$K = 20$	2.51	23	305.19	6.99	55.56	334.7
		35.32	34.79	35.26	0.1	0.08	0.06
	$K = 40$	2.65	23.14	306.12			
		35.42	35.39	35.38			
3D	$K = 5$	3.43	32.07	425.56			
		35.04	34.92	34.96			
	$K = 10$	3.5	32.16	420.18	6.58	34.84	421.77
		35.23	35.23	35.24	0.01	0.01	0.01
	$K = 15$	3.43	31.68	419.02			
		35.39	35.44	35.4			

Table 6: *Runtime comparison for solving the quadratic problem* (125) *on synthetic data.* (***Left***) *Reports of our IQT quantum formulation.* (***Right***) *Reports for the classical ICA counterpart . All values are averaged over* 10 *instances for each problem size* N*. Experimentally, IQT runtime is independent of the number of qubits* K*. Matrix preparation dominates the QPU/CPU access time for solving the system.*

Figure 6.4.2 presents registration results on the completion3D dataset [24] for point pairs without noise and with 50% outlier ratio. We observe that IQT finds a good transformation even in the presence of strong noise, with accuracy comparable to classical computation (ICT) while generating consistent (i.e., almost orthogonal) rotation matrices.

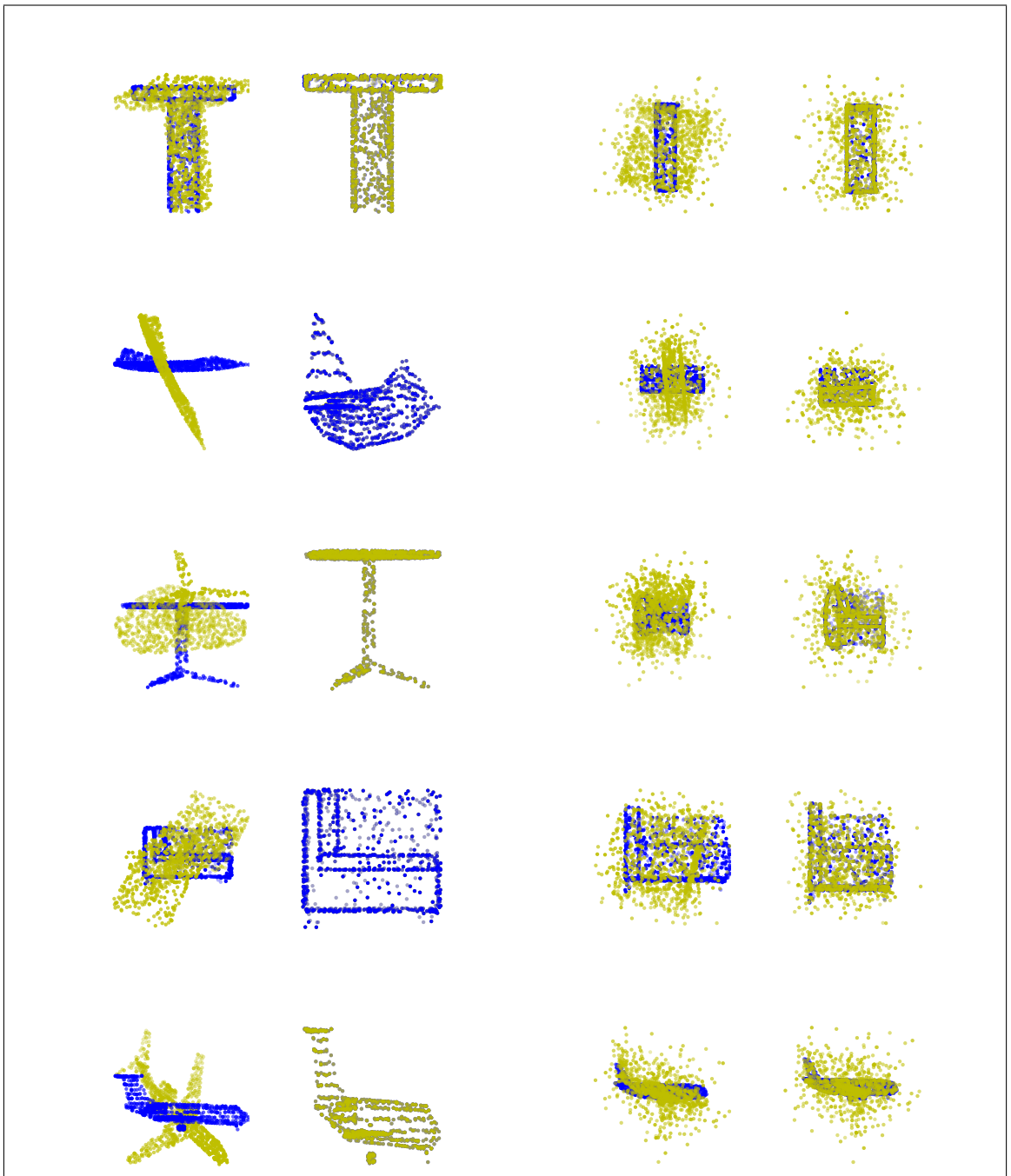
Computational Costs and Timing.

The most computationally expensive operation of our method, just as for QA [83], is the preparation of the couplers $\mathcal{D}_{kl}$ and biases $\mathcal{D}_{kk}$ at each iteration, see Equation (309), which require $O(N\xi^2K^2p^4)$ and $O(N\xi\xi'Kp^2)$ operations, where ξ and ξ' represent the cost for computing $\mathbf{R}_c y_i$ and $\mathbf{R}_c y_i - x_i$.

Table 6 compares the QPU access time of IQT and the CPU access time for ICA for solving the QP for various parameter configurations, averaged over 10 problem instances and for one iteration. For both IQT and ICT, we provide the time for the matrix preparations as well. We observe that both the QPU and CPU access times remain relatively constant with respect to the different combinations. On the contrary, the matrix preparation time grows with the number of points n and exceeds the QPU and CPU access time for moderate n.

Limitations.

We regard IQT as a first step towards efficient transformation estimation and ultimately point cloud registration on quantum hardware. Currently, the run



(a) *Noise-free data.* (b) *Noisy data.*

Figure 6.4.2: *Performance of IQT.* ***(a)*** *Registration results on perfect point sets pairs.* ***(b)*** *Registration results on template point sets with an outlier ratio of* 50% *to the reference point sets. The point sets used are 3D points from the completion3D dataset [24]. Blue points represent reference points and olive points represent template points. The initial alignment is shown on the left and the result of the registration on the right. Even with strong noise, the IQT algorithm provides a robust transformation estimation.*

time is dominated by preparation of the matrices both in the quantum as in the classical case. Limited availability of quantum hardware currently introduces additional time penalties for transfer to quantum computing service providers. Ultimately, it would be beneficial to have an intrinsical (circuit-model) quantum formulation that does not rely on AQC, which we leave for future work.

6.5 Conclusion

The proposed quantum approach for transformation estimation from point sets demonstrates the potential of D-Wave quantum annealers to solve practically relevant problems. While classical methods often relax combinatorial problems to continuous ones, mapping our classical optimization problem to a form adapted for quantum computing goes the reverse way, as current quantum annealers are more efficient in solving combinatorial problems. In an iterative and hybrid optimization regime, we classically compute the cost matrix of a QUBO problem that we solve on the quantum machine, which allows to register the point sets with refinement to improve accuracy.

We believe that transferring methods from one domain (classical) to another (quantum) and vice versa could be an impactful way to push forward both fields and improve adaptation. As quantum technologies continue to mature, we anticipate a transformative impact of the synergy between classical and quantum approaches in image processing.

CHAPTER 7

Further Quantum Optimization Perspectives

In this chapter, we apply the recently proposed quantum Hamiltonian descent (QHD) algorithm [15] to the non-convex rigid image registration problem. Prerequisites for this chapter are quantum evolution and Trotterization from Chapter 2, as well basics of gradient-based optimization from Chapter 3. The QHD method is derived from the path integral of dynamical system referring to the continuous time limit of classical gradient descent methods. While gradient descent methods can only move to a local minimum, the intrinsic quantum nature of quantum elements allow to explore classically prohibited paths and boost the performance of QHD to non-convex problems.

We begin the chapter by setting up the optimization framework. Then, we provide a brief review of gradient flows as classical dynamics, as well as quantum dynamics on which the QHD method is based. Lastly, we classically simulate and apply the method to the image registration problem, finding that the method can effectively target the global solution of the non-convex problem.

