



Technical University of Munich
School of Natural Sciences
Department of Physics

Scalability of Variational Quantum Optimization in the Presence of Noise

Adelina Bärligea

Thesis submitted within the Master's Program
Applied and Engineering Physics

December 14, 2024

In collaboration with the
Fraunhofer Institute for Cognitive Systems





Technische Universität München
School of Natural Sciences
Fakultät für Physik

Skalierbarkeit von variationeller Quantenoptimierung unter dem Einfluss von Rauschen

Abschlussarbeit im Masterstudiengang
Applied and Engineering Physics

14. Dezember 2024

Erstgutachter: Prof. Dr. Frank Pollmann

Zweitgutachterin: Dr. habil. Jeanette Miriam Lorenz

©2024 – ADELINA BÄRLIGEA
All rights reserved.

Adelina Bärligea

Scalability of Variational Quantum Optimization in the Presence of Noise

ABSTRACT

Recent advancements in quantum hardware development have marked the beginning of the Noisy Intermediate-Scale Quantum (NISQ) era, characterized by devices with a limited number of qubits and significant noise. Variational Quantum Algorithms (VQAs) have emerged as potential candidates for achieving quantum advantage in this noisy era, particularly in the highly anticipated field of optimization. These hybrid algorithms combine classical optimization with shallow, parameterized quantum circuits to minimize a loss function. Despite their promise, VQAs face significant challenges often overshadowed by results based on small-scale toy problems and narrow testing setups, leaving their practicality for real-world applications uncertain. This thesis provides a systematic investigation of the scalability of VQAs in solving combinatorial optimization problems. Through a comprehensive empirical study, it evaluates the performance of state-of-the-art classical optimizers in minimizing well-behaved VQA loss functions on random Quadratic Unconstrained Binary Optimization (QUBO) problem instances. The study considers varying levels of Gaussian noise and increasing system sizes, with an experimental setup thoroughly motivated throughout the work. The results demonstrate that, for fixed system sizes, the ability of all tested optimizers to find optimal or near-optimal solutions declines sharply once a critical noise threshold is surpassed. Notably, this threshold decreases on an exponential scale with increasing system sizes, far exceeding the impact expected from the well-documented barren plateau phenomenon. When translated into practical terms, such as error rates or measurement shot requirements, the findings indicate prohibitively high resource demands for solving problems of practical relevance, even under the assumption of fault-tolerant hardware. These demands starkly contrast with the efficiency of best-in-class classical solvers. This casts serious doubt on the viability of achieving practical quantum advantage in optimization with the variational approach. It, therefore, highlights the need to explore fundamentally different algorithms and problem domains to realize near-term quantum utility. The methodology presented in this thesis serves as a versatile framework for evaluating the scalability and feasibility of any quantum optimization algorithm and contributes to informed assessments of the limitations and potential of quantum computing in general.

Contents

1	Introduction	1
2	Theoretical Background	3
2.1	Quantum Computing	3
2.1.1	Working Principles	3
2.1.2	NISQ Hardware	5
2.1.3	Quantum Advantage	6
2.2	Combinatorial Optimization	7
2.2.1	QUBO and Ising Problems	7
2.2.2	Example: Traveling Salesperson Problem	8
2.3	Variational Quantum Algorithms	9
2.3.1	Theory	10
2.3.2	Prominent Examples: VQE and QAOA	11
2.3.3	Recent Proposal: BENQO	12
2.3.4	Parameter Optimization	14
2.3.5	Limitations and Challenges	16
3	Methodology	18
3.1	Problem Instances	20
3.2	Statistical Measures	20
3.3	Benchmark Metrics	22
4	Quantum-Inherent Challenges and Errors	25
4.1	Circuit Resources	25
4.2	Hardware Noise	28
4.3	Finite Sampling Error	31
4.4	Loss Landscapes	34
4.5	Preliminary Conclusions	38
5	Limitations of Classical Optimizer Performance	40
5.1	Related Work	40
5.2	Experimental Setup	42
5.3	Scaling of Solvability under Additive Gaussian Noise	44
5.4	Scaling of Solvability with System Size	49
5.5	Implications and Discussion	55
5.5.1	Hardware Development	55
5.5.2	The Curse of Finite Sampling	57
5.5.3	Relevant Classical Problems	60

Contents

6	Conclusions	62
7	Outlook	65
A	Supplementary Material for Chapter 2	67
A.1	Mapping QUBO to Ising Problems	67
A.2	Derivation of BENQO's Operator Representation	68
A.3	Algorithmic Details of Selected Optimizers	68
B	Supplementary Material for Chapter 4	70
B.1	IBM Quantum Backends	70
B.2	Distribution of Modelled Hardware Noise	70
B.3	Derivations for Statistical Error	71
C	Supplementary Material for Chapter 5	73
C.1	Local Qiskit Optimizers	73
C.2	Derivation of the Lower Confidence Limit	74
C.3	Ranking of Optimizers	75
C.4	Investigation of Decay Type via Fitting Experiment	76
	Bibliography	78
	Abbreviations	92
	List of Figures	92
	List of Tables	93
	Acknowledgements	95

Chapter 1

Introduction

Since its initial proposal [1], quantum computing has attracted considerable attention as a promising alternative for tackling computationally hard problems that are practically unsolvable with classical computers. This hope has been fueled by advancements on the algorithmic front, most famously with Shor’s algorithm for integer factorization [2], which provides exponential speed-up over classical methods and might endanger most of today’s cryptography. Other impactful quantum algorithms include Grover’s database search [3] with quadratic speed-up, quantum simulation techniques [4], and a quantum solver for linear systems of equations [5]. Alongside these theoretical advancements, considerable progress has been made in the physical realization of quantum computers using superconducting circuits [6], trapped ions [7], neutral atom systems [8], and photonic processors [9].

Despite these advancements, current quantum systems remain in their early stages, with yet a long way to go before scalable, fault-tolerant quantum computing becomes a reality. Preskill [10] coined the term Noisy Intermediate-Scale Quantum (NISQ) era to describe these early-stage devices, expressing cautious optimism about their potential to achieve practically relevant quantum advantage. Among the most anticipated applications, and arguably one of the most industrially relevant, is optimization, for which much of the hope lies in quantum approximation algorithms and heuristics [11]. A particularly interesting framework within this class are Variational Quantum Algorithms (VQAs) [12–14], which are often regarded as the leading candidates to provide near-term quantum advantage for practical applications.

VQAs are hybrid schemes that delegate the main computational workload of optimizing parameters to the classical processor, while the quantum device focuses on evaluating the loss function using shallow, parametrized quantum circuits (the ansatz). This hybrid architecture makes them especially suited for near-term hardware; some researchers even describe them as noise-resilient [13, 15, 16].

However, in light of widespread optimism and high expectations, it is equally important to rigorously assess and understand the limitations of such approaches. Establishing the boundaries between genuinely promising advancements and appealing but unproven claims is a crucial step for the field’s near-term progress.

Hoeffler et al. [17] has pointed out that most proposed NISQ algorithms offer, at best, polynomial speed-up, with crossover times to the best classical algorithms potentially stretching to days, months, or even years. Even with theoretically super-quadratic speed-ups, achieving practical performance gains may remain out of reach given current and

foreseeable quantum hardware limitations [18]. A growing body of research underscores the detrimental effects of noise [19–24], as well as other fundamental challenges [25–27] on these algorithms’ viability. This may help explain why, to date, no quantum advantage has been demonstrated for practically meaningful optimization problems with VQAs on near-term devices. Thus, a critical question remains: is quantum practicality at all possible, given the unavoidable presence of stochastic errors?

Rather than directly addressing this question, such as in the complexity-theoretic approach by Chen et al. [28], new heuristic quantum algorithms continue to emerge, often labeled “promising” based on limited testing with small toy problems and idealized, noise-free simulations. Such setups, however, often overestimate an algorithm’s true potential. As a result, it largely remains unclear whether these schemes can be scaled up effectively or still perform reliably in the presence of realistic noise; and how to even tackle these problems.

This thesis addresses the question of how promising VQAs truly are for solving practically relevant optimization problems. To explore this, it examines the scaling behavior of the classical component within the variational pipeline – namely, parameter optimizers – when applied to noisy loss functions across increasing system sizes. This investigation takes an empirical approach¹, providing experimental results that enhance the understanding and clearly demonstrate the practical limitations of VQAs. By implementing a scalable Gaussian error model on a well-behaved loss function of a representative VQA ansatz for typical optimization problems and applying this to a suite of state-of-the-art optimizers, this study conducts systematic experiments to analyze the scaling behavior of problem solvability across two dimensions: error level and system size. This allows for the identification of a boundary line beyond which problems become unsolvable, which holds significant implications for the general practicality of VQAs for solving real-world problems even beyond the constraints of near-term devices.

Therefore, this thesis’s contributions extend beyond a detailed study of the scaling behavior and limitations of VQAs, providing a robust and generalizable methodology for evaluating the potential and viability of quantum optimization algorithms in general.

To this end, the thesis is structured as follows: Chapter 2 provides the theoretical background for the relevant topics, including the basics of quantum computing, the formulation of combinatorial optimization problems, and a more comprehensive introduction to the VQA framework. Chapter 3 describes the methodology used in the numerical experiments and defines key measures for evaluation. In Chapter 4, the sources of errors and other challenges arising in VQAs are discussed, and their impact is experimentally analyzed across three representative algorithms. Chapter 5 presents the main findings of this work, including not only the results of systematic optimizer studies but also a thorough discussion of their broader implications. Finally, Chapter 6 summarizes these insights, and Chapter 7 suggests interesting directions for future research.

¹A theoretical counterpart to this work, similarly addressing the capabilities of classical optimizers, was recently published by Kungurtsev et al. [29], offering insights from a more analytical perspective.

Chapter 2

Theoretical Background

The following chapter provides a summary of essential background from the literature necessary for understanding the experiments and results presented in Chapters 4 and 5. It begins with an introduction to quantum computing, covering foundational concepts, current device capabilities, and definitions of quantum advantage (see Section 2.1). Next, it introduces combinatorial optimization problems – a well-established and promising application area for quantum computing (see Section 2.2). Finally, variational quantum algorithms, a leading approach for addressing these problems, are explained and discussed in detail (see Section 2.3).

2.1 Quantum Computing

Quantum computers harness the principles of quantum mechanics to manipulate quantum states, enabling them to potentially solve certain problems far more efficiently than classical computers [30]. One of the first practical demonstrations of this potential was Shor’s integer factoring algorithm [2], which is super-polynomially faster than any known classical algorithm, for which the problem was generally considered intractable. The promise of quantum computing is partly rooted in its ability to explore exponentially large state spaces that are fundamentally beyond classical simulation capabilities. To provide a foundation for understanding these concepts, this section provides an overview of the basic working principles and current topics of quantum computing, with more detailed explanations provided by Nielsen and Chuang [31].

2.1.1 Working Principles

Quantum computers are composed of qubits (i.e., quantum bits), represented by the quantum state

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad \text{with} \quad \langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1, \quad (2.1)$$

which generally defines the superposition of the two basis states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ of the two-dimensional Hilbert space. Using the normalization condition in Equation (2.1) and ignoring global phases, one can find ϑ and φ such that $\alpha = \cos \frac{\vartheta}{2}$ and $\beta = e^{i\varphi} \sin \frac{\vartheta}{2}$ represent the qubit state by a vector in the Bloch sphere (unit sphere spanned by ϑ, φ). The time evolution of a quantum state is governed by gates, represented by unitary matrices $\mathbf{U} \in \mathbb{C}^{n \times n}$ (with $\mathbf{U}^\dagger \mathbf{U} = \mathbf{U} \mathbf{U}^\dagger = \mathbf{I}$). The most elementary

single-qubit operations are the Pauli gates

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (2.2)$$

which anti-commute, i.e., $[\mathbf{X}, \mathbf{Z}] = -[\mathbf{Z}, \mathbf{X}]$. Another relevant single-qubit gate realizes a rotation about an axis $\vec{v} \in \mathbb{R}^3$ (with $\|\vec{v}\| = 1$) by any angle ϑ ,

$$\mathbf{R}_{\vec{v}}(\vartheta) = \exp\left(-i \frac{\vartheta}{2} \vec{v} \cdot \vec{\sigma}\right) = \cos\left(\frac{\vartheta}{2}\right) \mathbf{I} - i \sin\left(\frac{\vartheta}{2}\right) \vec{v} \cdot \vec{\sigma}, \quad (2.3)$$

with $\vec{\sigma} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})^\top$. For a universal set of quantum gates¹, a multi-qubit gate is needed. The most important two-qubit gate in quantum computing is the controlled-not (CNOT) gate

$$\mathbf{CX} |a\rangle \otimes |b\rangle = (|0\rangle \langle 0| \otimes \mathbf{I} + |1\rangle \langle 1| \otimes \mathbf{X}) |a\rangle \otimes |b\rangle, \quad (2.4)$$

which flips the target qubit $|b\rangle$ if and only if the control qubit $|a\rangle$ is in state $|1\rangle$. This, or other controlled operations², can be used to create entangled states, such as the Bell state $|\psi_{\text{Bell}}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. To graphically represent the evolution of an initial state $|\Psi\rangle$ (usually $\bigotimes_{i=1}^n |0\rangle$) defined by some $n \times n$ unitary \mathbf{U} (i.e., a quantum algorithm), circuit diagrams are used. Table 2.1 provides the circuit and matrix representation of a selection of practically relevant gates.

Finally, to retrieve any information from a quantum circuit, measurements must be conducted. A measurement is defined by a Hermitian operator \mathbf{M}_m , satisfying completeness $\sum_m \mathbf{M}_m^\dagger \mathbf{M}_m = \mathbf{I}$ with respect to all outcomes m . The probability for outcome m is defined as

$$p(m) = \langle \psi | \mathbf{M}_m^\dagger \mathbf{M}_m | \psi \rangle = \|\mathbf{M}_m | \psi \rangle\|^2, \quad (2.5)$$

leaving the state after measurement as $|\psi'\rangle = \mathbf{M}_m | \psi \rangle / \|\mathbf{M}_m | \psi \rangle\|$. For example, applying the measurement operators $\mathbf{M}_0 = |0\rangle \langle 0|$ and $\mathbf{M}_1 = |1\rangle \langle 1|$ to the single-qubit state (2.1) gives a probability of $|\alpha|^2$ to measure “0”, and $|\beta|^2$ to measure “1”, leaving the system in state $|0\rangle$ or $|1\rangle$. Lastly, note that any quantum system may also be described by a Hermitian, positive semidefinite density operator ρ with unit trace, most generally defined as

$$\rho = \sum_i p_i |\Psi_i\rangle \langle \Psi_i|, \quad (2.6)$$

representing an ensemble of quantum states $\{p_i, |\Psi_i\rangle\}$. After applying a unitary \mathbf{U} to the system, the corresponding density operator becomes $\rho' = \mathbf{U} \rho \mathbf{U}^\dagger$.

¹This means that any unitary operation (i.e., a quantum circuit) can be expressed as a finite sequence of gates from this universal set.

²The two-qubit entangling gate, native to many of IBM’s quantum processors, is the ECR gate:

$$\mathbf{U}_{\text{ECR}} = \frac{1}{\sqrt{2}}(\mathbf{I} \otimes \mathbf{X} - \mathbf{X} \otimes \mathbf{Y}) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 & i \\ 1 & 0 & -i & 0 \\ 0 & i & 0 & 1 \\ -i & 0 & 1 & 0 \end{pmatrix}.$$

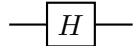
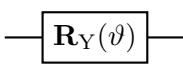
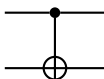
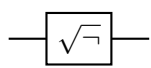
Name	Circuit Diagram	Matrix Representation
Hadamard gate		$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
RY gate		$\mathbf{R}_Y(\vartheta) = \begin{pmatrix} \cos \frac{\vartheta}{2} & -\sin \frac{\vartheta}{2} \\ \sin \frac{\vartheta}{2} & \cos \frac{\vartheta}{2} \end{pmatrix}$
CNOT gate		$\mathbf{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
SX gate		$\sqrt{\mathbf{X}} = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$

Table 2.1: Circuit Diagrams and Matrix Representation of Selected Quantum Gates. From top to bottom: the Hadamard gate changes the basis of a qubit from $|0\rangle$, $|1\rangle$ to $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$, which are called equal superposition states. The RY gate is often used when preparing parametrized quantum ansatzes (see Section 2.3.2), as it rotates single qubits about the y-axis by any given angle ϑ (see Equation (2.3)). The CNOT gate is used for entangling two qubits, with a dot marking the control and a cross marking the target qubit. The SX gate realizes the square root of \mathbf{X} , which is often included in the native gate sets of IBM quantum hardware (see Appendix B.1).

2.1.2 NISQ Hardware

After the original proposal of quantum computers by Feynman [1], their physical realization remained elusive for many years, leaving them as theoretical constructs. Today, we have entered what Preskill [10] termed the Noisy Intermediate-Scale Quantum (NISQ) era of quantum computing. This era is characterized by the development of small quantum computing devices, particularly those based on trapped ions and superconducting circuits, each reaching scales in the order of approximately 100 qubits. Preskill’s term “NISQ”, however, not only highlights the limited qubit counts of these early devices but, more critically, draws attention to the inherent noise in current quantum systems, which causes imperfect computations.

One of the most significant types of noise is caused by the depolarizing error, mathematically described by the channel³

$$\mathcal{E}(\rho) = (1 - \lambda)\rho + \lambda \operatorname{tr}(\rho) \frac{\mathbf{I}}{2^n} \quad (2.7)$$

for a system of n qubits, where $\lambda \in [0, 1]$ is the error parameter. For $\lambda = 1$, the depolarization turns the system into a totally mixed state. Furthermore, there is the amplitude damping channel

$$\mathcal{E}(\rho) = \mathbf{E}_0 \rho \mathbf{E}_0^\dagger + \mathbf{E}_1 \rho \mathbf{E}_1^\dagger \quad (2.8)$$

³Quantum channels are linear maps that are both trace-preserving and completely positive, used to describe the physical processes affecting a quantum system, including its evolution and interactions with the (noisy) environment.

with the Kraus operators $\mathbf{E}_0 = |0\rangle\langle 0| + \sqrt{1 - \lambda_{\text{AD}}}|1\rangle\langle 1|$, corresponding to an absence of decay, and $\mathbf{E}_1 = \sqrt{\lambda_{\text{AD}}}|0\rangle\langle 1|$, corresponding to the decay event. The error parameter $\lambda_{\text{AD}} \in [0, 1]$ again adjusts the strength of the damping with $\lambda_{\text{AD}} = 1$ turning the single-qubit system into its ground state $|0\rangle$. The phase damping channel has the same form as Equation (2.8), but with the Kraus operators $\mathbf{E}_0 = |0\rangle\langle 0| + \sqrt{1 - \lambda_{\text{PD}}}|1\rangle\langle 1|$, and $\mathbf{E}_1 = \sqrt{\lambda_{\text{PD}}}|1\rangle\langle 1|$. The thermal relaxation error in QISKIT [32] – IBM’s software package for simulating and running quantum circuits – combines these two channels, by relating the parameters,

$$\lambda_{\text{AD}} = 1 - e^{-\frac{t}{T_1}} \quad \text{and} \quad \lambda_{\text{PD}} = 1 - e^{-\frac{2T_1 + T_2}{T_1 T_2} t}, \quad (2.9)$$

to the hardware-specific relaxation times T_1 (thermal decoherence time) and T_2 (dephasing time). Generally, superconducting qubits exhibit shorter T_1 and T_2 times, which, while limiting coherence, enable faster gate operations. In contrast, ion traps benefit from significantly longer relaxation times but also have longer gate operation times.

Another critical error source in quantum computation is the readout error, which introduces a measurement inaccuracy with probability λ . This is caused by the fact that measuring a qubit takes significantly longer than performing a unitary operation on it, leaving a non-negligible probability of the state changing mid-measurement due to the abovementioned errors. An overview of typical values for all error parameters on currently available IBM quantum backends is provided in Appendix B.1.

2.1.3 Quantum Advantage

A central objective in quantum algorithm design is to attain a speed-up over the most advanced classical methods, often viewed as the core of quantum advantage. However, achieving computational speedups that are practically relevant may be challenging, as these advantages often manifest in impractically large system sizes or long runtimes [17]. As a result, alternative definitions of quantum advantage have emerged, which are well summarized by Ezzratty [18]:

- **space** – if the storage space of an n -qubit register, scaling as 2^n , exceeds the memory capacity of classical computers
- **quality** – if the quality of a solution provided by a quantum algorithm surpasses that of the classical counterpart by any chosen metric
- **energy** – if the fully-burdened quantum computation consumes less total energy than the best classical equivalent for the same problem
- **cost** – if the total cost of the quantum solution is lower than that of the best-in-class classical solution

As Ezzratty [18] suspects, real business benefits from quantum computing must come from a favorable balance across all these factors. However, as Grumbling and Horowitz

[33] point out, no application of commercial interest has yet demonstrated a clear advantage using the near-term systems outlined in Section 2.1.2. Consequently, current efforts are placed on both the hardware side, aiming to scale up qubit numbers and reduce error rates, and on the software side, developing algorithms and identifying applications that may offer such benefits.

2.2 Combinatorial Optimization

One of the most anticipated applications of quantum computing is optimization. This is because optimization problems are prevalent across various sectors such as finance, logistics, and life sciences, such that any improvement in state-of-the-art solutions could have a great impact [11]. In fact, in 2023, McKinsey & Company [34] projected that this impact would be disruptive in the long term for some of these fields. This optimism is driven by the hope that quantum computers, with their ability to explore exponentially large state spaces and entanglement capability, will enable some sort of quantum advantage, as described in Section 2.1.3. The focus is, therefore, often on combinatorial optimization problems [35], which are inherently difficult due to the exponential “explosion” of possible solutions with respect to problem size, classifying many as NP-hard⁴ according to complexity theory [36]. It is, however, important to recognize that these optimization problems are inherently classical: While quantum problems involve finding the ground state of a quantum mechanical system, many classical problems can be mapped onto an Ising Hamiltonian [37], where the ground state corresponds to a classical bit-string. As a result, the potential for quantum advantage is far more intuitive for the simulation of quantum systems [1, 4], whereas it remains a subject of significant debate for classical optimization problems [17, 19, 38]. Yet, their solvability with quantum devices is extensively studied in the literature [11, 39–42].

2.2.1 QUBO and Ising Problems

The established connection between NP-hard optimization problems and the statistical physics of Ising spin glasses [43] provides a foundation for representing many classically significant yet notoriously challenging optimization problems as an Ising Hamiltonian.