7.1 The problem

We consider the optimization problem

$$\inf_{x\in\mathcal{X}} f(x), \tag{314}$$

where the feasible domain $\mathcal{X}$ is a finite dimensional convex set and $f : \mathcal{X} \to \mathbb{R}$ is a convex and continuously differentiable function. We assume that f has a unique minimizer $x^\star \in \mathcal{X}$ satisfying the optimality condition $\nabla f(x^\star) = 0$. Many

gradient-based optimizers for minimizing f, some of which were mentioned in Chapter 2, are sequential algorithms generating a series of iterates

$$x_{k+1} := T_k(x_k), \tag{315}$$

where $T_k : \mathcal{X} \to \mathcal{X}$ is an update function that depends on the objective f and typically uses its derivatives. In Chapter 2, we intuitively thought about the series $(x_k), k = 0, 1, \ldots, \infty$ as different positions of a particle rolling downhill on a the energy function landscape. In this chapter, we seek to make use of classical, then quantum dynamics to understand the behaviour of the particle on the optimization landscape.

In classical dynamics [126], the particle follows a predefined *curve* or *flow*, that the iterates (x_k) discretize. Contrastingly, in quantum mechanics, as highlighted by the quantum Hamiltonian descent algorithm (QHD) by Leng et al. [15], the particle is characterized by a probability distribution across positions and moments in time. We will simulate the QHD algorithm and apply it to a challenging non-convex image registration problem.

7.2 Preliminaries

7.2.1 Classical Dynamic of Gradient-based Optimizers

A powerful way to understand the path of the "particle," as introduced by Wibisono et al. [126], is by considering the classical dynamic of the particle when it is subjected to external and internal energies.

The Bregman-Lagrangian.

We start by defining a distance metric on the feasible set $\mathcal{X}$:

Definition 7.2.1 (*Bregman distance [126]*)

Let $h : \mathcal{X} \to \mathbb{R}$ be a continuous, differentiable, convex function that satisfies $\|h(x)\|_2^2 \to \infty$ for $\|x\|_2 \to \infty$. The Bregman distance *is a function $D_h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as*

$$D_h(x, y) := h(y) - h(x) - \langle \nabla h(x), y - x \rangle . \tag{316}$$

The function h is called distance generating function.

Now, let $X : [0, T] \to \mathcal{X}$ denote the continuous-time curve of the particle that depends on the time $t \in [0, T]$ for some $T \in \mathbb{N}$. In classical dynamic, we will define a function, a so-called *Lagrangian*, that "captures" the energies in the system over the time. The notation $\square_t$ will be used as a short-cut for $\square(t)$.

Let

$$\mathcal{L}(t, X, \dot{X}) := e^{\alpha_t + \gamma_t} \left(-\beta_t f(X_t) + D_h(X_t + e^{-\alpha_t} \dot{X}_t, X_t) \right) \tag{317}$$

be that function, a smooth function of the time variable $t \in [0, T]$, of the continuous-time curve X and its temporal derivative $\dot{X}$. Wibisono et al. [126] use the term *Bregman-Lagrangian* for the Lagrangian $\mathcal{L}$ of the system.

The Bregman-Lagrangian is the sum of the potential energy of the particle, as described by the objective function f, and its kinetic energy, which is modelled by the Bregman-distance term $D_h : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The time-dependent functions $\alpha, \beta, \gamma : [0, T] \to \mathbb{R}$ are freely choosable damping functions satisfying the *ideal scaling condition*

$$\dot{\beta}_t \leq e^{\alpha_t} \text{ and } \dot{\gamma}_t = e^{\alpha_t}. \tag{318}$$

The Euler-Lagrange equation.

The laws of classical mechanics [16, 127] force the particle to take the path X according to the least action principle, which requires that the action functional

$$S(X) := \int_{t=0}^{T} \mathcal{L}(t, X_t, \dot{X}_t) dt \tag{319}$$

of the curve must remain in stationary points throughout the time evolution of the system. This constraint, which entails setting to zero the derivative of S with respect to X, captures the equation of motion of the particle in the Euler-Lagrange equation

$$\frac{\partial}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{X}}\right) - \frac{\partial \mathcal{L}}{\partial X} = 0, \tag{320}$$

which – for the Bregman-Lagrangian $\mathcal{L}$ and under the ideal scaling conditions in Equation (318) –reduces to the Bregman flow

$$\ddot{X}_t + (e^{\alpha_t} - \dot{\alpha}_t)\dot{X}_t + e^{2\alpha_t + \beta_t}\left[\nabla^2 h(X_t + e^{-\alpha_t}\dot{X}_t)\right]^{-1} \nabla f(X_t) = 0. \tag{321}$$

If f is convex, the following theorem states that, in ideal scaling conditions, the Bregman flow indeed conduces the particle to the minimizer $x^\star$ of f:

Theorem 7.2.1 (*Convergence of the Bregman flow [126, Theorem 1.1]*)

If the ideal scaling conditions Equation (318) are satisfied and f is convex, then the solution of the Bregman flow in Equation (321) satisfies

$$f(X_t) - f(x^\star) \leq O(e^{-\beta_t}). \tag{322}$$

Proof : See Wibisono et. al [126]. ■

Discussion.

An inherent limitation of gradient-based optimizers, as evident from Equation (321), is that the generating flows drive the particle to stationary points of the action

functional, such that the particle should stop rolling when the gradient $\nabla f(X_t)$ of f at the current position X_t of the particle vanishes, as we then lose any information about possible decent directions of f. In many practical optimization problems, the function f is non-convex and can have many such points, ranging form local minimizers, saddle points, to even local maximizers, at which the flow can thus get stuck.

7.2.2 Quantum Hamiltonian Descent

Quantum Hamiltonian Descent (QHD), introduced by Leng et al. [15], is a promising quantum method for escaping local minima and finding globally optimal solutions. In a way, the method can be seen as building on quantum dynamics; we will provide a brief overview in this section.

Unlike classical mechanics, where the rolling particle possesses, at each time step t, a defined position $X_t \in \mathcal{X}$ and velocity $\dot{X}_t$, in quantum mechanics, the location of the particle at each time is only known in terms of a complex-valued wave-function $\psi(x,t)$, whose squared modulus is the probability of observing the particle at each point.

In this quantum-mechanical setting, we are interested in the evolution of the particle from a point $a = X(t_a)$ at time t_a to $b = X(t_b)$ at time t_b. Intuitively, not only paths minimizing the action S and satisfying the Euler-Lagrange equation, but all trajectories contribute to the evolution from a to b [16, Section 2-2].

The transition probability of this is

$$p(a,b) := |K(b,t_b;a,t_a)|^2, \tag{323}$$

where

$$K(b,t_b;a,t_a) \propto \sum_{\text{path X from a to b}} e^{(i/\hbar)S(X)} \tag{324}$$

is a so-called *propagator*. We refer to the discussion by Feynman [16, Section 2-4] for a precise mathematical understanding of the sum over paths.

For simplicity, we assume that the position variable x is one-dimensional, but the results can be generalized to more dimensions. The wave function $\psi(x,t)$ at time t_b is constructed as

$$\psi(x,t_b) = \int_{\mathbb{R}} K(x,t_b;y,t_a)\psi(y,t_a)dy, \tag{325}$$

which can be regarded as a convolution of the wave function at time t_a with the propagator kernel [16, Equation 3.42] .

Derivation of the QHD Hamiltonian.

The evolution of the wave function is fully described by a Hamiltonian $\mathbf{H}(t)$ driving the Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\left|\psi(x,t)\right\rangle = \mathbf{H}(t)\left|\psi(x,t)\right\rangle. \tag{326}$$

The core idea of QHD [15] is to construct a Hamiltonian $\mathbf{H}(t)$ that forces the wave function to collapse at the position $x^\star$ minimizing the function f. We briefly review this construction next, referring the reader to [15, Appendix A] for a full discussion.

In QHD, a Lagrangian in defined as

$$\mathcal{L}(t, X_t, \dot{X}_t) := -e^{\chi_t} f(X_t) + e^{-\varphi_t}\left(\frac{1}{2}\|\dot{X}_t\|^2\right), \tag{327}$$

which is a special case of the Bregman-Lagrangian with the damping functions $\varphi_t = \alpha_t - \gamma_t$, $\chi_t = \alpha_t + \beta_t + \gamma_t$ and $h(\cdot) = \|\cdot\|^2$. With this choice of damping functions, the QHD ideal scaling condition requires that $\lim_{t\to\infty} e^{\varphi_t/\chi_t} = 0$, in which case the kinetic energy is gradually drained out from the system.

If we consider an infinitesimal time evolution from $t_a = t$ to $t_b = t + \epsilon$, then the state of the wave function sightly changes from $X_t = x$ to $X_{t+\epsilon} = y = x + \eta$, and:

1. The damping functions φ and χ can be treated as constants,
2. The action is approximately ϵ times the Lagrangian:

$$S(X) = \epsilon \cdot \mathcal{L}\left(t, \frac{x+y}{2}, \frac{x-y}{2}\right) \tag{328}$$

3. The propagator can by evaluated by

$$K(x,t;y,t+\epsilon) = \frac{1}{A}\exp\left\{\frac{i}{\hbar}\cdot\epsilon\cdot\mathcal{L}\left(t, \frac{x+y}{2}, \frac{x-y}{\epsilon}\right)\right\}, \tag{329}$$

   where A is a constant normalization factor.

Plugging this together into Equation (325) with the change of variable $y \mapsto x + \eta$ allows to write the wave function as:

$$\psi(x,t+\epsilon) = \int_{\mathbb{R}} \frac{1}{A}\exp\left\{\frac{i}{\hbar}\cdot\epsilon\cdot\left(-e^{\chi_t} f\left(x+\frac{\eta}{2}\right) + e^{-\varphi_t}\left(\frac{\eta^2}{2\epsilon^2}\right)\right)\right\}\psi(x+\eta,t)d\eta. \tag{330}$$

We approximate from Equation (330) the exponential term of the potential energy

and the potential energy itself as

$$\exp\left(-\frac{i\epsilon}{\hbar}\cdot e^{\chi_t} f\left(x+\frac{\eta}{2}\right)\right) \approx 1-\frac{i\epsilon}{\hbar}\cdot e^{\chi_t} f\left(x+\frac{\eta}{2}\right)+\ldots, \tag{331}$$

$$f\left(x+\frac{\eta}{2}\right) \approx f(x)+\eta\frac{\partial}{\partial\eta}f(x)+\ldots. \tag{332}$$

Keeping only the first two terms of Equation (331) and the first term of Equation (332), it turns out that

$$\psi(x,t+\epsilon) \approx \int_{\mathbb{R}} \frac{1}{A}\left\{\left(1-\frac{i\epsilon}{\hbar}\cdot e^{\chi_t} f(x)\right)\cdot\exp\left(-\frac{e^{-\varphi_t}}{2i\hbar\epsilon}\eta^2\right)\right\}\psi(x+\eta,t)d\eta. \tag{333}$$

The next step is to approximate the wave function in time using a first-order Taylor approximation, and in space using a second-order Taylor approximation:

$$\psi(x,t+\epsilon) \approx \psi(x,t)+\epsilon\cdot\frac{\partial}{\partial t}\psi(x,t) \tag{334}$$

$$\psi(x+\eta,t) \approx \psi(x,t)+\eta\cdot\frac{\partial}{\partial x}\psi(x,t)+\frac{\eta^2}{2}\cdot\frac{\partial^2}{\partial x^2}\psi(x,t). \tag{335}$$

With Equations (334) and (335), we can write Equation (333) as

$$\begin{aligned}\psi(x,t)+\epsilon\cdot\frac{\partial}{\partial t}\psi(x,t) \approx \frac{1}{A}&\left(1-\frac{i\epsilon}{\hbar}\cdot e^{\chi_t} f(x)\right)\\ &\cdot\left\{C_1\psi(x,t)+C_2\frac{\partial}{\partial x}\psi(x,t)+C_3\frac{\partial^2}{\partial x^2}\psi(x,t)\right\},\end{aligned} \tag{336}$$

where

$$C_1 := \int_{\mathbb{R}} \exp\left(-\frac{e^{-\varphi_t}}{2i\hbar\epsilon}\eta^2\right)d\eta = \sqrt{i2\pi\hbar\epsilon e^{\varphi_t}}, \tag{337}$$

$$C_2 := \int_{\mathbb{R}} \eta\exp\left(-\frac{e^{-\varphi_t}}{2i\hbar\epsilon}\eta^2\right)d\eta = 0, \tag{338}$$

$$C_3 := \int_{\mathbb{R}} \frac{\eta^2}{2}\exp\left(-\frac{e^{-\varphi_t}}{2i\hbar\epsilon}\eta^2\right)d\eta = \frac{i\hbar\epsilon e^{\varphi_t}}{2}\sqrt{i2\pi\hbar\epsilon e^{\varphi_t}}. \tag{339}$$

We solve for the constant A by equating the coefficients of the ϵ^0-term of Equation (336) and considering only the first term of the approximation w.r.t. η, yielding

$$\psi(x,t) = \frac{C_1}{A}\psi(x,t), \tag{340}$$

hence $A = \sqrt{i2\pi\hbar\epsilon e^{\varphi_t}}$. This simplifies Equation (336) to

$$\psi(x,t) + \epsilon \cdot \frac{\partial}{\partial t}\psi(x,t) \approx \left(1 - \frac{i\epsilon}{\hbar} \cdot e^{\chi_t} f(x)\right) \cdot \left\{\psi(x,t) + \frac{i\hbar\epsilon e^{\varphi_t}}{2}\frac{\partial^2}{\partial x^2}\psi(x,t)\right\}. \quad (341)$$

Finally, by equating the ϵ^1-terms it turns out that

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = \left(e^{\chi_t} f(x) - \frac{\hbar e^{\varphi_t}}{2}\frac{\partial^2}{\partial x^2}\right)\psi(x,t), \quad (342)$$

and we see that our desired evolution of the wave function is described by the Schrödinger equation, with the term in parentheses taking the place of the Hamiltonian.

In more dimension, this generalizes to

$$i\hbar\frac{\partial}{\partial t}\psi(x,t) = \left(e^{\chi_t} f(x) - \frac{\hbar e^{\varphi_t}}{2}\Delta_x\right)\psi(x,t), \quad (343)$$

where Δ_x is the d-dimensional Laplacian operator.

The following theorem guarantees that in the convex setting, evaluating the energy of the system at the end of the quantum evolution yields the minimal value of f:

Theorem 7.2.2 (*Convergence of QHD [15, Theorem 1]*)

Let f be a continuous differentiable convex function with a unique local minimizer $x^\star$ and the ideal scaling condition Equation (318) holds. Then, for any smooth initial wave function $|\psi(x,0)\rangle$, the solution $|\psi(x,t)\rangle$ at any time t of the Schrödinger equation (326) with the Hamiltonian

$$\mathbf{H}(t) := e^{\chi_t} f + e^{\varphi_t}\left(-\frac{\hbar}{2}\Delta_x\right) \quad (344)$$

such that $\chi_t = \alpha_t - \gamma_t$ and $\varphi_t = \alpha_t + \beta_t + \gamma_t$ satisfies

$$\int_{\mathcal{X}} f(x)\|\psi(x,t)\|^2 dx - f(x^\star) \leq O(e^{-\beta_t}). \quad (345)$$

Proof: See [15, Theorem 1]. ■

In the non-convex setting, it is also shown in [15, Theorem 2] that the wave function can still converge to the minimizer of f under some extra assumptions requiring in particular that the total evolution time T is large enough, that the parameters vary slowly enough, and that the evolution stays in a space spanned by the N eigenstates with the smallest energy [15, Assumptions 1-3].

Discussion.

The viewpoint above, formulated in [15], provides an interesting perspective that allows to think of gradient-based descent methods in terms of mechanical dynamics. The transition from classical to quantum dynamics potentially allows to also find global solutions of non-convex problems. In the following section, we will again consider the inherently non-convex image registration problem and investigate if it can be solved using the QHD approach.

7.3 Application to Rigid Image Registration

The goal of this section is to apply the QHD method to solving the rigid image registration problem. In the considered registration problem, two images $R, T : \Omega \to X$ of an object are given, where $\Omega := [0, m] \times [0, n] \subset \mathbb{R}^2$ is the image domain and $X \subset \mathbb{R}_+$ the range.