The Ising Hamiltonian of an $n \in \mathbb{N}$ spin particle system is described by the $2^n \times 2^n$ Hermitian operator

$$\mathbf{C} = \sum_{i=1}^n C_{ii} \mathbf{Z}_i + \sum_{i < j} C_{ij} \mathbf{Z}_i \mathbf{Z}_j. \quad (2.10)$$

Here, \mathbf{Z}_i denotes the Pauli- \mathbf{Z} operator with eigenvalues ± 1 acting on the i^{th} particle, and C_{ij} corresponds to the interaction energy between particles i and j . If the system is in the basis state $|\mathbf{q}\rangle = \bigotimes_{i=1}^n |q_i\rangle$ with $q_i \in \{0, 1\}$, the total energy $\langle \mathbf{C} \rangle := \langle \mathbf{q} | \mathbf{C} | \mathbf{q} \rangle$ is

⁴Note that while it is not widely expected that quantum computers will be able to solve NP-hard problem instances, some believe that they will be able to find better approximate solutions or find such approximate solutions faster [10, 11, 14].

given by

$$\langle \mathbf{C} \rangle = \sum_{i=1}^n (-1)^{q_i} \mathcal{C}_{ii} + \sum_{i < j} (-1)^{q_i + q_j} \mathcal{C}_{ij}. \quad (2.11)$$

The resulting problem of finding the ground state $|\mathbf{q}^*\rangle$ of the system is equivalent to minimizing the expectation value of Equation (2.11) with respect to $|\mathbf{q}\rangle$, which is known as the Ising problem. Note that when setting all self-connections $\mathcal{C}_{ii} = 0$, this corresponds to the weighted Maximum Cut Problem (MaxCut). Lucas [37] demonstrated how many NP-hard combinatorial optimization problems of practical relevance – from partitioning and covering to satisfiability – can be described within this formalism.

At around the same time, a mathematically equivalent formulation gained popularity: the Quadratic Unconstrained Binary Optimization (QUBO) problem. This formulation, like the Ising model, can represent a wide variety of important combinatorial optimization problems [44–47]. The QUBO problem is described by

$$\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{Q} \mathbf{x}, \quad (2.12)$$

where $\mathbf{x} \in \{0, 1\}^n$ are binary decision variables $\mathbf{x} \in \{0, 1\}^n$, and it is fully defined by the matrix \mathbf{Q} . To convert this formulation into its Ising equivalent, the binary variables \mathbf{x} are transformed into spin variables \mathbf{z} , resulting in the following cost operator:

$$\mathbf{C} = \sum_{i < j} \frac{Q_{ij}}{2} \mathbf{z}_i \mathbf{z}_j + \sum_{i=1}^n \left(\sum_{j=1}^n \frac{Q_{ij}}{2} \right) \mathbf{z}_i + \text{const.} \quad (2.13)$$

The constant offset can be disregarded during a minimization process. A full derivation of this mapping is provided in Appendix A.1.

The equivalence between the QUBO and Ising models is why QUBO has become a key focus in quantum optimization research and experimentation, particularly with quantum annealers [48, 49], such as those implemented by D-Wave systems.

2.2.2 Example: Traveling Salesperson Problem

The Traveling Salesperson Problem (TSP) is a famous NP-hard problem in combinatorial optimization [50], with numerous practically relevant applications even beyond logistics and routing. Despite its extensive study, classical approximation algorithms have yet to meet the problem’s theoretical inapproximability bounds⁵, leaving hope for quantum algorithms to close the gap [11]. But even besides these complexity-theoretic considerations, the TSP and its variants have become a paradigmatic benchmark for quantum computing [56–61].

⁵This specifically refers to the metric TSP, which belongs to a class of problems with polynomial-time approximation schemes (PTAS) [36, 51]. These are polynomial-time algorithms that can find an approximate solution within a factor of $(1 + \epsilon)$ of the optimum. Inapproximability bounds are theoretically derived limits of how closely this solution can approximate the optimal solution, as established for the metric TSP [52]. Classical approximation algorithms have yet to reach these bounds [53, 54]. For more background on the TSP, see Korte and Vygen [35], and for more details on complexity theory, refer to Ausiello et al. [36], Michael R. Garey, David S. Johnson [55].

The TSP is typically defined on a fully connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of n nodes, and \mathcal{E} refers to the set of weighted edges, each associated with a “distance” ω_{ij} between the i^{th} and j^{th} nodes. The objective is to find a Hamiltonian cycle with the minimum total distance, visiting each node exactly once. When fixing the starting position, the number of possible distinct cycles is therefore given by $\frac{(n-1)!}{2}$, which grows worse than exponentially with the problem size n .

An established QUBO formulation for this problem proposed by Lucas [37] is

$$C = \sum_{i,j=1}^n \omega_{ij} \sum_{\alpha=1}^n x_i^\alpha x_j^{\alpha+1} + P \sum_{i=1}^n \left(1 - \sum_{\alpha=1}^n x_i^\alpha\right)^2 + P \sum_{\alpha=1}^n \left(1 - \sum_{i=1}^n x_i^\alpha\right)^2, \quad (2.14)$$

where the paths are described by n^2 binary decision variables x_i^α , marking whether node i is visited at time step α . This corresponds to a one-hot encoding of the equivalent integer problem, where the two permutational constraints

- $\sum_i x_i^\alpha = 1 \quad \forall \alpha = 1, \dots, n$
(at each step, only one node can be visited)
- $\sum_\alpha x_i^\alpha = 1 \quad \forall i = 1, \dots, n$
(each node should be visited only once in the cycle)

are included as quadratic penalty terms with the penalty factor P , as suggested by Glover et al. [46]. The length of the chosen path is calculated in the first term of Equation (2.14). To ensure feasible solutions, P must be larger than a minimum threshold P_{\min} , which is known to be lower-bounded by $\max(\omega_{ij})$ [37]. A suitable choice for P can follow QISKIT’s default strategy, which is based on the absolute value range of the unconstrained quadratic cost function in Equation (2.12). Since the product of any two binary variables x_i^α in the first term of Equation (2.14) ranges between 0 and 1, the penalty factor can be chosen as $P = n \cdot \sum_{i,j} \omega_{ij}$. Moreover, note that the TSP can be reduced to $(n-1)^2$ variables by fixing the starting point of each cycle. This is done by setting $x_1^1 = 1$, $x_1^\alpha = 0$ for all $\alpha \neq 1$, and $x_i^i = 0$ for all $i \neq 1$, thereby eliminating the cyclic permutation symmetry inherent in the problem.

2.3 Variational Quantum Algorithms

Until fault-tolerant quantum computing becomes a reality, variational quantum algorithms (VQAs) have emerged as the leading heuristic for tackling computationally hard problems in the near term, such as combinatorial optimization (see Section 2.2), as well as various other applications [12–14]. In the rapidly growing body of literature, VQAs are predominantly described as the most promising candidates for achieving real-world quantum advantage. Despite these optimistic projections, no concrete evidence of any advantage for solving practically useful problems has been demonstrated yet.

2.3.1 Theory

VQAs are hybrid methods relying on both quantum and classical computing. The classical computers handle the optimization process, while the quantum computers evaluate the loss function – a task assumed to be inefficient on classical machines [13]. The quantum component consists of a parameterized quantum circuit that prepares a candidate state $|\Psi(\theta)\rangle$. Given a Hermitian operator \mathbf{C} , the quantum computer calculates the expectation value $\langle \mathbf{C} \rangle$ with respect to the prepared state. The corresponding classical minimization problem is formulated as:

$$\min_{\theta} \mathcal{L}(\theta) \quad \text{with} \quad \mathcal{L}(\theta) := \langle \Psi(\theta) | \mathbf{C} | \Psi(\theta) \rangle. \quad (2.15)$$

For the optimal parameters θ^* , the state $|\Psi(\theta^*)\rangle$ approximates the true ground state $|\mathbf{q}^*\rangle$. This is based on the variational principle, which asserts that the expectation value $\langle \psi | \mathbf{C} | \psi \rangle$ of a given Hamiltonian \mathbf{C} for an arbitrary state $|\psi\rangle$ is lower bounded by the ground state energy E_0 of the corresponding system [12]. The typical workflow of VQAs is depicted in Figure 2.1. This variational paradigm offers a convenient way to transform discrete problems — where a single change in the input parameters can lead to a dramatically different energy — into smoother problems with continuous loss functions.

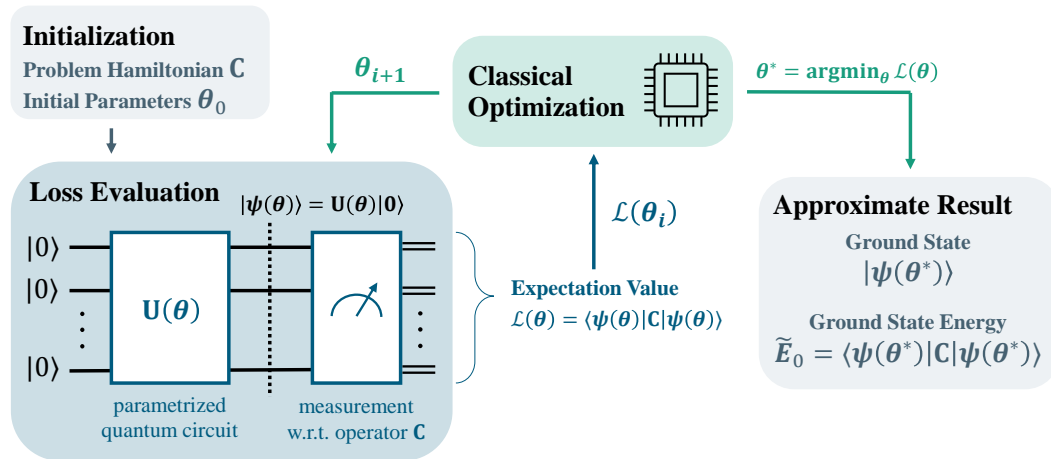


Figure 2.1: Representative Illustration of the VQA workflow. The algorithm is initialized by the set of parameters θ_0 and a Hamiltonian \mathbf{C} encoding the loss of the optimization problem. The loss value is evaluated using a parametrized quantum circuit $U(\theta)$ (the ansatz) and measurements with respect to the operator \mathbf{C} . This routine becomes the input of a classical optimizer, which iteratively searches for an optimal set of parameters θ^* that minimizes the loss function $\mathcal{L}(\theta)$. Using the final result, one can approximate the ground state and ground state energy of the given problem \mathbf{C} .

2.3.2 Prominent Examples: VQE and QAOA

The two most prominent and extensively studied examples for VQAs are the Variational Quantum Eigensolver (VQE) [62] and the Quantum Approximate Optimization Algorithm (QAOA) [63]. Originally, VQE was introduced for solving quantum chemistry problems, while QAOA was explicitly designed for combinatorial optimization. However, both algorithms are now widely recognized and established for optimization tasks and have been extensively studied in this context [64–67].

The general framework of VQE involves selecting a suitable parametrized ansatz circuit, represented by the unitary $\mathbf{U}(\boldsymbol{\theta})$, which prepares the trial state as $|\Psi(\boldsymbol{\theta})\rangle = \mathbf{U}(\boldsymbol{\theta})|0\rangle$. Various different ansatzes have been proposed in the literature [66], such that a common classification distinguishes between hardware-efficient ansatzes and problem-inspired ones. For instance, a widely adopted ansatz of the first kind are two-local circuits. These typically alternate between rotational layers, where single-qubit gates are applied to all qubits, and entangling layers, where two-qubit gates create entanglement between the qubits, as is illustrated in Figure 2.2.

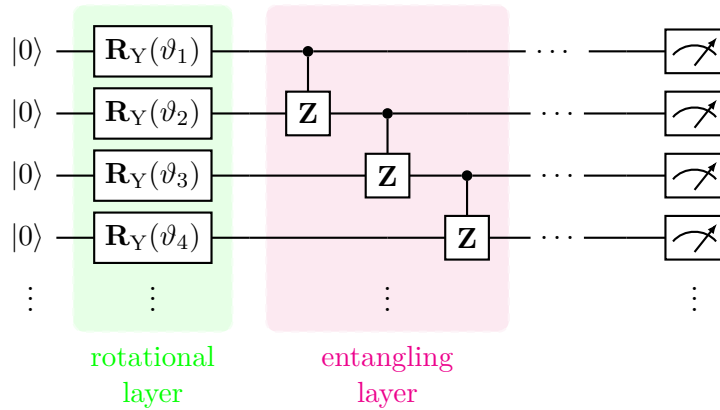


Figure 2.2: Circuit Representation of a Two-Local Ansatz for VQE. All qubits are initialized to $|0\rangle$, followed by an alternating set of layers of single-qubit rotations and entangling operations. The rotational layer consists of $\mathbf{R}_Y(\vartheta)$ gates applied to each qubit, where ϑ represents distinct parameters. The entangling layer features a linear arrangement of controlled- \mathbf{Z} gates, facilitating qubit entanglement. These layers are repeated until a final global measurement is performed to evaluate the circuit’s output.

As a representative example used in the experiments of this thesis, a two-local ansatz consisting of a single layer of \mathbf{R}_Y rotations followed by a linear entanglement layer of controlled- \mathbf{Z} gates was employed. After running the circuit, the qubits are measured with respect to a given Hermitian cost operator to obtain the loss value indicated in Equation (2.15).

The QAOA can be viewed as a specific ansatz in the VQE framework, inspired by Adiabatic Quantum Computing (AQC) [68]. AQC is based on the adiabatic theorem, which states that if a system is initialized in the known ground state $|\Psi_0\rangle$ of a (trial) Hamiltonian \mathbf{H}_M , and is evolved slowly enough towards a problem Hamiltonian \mathbf{H}_P , the

system will remain in its ground state throughout the evolution. After a sufficiently long time T , the final state $|\Psi_T\rangle$ will therefore have a high overlap with the actual ground state of \mathbf{H}_P , representing the solution to the problem.

QAOA implements a discretized (Trotterized) version of this evolution, defining the two unitaries:

- mixing operator⁶: $\mathbf{U}_M(\beta) = e^{-i\beta\mathbf{H}_M}$ with $\mathbf{H}_M = \sum_{i=1}^n \mathbf{X}_i$
- phase shift operator: $\mathbf{U}_P(\gamma) = e^{-i\gamma\mathbf{H}_P}$ with $\mathbf{H}_P = \mathbf{C}$

The resulting ansatz circuit (see Figure 2.3) combines these operators in p layers, parametrized by the $2p$ angles $\boldsymbol{\gamma} = \{\gamma_i\}_{i=1}^p$ and $\boldsymbol{\beta} = \{\beta_i\}_{i=1}^p$, as follows:

$$\mathbf{U}(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \mathbf{U}_M(\beta_p) \mathbf{U}_P(\gamma_p) \dots \mathbf{U}_M(\beta_1) \mathbf{U}_P(\gamma_1) \quad (2.16)$$

It can be shown that for $p \rightarrow \infty$, the final state $\mathbf{U}(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*) |+\rangle$ with optimized parameters $\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*$, becomes the true ground state of the system [63]. However, since such high values of p are computationally infeasible – particularly for the low-depth requirements of NISQ devices, for which this algorithm was designed – research is largely focused on enabling quantum advantage even with a very limited number of layers [67].

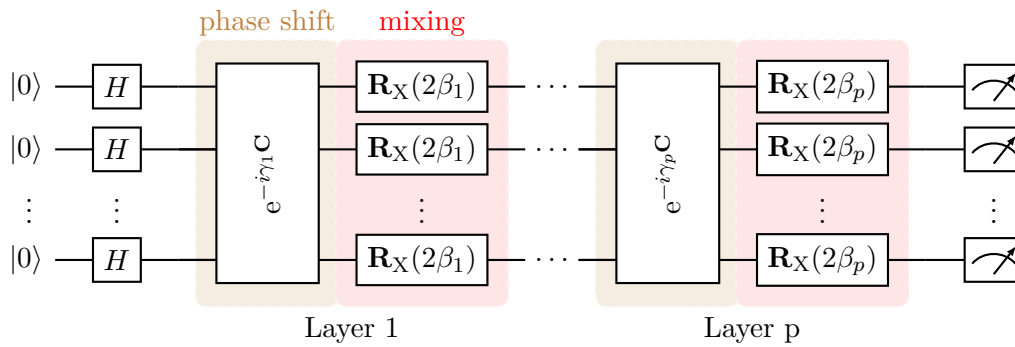


Figure 2.3: Circuit Representation of the QAOA Ansatz. All qubits are first initialized to the equal superposition state $|+\rangle$ using Hadamard gates. Then, p alternating layers of phase shift and mixing operators are applied, as defined in Equation (2.16). Finally, a global measurement is performed to evaluate the circuit’s output.

2.3.3 Recent Proposal: BENQO

Another ansatz was proposed in a recent paper [69], which presented the so-called “universal quantum algorithm” for Ising problems as a new hybrid quantum-classical optimization method. Later, this was picked up again by Bärligea et al. [70], who called it the Block Encoding Quantum Optimizer (BENQO), establishing it as a general VQA

⁶Note that the specific mixing Hamiltonian used here is a convenient default choice. However, in practice, any Hamiltonian can be chosen provided that its ground state is known.

framework. Its claim of promising results and specialization toward solving Ising problems made it an interesting candidate for the comparative investigations (see Chapters 4 and 5) of this thesis. The following theory is taken from Kuete Meli et al. [69].

The authors propose to encode the Ising cost operator of Equation (2.10) into a unitary, by embedding the $\sin(\hat{\mathbf{C}})$ into the larger $2^{n+1} \times 2^{n+1}$ unitary matrix

$$\mathbf{U} := \begin{bmatrix} \sin(\hat{\mathbf{C}}) & \cos(\hat{\mathbf{C}}) \\ \cos(\hat{\mathbf{C}}) & -\sin(\hat{\mathbf{C}}) \end{bmatrix} \quad \text{with} \quad \hat{\mathbf{C}} := \mathbf{C}/K, \quad (2.17)$$

where K is a constant scaling factor. This technique is commonly referred to as “block-encoding” [71], as one can retrieve the upper left block in (2.17) by setting the additional qubit to the state $|0\rangle$. The expectation of \mathbf{U} with respect to the basis state $|0, q\rangle := |0\rangle \otimes |q\rangle$ therefore becomes⁷

$$\langle \mathbf{U} \rangle := \langle 0, q | \mathbf{U} | 0, q \rangle \equiv \sin(\hat{\mathbf{C}}). \quad (2.18)$$

By choosing K such that it scales the entries of $\hat{\mathbf{C}}$ to lie within $[-\frac{\pi}{2}, \frac{\pi}{2}]$, one can assure that the sin-function is strictly increasing, making \mathbf{U} a monotone, bijective mapping of the cost \mathbf{C} . Kuete Meli et al. [69] choose $K = \frac{2}{\pi} \cdot C_{\max} := \frac{2}{\pi} \cdot \sum_{1 \leq i < j \leq n} |C_{ij}|$, which is adapted in this thesis. The exact cost value can then be retrieved as $\langle \mathbf{C} \rangle = K \arcsin\langle \mathbf{U} \rangle$.

Using trigonometric and rotational identities and inserting the expectation of the cost function (2.11), the unitary matrix (2.17) can eventually be written as

$$\mathbf{U}(\hat{\mathbf{C}}) = \prod_{i=1}^n \underbrace{\mathbf{X}^{q_i} \mathbf{R}_Y(-2\hat{C}_{ii}) \mathbf{X}^{q_i}}_{\text{unary terms}} \cdot \prod_{1 \leq i < j \leq n} \underbrace{\mathbf{X}^{q_i+q_j} \mathbf{R}_Y(-2\hat{C}_{ij}) \mathbf{X}^{q_i+q_j}}_{\text{quadratic terms}} \cdot \mathbf{X} \otimes \mathbf{I}^{\otimes n}. \quad (2.19)$$

For a proper derivation of this formulation, see Appendix A.2. As the above equation represents a product of simple unitary transformations, it can be easily implemented on a gate-based quantum computer (see Figure 2.4).

Since \mathbf{C} is Hermitian, the same holds for \mathbf{U} , and therefore, it works not only as a quantum operator but also as a measurement observable. However, as the number of individual Pauli observables in \mathbf{U} increases exponentially in system size n , the authors instead propose the use of a “Hadamard test” – based on the principle of implicit measurement [31, chapter 4.4] – to measure the expectation value of \mathbf{U} . The circuit displayed in Figure 2.5 illustrates the procedure: After the simple ansatz $|\psi(\boldsymbol{\theta})\rangle = \bigotimes_{i=1}^n \mathbf{R}_Y(\theta_i)|0\rangle$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ is applied, the quantum system is in the initial state

$$|0\rangle_a \otimes |\Psi_{\text{in}}\rangle = |0\rangle_a \otimes |0\rangle_c \otimes |\psi(\boldsymbol{\theta})\rangle, \quad (2.20)$$

including an ancillary qubit. With the Hadamard test, one can then project the entire system onto the eigenspace of \mathbf{U} via the projectors $\mathbf{P}_{\pm} = (\mathbf{I}_n \pm \mathbf{U})/2$. The resulting state is

$$|\psi_{\text{out}}\rangle = |0\rangle_a \otimes (\mathbf{P}_+ |\Psi_{\text{in}}\rangle) + |1\rangle_a \otimes (\mathbf{P}_- |\Psi_{\text{in}}\rangle). \quad (2.21)$$

⁷The last equality arises from the fact that \mathbf{C} is a diagonal matrix and $|q\rangle$ one of the computational basis states of the system.

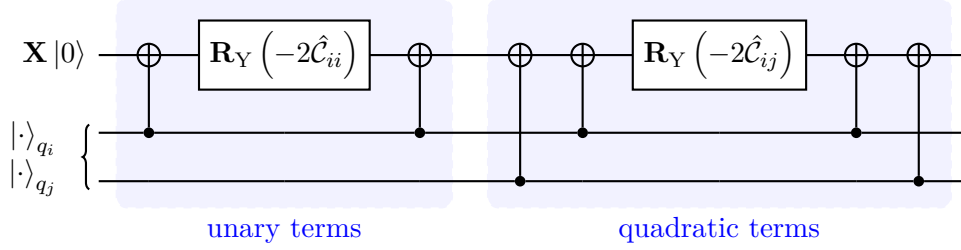


Figure 2.4: Circuit Representation of \mathbf{U} . The unitary of Equation (2.19) is represented by a sequence of unary and quadratic terms. The coupling weights \mathcal{C}_{ij} between nodes q_i and q_j each cause a rotation by $\mathbf{R}_Y(\pm 2\hat{\mathcal{C}}_{ij})$ if their qubits are in opposite (+) or the same (-) states⁸. The procedure for applying the self-weights \mathcal{C}_{ii} is analog.

This allows to “measure the operator” – by measuring the ancillary qubit – in terms of the probabilities $p_0 = \|\mathbf{P}_+|\psi_{\text{in}}\rangle\|^2$ and $p_1 = \|\mathbf{P}_-|\psi_{\text{in}}\rangle\|^2$. The desired expectation value is then simply $\langle \mathbf{U} \rangle \equiv p_0 - p_1$, leading to a cost value of

$$\langle \mathbf{C} \rangle = K \arcsin(p_0 - p_1). \quad (2.22)$$

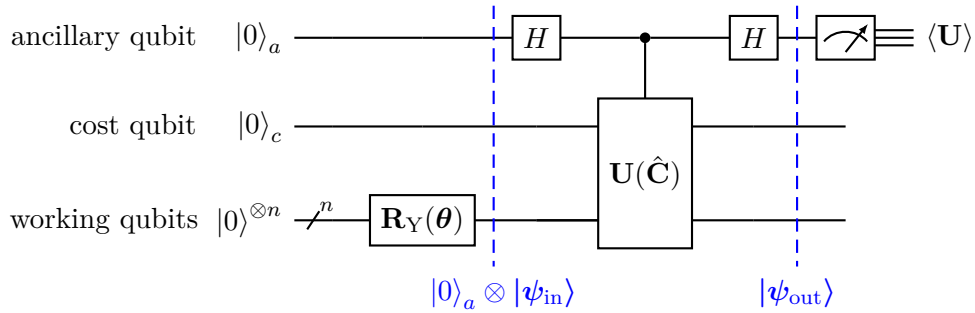


Figure 2.5: Hadamard Test for Measuring $\langle \mathbf{U} \rangle$. This circuit illustrates BENQO’s measurement procedure. After applying a Hadamard gate onto the ancillary qubit initialized as $|0\rangle_a$, the controlled- \mathbf{U} operation projects the eigenspace of \mathbf{U} onto the system. Thus, $\langle \mathbf{U} \rangle$ can be determined using p_0 and p_1 , which are revealed by the measurement.