We restrict ourselves to rigid deformations $\varphi_\omega : \mathbb{R}^2 \mapsto \mathbb{R}^2$,

$$\varphi_\omega(x) := \begin{pmatrix} \cos(\omega_1) & -\sin(\omega_1) \\ \sin(\omega_1) & \cos(\omega_1) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \omega_2 \\ \omega_3 \end{pmatrix},$$

parameterized by the vector $\omega := (\omega_1, \omega_2, \omega_3)$, where ω_1 is the rotation angle, and ω_2 and ω_3 form the translation vector.

In order to find the deformation that best aligns the two given images R and T, we formulate the "sum-of-squared-differences" (SSD) objective function

$$\texttt{SSD}(R, T \circ \varphi_\omega) := \int_\Omega \left(R(x) - T(\varphi_\omega(x))\right)^2 dx. \tag{346}$$

The goal is to minimize this energy over all parameter vectors $\omega := (\omega_1, \omega_2, \omega_3)^\top$, with $\omega_1 \in [0, 2\pi],\ \omega_2 \in [-m, m]$ and $\omega_3 \in [-n, n]$.

7.3.1 Numerical Simulation

Simulating quantum evolutions requires techniques such as product or Trotterization formulae as presented in Section 2.6. To numerically represent the wave function, we have to discretize the Schrödinger equation (326) both in space and time. The discretization schemes follow [15, Appendices E and F].

Space Discretization.

We use the standard regular grid discretization of continuous spaces. For each parameter space $\Omega_i = [a_i, b_i], i = 1, 2, 3$, we construct a regular mesh

$$\mathcal{M}_i := \left\{\omega_i^j : \quad \omega_i^j = a_i + j\delta_i, \quad j = 0, \ldots, N-1\right\}, \tag{347}$$

where $\delta_i := \frac{b_i - a_i}{N}$ for some $N \in \mathbb{N}$ is the step size. For convenience, we will let N be a power of 2. The spatially discretized wave function is a state vector $|\omega(t)\rangle := |\omega_1(t), \omega_2(t), \omega_3(t)\rangle$, constructed by concatenating three registers, each register $|\omega_i(t)\rangle := \sum_{\omega_i^j \in \mathcal{M}_i} c_i^j(t) |\omega_i^j\rangle$ with $i = 1, 2, 3$ corresponding to one parameter of the problem. The second order derivative of the wave function get discretized with finite differences, which, in practice is computed through the discrete 3-dimensional Laplace operator.

Time Discretization.

We first rewrite the QHD Hamiltonian $\mathbf{H}(t)$ from Equation (344) in the form $\mathbf{H}(t) = \mathcal{A}(t)\mathbf{H}(0) + \mathcal{B}(t)\mathbf{H}(T)$ with

$$\mathcal{A}(t) = e^{\varphi_t}, \tag{348}$$
$$\mathcal{B}(t) = e^{\chi_t}, \tag{349}$$
$$\mathbf{H}(0) = -\frac{\hbar}{2}\Delta_x, \tag{350}$$
$$\mathbf{H}(T) = f. \tag{351}$$

We recall that the final Hamiltonian $\mathbf{H}(T)$ is diagonal in the computational basis, such that $\mathbf{H}(T)|x\rangle = f(x)|x\rangle$ for some basis state $|x\rangle$. For the damping functions, we choose

$$\varphi(t) = \frac{2}{\epsilon + t^3}, \quad \text{and} \quad \chi_t = 2t^3, \tag{352}$$

where ϵ is a small positive constant to avoid the singularity of dividing by 0. These are the same damping functions as in [15, Equation C.4] and correspond to the weighting coefficients of Nesterov's accelerated gradient descent [128].

Time discretization consists in discretizing the continuous time variable $t \in [0, T]$ into discrete time steps $j \cdot \frac{T}{r}$ for $j = 0, \ldots, r$ and some $r \in \mathbb{N}$. Using the techniques from Section 2.6, we know that the unitary evolution of the system can be approximated by the operator

$$\mathbf{U}(T) = \prod_{j=1}^{r} e^{-i\frac{T}{r}\mathcal{A}(j\cdot\frac{T}{r})\mathbf{H}(0)} \cdot e^{-i\frac{T}{r}\mathcal{B}(j\cdot\frac{T}{r})\mathbf{H}(T)}. \tag{353}$$

The ultimate challenge is now to exponentiate the Hamiltonians in the above product. This is carried on by using operator functions [12, Section 2.1.8] for normal matrices. In other words, a given Hamiltonian $\mathbf{H}$ with spectral decomposition $\mathbf{H} = \Psi\Sigma\Psi^\dagger$ satisfies $\exp(\mathbf{H}) = \Psi \exp(\Sigma)\Psi^\dagger$, where $\exp(\Sigma) = \operatorname{diag}(\exp(\sigma^x))_{x=1}^N$.

Exponentiating the initial Hamiltonian $\mathbf{H}(0)$ is simple, as the 1-dimensional Laplacian is a Toeplitz matrix which can be diagonalized – for periodic boundary conditions – by the discrete Fourier transform, which is efficient on quantum computers [12, 129].

Exponentiating the final Hamiltonian $\mathbf{H}(T)$ is straightforward, as it is diagonal.

Update Rule.

Overall, defining a step size $s := \frac{T}{r}$ we get the update rule

$$|\omega(j+1)\rangle = \Psi \cdot \exp\left(-is\mathcal{A}\left(js\right)\Sigma\right) \cdot \Psi^{\dagger} \cdot \exp\left(-is\mathcal{B}\left(js\right)f\right)|\omega(j)\rangle \tag{354}$$

$j = 0, \ldots, r$, which is also known as pseudo-spectral method [15, 130] and is the standard numerical method for Schrödinger equations. In our experiments, we set $T = 1$ and $r = 10^5$ update steps.

7.3.2 Results

In this section, the QHD algorithm is classically simulated on the rigid image registration problem. The Laplace operator is implemented with periodic boundary condition. The images in the experiments are X-ray images of human hands from the FAIR software package [20].

Results with Rotation Only.

To illustrate the algorithm, we first consider the one-dimensional problem where we are searching only for the rotation angle, while fixing the translation to zero.

The wave function evolution is depicted in Figure 7.3.1. The rotation domain $[0, 2\pi]$ is discretized into a $2^8 = 256$-element grid. Clearly, the objective function is non-convex. We see that the measurement probability of the wave-function gradually contracts to a sharp peak at the global minimizer, demonstrating how accurate the QHD algorithm can solve non-convex optimization problems.

The angle with the highest probability of measurement is considered as the minimizer of the objective function. For this angle, the registration results are depicted in Figure 7.3.2, and can be verified to decrease the objective function value. However, the accuracy is still limited, due to the fact that the optimization is performed only for the rotation parameter while keeping the translation to zero.

Results with Rotation and Translation.

The 3-dimensional rotation-translation registration problem is solved in a similar manner as the 1-dimensional rotation-only case, with the exception that the Laplace operator is higher-dimensional and implemented with Neumann boundary condition. Due to computational constraints, we discretize each parameter space with $2^4 = 16$ basis states: The rotation parameter space is reduced to $[1.8 \cdot \pi, 2\pi]$, and the translation parameter space is a small $[-7, 8] \times [-7, 8]$ region cropped from the full $[-64, 64] \times [-64, 64]$ domain. We evaluate the operator f on those grid-points and evolve the wave function according to the QHD algorithm. The solution is computed as the basis state for which the wave function probability peaks. The visual registration results are presented in Figure 7.3.3 and can be verified to both globally minimize the objective function and accurately solve the underlying non-convex registration problem. We conjecture that a finer discretization would produce even better results.

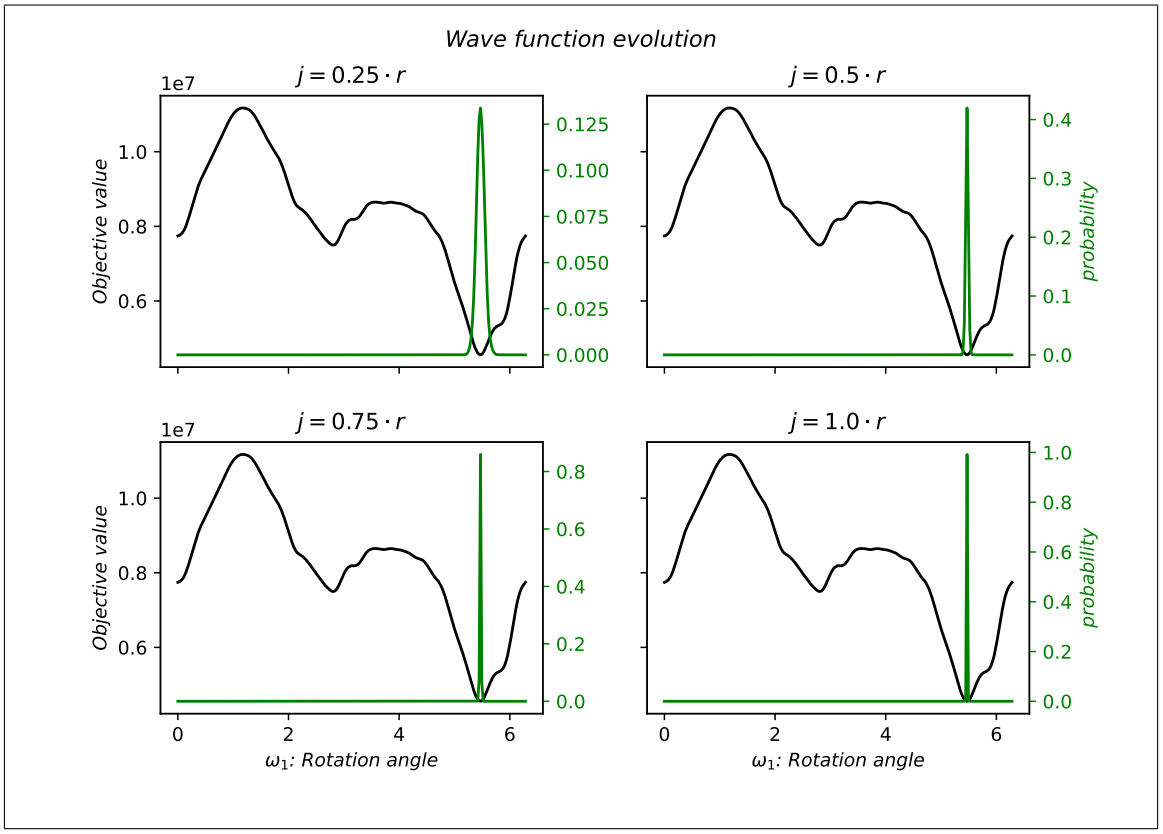


Figure 7.3.1: *Wave function evolution under the action of the QHD algorithm for the one-dimensional problem of finding the optimal angle that aligns two images.* ***(Left to right, top to bottom)*** *In green the evolution of the squared magnitude of the wave function, i.e., the probability of observing each angle at the final state, and in black the non-convex objective function. We see that the wave function probability clearly peaks at the global minimum, indicating the potential of QHD in solving non-convex problems. The transformations obtained in this way are visualized in Figure 7.3.2.*

Discussion.

It is very promising to see that in practice, the QHD method appears to be able to effectively drive the state towards a peak at the global minimizer even for non-convex image registration problems.

Simulating the Schrödinger equation as required by the method of course implicitly requires an efficient implementation of an operator for evaluating the objective function on basis states while running the algorithm. In our classical simulation approach, this operator is computed beforehand, which effectively means computing the objective function for all parameter values on a given grid. This is certainly not a viable approach for constructing better classical solvers and we do not claim so; we rather envision that ultimately, the specific Schrödinger equation will be able to be efficiently simulated on universal quantum computers.

For the method to run on such a universal quantum computer, one could think

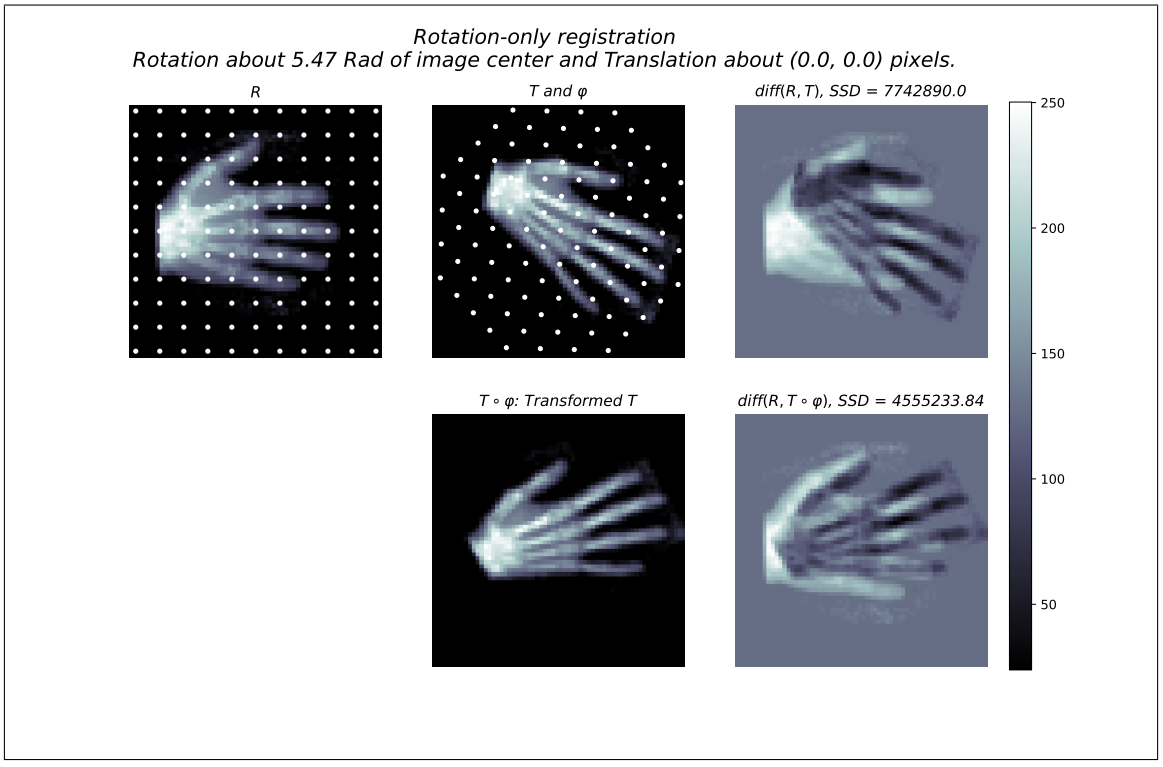


Figure 7.3.2: *Registration results using the QHD approach in Figure 7.3.1. (**Top row**) Reference image and template image on which the image grids before and after the deformation are overlaid, and the difference image before the registration. (**Bottom row**) Deformed template image and associated difference image. We see that QHD applied to this simple but non-convex registration problem finds better registration results compared to the initial situation, but the fact that the optimization is performed only for the rotation parameter while keeping the translation to zero still limits the accuracy. In Figure 7.3.3, the registration results for the optimization over the rotation and translation parameters is presented.*

about this operator as an oracle acting on a superposition state and evaluating the objective function for all basis states in parallel. At the current state of quantum hardware, implementing such an oracle for the image registration problem would be a serious challenge.

7.4 Conclusion

We have demonstrated the applicability of the quantum Hamiltonian descent [15] algorithm to the rigid image registration problem. The method works by evolving the wave function of a quantum system in order to bring the system in a state with the smallest potential energy, which corresponds to the minimal value of the objective function to minimize. For running on a universal quantum machine, the method requires the implementation of an efficient quantum oracle capable to evaluate the objective function on basis quantum states. We classically