2.3.4 Parameter Optimization

Now that it is clear how the loss function $\mathcal{L}(\theta)$ for a given problem can be evaluated on a quantum computer, the remaining challenge is how to minimize it (see Equation (2.15)). As mentioned earlier, this task is delegated to classical parameter optimizers within the variational framework. While in recent years, some methods have been developed specifically for the optimization of “quantum loss functions” [72], most state-of-the-art

⁸using that $\mathbf{X} \cdot \mathbf{R}_Y(-2\hat{\mathcal{C}}_{ij}) \cdot \mathbf{X} \equiv \mathbf{R}_Y(+2\hat{\mathcal{C}}_{ij})$.

algorithms used today were already established decades ago [73]. Generally, all local optimizers can be classified into two categories: gradient-based and gradient-free algorithms. Gradient-based methods are often preferred for their favorable convergence properties, but calculating derivatives can be computationally expensive or even infeasible, making gradient-free methods more suitable in other situations.

Gradient-based methods generally follow a straightforward iterative procedure:

1. Start with an initial guess $\boldsymbol{\theta}_0$.
2. Update $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \alpha_k \mathbf{d}_k$ until a convergence criterion is satisfied.

In this process, α_k represents the step size, which may vary with each iteration k , and \mathbf{d}_k is the search direction, determined by the gradient (and possibly the Hessian) of the loss function. The main goal in algorithm design is, therefore, to make a smart choice of α and \mathbf{d} . For example, in standard gradient descent, α is chosen as a constant hyperparameter, and the gradient itself serves as the search direction \mathbf{d} .

For computing the analytical gradient $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = (\partial_{\theta_1} \mathcal{L}(\boldsymbol{\theta}), \dots, \partial_{\theta_n} \mathcal{L}(\boldsymbol{\theta}))^\top$ of a parametrized quantum circuit, the parameter-shift rule was proposed [74, 75]. It is given by

$$\frac{\partial}{\partial \theta_i} \mathcal{L}(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta}_{i,+}) - \mathcal{L}(\boldsymbol{\theta}_{i,-})}{2} \quad (2.23)$$

where $\boldsymbol{\theta}_{i,\pm} := (\dots, \theta_i \pm \frac{\pi}{2}, \dots)$. Using this exact calculation method requires $2n$ function evaluations, where n is the number of parameters.

An interesting method, proposed as such by Kuete Meli et al. [69], is the normalized gradient descent (NGD) [76, 77], which uses a normalized gradient $\mathbf{d} = \nabla_{\boldsymbol{\theta}} \mathcal{L} / \|\nabla_{\boldsymbol{\theta}} \mathcal{L}\|_2$ as the search direction and an exponentially decaying step size, defined as

$$\alpha_k = \sqrt{\frac{\pi}{2}} n \cdot \exp\left(-\frac{4k^2}{k_{\max}^2}\right), \quad (2.24)$$

where the only hyperparameter is the number of total steps k_{\max} . This is the only algorithm examined in this thesis that is not part of the standard suite of local optimizers in QISKIT [32] or SCIPY [78]. Its inclusion here is motivated by its promising performance shown in related works [69, 70, 77].

Among the other methods studied, ADAM [79] and AMSGrad [80] rely on adaptive learning rates, using estimates of the first and second moments of the gradient, with bias correction influencing the search direction in each iteration. BFGS [81], on the other hand, employs an approximation of the inverse Hessian matrix combined with the exact gradient to determine the search direction, and uses line search to find the step size. The CG (conjugate gradient) method [82] equally uses a line search procedure for the step sizes, but transforms the gradients into conjugate directions for the direction updates. Both TNC [83] and SLSQP [84] tackle quadratic subproblems in each iteration to compute optimal search directions.

Lastly, as the SPSA method [85] plays a significant role in later discussions, it is worth highlighting its details. SPSA, while categorized as a gradient-based method, only

estimates the gradient using stochastic information based on simultaneous perturbation,

$$\mathbf{d}_k = (d_{1,k}, \dots, d_{n,k})^\top \quad \text{with} \quad d_{i,k} = \frac{\mathcal{L}(\boldsymbol{\theta}_k + c_k \boldsymbol{\Delta}_k) - \mathcal{L}(\boldsymbol{\theta}_k - c_k \boldsymbol{\Delta}_k)}{2c_k \Delta_{i,k}}, \quad (2.25)$$

where $\boldsymbol{\Delta}_k = (\Delta_{1,k}, \dots, \Delta_{n,k})^\top$ is a random perturbation vector, and c_k is a hyperparameter controlling the perturbation magnitude, decreasing as k increases. Notably, SPSA only requires two function evaluations per iteration, a significant improvement over the parameter-shift rule (2.23), which requires $2n$ function calls.

Gradient-free methods operate quite differently than the presented gradient-based scheme. Powell’s method [86] uses an iterative series of line searches along a set of mutually conjugate directions in the multidimensional parameters space, thereby minimizing the function without relying on gradients. The Nelder-Mead algorithm [87] uses a simplex (an n dimensional geometric shape with $n - 1$ vertices) to navigate the parameter space. By iteratively replacing the vertex with the highest function value with a new point, the simplex continuously adapts to the local landscape of the loss function, eventually contracting around the minimum. COBYLA [88] is another widely used method, where a series of linear approximations to the objective function, formed using the vertices of a simplex, are minimized iteratively in a sequential approach.

Specifically designed for quantum circuits, the NFT method [72] leverages the insight that the cost function for a single parameter resembles a sinusoidal curve and is, therefore, exactly minimizable. Using this fact, the problem can be split into a sequence of sinusoidal subproblems and solved iteratively. Another recently proposed optimizer is UMDA [89], a type of Estimation of Distribution Algorithm (EDA). UMDA samples solution candidates from univariate normal distributions, which are iteratively updated using the best-performing candidates from previous rounds.

For further details on the precise update mechanisms of some of these optimizers – both gradient-based and gradient-free – refer to Appendix A.3.

2.3.5 Limitations and Challenges

Despite the promise of the variational paradigm, reports of significant challenges and limitations continue to accumulate, with some of the most critical issues arising from the noise inherent in NISQ devices [19–22]. Notably, in comprehensive reviews of both VQE [66] and QAOA [67], the adverse effects of noise on algorithm performance are only briefly discussed towards the end. This thesis will address this issue in more detail. However, also other factors have been shown to limit the trainability of these algorithms [26, 27], with the most imminent one being the phenomenon of Barren Plateaus (BPs).

A VQA – representing a loss function $\mathcal{L}(\boldsymbol{\theta})$ – exhibits a *probabilistic* BP if the variance

$$\text{Var}_{\boldsymbol{\theta}}[\mathcal{L}(\boldsymbol{\theta})] \in \mathcal{O}\left(\frac{1}{b^n}\right) \quad \text{or} \quad \text{Var}_{\theta_i}[\partial_{\theta_i} \mathcal{L}(\boldsymbol{\theta})] \in \mathcal{O}\left(\frac{1}{b^n}\right) \quad \text{for some } \theta_i \in \boldsymbol{\theta} \quad (2.26)$$

for some $b > 1$, which implies that the loss function or its gradients concentrate exponentially around their mean as the number of qubits n increases. This behavior indicates

that resolving variations in the loss landscape for larger systems demands exponentially more resources, rendering training highly inefficient and practically infeasible. Loss landscapes with *probabilistic* BPs are generally flat and featureless, with occasional fertile valleys [90]. However, if

$$|\mathcal{L}(\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{\theta}}(\mathcal{L}(\boldsymbol{\theta}))| \in \mathcal{O}\left(\frac{1}{b^n}\right) \text{ for all } \boldsymbol{\theta}, \quad (2.27)$$

the plateau is termed *deterministic*, lacking any well-separated extrema and therefore presenting even more severe challenges. Detection of either BP type in a specific VQA requires analytical methods, well-summarized by Larocca et al. [91]. Causes of BPs are diverse and occur in nearly all known architectures. However, mitigation strategies remain limited, with recent work by Cerezo et al. [92] linking mitigation efforts to classical simulability, suggesting that solutions may be fundamentally constrained – with the most promising one being alternative initialization strategies. Due to this apparent intractability of the BP problem, it will be revisited in later studies.

For a thorough overview of these challenges and their implications, the NISQ review by Ezratty [18] is a recommended resource.

Chapter 3

Methodology

As outlined in the previous chapter, while variational quantum algorithms (VQAs) hold significant promise, they also come with various challenges. These include the inherent complexity of their loss landscapes [26, 27], the exponential concentration of loss values [91, 93], and stochastic noise [19, 20]. This noise is introduced not only at the circuit level by quantum devices (see Section 2.1.2) but also through the finite number of measurement samples that quantum computing inherently relies on, making it unavoidable in practice. Despite these obstacles, many comparative analyses – often included in new algorithmic proposals – report promising results based on narrowly defined testing setups that do not properly address these critical issues. This chapter introduces a systematic and generalizable methodology to evaluate the performance of VQAs, incorporating the unique challenges they face during the Noisy Intermediate-Scale Quantum (NISQ) era and beyond.

As Lavrijsen et al. [15] pointed out, the exact mechanisms of how noise impacts the results of a VQA remain an area of active research, and the same holds for the other challenges mentioned above. Figure 3.1 summarizes the VQA pipeline by schematically illustrating its three main components. Each of them can be changed or improved individually, making it intuitive that they may be affected by the stated challenges, such as stochastic noise, in distinct ways. Individually evaluating each part should, therefore, result in a clearer picture of the origins of any observed advantages or limitations in algorithm performance.

Some studies focus on opportunities in problem encoding [94–98], demonstrating that reformulating problems and applying constraints in different ways can effectively alter the size of feasible subspaces and reduce the number of required variables or qubits. While these methods can enhance performance and mitigate some scaling challenges, the most pressing obstacles arise in the two other parts of the VQA pipeline. An increasing body of theoretical research, therefore, examines the impacts of noise and other issues on the trainability of parametrized quantum ansatzes.

For example, Wang et al. [20] showed that noise can cause vanishing gradients in the training landscape of VQAs, imposing serious limitations on VQA performance even beyond the barren plateaus discussed in Section 2.3.5. Similarly, Stilck França and García-Patrón [19] demonstrated that noise fundamentally restricts the potential for quantum computational advantage, regardless of substantial increases in qubit numbers. González-García et al. [21] further highlighted that the single-qubit error rate necessary for achieving quantum advantage scales with the inverse of the product of circuit depth

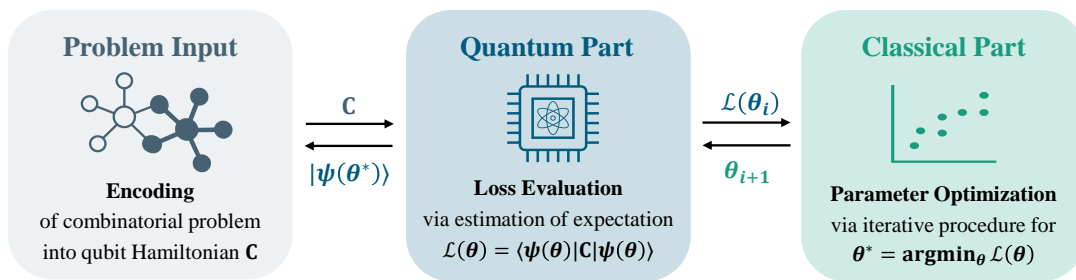


Figure 3.1: Components of the Performance Assessment of a VQA. First, the optimization problem is encoded into a Hamiltonian \mathbf{C} , which is embedded into a quantum circuit. Then, the circuit outputs the expectation value of \mathbf{C} with respect to a prepared quantum state $|\psi(\theta)\rangle$, implementing the loss function $\mathcal{L}(\theta)$. A classical optimization method iteratively adjusts the parameters θ to find the set θ^* which minimizes the loss. Finally, the state $|\psi(\theta^*)\rangle$ is used to derive the classical solution to the original problem. Each component (problem encoding, loss evaluation, and parameter optimization) can be modified independently and should be investigated as such.

and width, severely constraining algorithm design. Additionally, Wang et al. [99] showed that even with error mitigation, the trainability of VQAs remains limited and requires exponential resources for effective parameter optimization.

Beyond noise, these algorithms also face intrinsic challenges. Bittel and Kliesch [26] showed that the classical optimization problem of VQAs is NP-hard, as their loss landscape can have many persistent local minima far from the global optimum, which was also brought up by Anschuetz and Kiani [27].

The key question is how all these findings related to the quantum side of the variational pipeline impact the classical part – the parameter optimization, which is often regarded as the main computational driver in variational quantum computing. Given the present challenges, classical optimizers may experience early converging, miss the global minimum, become trapped in noise-induced local minima, or fail to converge altogether [15]. This thesis adopts a systematic approach to empirically demonstrate how these substantial challenges translate to performance limits of classical optimizers, directly affecting the viability of variational quantum optimization.

To this end, Chapter 4 presents a comparative investigation of two prominent VQA ansatzes – VQE and QAOA (see Section 2.3.2) – along with a new proposal, BENQO (see Section 2.3.2). By comparing circuit resources, testing hardware noise, investigating the statistical error of finite sampling, and the general loss landscapes, Chapter 4 highlights the various challenges these algorithms face even beyond the NISQ era. Its analyses particularly demonstrate how the total errors accumulated in those algorithms may, in worst-case scenarios, be effectively modeled as additive Gaussian noise. This noise model has the advantage of being well suited for testing and simulation.

Therefore, in Chapter 5, this simulated noise setting is applied to a suite of state-of-the-art optimizers available in QISKIT [32]. From extensive numerical studies with the BENQO algorithm, the scaling properties of problem solvability are examined across

both the total noise level and system size. This allows for the derivation of a boundary line, after which finding high-quality solutions becomes infeasible. The resulting performance limits are compared against both the actual noise conditions in current hardware and the statistical shot noise remaining in fault tolerance, revealing key insights into the general usefulness of variational quantum computing for classical problems of practical interest.

All experiments in this work were conducted using simulators, as access to real hardware was beyond the scope of this thesis.

3.1 Problem Instances

A wide variety of combinatorial optimization problems can be described by the Quadratic Unconstrained Binary Optimization (QUBO) formulation [45, 46], as outlined in Section 2.2.1. Rather than focussing on a particular graph problem, such as the TSP, and mapping it to its Ising equivalent [37], this study uses random problem instances for the experiments. Each instance is created with a unique seed by sampling all entries of the $n \times n$ QUBO-matrix \mathbf{Q} from a discrete, uniform distribution in the interval $[1, 10]$. The exact magnitude or precision of the weights is not critical, as the resulting loss values are normalized across all experiments. By starting the sampling range at 1 instead of 0, the QUBO matrix represents a fully-connected graph problem, including self-connections. Since many practically relevant problems involve constraints that translate to quadratic penalty terms (see Section 2.2.2) – often leading to negative diagonal terms in \mathbf{Q} – the diagonal of the random matrix is similarly multiplied by -1 to reflect the structure of realistic problems. To give an example, Figure 3.2 visualizes such a QUBO matrix and its corresponding problem graph for a system of size $n = 10$.

3.2 Statistical Measures

This section introduces the notation and definitions of the key statistical measures used in Chapter 4. Given the loss function $\mathcal{L}(\boldsymbol{\theta})$, the total range of loss values (the difference between the maximum and minimum) across the landscape $\boldsymbol{\theta} \in [-2\pi, 2\pi]^n$ increases with the system size n . To enable quantitative comparison across different system sizes, the loss function must be normalized (cf. Larocca et al. [100]). The formulation

$$\hat{\mathcal{L}}(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta})}{|\mathcal{L}(\boldsymbol{\theta})|_{\max}} \quad \text{with} \quad |\mathcal{L}(\boldsymbol{\theta})|_{\max} = \max_{\boldsymbol{\theta} \in \mathbb{R}^n} |\mathcal{L}(\boldsymbol{\theta})| \quad (3.1)$$

is therefore used in the following, as it guarantees that across the loss landscape $|\mathcal{L}(\boldsymbol{\theta})| \leq 1$ independent of the problem size n . Any variance or error off the loss function is thus computed with respect to the normalized loss $\hat{\mathcal{L}}$.

Generally, the variance of a random variable X is given by $\text{Var}[X] = \text{E}[(X - \text{E}[X])^2]$, corresponding to the dispersion of X around its mean $\text{E}[X]$. However, unless the exact

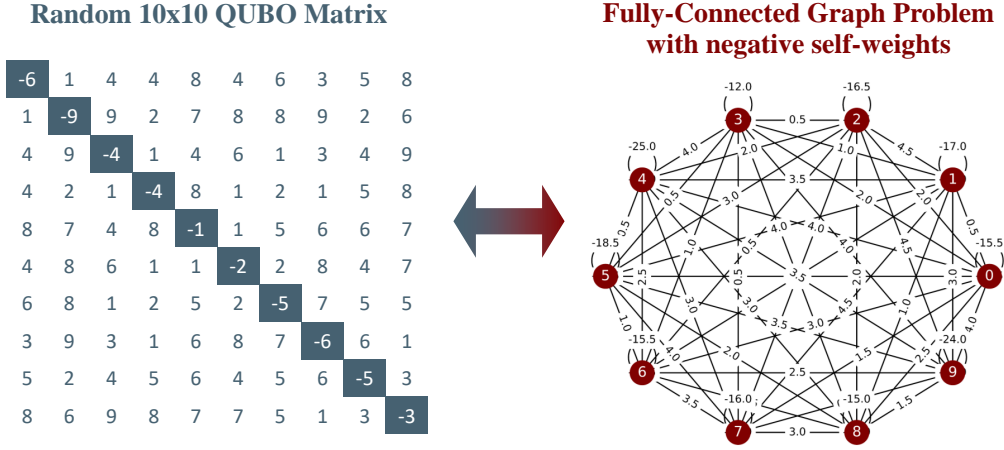


Figure 3.2: Example of a General QUBO Problem. The entries of the QUBO matrix \mathbf{Q} are sampled uniformly within $[1, 10]$ (on the off-diagonals) and $[-10, -1]$ (on the diagonal). \mathbf{Q} can be translated to the weights of the corresponding Ising problem using Equation (2.13), which is visualized as a fully-connected graph with negatively weighted self-connections. Many practically relevant problems have this form.

probability distribution of X is known, the variance can only be estimated via the so-called sample variance

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (3.2)$$

of a given sample $\{x_1, \dots, x_N\}$ of size N around its sample mean \bar{x} . As sample mean, variance and the corresponding standard deviation s are often used in the following analyses, note that their statistical uncertainties are given by $\Delta\bar{x} = s/\sqrt{N}$ for the mean, and $\Delta s^2 = \sqrt{\frac{2}{N-1}} s^2$ for the variance [101]. Furthermore, the mean absolute deviation (MAD) is defined as

$$d_{\text{mad}} = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}| \quad (3.3)$$

which is related to the sample standard deviation via $d_{\text{mad}} = \sqrt{\frac{2}{\pi}} s$. A closely related measure is the mean squared error (MSE)

$$d_{\text{mse}} = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (3.4)$$

which can be used as a goodness-of-fit measure.

When it comes to quantifying errors in the loss function, $\tilde{\mathcal{L}}$ generally defines an erroneous normalized loss function, such that $\delta\mathcal{L}(\boldsymbol{\theta}) = \hat{\mathcal{L}}(\boldsymbol{\theta}) - \tilde{\mathcal{L}}(\boldsymbol{\theta})$ gives the total error at a certain point in parameter space. For a sample of loss values at N different points, one

can investigate the relative absolute error (RAE)

$$\epsilon(\mathcal{L}, \tilde{\mathcal{L}}) = \frac{\sum_{i=1}^N |\tilde{\mathcal{L}}(\boldsymbol{\theta}_i) - \mathcal{L}(\boldsymbol{\theta}_i)|}{\sum_{i=1}^N |\mathcal{L}(\boldsymbol{\theta}_i) - \bar{\mathcal{L}}|} \quad (3.5)$$

with $\bar{\mathcal{L}}$ being the sample mean. The RAE can be seen as an absolute error measure normalized by the mean absolute deviation of the values. This means that an $\epsilon = 1$ indicates that the total range of the error is the exact same as the range of the loss values themselves. Note that as this is a relative measure, the normalization of $\tilde{\mathcal{L}}$ becomes obsolete, such that one might as well use the loss value \mathcal{L} in the original scale for comparison. Further, to quantify the linear correlation between two sets of samples (e.g. noisy and exact loss values), the Pearson correlation coefficient (PCC) of a sample of paired points $\{(x_1, y_1), \dots, (x_N, y_N)\}$ is defined as

$$\rho(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \in [-1, 1], \quad (3.6)$$

for which a PCC of +1 (or -1) represents a perfectly linear (or antilinear) relationship.

3.3 Benchmark Metrics

The main results of this thesis concern the behavior of the classical side of the optimization part of variational quantum optimization (see Chapter 5). An evaluation of optimizer performance on a given problem crucially depends on the chosen benchmark metrics. Meaningful metrics should enable a fair comparison of resource cost, run time, solution quality, or problem complexity between different solver settings [11]. Instead of fixing the desired solution quality and investigating the scaling of the resource cost and run time, as was done by Lavrijsen et al. [15], Sung et al. [102], the following analysis will focus on the scope of the achievable quality of the quantum solutions to instead make a statement of the general scaling behavior of VQAs' performance.

The solution of a variational optimization task is given by the n -dimensional parameter vector

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{\theta}), \quad (3.7)$$

which (approximately) minimizes the quantum loss function $\mathcal{L}(\boldsymbol{\theta})$, such that $|\psi^*\rangle = |\psi(\boldsymbol{\theta}^*)\rangle$ is considered the proposed solution state of the respective VQA. A widely established measure to evaluate such a solution is the approximation ratio (AR)

$$Q_{\text{AR}}(|\psi^*\rangle) = \frac{\langle \psi^* | \mathbf{C} | \psi^* \rangle - C_{\text{max}}}{C_{\text{min}} - C_{\text{max}}} \in [0, 1], \quad (3.8)$$

which expresses how far the discovered solution $|\psi^*\rangle$ differs from the optimal state, $|q^*\rangle$, in terms of the energy of the underlying quantum system. Here, \mathbf{C} is the Hamiltonian cost operator (i.e., Equation (2.10)) and the minimum energy is given by $C_{\text{min}} = \langle q^* | \mathbf{C} | q^* \rangle$. For the system sizes discussed later, all terms in Equation (3.8), including

the expectation value of the resulting cost, can be determined classically. In itself, the AR is a more meaningful measure than merely observing the loss value $\mathcal{L}^* = \mathcal{L}(\theta^*)$ or the absolute energy difference $\Delta\mathcal{L} = C_{\min} - \mathcal{L}^*$.

However, the AR is still not very representative of the quality of quantum state solutions, as even the uniform superposition $|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=1}^{2^n} |q_i\rangle$ of all basis states $\{|q_i\rangle\}_{i=1,\dots,2^n}$, which can be considered a worst-case solution, will achieve ARs way above 0¹, such that any experimentally measured Q_{AR} will effectively fall within $[Q_{\text{AR}}(|\phi\rangle), 1]$. One should, therefore, rather look at a normalized approximation ratio (NAR)

$$Q_{\text{NAR}}(|\psi^*\rangle) = \frac{\langle\psi^*|\mathbf{C}|\psi^*\rangle - \langle\phi|\mathbf{C}|\phi\rangle}{C_{\min} - \langle\phi|\mathbf{C}|\phi\rangle} \in [0, 1], \quad (3.9)$$

as a measure to distinguish the solution from a totally mixed quantum state.

Another issue with ARs is that for classical optimization problems, the optimal state $|q^*\rangle$ is actually a computational basis state of the system, meaning that it can be written as a binary string. Therefore, the most probable basis state $|\psi_{\max}\rangle = \operatorname{argmax}_{\psi_i} |\langle\psi^*|\psi_i\rangle|^2$ of the full quantum output $|\psi^*\rangle$ should be considered the actual solution proposal of a VQA, instead of $|\psi^*\rangle$ which is evaluated by the AR. One can monitor the probability of this basis state within $|\psi^*\rangle$ via

$$c(|\psi_{\max}\rangle) = |\langle\psi_{\max}|\psi^*\rangle|^2 \quad (3.10)$$

which is named “confidence” here, as it measures how confident the resulting classical solution is within the optimized quantum state.

These considerations give way to a more representative performance indicator, the approximation index x_t , which marks whether the AR of $|\psi_{\max}\rangle$ lies above a given threshold $t \in [0, 1]$:

$$x_t(|\psi_{\max}\rangle) = \begin{cases} 1, & \text{if } \frac{|\langle\psi_{\max}|\mathbf{C}|\psi_{\max}\rangle - C_{\max}|}{|C_{\min} - C_{\max}|} \geq t \\ 0, & \text{otherwise} \end{cases} \quad (3.11)$$

This means, $x_1 = 1$, if and only if the most probable basis state $|\psi_{\max}\rangle$ is the optimal solution $|q^*\rangle$. Using this index, one can estimate the success probability of finding an optimal (or near-optimal) solution by

$$\hat{p}_t = \frac{1}{N} \sum_{i=0}^N x_t(|\psi_{\max}\rangle_i) \in [0, 1], \quad (3.12)$$

corresponding to the measured proportion of “successful” solutions out of a sample size N . In the classical optimizer studies presented in Chapter 5, the probability of optimal solutions p_1 is monitored as \hat{p}_{opt} , and solutions within a 5% and 1% margin of the

¹This argument will be revisited and explained in more detail at the end of Section 5.4, where Figure 5.8 illustrates the respective distribution of possible loss values.

optimal are monitored as $\hat{p}_{95\%}$ and $\hat{p}_{99\%}$. Note, that the standard error for the estimated proportion in Equation (3.12) is given by

$$\sigma(\hat{p}(t)) = \sqrt{\frac{\hat{p}(t)(1 - \hat{p}(t))}{N}}, \quad (3.13)$$

using the normal approximation for a binomial distribution, which is valid for large N .

Further, one can investigate the total running time t , the classical part needs to execute the optimization. As the performance of computing systems may vary largely, Scriva et al. [24] proposed to instead monitor a general lower bound for t , given by

$$t < n_{\text{shots}} \cdot n_{\text{calls}} \cdot D \cdot t_{\text{gate}}. \quad (3.14)$$

Here, n_{calls} are the number of calls to the quantum function $\mathcal{L}(\boldsymbol{\theta})$, measured using n_{shots} , D corresponds to the circuit depth, and t_{gate} is a hardware-specific measure for the time it takes to execute a gate in the circuit. Note, that n_{calls} does not necessarily correspond to the number of iterations n_{iter} used by an optimizer, as one iteration might need not only an evaluation of the loss function itself but also of its gradient, which adds overhead (e.g., $2n$ calls for a gradient calculated via the parameter-shift rule (2.23)). Finally, Scriva et al. [24] pointed out, that even an optimal combination of $n_{\text{shots}} \cdot n_{\text{calls}}$ must strictly stay below 2^n in order to avoid quantum disadvantage, without even accounting for t_{gate} .

Chapter 4

Quantum-Inherent Challenges and Errors

In an effort to examine the performance limitations of the quantum and classical parts of variational quantum algorithms (VQAs) separately, this section begins with a demonstrative investigation of the challenges inherent to the quantum part. Therefore, all its results are solely based on the quantum loss function $\mathcal{L}(\boldsymbol{\theta})$, describing general QUBO (Quadratic Unconstrained Binary Optimization) problem instances as in Section 3.1, calculated with a parametrized quantum circuit ansatz. For a VQA to be called NISQ-friendly, one has to show its relative robustness against hardware-specific challenges, such as its resource cost or the errors arising from finite sampling and hardware noise (see Section 2.1.2). At the same time, the general “optimizability” of the parameters $\boldsymbol{\theta}$ in the quantum loss function should be investigated, as some quantum architectures are more prone to lead to infeasible loss landscapes [26, 91].

In the following analyses, three representative Variational Quantum Algorithms are compared regarding the aforementioned aspects: The well-established Variational Quantum Eigensolver (VQE) [62], Quantum Approximate Optimization Algorithm (QAOA) [63] (outlined in Section 2.3.2), and the newer proposal of a Block Encoding Quantum Optimizer (BENQO) [69, 70], described in Section 2.3.3. Their theoretical and hardware-specific circuit requirements are analyzed in Section 4.1, which holds important implications for their noise susceptibility when executed on real quantum hardware. Section 4.2 then examines the effect of modeled hardware errors on the algorithms’ outputs and characterizes the resulting error distribution. The analysis is repeated for the statistical errors introduced by finite measurement sampling in Section 4.3. Finally, the structure of the loss functions is explored in Section 4.4, offering insights into each algorithm’s general optimizability. This broad comparison not only highlights the potential strengths and weaknesses of each method but also motivates the experimental approach behind the results presented in Chapter 5.

4.1 Circuit Resources

As errors in the execution of quantum circuits increase significantly with circuit depth [21], any proposed VQA must operate within a regime where the circuit depth D and number of qubits n (i.e., width) are kept sufficiently low. González-García et al. [21] estimated that quantum advantage in variational quantum optimization requires at least a single-qubit error rate of $1/nD$. Given that for the problem encoding described in Section 2.2.1, the number of required qubits corresponds directly to the problem size

(i.e., the number of nodes in a graph), the depth of the parametrized quantum circuit would need to scale sublinearly in depth relative to problem size to stay feasible for larger instances.

Table 4.1 compares the number of different basis gates required by each algorithm to represent the loss function of the QUBO problem described in Section 3.1, which assumes a fully-connected graph with n nodes and m connections. To ensure a fair comparison, the setup of all three algorithms was configured to have the same number of parameters n , fixing the number of dimensions in the loss landscape.

	VQE	QAOA	BENQO
# qubits	n	n	$n + 2$
# CNOT gates	$n - 1$	$p(n^2 - n)$	$3n^2 + 2n$
# single-qubit rotations	n	$p(n^2 + 3n)/2$	$n^2 + 2n$
# Hadamard gates	$2n - 2$	n	2
# measurement bases	$n(n + 1)/2$	$n(n + 1)/2$	1
qubit connectivity	<i>1 dim. nearest neighbor</i>	<i>graph-dependent</i>	<i>one-to-all</i>

Table 4.1: Comparison of Quantum Resources of three VQAs. A hardware-efficient VQE-ansatz, a conventional QAOA ansatz with p layers, and BENQO (all as described in Sections 2.3.2 and 2.3.3) are compared. The values are computed for Ising cost functions defined on a fully connected, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $n \equiv \|\mathcal{V}\|$ the number of nodes, and $m \equiv \|\mathcal{E}\| = n(n + 1)/2$ the number of edges (including self-connections). For better comparability, all entries are given in terms of n .

The hardware-efficient VQE ansatz consists of a single layer of rotational gates, followed by a linear entangling layer of controlled- \mathbf{Z} gates (i.e., a two-local ansatz, see Section 2.3.2). The controlled- \mathbf{Z} gates are decomposed using the identity $\mathbf{Z} = \mathbf{H}\mathbf{X}\mathbf{H}$, which yields an additional $2(n - 1)$ Hadamard gates.

For the conventional QAOA ansatz to have the same number of parameters as qubits, $p = n/2$ layers must be used, which requires an even number of qubits. The state is first initialized to a uniform superposition by Hadamard gates. In every layer, the mixing operator is applied to the qubit as n single-qubit X-rotations, while the phase operator applies two CNOTs and one rotation per connection among the m edges. For self-connections, the two CNOT gates cancel out, leading to a total of $p(2m - 2n)$ CNOT gates and $p(n + m)$ single-qubit rotations.

To construct the unitary operator in Equation (2.19) for the BENQO algorithm, $4m - 2n$ CNOT gates and n single-qubit rotations are required (see Figure 2.4). For the measurement in Figure 2.5, a controlled version of the operator \mathbf{U} needs to be implemented. For this, Equation (2.19) can be expressed as a product of CNOTs and controlled rotations, which can generally be decomposed as $[\mathbf{R}_Y(\theta)]^a = \mathbf{X}^a \cdot \mathbf{R}_Y(-\theta/2) \cdot \mathbf{X}^a \cdot \mathbf{R}_Y(\theta/2)$ where a is the value of the control qubit [69]. This decomposition doubles the number of single-qubit rotations for each connection and adds $2m$ additional CNOT gates. The remaining n rotation gates come from the single-layer ansatz structure of BENQO. In

a fully-connected graph, there are $m = n(n + 1)/2$ edges including self-connections, ultimately giving the values listed in Table 4.1.

Clearly, the circuit resources, which directly translate to circuit depth, vary significantly across the three algorithms. While the number of CNOT gates scales linearly with the number of qubits n in VQE, it scales quadratically for BENQO and even cubically for QAOA, since $p \propto n$. However, BENQO requires measuring in only a single basis, thanks to the use of the Hadamard test. In contrast, VQE and QAOA must measure all m Pauli terms of the observable cost operator (2.10) individually.

Implementing these circuits on actual quantum hardware requires catering to its connectivity, which can further increase the effective circuit depth. For instance, on IBM’s hardware based on superconducting qubits, nearest-neighbor connections (as required by the VQE ansatz) are more suitable than the one-to-all connectivity needed for BENQO, where ion trap architectures might be a better fit.

Figure 4.1 summarizes these considerations and highlights the consequences of super-linear scaling combined with weakly connected hardware. Although the original proposal of BENQO [69] suggested that BENQO becomes more gate-efficient than QAOA when $p \geq 3$ (or $n \geq 6$) (cf. Table 4.1), this advantage diminishes once their circuits are transpiled to superconducting hardware. Furthermore, considering the connection between circuit width, depth, and required error rates made by González-García et al. [21], it becomes evident that none of the compared architectures meet the ambitious requirements for quantum advantage, yet. For example, for a 10-qubit system, BENQO’s depth in Figure 4.1 is approximately 8000, necessitating a single-qubit error rate on the order of $\mathcal{O}(10^{-5})$, which is already an order of magnitude below current hardware capabilities (see Appendix B.1).

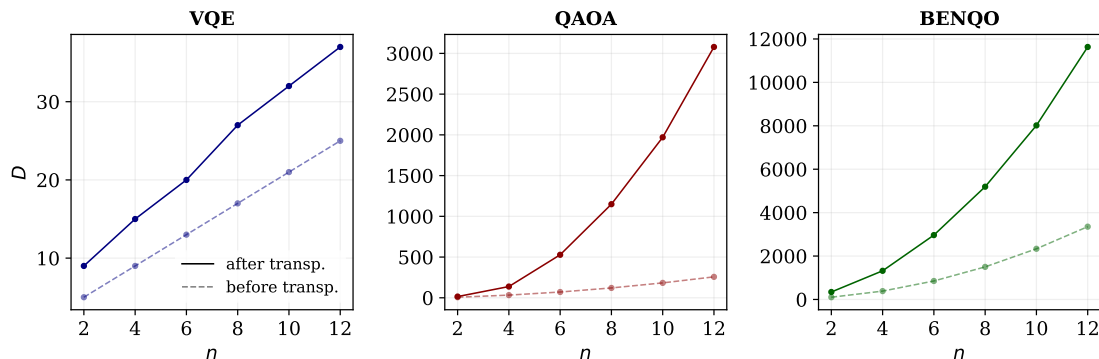


Figure 4.1: Comparison of Circuit Depths of three VQAs. Dashed lines represent the circuit depth D of the algorithms before transpilation¹ to specific backends (in this case, `ibm_sherbrooke`; see Appendix B.1 for details) as a function of system size n . Solid lines show the effective depth after transpilation to actual hardware.

¹Transpilation was performed using QISKIT [32] with an `optimization_level` of 2, which corresponds to a medium level of layout and routing optimization.

4.2 Hardware Noise

Noise is the primary constraint on algorithm executability imposed by NISQ hardware. This section examines how exactly this noise impacts the output of quantum algorithms, directly linking to the discussion on circuit resource constraints in the previous section. Since access to real quantum hardware was outside the scope of this thesis, noise models sampled from IBM quantum devices available through QISKIT [32, version 0.46.2] are used instead. These models account for the following types of errors:

- single-qubit gate errors, including single-qubit depolarization and thermal relaxation errors,
- two-qubit gate errors, including two-qubit depolarization and single-qubit thermal relaxation errors, and
- single-qubit readout errors.

Each error type reflects the hardware specifications of the chosen devices. For a brief overview of how these errors are defined and where they come from, refer to Section 2.1.2. The results presented in this section are based on the hardware models of the `ibm_brisbane`, `ibm_kyoto`, and `ibm_sherbrooke` backends (see Appendix B.1 for further details). Using simulated noise models to calculate the output of parametrized quantum circuits allows separating the effects of hardware noise and finite sampling errors, the latter being discussed in Section 4.3.

Figure 4.2 provides a graphical summary of the empirical behavior of hardware noise for the different modeled backends on all three algorithms under consideration. The errors are quantified using three key metrics: the relative absolute error (RAE) $\epsilon(\mathcal{L}, \tilde{\mathcal{L}})$ (see Equation (3.5)) between the exact loss values \mathcal{L} and the noisy values $\tilde{\mathcal{L}}$; the sample standard deviation $s[\delta\mathcal{L}]$ (see Equation (3.2)) of the total absolute error $\delta\mathcal{L} = \hat{\mathcal{L}} - \tilde{\mathcal{L}}$ with respect to the normalized loss value $\hat{\mathcal{L}}$; and the Pearson correlation coefficient (PCC) $\rho(\mathcal{L}, \tilde{\mathcal{L}})$ (see Equation (3.6)), which quantifies the linear correlation between exact and noisy values.

The RAE of all loss functions increases as expected with system size, given the analysis of circuit depths in the previous section. However, for both VQE and BENQO, the RAE tends to converge toward a constant value across all three backends. In contrast, it appears unbounded for QAOA. This behavior must be linked to changes in the absolute deviation of the loss function with increasing system size, as explored in more detail in Section 4.4. A constant RAE suggests that any changes in the absolute magnitude of error (numerator of the RAE), which is represented by $s[\delta\mathcal{L}]$ in Figure 4.2, is compensated by the changes in the overall deviation of loss values (denominator of the RAE).

The behavior of the PCC in Figure 4.2 with increasing system size is also distinct across all three VQAs. While VQE maintains a nearly perfect linear relationship between noisy and exact loss values, even as n grows, both QAOA and BENQO show a rapid decline in PCC, eventually approaching zero. This indicates the complete breakdown of any linear correlation between noisy and exact loss values, which is likely tied to the larger

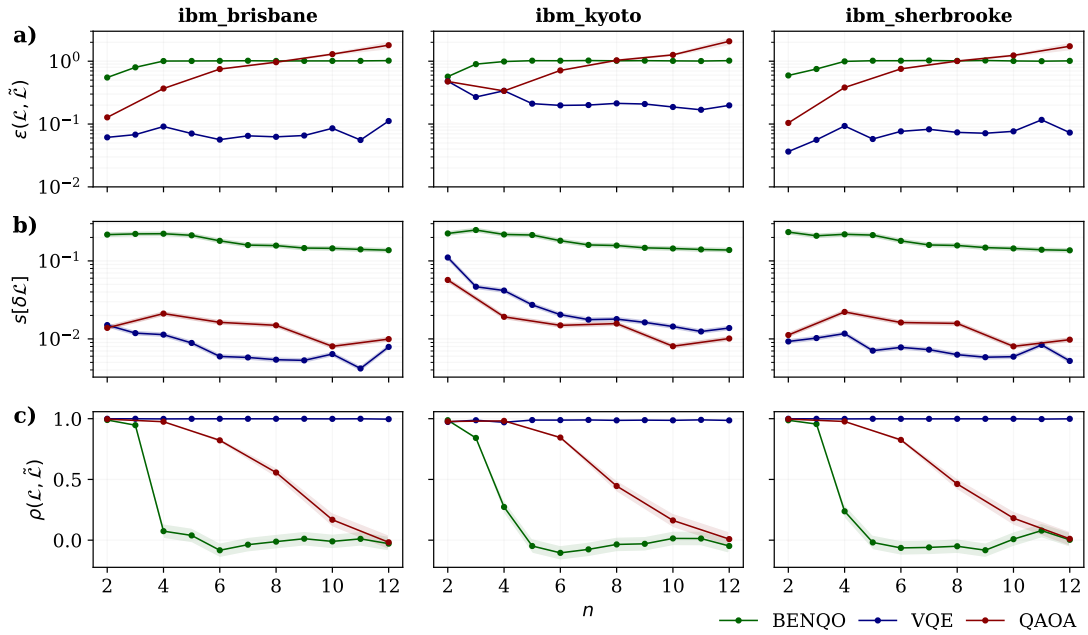


Figure 4.2: Comparison of Hardware-Induced Errors in the Loss Functions of three VQAs. a) illustrates the evolution of the RAE ϵ across varying system sizes (number of qubits) for each algorithm, using simulated noise models from three representative quantum backends (by IBM). b) displays the corresponding standard deviation s of the total absolute (normalized) errors, and c) shows the Pearson correlation coefficient between the exact and noisy loss values. All metrics are based on 1000 noisy loss evaluations at uniformly sampled parameters $\theta \in [-2\pi, 2\pi]^n$ for system sizes $n \in [2, 12]$. Note the logarithmic y-axes for ϵ and s .

circuit depths for increasing system size for those algorithms (refer again to Figure 4.1). As the circuit depth increases, the relationship between gate errors and the final output becomes more complex and nonlinear [21].

To bridge the gap to the main experiments of this thesis, described in Chapter 5, it is essential to characterize how these hardware-induced errors are distributed across each point of loss evaluation in the parameter space. Figure 4.3 illustrates that these errors, resulting from the hardware model, approximately follow a normal distribution at each evaluation point. However, this noise is not centered around zero. This is because systematic hardware noise sources often introduce bias – for example, a single qubit tends to relax towards the ground state (i.e., decay from $|1\rangle$ to $|0\rangle$) due to thermal relaxation. Additionally, gate errors accumulate in a consistent manner, gradually shifting the output values, which reduces the probability of measuring the correct state. Rather than being symmetrically distributed, these errors tend to shift the measurement outcomes in a specific direction. Furthermore, single-qubit readout errors may also introduce bias, as the probability of misreading a $|1\rangle$ as $|0\rangle$ is not necessarily equal to the reverse. Despite this localized bias in the noise at each point of the loss landscape, the global distribution

of noise, aggregated across the entire parameter space, is approximately centered around zero (see Appendix B.2). This property will be leveraged in later discussions.

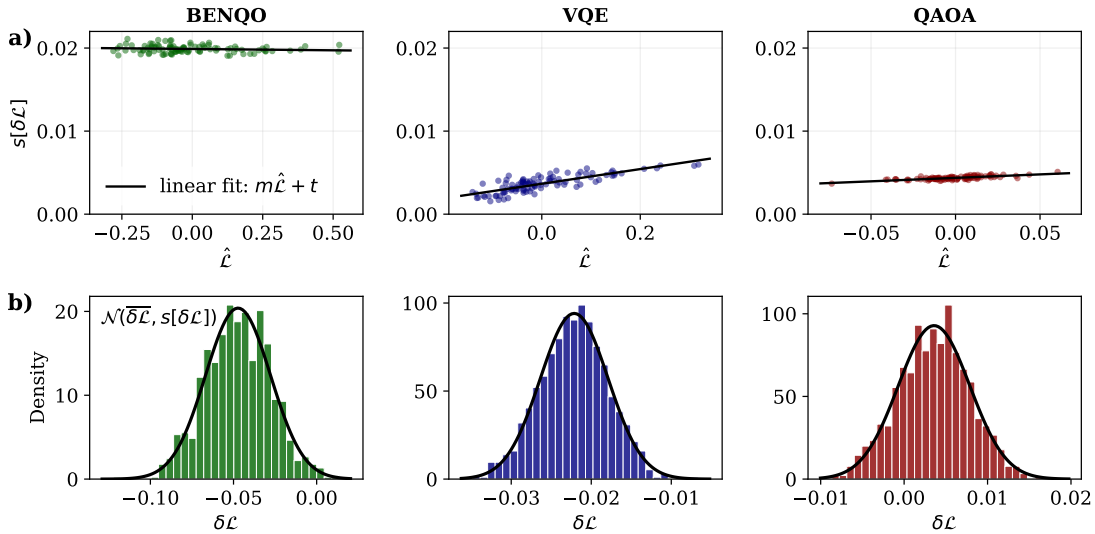


Figure 4.3: Distribution of Hardware-Induced Error Across Loss Landscape. a) shows the standard deviation of the errors of 1000 sample loss evaluations done with the `ibm_sherbrooke`² backend model at 100 randomly selected points across the loss landscape of a $n = 6$ problem instance against the normalized loss value at those points. Linear fits show the relationship between the loss value and the overall magnitude of the error for each of the three algorithms. b) are the probability density histograms of 1000 absolute normalized errors for one of these points, including the theoretic curves of a normal distribution with the same mean and standard deviation.

Figure 4.3 also shows the relationship between the overall magnitude of errors, $s[\delta\mathcal{L}]$, and the loss value at each evaluation point. These plots can help to determine whether modelled hardware noise is additive or multiplicative. To this end, linear fits of the sample standard deviations provide the optimal values for the slope m and y-intercept t of the lines. If $m \ll t$, the noise at an arbitrary point in parameter space is predominantly additive, whereas if $m \gg t$, the error behaves in a multiplicative manner.

Only for BENQO, the y-intercept is significantly larger than the slope, thus m becomes negligible. Hence, the error introduced by hardware in BENQO’s loss evaluation is predominantly modeled by an additive random variable. This is not the case for the other two algorithms, suggesting their errors are likely a combination of both multiplicative and additive noise. Nevertheless, as will be discussed in Section 4.4, additive errors tend to affect the loss landscape more severely than multiplicative ones³. Therefore, when modeling total hardware-induced errors, a worst-case scenario can always be represented by simply adding Gaussian noise to the loss function.

²This particular backend was chosen as a representative example, as the other two demonstrated comparable behavior, leading to identical conclusions.

³To understand why this is intuitive, refer to the next section, in particular Figure 4.7.

4.3 Finite Sampling Error

Even if fault-tolerant quantum computing became feasible in the near term, one persistent source of statistical error remains: finite sampling. To extract any information from a quantum computer’s calculation, measurements must be performed. The number of measurement “shots”⁴ N directly affects the precision of the estimated expectation $\mathbb{E}_N[\mathbf{C}]$ relative to the true loss value $\langle \mathbf{C} \rangle$.