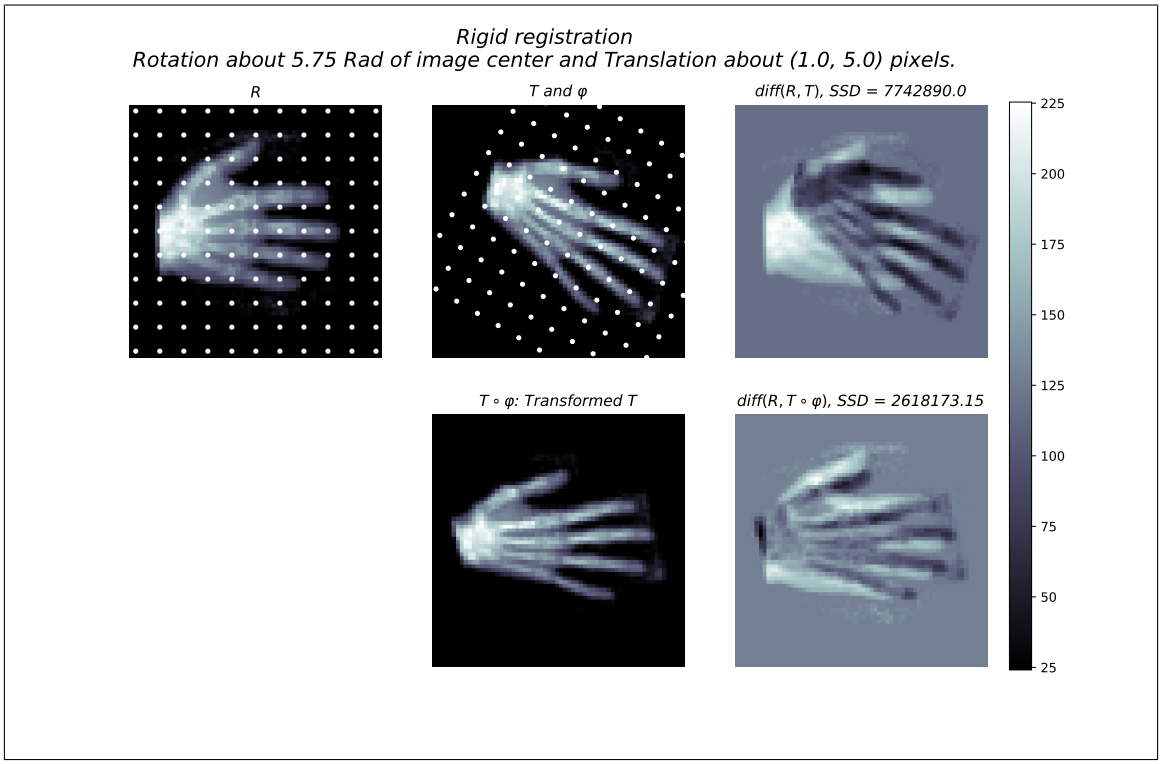


Figure 7.3.3: *Registration results using the QHD approach in Figure 7.3.1 for the 3-dimensional case. (**Top row**) Reference image and template image on which the image grids before and after the deformation are overlaid, and the difference image before the registration. (**Bottom row**) Deformed template image and it associated difference image. We see that QHD applied to this simple but non-convex registration problem finds a visually very accurate registration despite the local minima.*

simulated the algorithm and found that it is robust in solving the non-convex registration problem, as it escapes local minima and target the global solution. We hope that our results motivate further research on efficient quantum image representations and transformations for image processing problems, to benefit from the full potential of quantum optimization.

CHAPTER 8

Conclusion

In this thesis, we have introduced several quantum algorithms with applications in image processing. The main focus has been on deriving new mathematical models and optimization methods amenable to quantum computation for solving binary combinatorial and image registration problems.

- Variational Quantum Computing.

  We developed in Chapter 4 a hybrid quantum-classical variational quantum algorithm for solving the Ising problem. Our proposed UQIsing algorithm efficiently block-encodes the Ising problem into a unitary operator acting on a parameterized ansatz. We then used gradient descent for optimizing a set of rotation angles to bring the ansatz into the basis-state that locally solves the optimization problem. The resulting circuit is low-depth, and requires a one-to-all qubits connectivity, properties which are compatible with actual noisy intermediate scale quantum devices. Due to the non-convexity of the resulting objective function, the method does not guarantee to return a global optimum, but experimentally finds at least a good approximation.

- Quantum Amplitude Amplification.

  Moving from hybrid to fully-quantum approaches, we introduced in Chapter 5 a fully universal quantum algorithm for solving the Ising problem. The proposed PM-NBAA method employs quantum amplitude amplification techniques to iteratively amplify the probability of the state encoding the solution of the problem. At the end of an optimal number of iterations, it guarantees that the quantum system collapses into the optimal state with a higher probability than for any non-optimal state.

- Adiabatic Quantum Computing.

  We introduced in Chapter 6 an iterative quantum algorithm for iteratively estimating rigid transformation from point sets using adiabatic quantum computing. Our proposed IQT algorithm operates on the Lie algebra of rotation matrices. The method performs well on several 2D and 3D point sets, and robustly aligns the points even in the presence of outliers. In contrast to the existing QUBO-based approach for the task, it guarantees to deliver consistent and orthogonal rotation matrices, and provides higher accuracy.

- Quantum Hamiltonian Descent.

  We explored in Chapter 7 the quantum Hamiltonian descent (QHD) algorithm as promising method for solving the non-convex rigid image registration problem. We classically implemented and simulated the algorithm, and found that it accurately converges towards a global solution of the problem.

Future Work.

The algorithms developed in this thesis leave some questions unanswered, revealing what can and should be done to unlock the potentials of quantum computing for solving image processing problems.

The performance of the UQIsing algorithm for solving the Ising problem still depends on the quality of the underyling classical optimization scheme, and it is easy to imagine that advances in this area can lead to better solutions. Similarly, warm-start techniques could be derived to prepare the ansatz in a better initial state for reaching global convergence.

The most important unanswered question regarding the PM-NBAA method concerns the stopping criterion: The iteration should be stopped when the $\cos(\theta_k)$ term – corresponding to a weighted average – becomes negative. The big question here is how to find an explicit expression for the number of iterations when this happens, and we have to leave this for future work.

The IQT method could be extended and applied to other constrained optimization problems with Lie algebra structure such as optimizations problems over permutation matrices. Also, the practical availability and trajectory of AQC hardware size for solving QUBOs could make it feasible to solve realistic even less restricted – most importantly, non-linear – image registration problems using the method in near- to mid-term.

Applying the QHD algorithm to realistic image processing and optimization problems requires developing low-costs quantum image representations and transformation methods, both universal and adiabatic, and possibly quantum devices capable to evolve the wave function under arbitrary Hamiltonian operator on the hardware level. While this will certainly have a longer time horizon before practicability, it is theoretically probably the most exciting avenue.

Bibliography

[1] Richard P Feynman et al. Simulating physics with computers. *Int. j. Theor. phys*, 21(6/7), 1982.

[2] Eleanor Rieffel and Wolfgang Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32, 09 1998.

[3] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1), Jan 2018.

[4] Charles H Bennett. Logical reversibility of computation. *IBM journal of Research and Development*, 17(6):525–532, 1973.

[5] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[6] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

[7] IBM Quantum. https://quantum-computing.ibm.com, June 2023.

[8] D-Wave Systems. QPU Solver Datasheet. https://docs.dwavesys.com/docs/latest/doc_qpu.html, June 2023.

[9] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[10] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[11] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods: nonlinear programming*, volume 1. Springer Science & Business Media, 2006.

[12] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th edition, 2010.

[13] Robert S Sutor. *Dancing with Qubits: How quantum computing works and how it can change the world*. Packt Publishing Ltd, 2019.

[14] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J Egger, Bruce G Elmegreen, et al. Quantum optimization: Potential, challenges, and the path forward. *arXiv preprint arXiv:2312.02279*, 2023.

[15] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu. Quantum hamiltonian descent. *arXiv preprint arXiv:2303.01471*, 2023.

[16] Richard P Feynman, Albert R Hibbs, and Daniel F Styer. *Quantum mechanics and path integrals*. Courier Corporation, 2010.

[17] Ioannis Pitas. *Digital image processing algorithms and applications*. John Wiley & Sons, 2000.

[18] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley world student series. Addison-Wesley, 1992.

[19] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.

[20] Jan Modersitzki. FAIR: Flexible Algorithms for Image Registration. *Proc SIAM*, 2009.

[21] Point Cloud Mnist 2D. https://www.kaggle.com/cristiangarcia/pointcloudmnist2d.

[22] Bo Li, Aleksandar Jevtic, Ulrik Söderström, Shafiq Ur Réhman, and Haibo Li. Fast edge detection by center of mass. *ICISIP*, pages 103–110, 2013.

[23] The stanford 3d scanning repository. http://graphics.stanford.edu/data/3Dscanrep/.

[24] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR proceedings*, pages 383–392, June 2019.

[25] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. A universal quantum algorithm for weighted maximum cut and ising problems. *Quantum Information Processing*, 22(7):279, 2023.

[26] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[27] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[28] Prasanth Shyamsundar. Non-boolean quantum amplitude amplification and quantum mean estimation. *Quantum Information Processing*, 22(12):1–45, 2023.

[29] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. An iterative quantum approach for transformation estimation from point sets. In *Computer Vision and Pattern Recognition*, pages 529–537, 2022.