For the measurement of a single qubit with N shots, the count of “0” outcomes (k_0) (and analogously for “1”) follows a binomial distribution. According to the central limit theorem [103], as $N \rightarrow \infty$, this distribution approaches a normal distribution with mean $\mu(k_0) = Np_0$ and standard deviation $\sigma(k_0) = \sqrt{Np_0(1-p_0)}$, where p_0 is the probability of measuring “0”. For the measured probability $\hat{p}_0 = \frac{k_0}{N}$, the standard error is given by

$$\sigma(\hat{p}_0) = \sqrt{\frac{p_0(1-p_0)}{N}}. \quad (4.1)$$

Recalling that BENQO’s loss value is described by Equation (2.22), it can be estimated as $\mathbb{E}_N[\mathbf{C}] = K \arcsin(\hat{p}_0 - \hat{p}_1)$, where the approximate statistical uncertainty is

$$\sigma_{\text{BENQO}}(\mathbb{E}_N[\mathbf{C}]) \approx \frac{K}{2\sqrt{N}} \quad (4.2)$$

(see Appendix B.3 for details). This demonstrates that the size of the statistical error in BENQO’s loss evaluation critically depends on the choice of K , which itself is linked to the system size n (see Section 2.3.3).

In the usual measurement paradigm for VQAs, such as VQE and QAOA, the cost operator – a linear combination of Pauli terms⁵, such as $\mathbf{C} = \sum_{i=1}^m \alpha_i \mathbf{P}_i$ – is directly used as the measurement observable. Therefore, the expectation value $\langle \mathbf{C} \rangle$ becomes a weighted sum of the expectation values of individual Pauli terms, which can be written as $\langle \mathbf{P}_i \rangle = \sum_{j=1}^{2^n} p_j \langle q_j | \mathbf{P}_i | q_j \rangle$, where $p_j = |\langle \psi(\theta) | q_j \rangle|^2$ represents the probability of finding the system $|\psi(\theta)\rangle$ in the basis state $|q_j\rangle$. These probabilities can only be estimated on a quantum device, with the measurement counts of the basis states following a multinomial distribution. Each measured proportion \hat{p}_j still carries a standard error, as described in Equation (4.1). Using the same error propagation formula as before, the statistical uncertainty of the estimated loss value is given by

$$\sigma_{\text{VQE}}(\mathbb{E}_N[\mathbf{C}]) \approx \sqrt{\sum_{j=1}^{2^n} \left(\sum_{i=1}^m \alpha_i \langle q_j | \mathbf{P}_i | q_j \rangle \right)^2 \sigma(\hat{p}_j)^2}. \quad (4.3)$$

Since the two analytical expressions depend on the loss value itself, which varies across the three algorithms, they do not allow for a straightforward general comparison. Therefore, experimental studies were conducted to provide an empirical comparison. Figure 4.4

⁴Note, that the notation for measurement shots in Chapter 3 is n_{shots} . Instead, N is used here for better readability.

⁵This refers to a tensor product of Pauli matrices (\mathbf{X} , \mathbf{Y} , \mathbf{Z} , and \mathbf{I}) acting on different qubits (see Section 2.1.1).

illustrates the behavior of the RAE (3.5) for each VQA under consideration, as a function of the number of shots N (at a fixed system size) and the system size n (at a fixed number of shots). As expected, the error decreases⁶ proportional to $1/\sqrt{N}$ for fixed n . Interestingly, although the errors of VQE and QAOA are described by the same formula (see Equation (4.3)), their RAE for increasing shots differ by orders of magnitude in both visualizations. This discrepancy stems from the mean absolute deviation (MAD) (3.3), which is related to the variance of the algorithms' loss functions – the denominator in Equation (3.5). The smaller the variance of a loss function, the larger the RAE will be when influenced by an error source.

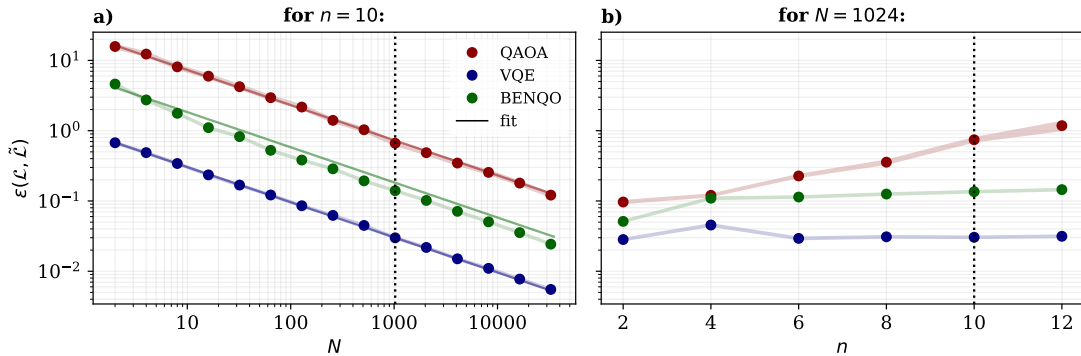


Figure 4.4: Comparison of Finite Sampling Errors of three VQAs. The relative absolute error arising from finite sampling of the loss function is plotted on a logarithmic scale. a) shows the RAE with varying numbers of shots N for $n = 10$, with their corresponding $1/\sqrt{N}$ -fit functions. b) displays the RAE for increasing system sizes n at $N = 1024$ shots (which is a default in many experimental computations). The vertical dotted lines indicate the region of overlap between the plots. All displayed RAE values are calculated from 1000 loss evaluations of a particular problem instance (for each n), sampled uniformly across the full parameter space.

While the scaling of the error in N can be derived directly from the analytical expressions in Equations (4.2) and (4.3), the scaling with respect to system size n is more intricate (see right plot in Figure 4.4). The errors increase linearly for all algorithms, with QAOA's loss function exhibiting the largest finite sampling error. However, as this is a semi-log plot, the linear curves actually indicate an exponential increase in statistical error given a fixed number of shots as system size grows. This sharp increase likely stems from the behavior of the loss function itself, particularly its underlying MAD. For a quantitative analysis of how the mean absolute deviation scales with system size across the three algorithms explaining this behavior, see Section 4.4. This exponential scaling and its detrimental effects will be revisited in Section 5.5.2.

Again, it is crucial to characterize how the statistical error is distributed at each point of loss evaluation in parameter space. Figure 4.5 shows the distribution of the absolute normalized errors $\delta\mathcal{L}$ when sampling the loss at a randomly chosen point $\theta \in \mathbb{R}^n$ for

⁶For a log-log-plot like Figure 4.4, lines represent power-law functions, such as $f(N) \propto N^{-1/2}$.

an example system of size $n = 6$ with $N = 1024$ shots, across all three VQAs. These distributions are compared graphically to a normal distribution with a mean of zero and the same standard deviation as the data. The close visual agreement between the histogram and the theoretical curve validates the central limit theorem, which predicts that for a sufficiently large number of shots, the binomial (or multinomial) distribution from finite sampling approaches a normal distribution.

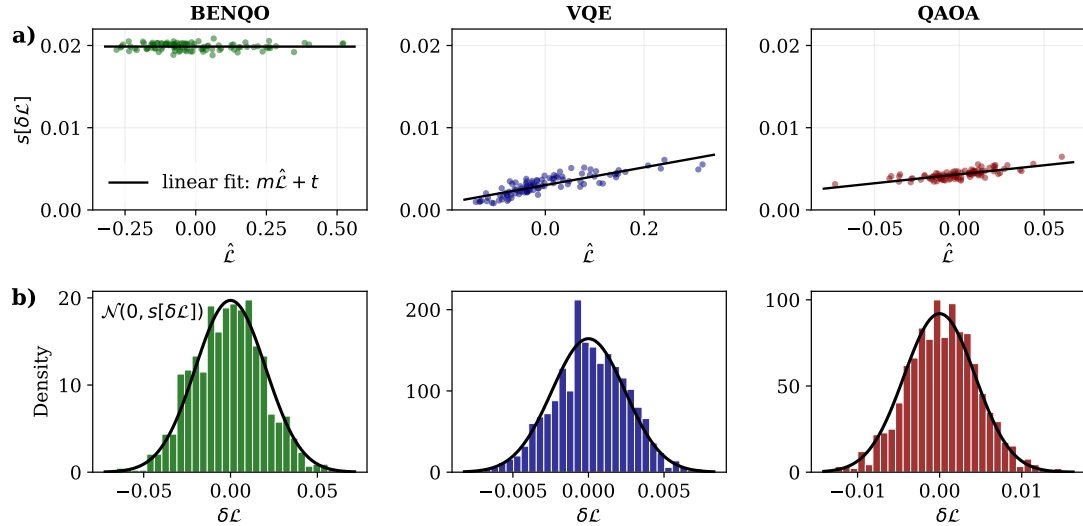


Figure 4.5: Distribution of Finite Sampling Errors Across Loss Landscape. a) shows the standard deviation of the errors of 1000 sample loss evaluations done with 1024 measurement shots for 100 randomly selected points across the loss landscape of a $n = 6$ problem instance against the normalized loss value at those points. Linear fits indicate the relationship between the loss value and the overall magnitude of the finite sampling error for each of the three algorithms. b) are the probability density histograms of 1000 absolute normalized errors for one of these points, including the theoretic curves of a normal distribution around 0 with the same standard deviation.

The remaining question is whether this approximately Gaussian error behaves in an additive or multiplicative manner. Analogous to the approach of the previous section, the standard deviation of the absolute normalized errors $\delta\mathcal{L}$ (over 1000 trials) was calculated for 100 points across the loss landscape and plotted against the corresponding exact normalized loss values $\hat{\mathcal{L}}$ at the same points, as shown in Figure 4.5. For both VQE and QAOA, the relationship between the error magnitude and the underlying loss value is predominantly multiplicative, as the slope of the fitted line is larger than the y-intercept. This observation aligns with the analytical expression of the error in Equation (4.3), which explicitly indicates proportionality between the loss at a given point and the resulting statistical error.

In contrast, BENQO exhibits an additive error pattern, as the magnitude of its errors

(reflected in $s[\delta\mathcal{L}]$) remains nearly constant across the entire⁷ range of loss values. This additive nature is also supported by the analytical formulation in Equation (4.2), which depends only on K , a constant that is independent of the evaluation point. It is intuitive that additive errors will likely cause greater distortion to the exact loss landscape than multiplicative errors. This point is revisited and illustrated visually in the next section.

In conclusion, this analysis suggests that the statistical error in the loss value arising from finite sampling (inherent in quantum evaluations) can, for a realistic worst-case scenario, be reasonably modeled by additive Gaussian noise.

4.4 Loss Landscapes

The optimizability of a variational quantum algorithm, which is the focus of the next chapter, is fundamentally tied to the underlying loss landscape – a topic that has been extensively explored in the literature [20, 27, 93, 104, 105]. While this section does not seek to provide the same level of depth as these works, its aim is to offer an intuition into the optimization challenges posed by VQAs, emphasizing that their difficulties extend beyond merely managing noise.

Figure 4.6 shows illustrative examples of loss landscapes of the three algorithms for different problem sizes n . For these visualizations, the loss $\mathcal{L}(\boldsymbol{\theta})$ was plotted along two random orthonormal directions, \mathbf{v}_1 and \mathbf{v}_2 , projected onto the parameter space $\boldsymbol{\theta} \in \mathbb{R}^n$, like

$$\boldsymbol{\theta}(\vartheta_1, \vartheta_2) = \boldsymbol{\theta}_0 + \vartheta_1 \mathbf{v}_1 + \vartheta_2 \mathbf{v}_2, \quad (4.4)$$

with the origin chosen as the base point $\boldsymbol{\theta}_0$. Note that for the particular plots in Figure 4.6, the overall parameter space of $\boldsymbol{\theta}$ was restricted to the range⁸ $[-4\pi, 4\pi]^n$ and the loss value $\mathcal{L}(\vartheta_1, \vartheta_2)$ was calculated at about 10 000 points $(\vartheta_1, \vartheta_2)$. Note that the same technique has been employed in prior studies [27, 59, 106], with the advantage of offering a way to examine the general structure of the landscape without bias from the underlying parameter relationships compared to PCA⁹.

When examining the plots in Figure 4.6, several observations can be made: First, all landscapes exhibit point-symmetry around the origin. This symmetry is intuitive, as the parameter values in VQAs represent angles of rotational gates, which are based on trigonometric functions that are inherently periodic. This periodicity naturally introduces symmetry and leads to the degeneracy of local extrema. Additionally, one can observe that the landscapes of VQE and BENQO are nearly identical in their structure, though their underlying loss ranges differ. This similarity stems from the role of ansatz parameters in the loss evaluation, which is the same for both algorithms (cf. Sections 2.3.2 and 2.3.3). In contrast, the QAOA loss landscape appears way less structured, to the

⁷Note that, since only 100 points were investigated for this proof-of-concept experiment, the observable range is smaller. If this were the full range, the normalized loss values $\hat{\mathcal{L}}$ would extend up to 1.

⁸This extended range, centered around the origin, was chosen, such that possible symmetries in the landscape would be more clearly visible.

⁹PCA (principal component analysis) is a dimensionality reduction technique that reduces the parameter space to the directions of maximum variance.

point where it is difficult to visually identify any distinct local minima. This chaotic landscape has been noted in other studies [100, 105, 107], highlighting the challenges QAOA faces in optimization. Lastly, note that although these plots range from $n = 4$ to $n = 10$, the landscapes of different system sizes are not directly comparable. This is because each plot corresponds to a different problem instance and is projected onto distinct axes within the respective parameter spaces. However, there are proposals on how to conduct a statistical analysis of the structural changes in loss landscapes that arise as the parameter space (or system size) increases [105]. Such an analysis, though relevant, was beyond the scope of this thesis.

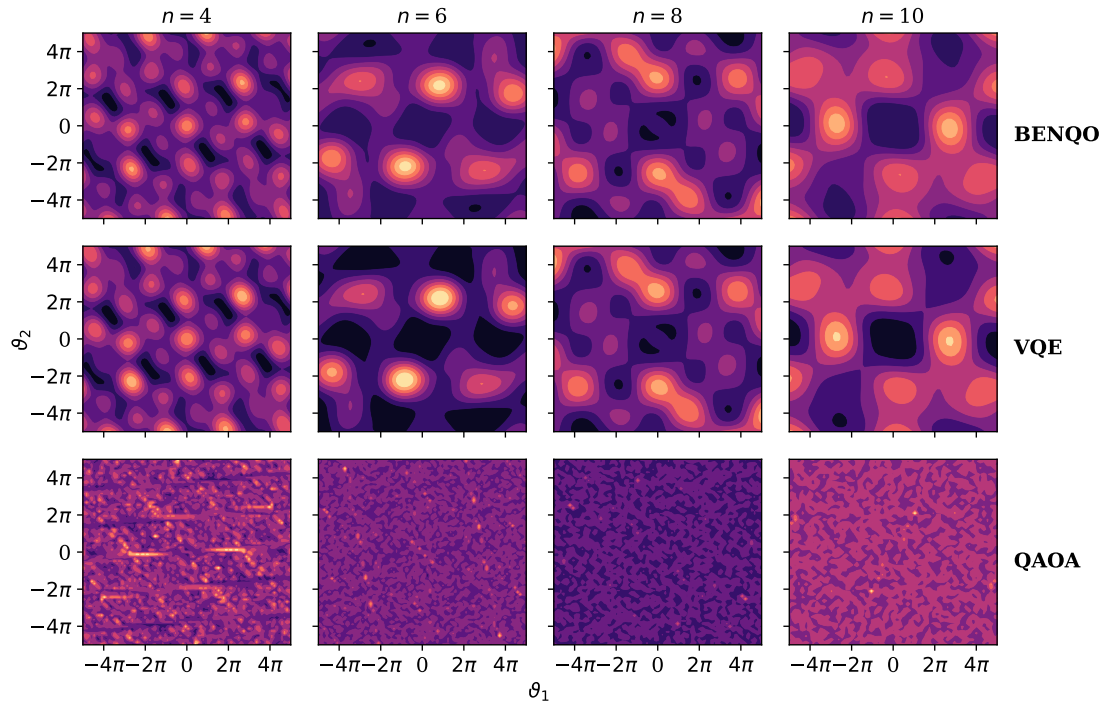


Figure 4.6: Illustrative Loss Landscapes of three VQAs Contour plots of the loss values along two randomly chosen orthonormal axes in parameter space are shown for random Ising problem instances of size n . Light tones represent higher values, while darker tones indicate lower values. Each column uses the same problem instance and the same orthonormal axes through the parameter space, to enable a direct comparison between the algorithms. Note that the scale of the colorbar is specific to each plot.¹⁰

In the previous two sections, it was explored how noise from hardware (see Section 4.2) and finite sampling (see Section 4.3) affects the output of a variational quantum algorithm. Both analyses distinguished between multiplicative and additive Gaussian noise.

¹⁰This decision was made due to the significantly different distributions of loss values across the three algorithms. A shared color scale would have obscured many subtle contours in QAOA. Each plot, therefore, has its own colorbar, ranging from the minimum to maximum possible loss value within its respective space. For a quantitative comparison of the loss values, refer to the end of this section.

To better understand these two representations of noise, Figure 4.7 illustrates their differing effects on an exemplary loss landscape. From this visualization, it becomes immediately clear that Gaussian noise with the same standard deviation σ has a far greater impact when added directly than when applied multiplicatively. This makes sense, as the effect of multiplicative noise depends on the loss value itself; when the loss value is zero, the noise has no impact. However, this characteristic might make multiplicative noise less realistic or representative, as was previously argued.

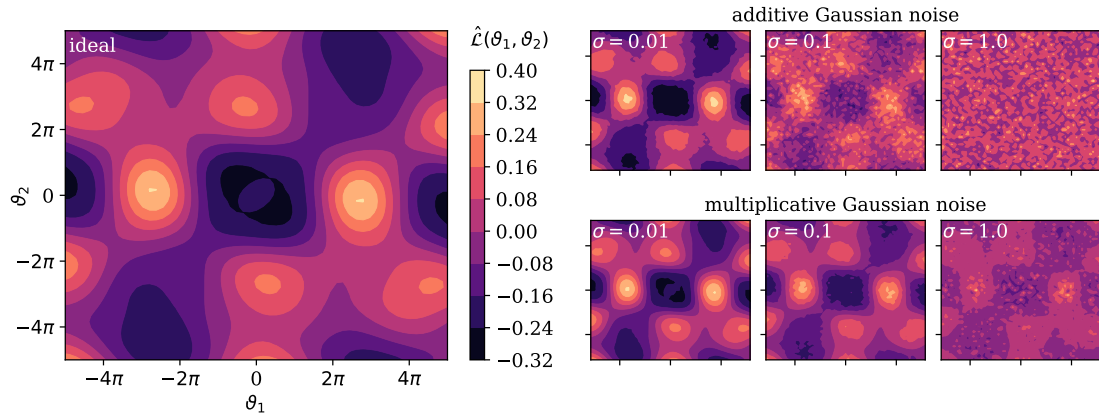


Figure 4.7: Illustrative Loss Landscape with Two Representations of Noise. The left plot corresponds to the exactly calculated normalized loss values $\hat{\mathcal{L}}$ along the two random orthonormal axes, as outputted by the 10-qubit BENQO of an arbitrary Ising problem (cf. Figure 4.6). On the right side, Gaussian noise $\sim \mathcal{N}(0, \sigma)$ is applied at different scales σ . In the upper part, the noise is directly added to the exact loss values, while in the lower part, the noise is first added to 1 and then multiplied with the exact values.

Finally, it is not only the structure of the landscape that can hinder optimization, but also the total range of values, which can negatively impact the optimizability of the underlying parameters. As comprehensively reviewed by Larocca et al. [91], the Barren Plateau (BP) phenomenon (see Section 2.3.5), describing the exponential concentration of the loss function and its gradient, is a significant challenge in variational quantum architectures. Several methods exist for analyzing the variance of a quantum loss function to detect a BP [91], with the simplest approach being an empirical study of how the sample variance of the function and its gradient scales with the number of qubits, as demonstrated by McClean et al. [93], Larocca et al. [100]. Given a BPs detrimental impact on the trainability of parameterized quantum circuits, this heuristic analysis was also applied to the algorithms under scrutiny.

Figure 4.8 presents the results of this examination for the three individual loss functions. For consistency with Chapter 5, the MAD (see Equation (3.3)) was chosen for the depiction, rather than the sample variance (3.2). Both an exponential and a power-law function were fitted to the data to assess the decay behavior of the MAD values:

- $f_{\text{exp}}(n, k, \gamma) = k \cdot e^{-\gamma n}$
- $f_{\text{pl}}(n, k, \gamma) = k \cdot n^{-\gamma}$

To confirm the presence of a BP in a loss function, both the loss and its partial derivatives should exhibit an exponential concentration with increasing system size (cf. Section 2.3.5).

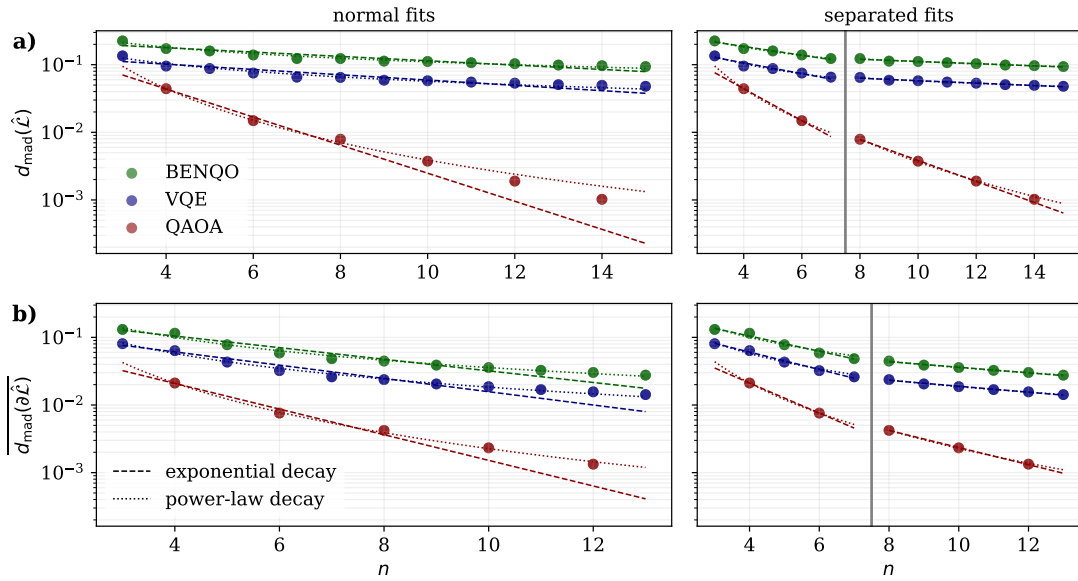


Figure 4.8: Mean Absolute Deviation of Loss Values over System Size. a) shows the MAD d_{mad} (as defined in Equation (3.3)) of the normalized loss values $\hat{\mathcal{L}}(\theta)$ across 10 000 different parameters sets θ , uniformly sampled within $[-2\pi, 2\pi]^n$, for increasing system sizes n . b) shows the mean MAD values over all n partial derivatives $\partial_{\theta} \hat{\mathcal{L}}(\theta)$, calculated as in Equation (2.23), at the same 10 000 points. Both plots include fitted theoretical curves representing exponential and power-law decay. On the right, the data for $n \geq 8$ is treated separately, where the fitted theoretical curves align more closely with the observed data.

Table 4.2 shows the mean squared error (MSE) (3.4) of the corresponding fits, providing a quantitative comparison between exponential and power-law decays. Interestingly, a slight change in decay rate can be observed for system sizes $n \geq 8$ in Figure 4.8. When applying separate fits for the two regions, the underlying curve is therefore captured more effectively (see also MSE-values in Table 4.2). This shift could be related to the nature of the loss function or the characteristics of the problem at larger sizes. The potentially slower decay for larger n (see also Table 4.3) could be more favorable for optimization. Both sets of fit parameters will therefore be discussed further in Section 5.5.

As evident from Table 4.2, the power-law curve consistently results in smaller fit residuals than the exponential for nearly all VQAs. The only exception is the QAOA for system sizes $n \geq 8$, where the exponential curve provides a better fit. This observation holds for both the MAD of the loss values and the mean MAD of the partial derivative values. Consequently, it is not possible to conclusively demonstrate the presence of a barren plateau for any of the circuit architectures tested. However, based on their structural design and the summary of BP causes provided by Larocca et al. [91], it is highly likely that the quantum loss functions of all considered algorithms exhibit

curve		BENQO		VQE		QAOA	
		all n	$n \geq 8$	all n	$n \geq 8$	all n	$n \geq 8$
$\hat{\mathcal{L}}$	exp	1.8×10^{-4}	2.6×10^{-6}	0.9×10^{-4}	0.8×10^{-6}	1.5×10^{-6}	5.5×10^{-9}
	pl	3.7×10^{-5}	1.6×10^{-6}	2.2×10^{-5}	4.6×10^{-7}	2.3×10^{-7}	5.9×10^{-9}
$\partial \hat{\mathcal{L}}$	exp	5.9×10^{-5}	3.7×10^{-7}	2.0×10^{-5}	1.2×10^{-7}	5.7×10^{-7}	4.2×10^{-10}
	pl	3.0×10^{-5}	1.1×10^{-7}	5.1×10^{-6}	3.2×10^{-8}	3.7×10^{-8}	1.3×10^{-9}

Table 4.2: Mean Squared Error of Fitted MAD Decay Curves. The table summarizes the mean squared error (MSE) for both the exponential (exp) and power-law (pl) fits, as presented in Figure 4.8. The results are shown for the full data set and the separated scenario ($n \geq 8$), across all three VQAs. The goodness-of-fit measure is reported for both the decay of the loss values and the partial derivatives.

exponential concentrations as the system size n increases.

	BENQO		VQE		QAOA	
	all n	$n \geq 8$	all n	$n \geq 8$	all n	$n \geq 8$
exponential	0.07(1)	0.04	0.09(1)	0.04	0.48(3)	0.36(1)
power-law	0.55(3)	0.41(2)	0.65(4)	0.46(2)	2.65(7)	3.49(9)

Table 4.3: MAD Decay Parameters of Different Theory Curves. The γ parameters of both fitted theory functions, $f_{\text{exp}}(n, k, \gamma)$ and $f_{\text{pl}}(n, k, \gamma)$ are shown with their respective uncertainties. The results are shown for the fits on the full data set and the separated scenario ($n \geq 8$), across all three VQAs.

Regardless of this, the decay parameters for $\hat{\mathcal{L}}$, summarized in Table 4.3, are noteworthy. Among the algorithms, BENQO demonstrates the most favorable behavior with respect to the variance of its loss values across parameter space, even as the system size grows. Conversely, QAOA shows the worst loss concentration.

4.5 Preliminary Conclusions

This chapter explored the error sources and general behavior of loss functions produced by three representative variational quantum algorithms. Through extensive simulations, the impact of both hardware noise models and finite sampling errors was investigated, enabling not only a direct comparison of the considered VQAs, but also providing insights into an efficient way to model the total error in a quantum loss function. Before proceeding to the main chapter of this thesis, the key results and findings of these analyses are summarized:

Hardware errors are often biased, resulting in approximately Gaussian distributions that are not centered around zero for each point in parameter space. It turned out that these errors tend to have a higher multiplicative component; however, for some algorithms, such as BENQO, the additive component clearly dominated.

In realistic scenarios, the statistical error from finite sampling must also be accounted for. It was argued and shown experimentally that this error can be well-approximated by a Gaussian distribution with zero mean. Particularly for the loss function of BENQO, the total error remained predominantly additive. Given the trend of hardware improvements towards reducing error rates, it can be argued that in the fault-tolerant future, finite sampling error should dominate all other noise sources. Thus, modeling the total error as additive Gaussian noise becomes a realistic approach, which should well capture the severity of errors for the performance of classical optimizers when faced with a noisy quantum loss function.

While the examination of quantum resources through the comparison of circuit complexities showed that BENQO might not be an efficient algorithm in practice (especially not on superconducting hardware), the investigation of quantum loss landscapes revealed that BENQO has the most favorable landscape characteristics for classical optimization. It exhibits the highest variance and slowest decay in the variance of its loss function among the algorithms studied, making it the best choice for the classical optimizer studies presented in Chapter 5.

Finally, the methodology applied in this chapter provided a general framework for comparing multiple VQAs. If neither the quantum loss landscape nor its resilience to noise exhibits favorable characteristics, it raises the fundamental question of whether such algorithms can be expected to perform well on complex problems. Understanding this early on is crucial to avoid unnecessary comparisons of algorithms that are unlikely to yield meaningful results for practical applications.

Chapter 5

Limitations of Classical Optimizer Performance

This chapter presents the main contribution of this thesis: a detailed investigation into the scaling behavior of the performance of classical optimizers within the variational pipeline (cf. Figure 3.1) with respect to system solvability. Building on the analysis of quantum-inherent challenges and errors in the previous chapter, it was demonstrated that the total error in variational quantum loss functions can – to account for worst-case scenarios – be modeled effectively as additive Gaussian noise. Now, the focus shifts to the performance of classical optimizers when confronted with such noisy functions. This study uses the well-behaving quantum loss function generated by BENQO (see Section 2.3.3) as was motivated in Section 4.5.

To frame this investigation, Section 5.1 provides a brief overview of related research, identifying a key gap that this work addresses: the lack of dynamic, scaling-focused analyses under varying conditions. Rather than simply comparing different classical optimizers under a fixed setup, here, the solvability of problems is examined across the two critical dimensions: noise level (in Section 5.3) and system size (in Section 5.4). This two-fold analysis leads to the identification of a threshold beyond which classical optimizers become unable to find the minimum. Finally, Section 5.5 explores the broader implications of this threshold, considering both the current state of quantum hardware and the capabilities of classical optimization techniques.

5.1 Related Work

This section demonstrates a gap in the current body of research. Most studies on the classical side of variational quantum optimization focus on comparing various parameter optimizers based on their performance in solving specific optimization problems. Table 5.1 provides an overview of these studies, emphasizing their testing setup.

Nakanishi et al. [72] proposed a new hyperparameter-free optimization method specifically designed for VQAs (see also Section 2.3.4), claiming it converges quickly and is robust against statistical errors. To support this claim, they compared its convergence against five other state-of-the-art optimizers, testing across 1024, 2048, 4096, and 8192 measurement shots for a 5-qubit VQE. However, while one can observe a clear decline in convergence capabilities as statistical errors increased (with fewer measurement shots), this effect was not quantified.

Reference	VQA	Problem	n	N	Noise Tested				Metrics
					FS	GL	HM	RH	
Nakanishi et al. (2020)	VQE	ground state	5	100	✓	✗	✗	✗	\mathcal{L}^* , $\Delta\mathcal{L}$
Sung et al. (2020)	QAOA	MaxCut, Ising	8	50	✓	✓	✗	✗	t , \hat{p}
Lavrijsen et al. (2020)	VQE	ground state	4, 8	100	✗	✓	✗	✗	n_{iter} , $\Delta\mathcal{L}$
Pellow-Jarman et al. (2021)	VQLS	linear system	3-5	100	✓	✗	✓	✗	\mathcal{L}^*
Soloviev et al. (2022)	QAOA	MaxCut	10	25	✗	✗	✗	✗	\mathcal{L}^* , t
Singh et al. (2023)	VQE	ground state	2-10	100	✓	✗	✓	✗	n_{iter} , $\Delta\mathcal{L}$
Palackal et al. (2023)	VQE	TSP	9, 16, 25	50	✗	✗	✗	✓	\mathcal{L}^* , n_{calls}
Bonet-Monroig et al. (2023)	VQE	ground state	8	15	✓	✗	✗	✗	Q_{AR} , $\Delta\mathcal{L}$
Pellow-Jarman et al. (2024)	QAOA	vertex cover	5	100	✓	✗	✓	✗	Q_{AR}

Table 5.1: Summary of Related Optimizer Studies. Overview of research papers that include comparisons of standard QISKIT and SCIPY optimizers regarding their performance within a VQA for a system of n qubits in different error scenarios (with FS: finite sampling error, GL: errors on gate-level, HM: hardware-models, and RH: real-hardware). The rightmost column indicates the main measures of performance analyzed in each study (see Section 3.3 for explanations), and N refers to the number of experimental trials.

Similarly, Sung et al. [102] introduced two new surrogate-model-based optimization methods, comparing their performance in the presence of statistical errors (from finite sampling) and gate errors¹ using an 8-qubit QAOA problem. Each optimizer was fine-tuned with varying resources (e.g., different numbers of measurement shots) and compared based on their convergence to the optimal energy within a target precision. Although this study considered the upscaling of gate errors, it did not specifically address or analyze the common qualitative convergence behavior across all optimizers.

In contrast, Lavrijsen et al. [15] focused directly on the effects of quantum noise on the classical optimizers by examining the noise-affected optimization surfaces for 4- and 8-qubit ground state problems. Their results underscored the weaknesses of certain methods in the presence of noise and emphasized the importance of selecting an appropriate classical optimizer for each task. However, they did not assess the resulting

¹In their analysis, gate error was modeled by a Gaussian random variable added to each evaluation point in parameter space, with the standard deviation as a tuning parameter.

limitations of these methods, instead suggesting that the overall performance of VQAs is likely rather constrained by the quantum part.

Pellow-Jarman et al. [108] provided another comparison of relevant optimizers for specific noisy 3- to 5-qubit problems with the Variational Quantum Linear Solver (VQLS), focussing on loss values. These tests were later extended to a 5-qubit QAOA, where the approximation index was used as a key performance measure [111]. Both studies concluded that the choice of classical optimizer has a significant impact on a VQA's performance, but neither quantified this effect nor provided limiting conclusions based on the observed deficiencies. This pattern is also evident in a comparative analysis by Palackal et al. [59], which evaluated the VQE on real noisy hardware, or Bonet-Monroig et al. [110] who tested 8-qubit ground state problems with finite sampling noise.

Finally, Singh et al. [109] presented a comprehensive benchmark study of optimizers for VQE applications in quantum chemistry, comparing their performance in noisy and ideal settings. Their conclusion suggested that the challenges posed by noise could be mitigated either by employing error correction methods or by developing a new suite of classical optimizers altogether.

As evident from this summary, numerous efforts have been made to analyze the performance of different classical optimizers within the VQA pipeline. However, these studies remain largely static, examining solution methods under fixed conditions. Consequently, none are able to provide a quantitative analysis of how the chosen performance metrics scale with varying error levels or system sizes – both of which are often limited in scope. Note that without a focus on scaling, there is also no way to infer potential limitations of the classical optimization process. Additionally, due to the diverse types of errors considered, the results often lack generalizability, making it difficult to apply them to other error scenarios.

To address these gaps, this thesis tries to give a systematic evaluation of the behavior of well-established classical optimizers when dealing with noisy quantum loss functions of general QUBO problems. In contrast to the static setup of most related research, this thesis introduces a two-fold scaling analysis that includes both the level of error and the size of the system to be solved to quantify the effects on the solution-finding capabilities of the optimizers. The details of the experimental setup are described in the following section.

5.2 Experimental Setup

In this study, noise is introduced as a post-processing step before the loss function is passed to the classical optimizers, leaving the original, simulated, and exact quantum evaluation unchanged. Note that this is different from approaches that model errors on a gate level (see GL in Table 5.1), for which the total error of the output loss will highly depend on the chosen quantum ansatz. Instead, consider the exact quantum loss function $\mathcal{L}(\theta)$ for an arbitrary QUBO problem of size n , as defined in Section 3.1 and produced by BENQO, which is parameterized by $\theta \in \mathbb{R}^n$. Since the optimizer responds only to the total error magnitude, regardless of its source, it is reasonable to model

the combination of arbitrary error types by additive Gaussian noise, as was motivated in Chapter 4. The normalized loss function $\hat{\mathcal{L}}(\boldsymbol{\theta})$, defined in Equation (3.1), is hence disturbed as follows:

$$\tilde{\mathcal{L}}_{\sigma}(\boldsymbol{\theta}) = \hat{\mathcal{L}}(\boldsymbol{\theta}) + \lambda_{\sigma} \quad \text{with} \quad \lambda_{\sigma} \sim \mathcal{N}(0, \sigma), \quad (5.1)$$

In this formulation, the random variable λ_{σ} is drawn from a normal distribution with mean 0 and standard deviation σ . For example, with $\sigma = 0.1$, more than 95% of the sampled values will lie within the range $[-0.2, 0.2]$.

While this method does not capture error propagation at the hardware or circuit level, as explored by studies like Lavrijsen et al. [15], González-García et al. [21], Sung et al. [102], it offers several advantages. It is not only easily scalable but also provides a highly generalizable evaluation by collapsing all potential sources of error into a single noise term. This way, other studies can approximately map their findings to a noise level σ by calculating the standard deviation of their total error, independent of the used hardware or noise model.

For the classical part, an initial screening of 13 state-of-the-art local optimizers available in QISKIT (see Appendix C.1 for reference) was conducted, along with the normalized gradient descent method [69, 77] (as described in Section 2.3.4). They were evaluated based on their performance in solving a specific problem instance, considering both the number of function evaluations required for convergence and their success probability. As a result, five optimizers were deemed unfit for further analysis due to their comparatively bad performance in one of these categories. The remaining eight optimizers – established in both QISKIT [32] and SCIPY [78] – were selected for systematic assessment in optimizing the noisy loss function defined in Equation (5.1). These chosen optimizers, representative of various methodical paradigms, are summarized in Table 5.2.

Name	Description	in QISKIT
NGD	normalized gradient descent	<i>custom</i>
BFGS	Broyden-Fletcher-Goldfarb-Shannon algorithm	L_BFGS_B
CG	Conjugate Gradient method	CG
SLSQP	Sequential Least Squares Programming	SLSQP
SPSA	Simultaneous Perturbation Stochastic Approximation	SPSA
COBYLA	Constrained Optimization By Linear Approximation	COBYLA
NFT	Nakanishi-Fuji-Todo algorithm	NFT
Powell	conjugate direction method	POWELL

Table 5.2: Overview of Tested Qiskit Optimizers. Eight classical parameter optimizers were filtered in an initial screening to be used in the main experiments. Their respective class names in QISKIT are shown in the right column.

To ensure a fair evaluation, the available resources for these optimizers were kept fixed², using the default settings provided in QISKIT (version 0.46.2). For the NGD

²For an alternative analysis focused on resource requirements for achieving a fixed target precision, refer to Sung et al. [102].

optimizer, the number of steps was set to 20, as recommended by Kuete Meli et al. [69]. The initial parameters for each run were sampled from a multivariate standard normal distribution centered at $\mathbf{0} \in \mathbb{R}^n$, which has proven effective in previous analyses [69, 70]. Each experiment – comprising one optimization run with a fixed optimizer, noise level σ , and system size n – was repeated $N = 100$ times using different random problem instances. The primary performance metric used in this study is the probability of success, as in Equation (3.12), which provides direct insights into system solvability.

5.3 Scaling of Solvability under Additive Gaussian Noise

All the experiments described in the following were conducted for system sizes $n \in [3, 10]$ and error levels $\sigma \in (10^{-3}, 10^1)$. For illustrative reasons, all results shown in this section are based on problems of the fixed system size $n = 6$. Section 5.4 later completes these findings with the corresponding scaling analysis in n .

Figure 5.1 displays the behavior of success proportions, \hat{p}_{opt} , $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ (cf. Equations (3.12) and (3.13)), over varying levels of noise σ added to the respective cost function (cf. Equation (5.1)). Note that BFGS and CG are not displayed in Figure 5.1, as they did not find a single optimal solution when faced with noise (even at low levels of $\sigma \sim 10^{-3}$ around 0). Also, SLSQP did not perform well in any of the tested noise settings, so in the following analyses, the results for these optimizers are often not displayed. Interestingly, all the excluded optimizers are gradient-based. The smaller noise resilience of optimizers relying on exact gradients will be revisited and explained in later discussions in Section 5.4.

For the remaining optimizers in Figure 5.1, one can visually identify – using a logarithmic scale on the noise level – that the curves resemble sigmoidal functions constrained by a pair of horizontal asymptotes (as $\log(\sigma) \rightarrow \pm\infty$). A typical example of such a sigmoidal function is the hyperbolic tangent $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, which was used to fit³ the measured curves via

$$p_{\text{fit}}(\sigma, p_u, p_l, b, c) = \frac{p_u - p_l}{2} \tanh(-b \log(\sigma) + c) + \frac{p_u + p_l}{2}. \quad (5.2)$$

This formulation allows estimating the upper and lower limits of performance via the parameters p_u and p_l , as they correspond to the asymptotes

$$\lim_{\sigma \rightarrow +\infty} p_{\text{fit}}(\sigma) = p_l, \quad \lim_{\sigma \rightarrow 0} p_{\text{fit}}(\sigma) = p_u, \quad (5.3)$$

while the parameters b and c quantify the gradient and location of decrease on the logarithmic scale.

To validate the goodness of these tanh-fits, Figure 5.2 shows the distribution of residuals $\hat{p}(\sigma) - p_{\text{fit}}(\sigma)$ aggregated across all $n \in [3, 10]$ (upper half) and their corresponding normal probability plot⁴ (lower half), commonly used as a graphical test for normality.

³It must be noted that as COBYLA's drop seemed to start much earlier than for the other optimizers, noise levels down to $\sigma \sim 10^{-5}$ were added for its fit to derive more reliable parameters.

⁴A normal probability plot compares the theoretical percentiles of the normal distribution versus the sample percentiles, ideally aligning on a straight line.

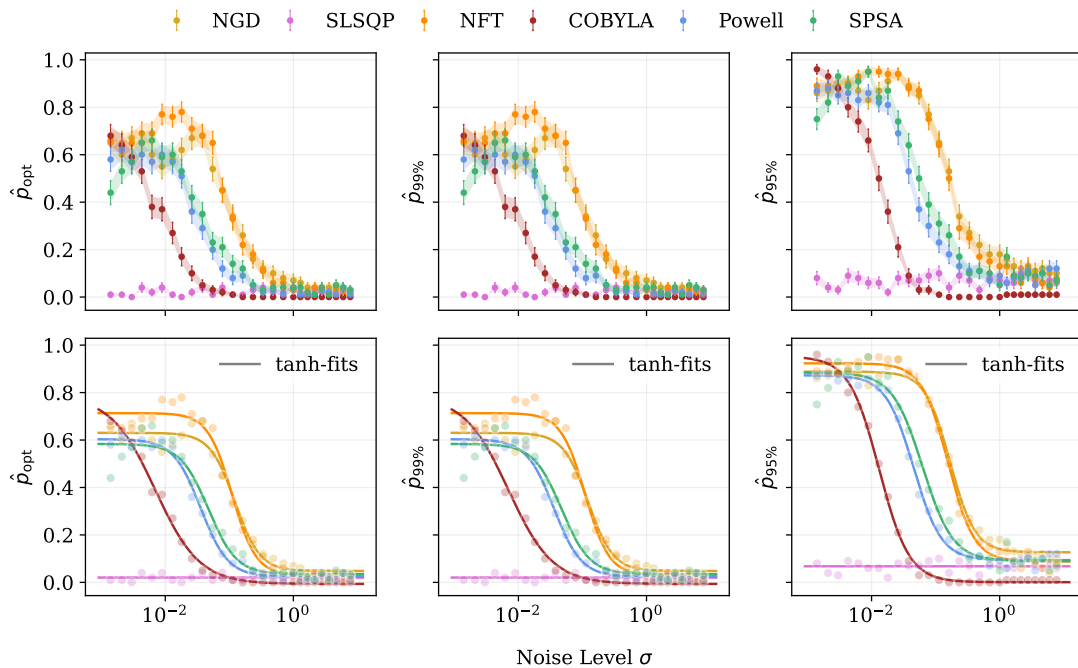


Figure 5.1: Scaling of the Success Probability with the Noise Level. The proportions \hat{p} of optimal (left) and near-optimal (center, right) solutions with increasing noise levels σ on a logarithmic scale are displayed for a suite of classical optimizers. The bottom row shows tanh-fits of the measured data, according to Equation (5.2). Note that due to the bad performance of the SLSQP optimizer under noise, it is fitted by a constant here.

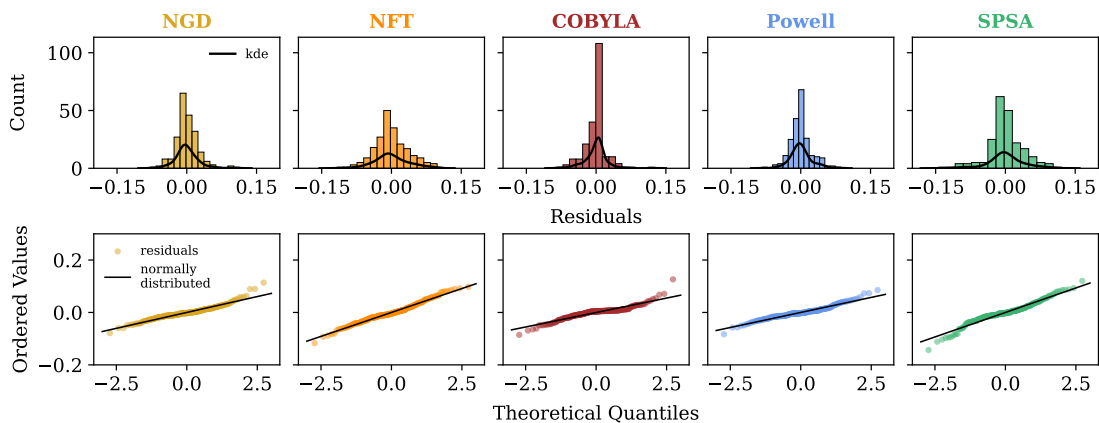


Figure 5.2: Graphical Goodness-of-Fit Analysis. The top row shows the histograms of the fit-residuals for each of the optimizers cumulated for all $n \in [3, 10]$ and their kernel density estimates (KDE) [112]. The bottom row displays the deviation of these residuals from a normal distribution via a normal probability plot (see text).