[30] A Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993.

[31] AM Turing. On computable numbers, with an application to the entscheidungsproblem. a correction. *Proceedings of the London*, 1938.

[32] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2020 edition, 2020.

[33] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[34] David Elieser Deutsch. Quantum computational networks. *Proceedings of the royal society of London. A. mathematical and physical sciences*, 425(1868):73–90, 1989.

[35] Colin P Williams and Colin P Williams. Quantum gates. *Explorations in Quantum Computing*, pages 51–122, 2011.

[36] David P DiVincenzo. Topics in quantum computers. In *Mesoscopic electron transport*, pages 657–677. Springer, 1997.

[37] Sabine Jansen, Mary-Beth Ruskai, and Ruedi Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10):102111, Oct 2007.

[38] Paul Ehrenfest. Adiabatische invarianten und quantentheorie. *Annalen der Physik*, 356(19):327–352, 1916.

[39] Max Born and Vladimir Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3-4):165–180, 1928.

[40] Bruno Apolloni, C Carvalho, and Diego De Falco. Quantum stochastic optimization. *Stochastic Processes and their Applications*, 33(2):233–244, 1989.

[41] Tosio Kato. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6):435–439, 1950.

[42] Dorit Aharonov, Wim Van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[43] Wim Van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In *Proceedings 42nd IEEE symposium on foundations of computer science*, pages 279–287. IEEE, 2001.

[44] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. American Mathematical Soc., 2002.

[45] Quantum AI. https://quantumai.google/, June 2023.

[46] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[47] Ryan Murray, Brian Swenson, and Soummya Kar. Revisiting normalized gradient descent: fast evasion of saddle points. *IEEE Trans. Autom. Control*, 64(11):4818–4824, 2019.

[48] Yudai Suzuki, Hiroshi Yano, Rudy Raymond, and Naoki Yamamoto. Normalized gradient descent for variational quantum algorithms. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 1–9, 2021.

[49] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436. Springer, 2012.

[50] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, 2020.

[51] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 2021.

[52] Sebastian Soritz, Daniel Moser, and Heidrun Gruber-Wölfler. Comparison of derivative-free algorithms for their applicability in self-optimization of chemical processes. *Chemistry-Methods*, 2(5):e202100091, 2022.

[53] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[54] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

[55] M.J.D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, Dordrecht, 1994.

[56] Stuart Harwood, Claudio Gambella, Dimitar Trenev, Andrea Simonetto, David Bernal, and Donny Greenberg. Formulating and solving routing problems on quantum computers. *IEEE Transactions on Quantum Engineering*, 2:1–17, 2021.

[57] Colin Campbell and Edward Dahl. Qaoa of the highest order. In *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*, pages 141–146. IEEE, 2022.

[58] Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4:193–201, 1973.

[59] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[60] Stefano Mangini. Variational quantum algorithms for machine learning: theory and applications, 2023.

[61] Xin Wang, Zhixin Song, and Youle Wang. Variational quantum singular value decomposition. *Quantum*, 5:483, 2021.

[62] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[63] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

[64] James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

[65] Michael C Fu and Stacy D Hill. Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE transactions*, 29(3):233–243, 1997.

[66] Qi Wang and James C Spall. Discrete simultaneous perturbation stochastic approximation on loss function with noisy measurements. In *Proceedings of the 2011 American Control Conference*, pages 4520–4525. IEEE, 2011.

[67] Julien Gacon, Christa Zoufal, Giuseppe Carleo, and Stefan Woerner. Simultaneous perturbation stochastic approximation of the quantum fisher information. *Quantum*, 5:567, 2021.

[68] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[69] Barry M McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Courier Corporation, Cambridge, 2014.

[70] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[71] Svatopluk Poljak and Franz Rendl. Solving the max-cut problem using eigenvalues. *Discrete Applied Mathematics*, 62(1-3):249–278, 1995.

[72] Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

[73] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–159, 2004.

[74] Chris HQ Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE international conference on data mining*, pages 107–114, 2001.

[75] Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

[76] Andrew Lucas. Ising formulations of many np problems. *Frontiers in physics*, 2:5, 2014.

[77] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 51(2):271–279, 1989.

[78] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[79] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 1970.

[80] Hiroshi Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1333–1336, 2003.

[81] Jörg H Kappes, Bjoern Andres, Fred A Hamprecht, Christoph Schnörr, Sebastian Nowozin, Dhruv Batra, Sungwoong Kim, Bernhard X Kausler, Thorben Kröger, Jan Lellmann, et al. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, 115(2):155–184, 2015.

[82] Dimitri Bertsekas. *Network optimization: continuous and discrete models*, volume 8. Athena Scientific, 1998.

[83] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. *CVPR*, abs/1912.12296, 2019.

[84] Clarence Zener. Non-adiabatic crossing of energy levels. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 137(833):696–702, 1932.

[85] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[86] Gian Giacomo Guerreschi and Anne Y Matsuura. Qaoa for max-cut requires hundreds of qubits for quantum speed-up. *Scientific reports*, 9(1):1–7, 2019.

[87] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.

[88] Rebekah Herrman, James Ostrowski, Travis S Humble, and George Siopsis. Lower bounds on circuit depth of the quantum approximate optimization algorithm. *Quantum Information Processing*, 20(2):1–17, 2021.

[89] Ruslan Shaydulin and Yuri Alexeev. Evaluating quantum approximate optimization algorithm: A case study. In *2019 tenth international green and sustainable computing conference (IGSC)*, pages 1–6, 2019.

[90] Andreas Bärtschi and Stephan Eidenbenz. Grover mixers for qaoa: Shifting complexity from mixer design to state preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 72–82. IEEE, 2020.

[91] Zhihui Wang, Nicholas C Rubin, Jason M Dominy, and Eleanor G Rieffel. X y mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Physical Review A*, 101(1):012320, 2020.

[92] Zichang He, Ruslan Shaydulin, Shouvanik Chakrabarti, Dylan Herman, Changhao Li, Yue Sun, and Marco Pistoia. Alignment between initial state and mixer improves qaoa performance for constrained optimization. *npj Quantum Information*, 9(1):121, 2023.

[93] Daniel J Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. *Quantum*, 5:479, 2021.

[94] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019.

[95] D. Camps and R. Van Beeumen. Fable: Fast approximate quantum circuits for block-encodings. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 104–113, sep 2022.

[96] Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 427–436, 2006.

[97] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

[98] Stefano Polla, Gian-Luca R Anselmetti, and Thomas E O'Brien. Optimizing the information extracted by a single qubit measurement. *Physical Review A*, 108(1):012403, 2023.

[99] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), 2008.

[100] D-Wave Systems. D-Wave Leap. https://www.dwavesys.com/take-leap, Oct 2021.

[101] D-Wave Systems. D-Wave Ocean Software Documentation. https://docs.ocean.dwavesys.com/, Oct 2021.

[102] Madita Willsch, Dennis Willsch, Fengping Jin, Hans De Raedt, and Kristel Michielsen. Benchmarking the quantum approximate optimization algorithm. *Quantum Information Processing*, 19(7):1–24, 2020.

[103] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[104] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[105] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum counting. In *Automata, Languages and Programming: 25th International Colloquium, ICALP'98 Aalborg, Denmark, July 13–17, 1998 Proceedings 25*, pages 820–831. Springer, 1998.

[106] Gui Lu Long, Yan Song Li, Wei Lin Zhang, and Li Niu. Phase matching in quantum searching. *Physics Letters A*, 262(1):27–34, 1999.

[107] Gui-Lu Long, Xiao Li, and Yang Sun. Phase matching condition for quantum search with a generalized initial state. *Physics Letters A*, 294(3-4):143–152, 2002.

[108] Panchi Li and Shiyong Li. Phase matching in grover's algorithm. *Physics Letters A*, 366(1-2):42–46, 2007.

[109] Eli Biham, Ofer Biham, David Biron, Markus Grassl, and Daniel A Lidar. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. *Physical Review A*, 60(4):2742, 1999.

[110] D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9, March 1997.

[111] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A*, 5(7):1127–1135, Jul 1988.

[112] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2nd edition, 2003.

[113] Marcel Seelbach Benkner, Vladislav Golyanik, Christian Theobalt, and Michael Moeller. Adiabatic quantum graph matching with permutation matrix constraints. In *2020 International Conference on 3D Vision (3DV)*, pages 583–592. IEEE, 2020.