Both visualizations suggest that the residuals exhibit near-normal behavior, with no significant anomalies visible. It must be noted, though, that for the NFT and SPSA optimizers, the curve of \hat{p} does not strictly follow a sigmoidal function. Instead, both optimizers display a clear peak in performance before the decline (see Figure 5.1), indicating that a small level of noise actually enhanced their ability to find optimal solutions. In this context, noise can be seen as a beneficial feature rather than an obstacle. Consequently, even though $p_{\text{fit}}(0) = p_u$, it is possible that p_u is larger than the estimated $\hat{p}(0)$. This deviation from the expected no-noise scenario was also observed for the Powell and NGD optimizer, suggesting that this phenomenon is not unique to specific local optimizers. In fact, some noise may be able to assist in escaping local minima during the optimization process, as also reported by Branke and Schmidt [113], Stich and Harshvardhan [114], which is why certain algorithms, like SPSA [115], intentionally introduce noise as part of their procedure.

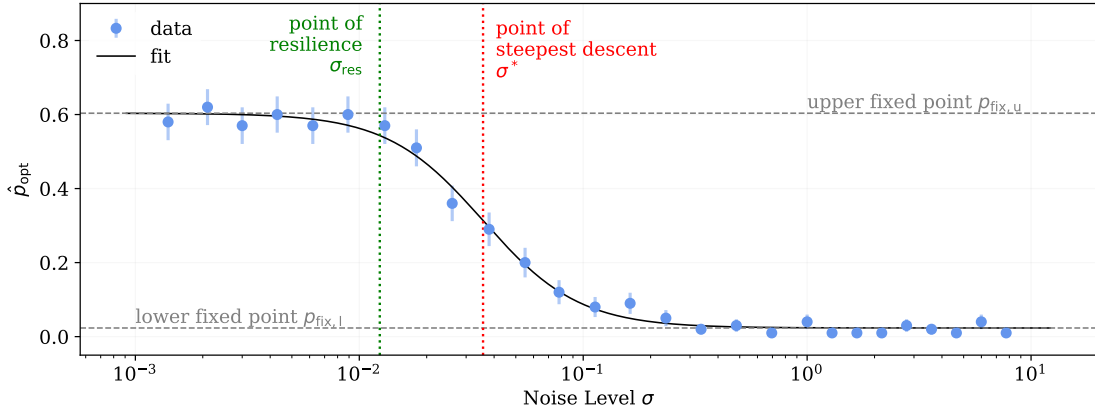


Figure 5.3: Visualization of Important Fit-Parameters. Exemplary tanh-fit of \hat{p}_{opt} of the Powell optimizer with the parameters of Equations (5.4) to (5.8) indicated.

To quantify the scaling behavior of the solvability measure \hat{p} , several measures can be derived from the fit parameters defined in Equation (5.2) (illustrated in Figure 5.3):

- upper fixed point : $p_{\text{fix},u} = p_u$ (5.4)

- lower fixed point : $p_{\text{fix},l} = p_l$ (5.5)

- point of steepest descent : $\sigma^* = \exp \frac{c}{b}$ (5.6)

- slope at steepest descent : $m^* = \frac{b(p_l - p_u)}{2} \exp \left(-\frac{c}{b} \right)$ (5.7)

- point of resilience : $\sigma_{\text{res}} = p_{\text{fit}}^{-1}(0.9 p_u)$ (5.8)

The lower asymptote $p_{\text{fix},l}$ does not necessarily have to be zero. Even in a uniform superposition state, a random basis state (with the highest probability) could be the

optimal solution. However, the likelihood of this is decreasing as $1/2^n$ with the system size n , as also discussed in Section 5.4. The point and slope of the steepest descent, σ^* and m^* , quantify the position and severity of the performance drop. The slope m^* thereby corresponds to the derivative of $p_{\text{fit}}(\sigma)$ evaluated at $\sigma = \sigma^*$. Additionally, the point of resilience, σ_{res} , is introduced as an optimizer-specific performance indicator. It specifies the amount of noise an optimizer can tolerate before its performance drops to 90% of the no-noise level $p_{\text{fix},u}$, marking the onset of performance degradation⁵. Finally, the uncertainty of any such quantity ζ , which depends on the underlying fit parameters p_u, p_l, b, c , is approximated by the Gaussian error propagation formula⁶ of the fit uncertainties $\Delta p_u, \Delta p_l, \Delta b, \Delta c$ as

$$\Delta\zeta = \sqrt{\left(\frac{\partial\zeta}{\partial p_u}\Delta p_u\right)^2 + \left(\frac{\partial\zeta}{\partial p_l}\Delta p_l\right)^2 + \left(\frac{\partial\zeta}{\partial b}\Delta b\right)^2 + \left(\frac{\partial\zeta}{\partial c}\Delta c\right)^2}. \quad (5.9)$$

For $n = 6$, these quantities of all optimizers for the tanh-fits of \hat{p}_{opt} can be found in Table 5.3. The results for $p_{\text{fix},l}$ are not listed due to the high uncertainties of \hat{p} -values near zero. Even though the exact values vary over different problem sizes and when considering $\hat{p}_{99\%}$ or $\hat{p}_{95\%}$ instead (see Section 5.4), this allows for a qualitative comparison of the optimizers which is valid across all variables. Generally, with the hyperparameter settings considered, the NGD and NFT optimizers seem most resilient to noise, having their drop in performance almost an entire order of magnitude of σ later than the rest. While COBYLA starts with the highest success rates in the no-noise limit, its performance drops most rapidly⁷.

	$p_{\text{fix},u}$	σ^*	m^*	σ_{res}
NGD	0.63 ± 0.05	$(1.2 \pm 0.5) \times 10^{-1}$	-2.7 ± 0.9	$(4.8 \pm 2.4) \times 10^{-2}$
SPSA	0.58 ± 0.05	$(4.7 \pm 3.8) \times 10^{-2}$	-5.8 ± 4.0	$(1.6 \pm 1.6) \times 10^{-2}$
COBYLA	0.79 ± 0.04	$(7.0 \pm 3.3) \times 10^{-3}$	-35.7 ± 15.0	$(1.2 \pm 0.7) \times 10^{-3}$
NFT	0.72 ± 0.04	$(1.1 \pm 0.5) \times 10^{-1}$	-3.6 ± 1.2	$(4.3 \pm 2.3) \times 10^{-2}$
Powell	0.60 ± 0.05	$(3.6 \pm 1.4) \times 10^{-2}$	-8.24 ± 2.7	$(1.2 \pm 0.6) \times 10^{-2}$

Table 5.3: Relevant Fit Quantities for \hat{p}_{opt} . The values of $p_{\text{fix},u}$, σ^* , m^* , and σ_{res} (see Equations (5.4) to (5.8)), retrieved from the corresponding tanh-fits of \hat{p}_{opt} , are listed for all optimizers. The “best” values of each column are marked in green.

At last, an explanation for the scaling behavior of the success rates \hat{p} , representative for the solvability of these systems, is derived. With increasing σ , every loss evaluation of $\tilde{\mathcal{L}}(\theta)$ (including those needed to calculate gradients) becomes more and more noisy, such that at a certain level the original loss landscape of the VQA becomes unsolvable.

⁵Note that this is an arbitrary threshold set by the author of this thesis.

⁶Note that Gaussian error propagation assumes uncorrelated errors, which is not necessarily given for the fit parameters. It is instead used as an estimation for the overall magnitude of error.

⁷Note, that COBYLA is only compared to the other four well-performing optimizers here. In general, BFGS, CG and SLSQP must have had the main performance drop for way smaller noise levels, such that no statement about the slope of their descents can be made.

As soon as the standard deviation of the random error λ_σ is higher than that of the loss function itself, any “optimized” result state must correspond to a random guess.

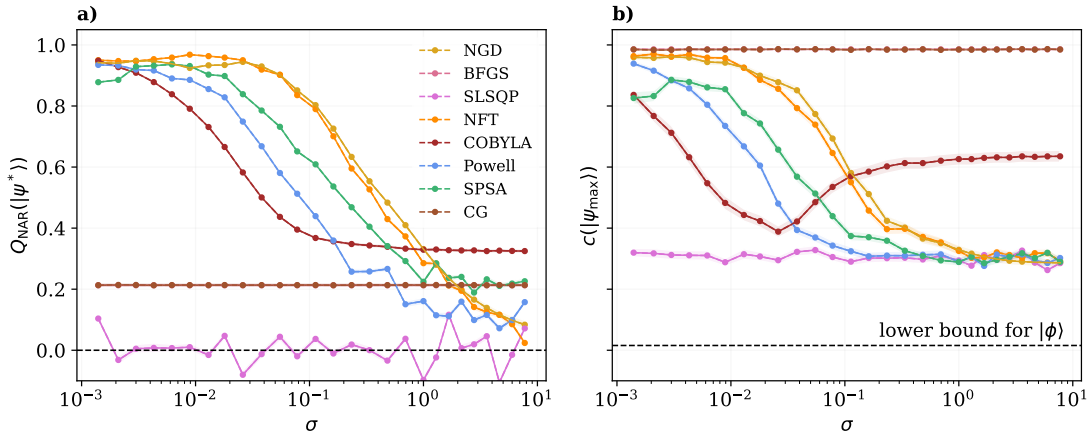


Figure 5.4: Mean Approximation Ratio and Confidence over Error Level. a) displays the mean normalized approximation ratio $Q_{\text{NAR}}(|\psi^*\rangle)$ (3.9) of the optimizers for increasing noise σ (for $n = 6$) over 100 runs. The zero-level of the NAR, corresponding to the uniform superposition solution $|\phi\rangle$, is marked by a dashed line. b) shows the mean confidence $c(|\psi_{\text{max}}\rangle)$ (3.10), respectively. The highest probability of any basis state in $|\phi\rangle$ is $1/2^6 \approx 0.016$, which is marked as a lower bound. Note that the curve of BFGS is hidden behind that of the CG method in both plots.

Figure 5.4 displays the normalized approximation ratio (NAR) of all optimizers’ final solutions $|\psi^*\rangle$ for increasing noise levels σ and the respective confidence c (see Equation (3.10)) of the proposed classical solution $|\psi_{\text{max}}\rangle$, which corresponds to the sample probability of the most probable basis state. The ratio $Q_{\text{NAR}}(|\psi^*\rangle)$ is decreasing similarly to the success probabilities (cf. Figure 5.1), but it is less steep, due to the superposition of various solutions going into the calculation of an AR. While CG, BFGS, COBYLA, and SPSA all seem to have NARs, which converge into some fixed point in the loss landscape, the other optimizers seem to find more and more random solutions. Intuitively, one might expect this to come from uniform superposition states $|\phi\rangle$, which have equal probabilities for every basis state, and therefore a confidence of $c(|\psi_{\text{max}}\rangle) = 1/2^n$. However, Figure 5.4 shows, that instead of converging to a probability of $1/2^n$, which is approximately 0.016 for $n = 6$, the confidence of most of the optimizers converges to the value 0.303 ± 0.002 in the noisy limit. This is because these solutions actually converge to a random parameter state instead of $|\phi\rangle$. Remembering that any solution state $|\psi^*\rangle$ is parametrized by the n independent variables $\theta^* \in \mathbb{R}^n$, one easily sees that the randomness does not come from sampling the probabilities $\{|\langle q_i | \psi^* \rangle|^2\}_{i=1, \dots, 2^n}$ itself, but from uniformly sampling random angles for θ^* . This results in a confidence limit of $(0.5 + \frac{1}{\pi})^n$ (mean maximum state probability), which is approximately 0.3 for $n = 6$, matching the empirical finding. For a derivation of this analytical expression, see Appendix C.2.

5.4 Scaling of Solvability with System Size

In this section, the results of solvability with increasing noise levels are extended to the full set of tested system sizes $n \in [3, 10]$. Figure 5.5 shows a summary of the united results in form of two-dimensional grids with the experimentally measured success probabilities, \hat{p}_{opt} , $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ applied as contours. These visualizations reveal not only the large differences in the behavior of classical optimizers, but also the common trends among all of them: The areas in which finding optimal solutions is feasible and the areas in which this probability rapidly falls to nearly 0 seem to be divided by a diagonal line. This separating line represents the change in position of the point of steepest descent σ^* across different system sizes n . While for \hat{p}_{opt} , even for low noise levels, the success rates are decreasing with growing system sizes, for $\hat{p}_{99\%}$ and especially for $\hat{p}_{95\%}$, these seem to be improving slightly for larger n . This phenomenon will be explored in greater detail later in this section.

First, the transition line dividing the solvable from the unsolvable region is investigated. Instead of taking the values at the discrete measurement points, the parameters derived from the tanh-fits (see Section 5.3) are used to be able to quantify the behavior more exactly. Figure 5.6 shows how the point of steepest descent σ^* , its slope m^* and the point of resilience σ_{res} (cf. Figure 5.3) behave over increasing system sizes. Together, they measure the noise resilience of each optimizer, therefore the visualization allows for a clear ranking regarding their ability to handle noise. NGD and NFT still share the first place, followed by SPSA, Powell, and finally COBYLA (cf. Table 5.3). Such findings on the noise resilience of the optimizers⁸ can also be found in related work [59, 72, 108, 109]. Interestingly, though the NGD and SPSA optimizers are based on gradient information, their way of calculating the derivatives is more resilient to noise than the other gradient-based methods. While the most noise-susceptible methods⁹ rely on exact gradient information, SPSA merely calculates a stochastic gradient and NGD uses a normalized version (see Section 2.3.4). Both seemed to somehow make the algorithms more robust to noise in the loss landscape. For a detailed ranking of the tested optimizers, see Appendix C.3.

Instead of merely ranking the various classical optimizers, a more meaningful analysis concerns their general scaling behavior. As visible in Figure 5.6, this is not only common across all methods, but also was very similar along the different fits of \hat{p}_{opt} , $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$, which is why only the results of the former are shown. The point of steepest descent σ^* and the point of resilience σ_{res} should scale equally¹⁰, as both are specific x-positions of the same tanh-curve. All metrics, including the slope of the descent's tangent line m^* , are clearly decreasing with system size. The remaining question is to determine the nature of this decay, i.e., whether it follows an exponential or power-law pattern. This distinction is crucial, as each decay type carries significant implications for

⁸Except for NGD, which is a custom implementation based on Kuefe Meli et al. [69], and therefore not publically available on SCIPY or QISKIT

⁹Here: CG, BFGS, and SLSQP.

¹⁰Note that σ^* marks the point at which $p = 0.5 p_u$, while σ_{res} is defined as the point where $p = 0.9 p_u$, which should clarify their similarity.

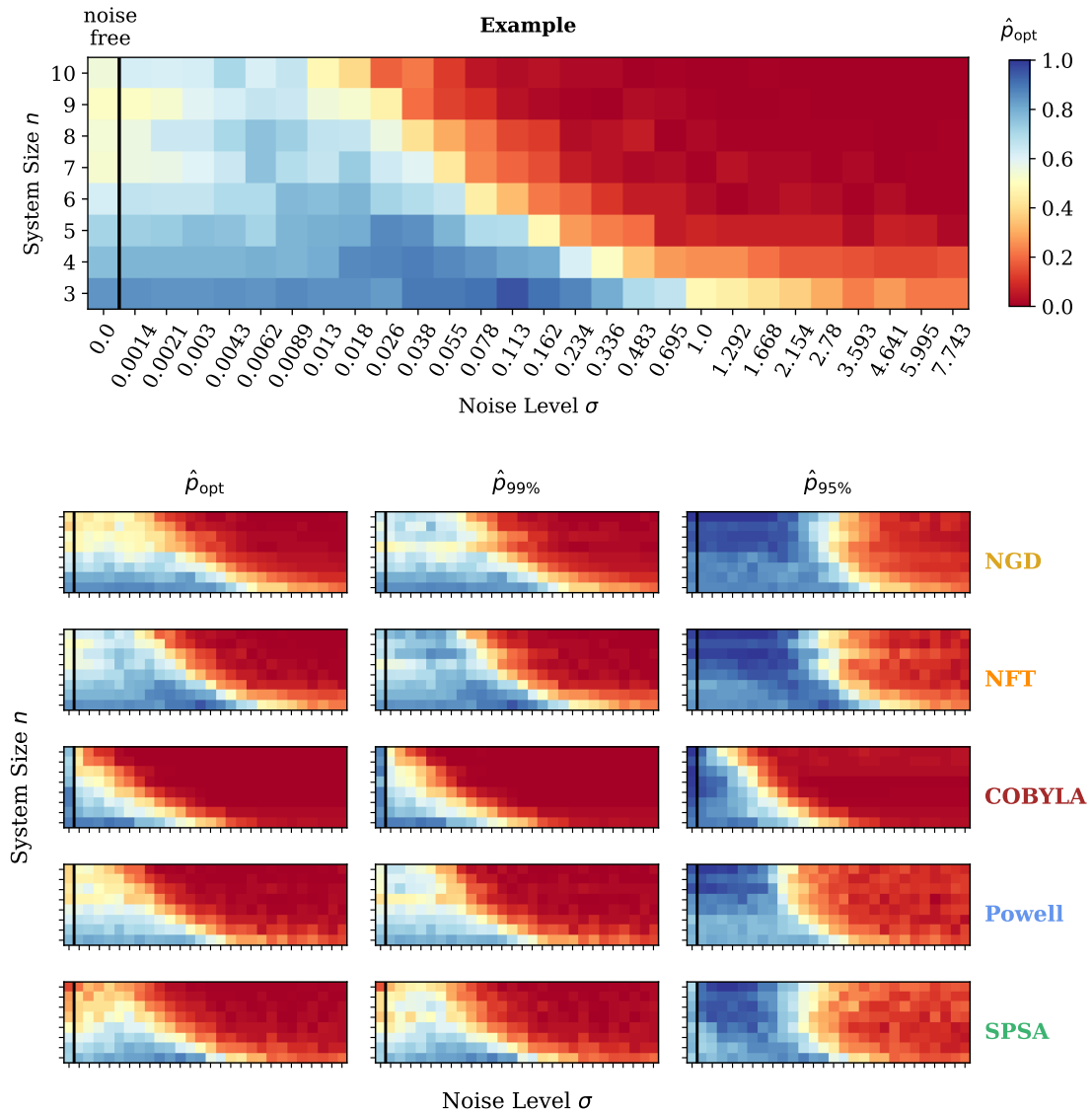


Figure 5.5: Success Probability Across Noise Level and System Size. Contours are depicting the proportions \hat{p}_{opt} , $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ for varying noise levels σ (x-axis) and system sizes n (y-axis). The upper part shows an example of \hat{p}_{opt} of the NFT optimizer with proper axis inscriptions and the corresponding colorbar for all the remaining plots below. Note the logarithmic x-axes of all plots. For completeness, the leftmost grid column of each plot depicts the no-noise results which are not part of the log-scale.

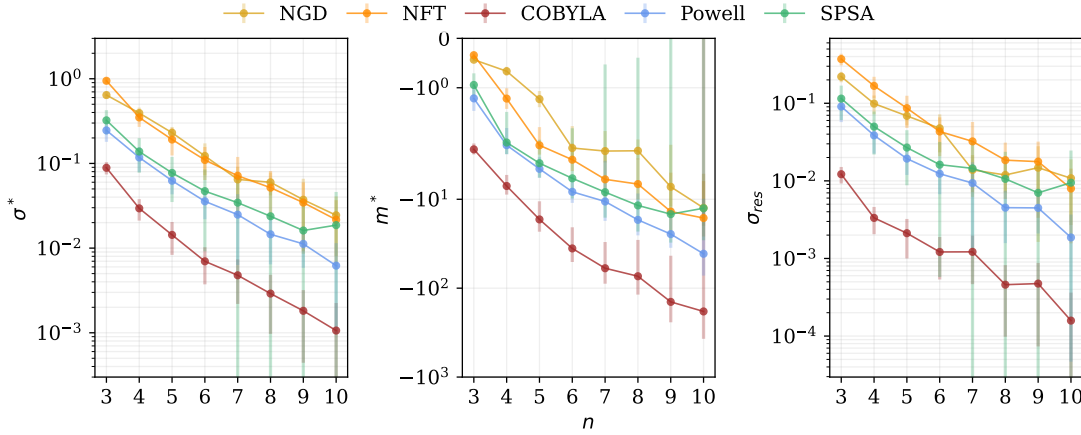


Figure 5.6: Noise Resilience over System Size. The parameters σ^* (point of steepest descent), m^* (slope at steepest descent), and σ_{res} (point of resilience) derived from the tanh-fits of \hat{p}_{opt} (see Section 5.3) are plotted for the experimentally tested system sizes n . Note the logarithmic y-axes of all plots.

the general feasibility of solving relevant problems with VQAs on near-term hardware and beyond (see Figure 5.10).

However, eight data points are generally not enough to make a statement on the exact scaling of a curve, especially not about the asymptotic behavior. It is however possible to compare how well different decay curves fit the measured data. In the following analysis, the behavior of σ^* was chosen as the representative for the fit experiments. Based on the observation of Figure 5.6, three types of decaying functions were tested:

- exponential decay: $f_{\text{exp}}(n, k, \gamma) = k \cdot \exp(-\gamma \cdot n)$
- power-law decay: $f_{\text{pl}}(n, k, \gamma) = k \cdot n^{-\gamma}$
- logarithmic decay: $f_{\text{log}}(n, k, \gamma) = k \cdot \log(n)^{-\gamma}$

As a measure of the goodness of fit, the mean squared error (MSE) (see Equation (3.4)) between the measured $\sigma^*(n)$ and the fitted $f(n)$ over all n is calculated. Table 5.4 presents the results of these experiments (see Appendix C.4 for the corresponding figures). Unfortunately, as the MSE is inconclusive for each optimizer and the suitability also changes between the optimizers, this does not allow for a final statement on the decay of noise resilience with system size¹¹. This may mean that the different optimization methods inherently behave differently regarding their noise susceptibility, but more data (possible with yet more optimizers) would be needed to confirm that. The only conclusion that this data allows, is that no matter the type of decay, it is rapid. For this, regard the sizable γ -parameters which are describing the speed of decay, which were $\gamma_{\text{exp}} \in (0.53, 0.99)$, $\gamma_{\text{pl}} \in (2.2, 3.8)$, and $\gamma_{\text{log}} \in (2.9, 5.0)$ considering all optimizers.

¹¹For this reason, the experiment was also not repeated for m^* and σ_{res} .

	NGD	NFT	COBYLA	Powell	SPSA
exponential	1.0×10^{-4}	1.1×10^{-3}	5.0×10^{-6}	2.8×10^{-5}	1.6×10^{-4}
power-law	6.7×10^{-4}	1.3×10^{-4}	4.1×10^{-7}	5.6×10^{-6}	1.5×10^{-5}
logarithmic	1.6×10^{-3}	8.4×10^{-5}	8.8×10^{-7}	5.5×10^{-5}	1.3×10^{-5}

Table 5.4: Mean Squared Error from the Fit of Decay Curves. The MSE from the fits of various theory curves (left column) are presented for each optimizer’s noise resilience, measured in σ^* . The lowest value in each column is marked for better comparability.

A potential cause for the observed drop in noise resilience with increasing system size is the corresponding decrease in the variance of the loss function (cf. Section 4.4). However, this is not the only factor. Even after normalizing the noise level’s σ parameter to account for the shrinking standard deviation of the loss function across the full parameter space – by using the relative absolute error (RAE) (see Equation (3.5), and the discussion in Section 5.5) – this decline in resilience persists. This becomes evident when comparing the γ parameters from Table 4.3, which are nearly an order of magnitude smaller than those derived here. Therefore, the shrinking variance alone is not the sole factor lowering the noise resilience of classical optimizers:

For instance, Fontana et al. [116] demonstrated that quantum noise can disrupt parameter symmetries, thereby eliminating degeneracies in the loss landscape. Additionally, Wang et al. [20] highlighted the phenomenon of noise-induced barren plateaus, which severely limit the trainability of VQAs. Other factors, such as the growing number of local minima as system size increases in variational loss landscapes [105], and the issue of overparameterization [104], may also play a role. Generally, Leymann and Barzen [25] provide a summary of the challenges associated with the trainability of VQAs, beyond the issue of barren plateaus.