[114] Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Moeller. Q-match: Iterative shape matching via quantum annealing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7586–7596, 2021.

[115] Tolga Birdal, Vladislav Golyanik, Christian Theobalt, and Leonidas J Guibas. Quantum permutation synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13122–13133, 2021.

[116] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13:119–152, 1994.

[117] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992.

[118] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *PAMI*, 32(12):2262–2275, 2010.

[119] Jean H. Gallier and Jocelyn Quaintance. Lie groups, lie algebra, exponential map. In *Differential Geometry and Lie Groups*, pages 537–570. Springer, 2019.

[120] Pei Yean Lee. Geometric optimization for computer vision. *Ph.D. thesis. Australian National University*, Apr 2005.

[121] Shaoyi Du, Nanning Zheng, Shihui Ying, and Jianyi Liu. Affine iterative closest point algorithm for point set registration. *Pattern Recognition Letters*, 31(9):791–799, 2010.

[122] Michael L. Rogers and Robert L. Singleton. Floating-point calculations on a quantum annealer: Division and matrix inversion. *Frontiers in Physics*, 8:265, 2020.

[123] Giovani G. Pollachini, Juan P. L. C. Salazar, Caio B. D. Góes, Thiago O. Maciel, and Eduardo I. Duzzioni. Hybrid classical-quantum approach to solve the heat equation using quantum annealers. *Phys. Rev. A*, 104:032426, Sep 2021.

[124] Paul I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, Elena Tolkacheva, Fabio Altomare, Andrew J. Berkley, Richard Harris, Jeremy P. Hilton, Trevor Lanting, Anthony J. Przybysz, and Jed Whittaker. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE TAS*, 24(4):1–10, 2014.

[125] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. Next-generation topology of d-wave quantum processors, 2020.

[126] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

[127] Charles Fox. *An introduction to the calculus of variations*. Courier Corporation, 1987.

[128] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: theory and insights. *Advances in neural information processing systems*, 27, 2014.

[129] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the poisson equation. *New Journal of Physics*, 15(1):013021, 2013.

[130] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

Natacha **Kuete Meli**

Curriculum vitae

Objective

Dedicated and results-driven research scientist. Seeking to contribute to the research in the field of quantum computing, computer vision and image processing.

Profile

Name: Natacha Kuete Meli
Gender: Female
Nationality: Cameroon
Date of birth: Jan 1st, 1996
Place of birth: Galim, Cameroon
Marital status: Single

Contact

Tel: +49 176 74908938
E-mail:
natachakuete@gmail.com
Address: Kleine Klosterkoppel 11, 23562 Luebeck, Germany

Language skills

French
C2
Mother tongue

English
Good

German
C1

Degrees

2021–2024
PhD in computational life science
University of Luebeck · Germany
Thesis: Quantum Algorithms for Binary Problems with Applications to Image Processing.
Supervisor: Prof. Dr. Jan Lellmann.
Variational quantum circuits • QUBO/Ising models • Quantum annealing.

2019–2021
Master in computational life science
University of Luebeck · Germany
Thesis: Trainable Detection Methods for Industrial Materials Testing.
Supervisor: Prof. Dr. Jan Lellmann.
Object detection • image segmentation • deep learning • R-CNN • U-Net.

2012–2015
Bachelor in mathematics and computer science
University of Dschang · Cameroon
Thesis: Backend Programming of a real estate web application in Java EE.
Supervisor: Prof. Maurice Tchoupe Tchendji.
Jave EE • HTML • PHP • UML.

Competencies

Quantum computing

- **Adiabatic quantum computing**
  Modeling and discrete optimization • QUBO/Ising models • Quantum annealing.
- **Universal quantum computing**
  Quantum circuits • variational quantum circuits • hybrid quantum algorithms • approximate quantum algorithms.

Image processing

- **Classical image processing**
  Intensity transformations • morphological operations • principal component analysis • image reconstruction.
- **Image segmentation**
  Active contour models • statistical shape models • atlas-based segmentation.
- **Image registration**
  Parametric image registration • non-parametric image registration.

Machine learning

Image classification • object detection • image segmentation.

Mathematics

Linear algebra • analysis • numerics • optimization • signal processing • image processing • graph theory • approximation theory • relativity theory • test and estimation theory • stochastic processes.

Research Interest

I specialize in quantum computing's application to image processing. This involves developing quantum algorithms for a range of image processing tasks, both combinatorial and non-combinatorial. Quantum computing offers a powerful solution to overcome challenges posed by non-convex energies, limited storage, and high computational demands, promising more accurate and cost-effective solutions for real-world image processing problems.

Programming skills

MATLAB
PYTHON
C++
JAVA
LATEX

Quantum programming

D-WAVE LEAP
QISKIT

TEACHING ACTIVITIES / SCIENTIFIC ASSISTANT

UNIVERSITY OF LUEBECK · Germany
Responsible for creating exercise sheets and conducting student assessments to complement lectures in the following subjects:

Quantum image processing	Winter terms 2021, 2022 and 2023
Optimization	Summer terms 2022 and 2023
Linear algebra	Summer term 2021 and winter term 2023
Seminar Numerical Optimization	Winter terms 2021 and 2022
Mathematical methods of image processing	Winter term 2021
Computer graphics	Summer term 2020

UNIVERSITY OF LUEBECK · Germany
Responsible for database maintenance, organizing, and overseeing Erasmus applications as a scientific assistant at the International Office of the University of Luebeck.
Winter term 2019 - summer term 2021

REVIEWING ACTIVITIES

Participated in the following events and conferences as a reviewer:

QDSM 2023	International workshop on quantum data science and management
SSVM 2023	International conference on scale space and variational methods in computer vision
CVPR 2023	IEEE/CVF conference on computer vision and pattern recognition
BVM 2021	Bilbverarbeitung für die Medizin

CERTIFICATES

Continuously advancing my knowledge through online courses, I have earned the following certificates:

Sep. 2023
Qiskit global summer school 2023 - Quantum excellence
IBM · Online
For exploring the world of quantum computing through physics, math, and python via Qiskit, to bridge the gap between quantum theory and real-world implementation.
Quantum teleportation • iterative phase estimation • qiskit runtime.

Aug. 2023
QBronze
QWORLD · Online
For successfully completing the online training in Quantum Computing & Programming using QWorld's introductory tutorial Bronze-Qiskit in the Womanium Global Quantum Program.
Quantum entanglement • quantum search • quantum annealing.

Mai 2023
IBM Quantum Challenge: Spring 2023
IBM · Online
For understanding of how to create circuits that perform measurements mid-circuit and dynamically decide what the next steps should be.
Dynamic circuits • phase estimation • Quantum teleportation • error correction.

CONFERENCES

Presented research findings at the folowing conferences:

MIA 2023
Mathematics and Image Analysis
BERLIN · Germany
Poster: Iterative quantum transformation estimation.

CVPR 2022
Computer vision and pattern recognition
NEW ORLEANS · USA
Poster: An iterative quantum approach for transformation estimation from point sets.

Research skills

Oral presentation
Literature review
Academic writing
Team work

Connect

Google scholar
• ResearchGate • ORCID
• GitHub

Students

Supervised final thesis of the following students:

Nov. 2023
Josephine Elisabeth Oettinger (B.Sc.)
University of Luebeck · Germany
Thesis: Non-Boolean quantum amplitude amplification for discret optimization.
Quantum search • Ising model • discrete optimization.

Volonteer

Worked on a voluntary basis on the following projects:

2022-2024
Mentoring-programm CyberMentor
Germany · Online
Provided online mentoring to an 8th-grade students while actively promoting science to schoolgirls. Served as a STEM role model and supported STEM projects that focused on understanding online shops, mandelbrot sets and digital images.

Peer-reviewed publications

Authored or co-authored the following peer-reviewed publications:

- **Kuete Meli, N.**, Mannel, F. & Lellmann, J. A universal quantum algorithm for weighted maximum cut and Ising problems. Quantum Inf Process 22, 279 (2023). https://doi.org/10.1007/s11128-023-04025-x.
- **Kuete Meli, N.**, Mannel, F. & Lellmann, J. An iterative quantum approach for transformation estimation from point sets. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 529-537 (2022). Paper url.

Hobbys

Reading • Sewing clothes • Jogging • Watching documentaries about the universe

Natacha **Kuete Meli** University of Lübeck Germany +49 176 74908938
natachakuete@gmail.com