Finally, the scaling behavior of the performance limits of BENQO’s problem solvability, derived from the parameters $p_{\text{fix},u}$ and $p_{\text{fix},l}$, is discussed. To this end, Figure 5.7 displays the curves of both fitted fixed points p_u and p_l , as well as the measured zero-noise success proportions $\hat{p}(0)$ for (near) optimal solutions. This form of presentation allows for a quantitative analysis expanding on the qualitative arguments from Figure 5.5. Interestingly, three of the best-performing optimizers in the zero-noise scenario (i.e., BFGS, SLSQP, and CG) are failing when even a minor amount of noise is added to the loss function¹². Apart from the apparent deviation of $\hat{p}(0)$ and $p_{\text{fix},u}$ for SPSA (as discussed in Section 5.3), both the fitted noiseless limit and the noiseless experimental results are qualitatively behaving the same. The proportion of optimal solutions \hat{p}_{opt} seems to decrease approximately linearly, meaning that not only a constant rate of decay α can be derived from these curves, but also the maximal system size n_{max} , after which finding the optimal solution becomes infeasible.

The resulting linear fit parameters can be found in Table 5.5. Notice that considering the uncertainty, the biggest system possibly solvable even with exactly simulated noise

¹²This apparent trade-off between noise-resilience and no-noise performance is further discussed in Appendix C.3.

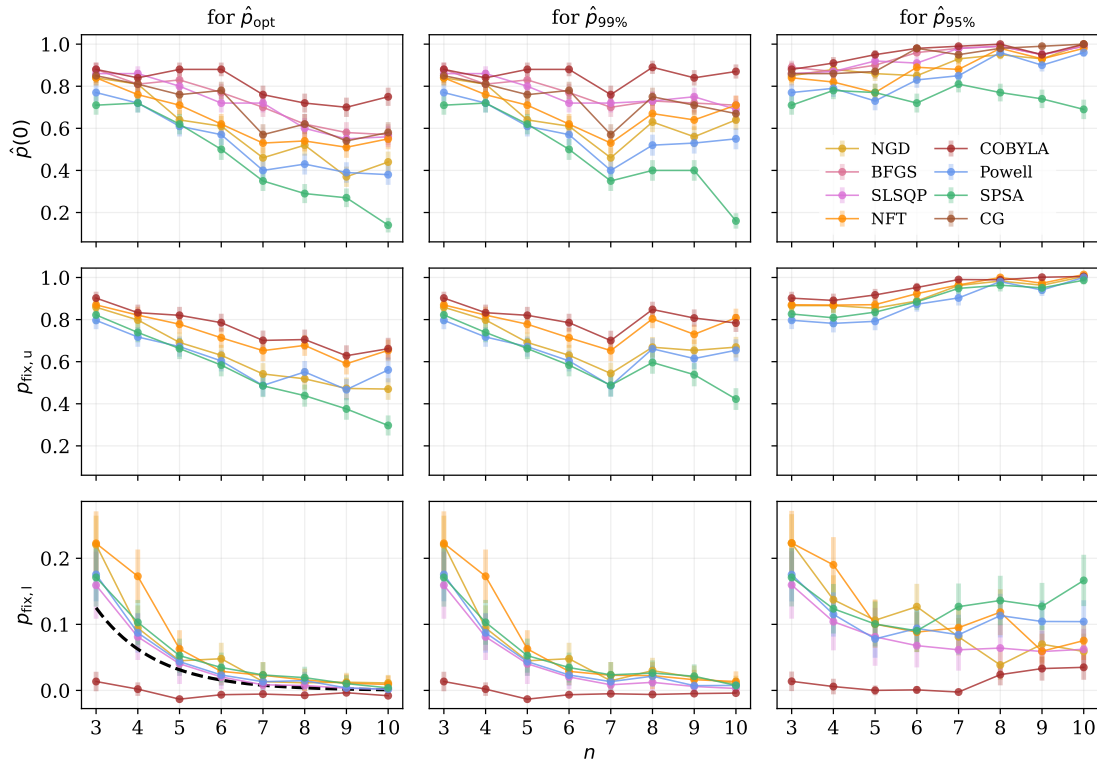


Figure 5.7: Performance Limits over System Size. The measured no-noise success probabilities $\hat{p}(0)$ and the fitted parameters $p_{\text{fix},u}$ and $p_{\text{fix},l}$ for the experimental data of \hat{p}_{opt} , $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ are plotted over system size n . The black dashed line in the lower left plot depicts the theoretical behavior of $1/2^n$, corresponding to random guessing.

values is at $n = 31$ (using the NFT optimizer). This rapid decay can be explained by the exponentially increasing search space the optimizer has to explore to find the minimum. Despite the visual hints for a linear decay, one may still argue that the true asymptotic behavior of these curves is not linear but that they instead converge towards a constant value for even larger system sizes, as it was demonstrated for related experiment with the VQE solver [117]. However, in the study by Díez-Valle et al. [117], the number of quantum function evaluations was not fixed, but instead grew exponentially with system size to achieve convergence.

In contrast to this behavior of exact solvability in the no-noise limit, the success probabilities $\hat{p}_{99\%}$ and $\hat{p}_{95\%}$ seem to be increasing with n for some of the optimizers (see Figure 5.7). A possible explanation of this behavior is the following: Without noise, each of these optimizers should converge exactly to a local minimum of the underlying loss landscape. The question is whether this minimum is in the vicinity of the global minimum. For the chosen normalization of loss values, the maximum always remains 1, while the rest of the loss values concentrate (possibly exponentially) around the mean, which is zero (see discussion in Section 4.4). This concentration has the effect, that most

	NGD	BFGS*	SLSQP*	NFT	COBYLA	Powell	SPSA	CG*
$-\alpha \cdot 10^2$	5.9(6)	4.7(4)	5.1(5)	3.6(6)	3.7(4)	4.0(10)	7.5(2)	4.6(8)
n_{\max}	17(2)	21(2)	20(2)	26(5)	26(3)	21(5)	13(1)	21(4)

Table 5.5: Linear Fit Parameters of No-Noise Solvability with System Size. The upper row displays the percentage decrease of optimal solutions found for each unit increase in n , along with the associated uncertainties. The lower row represents the x-axis intercepts of the linear fits, rounded down to the nearest integer, with uncertainties. Optimizers with an asterisk (*) were fitted using $\hat{p}_{\text{opt}}(0)$, while the rest are based on $p_{\text{fix},u}$ (see center left plot of Figure 5.7).

minima in the landscape will likewise get closer to each other as the system increases, moving most (if not all) of them towards the vicinity of the global optimum, according to Equation (3.11). This argument is revisited shortly.

Lastly, the behavior of the lower performance limit $p_{\text{fix},l}$ in Figure 5.7 is examined. As already detailed in the previous section, in the case where the level of noise exceeds the loss function’s own variance, any solution path should correspond to random guessing. The black dashed line in Figure 5.7 depicts the probability of finding an optimal solution state when drawing randomly from the 2^n sized solution space. One immediately notices that COBYLA’s points are significantly lower than that threshold. This means that, instead of random guessing, it must have converged to a fixed point in parameter space, which is neither optimal nor in the immediate vicinity of optimality. This is confirmed when reviewing Figure 5.4. All other optimizers, however, lie slightly above the given threshold. This is likely due to uncertainties in the tanh-fits, as the modeled noise levels remained strictly below 10^1 , which does not necessarily capture the behavior at infinite noise levels, even though it is already way above $s(\hat{\mathcal{L}})$ (see Section 4.4).

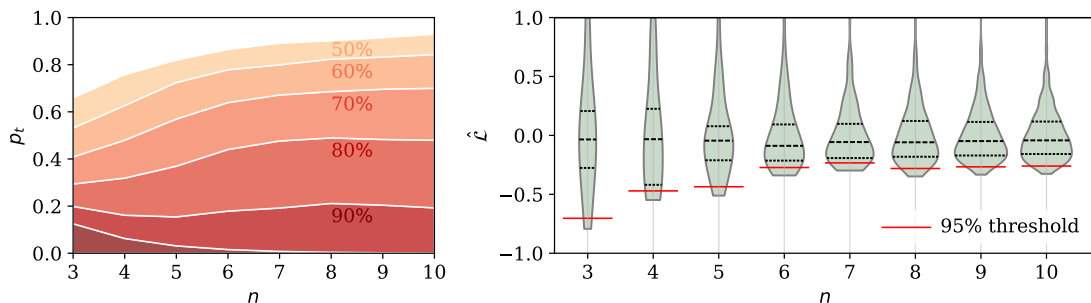


Figure 5.8: Distribution of the Optimality of Solution States. On the left, the mean percentage p_t of solutions resulting in cost values in a relative threshold of t percent near the optimal across system sizes n is depicted as contours. The right shows violin plots representing the distribution of the 2^n possible solution states across the normalized loss landscape $\hat{\mathcal{L}}$. The loss value corresponding to a 95% threshold is marked for each n .

The higher thresholds visible for the lower limits of $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ can be explained by the distribution of solutions regarding their cost value, depicted in Figure 5.8. The

data used for these plots resulted from the same 100 problem instances that were used in the optimizer studies. The left figure shows the percentage of solutions p_t for different thresholds t (cf. Equation (3.11)) near the optimum, calculated as in Equation (3.12), within all 2^n possible classical solution states. Interestingly, this percentage seems to converge for system sizes larger than 7 or 8. More importantly, though, is the finding that almost 20% of all possible (classical) solution states still produce solutions in the 90% vicinity of the optimal cost value. This is due to the distribution of their cost values in between C_{\max} and C_{\min} , of which a normalized version is shown in the right figure. Here, as well, the normalized cost distribution over all basis states seems to converge to a fixed shape, which is left skewed towards the minimum. This non-uniform, non-normal distribution explains why solutions above high t values are more likely (than expected from a normal distribution). This explains the behavior of $\hat{p}_{99\%}$, and $\hat{p}_{95\%}$ in the infinite noise limit.

However, this also raises the question of how practically useful such 99% or 95% optimal solutions truly are and, correspondingly, whether there is a more effective way to define the “success probability” of an optimizer than what was proposed in Equation (3.12). Instead of the true approximation ratio of the most probable solution string, which is used for the success index in Equation (3.11), one may, for example, rather use the normalized version of Equation (3.9) which takes the energy of a totally mixed state as the lower limit. This may already help to lift the meaning of the threshold t . Such considerations will be discussed further in Section 5.5.3.

5.5 Implications and Discussion

In recent years, numerous significant findings have been published regarding the practical limitations of variational quantum algorithms [16, 19–21, 24, 26, 27, 118]. However, in many cases, the task of fully demonstrating the implications of these findings is often left to the reader or deferred to future research. This section, therefore, aims to not only discuss the key insights of this chapter but also illustrate their practical relevance.

5.5.1 Hardware Development

In 2018, Preskill [10] introduced the concept of Noisy Intermediate-Scale Quantum (NISQ) computers and highlighted their potential to solve “real-world” problems with commercial value faster than conventional computers. While significant progress has been made since then, five years later, Ezratty [18] reviewed the current state of quantum devices and emphasized the persistent critical limitations and challenges. To date, no practical use case demonstrating quantum advantage with a NISQ device has been successfully implemented. Despite these hurdles, many researchers and companies still hold onto the belief in the potential of quantum advantage for optimization problems, when using NISQ solutions like variational quantum algorithms.

This chapter has demonstrated how the optimization of VQAs can deal with less and less noise when scaling up the system size. This decreasing noise resilience was quantified in terms of the standard deviation of the total error on the loss function,

after which the optimizer’s performance dropped to less than 50% of its ideal no-noise performance. Among the tested optimizers, the NGD (normalized gradient descent) algorithm, described in Section 2.3.4, was found to be the most noise-resilient.

To generalize these findings, the relative absolute error (RAE) (see Equation (3.5)) is used as an error measure in this section, as it also accounts for the growing concentration of loss values with system size. For the following analysis, the results of the best-performing optimizer, NGD, are considered as the best-case scenario for problem solvability. Figure 5.9 showcases, where the noise produced by IBM hardware models¹³ lies in this respect. According to this graph, problems of more than 6 nodes may already be unsolvable¹⁴ with current hardware. Note, that the error produced by other qubit technologies, such as trapped ions, may be lower for the algorithm under consideration (as discussed in Table 4.1). Nevertheless, the implications of near-exponential scaling in the required precision of loss values are evident: Increasing the number of available qubits is not always the solution; instead, a drastic reduction of error rates would be required. This critical relationship between system size and noise levels was previously highlighted by Stilck França and García-Patrón [19].

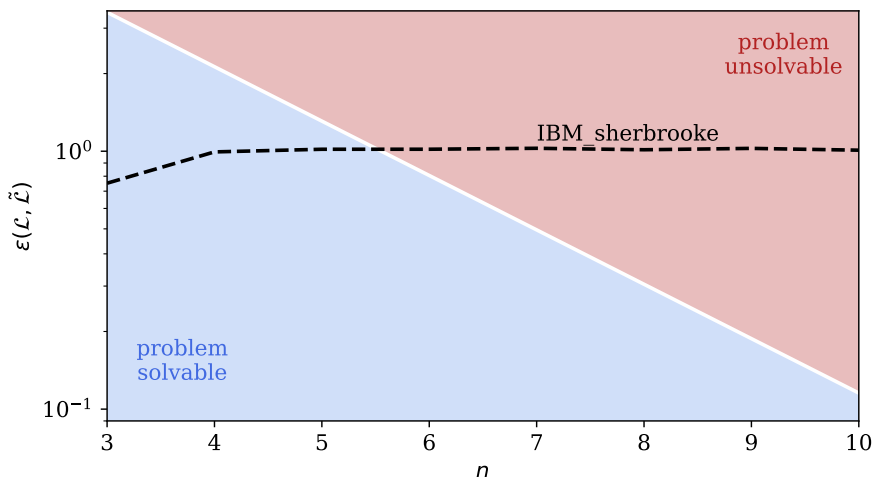


Figure 5.9: Exemplary Hardware Error and Regions of Problem Solvability.

The solid line in the middle represents the exponential fit of the critical error threshold σ^* for the most noise-resilient optimizer, NGD (see Section 5.4), converted into the RAE metric ϵ based on the measurements presented in Section 4.4. Above this line, the probability of the classical optimizer finding an optimal solution becomes highly unlikely or effectively impossible. The dashed line indicates the empirically measured hardware error obtained using the `ibm_sherbrooke` backend (refer to Figure 4.2).

According to IBM’s Development & Innovation Roadmap [119], the Starling processor,

¹³Note, that the noise produced by such a model only captures a very limited amount of error sources. Real hardware may produce even higher errors.

¹⁴In this context, “unsolvable” refers to the classical optimizer consistently finding significantly fewer optimal solutions – less than half of those it would identify in an idealized, noise-free setting. Note that every other threshold would have resulted in the same scaling behavior.

set for release in 2029, is expected to be error-corrected with 200 logical qubits. Similarly, Quantinuum’s roadmap projects a nearly fault-tolerant quantum computer with 1000 logical qubits by 2030 [120]. Similar goals have been outlined by companies such as IonQ, IQM, and Google. This raises the question of whether fault-tolerant hardware will be sufficient to overcome the challenges posed by the scaling relationships and noise constraints discussed in this chapter.

5.5.2 The Curse of Finite Sampling

As pointed out several times, and examined in more detail in Section 4.3, the error produced by finite sampling is unavoidable when using quantum computing architectures. The question is, how do the error boundaries to problem solvability translate to the required number of measurement shots, and what implications this scaling has on the realization of any quantum speed-up? In the following, the results of all previous scaling analyses will be combined, using the most optimistic, best-case scenarios to highlight the detrimental implications these findings have on the feasibility of achieving quantum advantage with variational quantum optimization.

Based on the experiments of Section 5.4, the decline in an optimizer’s noise resilience as the system size n increases, denoted by $\sigma^*(n)$, may follow an exponential, logarithmic, or power-law trend (see again Figure 5.6 and Appendix C.4). To convert the standard deviation σ^* of the most optimistically performing optimizer, NGD, into the general measure of the RAE (3.5), it needs to be normalized by the mean absolute deviation (MAD) of the loss function, measured in Section 4.4 for the BENQO algorithm. The most optimistic scaling for the MAD, which yet provided a good fit, followed a power-law decay at $n \geq 8$, which was less steep compared to when smaller system sizes were included. Using the fitted parameter from Table 4.3, the maximum RAE $\epsilon^*(n)$ an optimizer can handle for a given system size n depends on the scaling law:

$$\epsilon^*(n) = \frac{\sqrt{\frac{2}{\pi}} \sigma^*(n)}{d_{\text{mad}}(n)} \approx \begin{cases} 8.5 \cdot n^{0.4} \cdot \exp(-0.5 n) & \text{for exponential scaling} \\ 22.0 \cdot n^{-1.8} & \text{for power-law scaling} \\ 2.5 \cdot n^{0.4} \cdot \log(n)^{-3.2} & \text{for logarithmic scaling} \end{cases} \quad (5.10)$$

These results, though based on the combination of uncertain numerical fits, emphasize that the drop in noise resilience can not merely be due to the decay in the variance of the loss function but is present regardless of how the empirical data is fitted¹⁵.

Next, one has to derive the number of shots n_{shots} needed for each system size n to achieve an error as small as $\epsilon^*(n)$ for an otherwise error-free quantum computer, which would assure the successful performance of a classical parameter optimizer. Based on the calculations and experiments made in Section 4.3, the RAE of finite sampling at

¹⁵The only scenario where – based on the experimental data – the classical optimizers would become more noise-resilient with increasing system size n , would be if the MAD of BENQO decreased exponentially with n , while the noise resilience σ^* did not – a speculative outcome. However, even this scenario would not alter the overall conclusions of this section.

fixed n behaves as

$$\epsilon_{\text{FS}}(n = 10, n_{\text{shots}}) \approx \frac{5.8}{\sqrt{n_{\text{shots}}}}, \quad (5.11)$$

while the error for a fixed number of shots increases exponentially (cf. Figure 4.4) as

$$\epsilon_{\text{FS}}(n, n_{\text{shots}} = 1024) \approx 0.1 \cdot \exp(0.04 n). \quad (5.12)$$

Assuming that this general scaling behavior is consistent across all sampling rates and system sizes, and defining ϵ_{fix} as the error at $(n = 10, n_{\text{shots}} = 1024)$, where both experimental curves meet, one can derive a function for the RAE in the loss function, that is caused by sampling with n_{shots} at a system size of n :

$$\begin{aligned} \epsilon_{\text{FS}}(n, n_{\text{shots}}) &\approx \epsilon_{\text{FS}}(n = 10, n_{\text{shots}}) \frac{\epsilon_{\text{FS}}(n, n_{\text{shots}} = 1024)}{\epsilon_{\text{fix}}} \\ &\approx 4.0 \cdot \frac{\exp(0.04 n)}{\sqrt{n_{\text{shots}}}}. \end{aligned} \quad (5.13)$$

Note, that while the parameters in this equation are only rough estimates and depend on the reference point ϵ_{fix} , the general scaling trend is representative. For a sufficiently small error in the loss values outputted by a VQA – in this case BENQO –, the condition $\epsilon_{\text{FS}}(n, n_{\text{shots}}) < \epsilon^*(n)$ should hold across all n . By substituting Equation (5.10) and Equation (5.13), it becomes clear that the required number of shots, for a problem to be solvable with a VQA, grows exponentially with system size. Specifically,

$$n_{\text{shots}}(n) > \left(\frac{4.0 e^{0.04 n}}{\epsilon^*(n)} \right)^2 \quad (5.14)$$

is always $\mathcal{O}(k^n)$ for some $k > 1$, regardless of the true scaling of Equation (5.10). Similar findings of an exponential scaling of the required resources are reported in the literature [24, 99, 121].

As pointed out at the end of Section 3.3, the number of shots multiplied by the number of quantum function calls required for one optimization run must remain strictly below 2^n to avoid quantum disadvantage – i.e., the point where classical brute-force solving becomes faster than using a quantum device. The NGD, being a highly resource-efficient optimizer, requires only 20 iterations regardless of n , with each iteration involving one quantum call to evaluate the loss function and $2n$ calls to evaluate the gradient (see Equation (2.23)). For the NGD, the maximum number of allowed shots before encountering quantum disadvantage is therefore given by:

$$n_{\text{shots}}(n) < \frac{2^n}{40n + 20}. \quad (5.15)$$

This means that for problem sizes below 9, a quantum speed-up is impossible, as brute-force solutions are always faster. Figure 5.10 graphically summarizes all these findings for system sizes ≥ 10 .

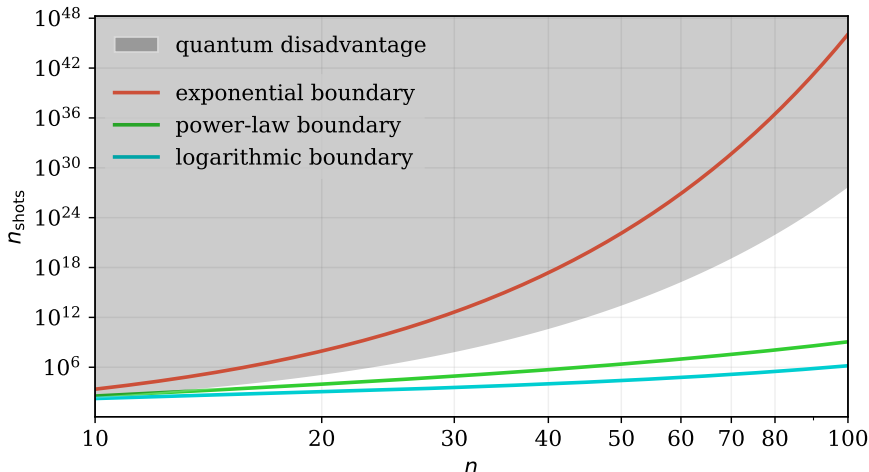


Figure 5.10: Number of Required Shots for Reliably Solving a QUBO Problem of Size n . The minimum number of shots required to reach the three possible error boundaries for the NGD solver are plotted as solid lines (derived from the findings in Sections 4.4 and 5.4). For any value above these theory curves, finding an optimal solution to a QUBO problem of size n with the BENQO solver is feasible. In the shaded area, however, the number of measurement samples exceeds that of brute-force trials, making the quantum solution more inefficient than the worst classical solver.

The figure clearly demonstrates that if the boundary lines identified in Figure 5.6 of Section 5.4 were to follow an exponential drop, the number of shots required for the optimizers to reliably find a solution would grow faster than the number of brute-force trials needed to solve the problem classically. This would suggest that quantum advantage may be fundamentally unattainable for optimization tasks using variational quantum algorithms. In the other two cases (with non-exponential scaling), there remains a narrow window below the critical curve of 2^n for sizes $n \geq 20$ (for the logarithmic boundary) and $n \geq 25$ (for the power-law boundary). This indicates that some sort of advantage is theoretically possible but likely unpractical, considering the exponentially large resources needed to produce it. Considering Equation (5.14), any potential advantage would, at most, represent a polynomial speedup, which only starts at specific system sizes n . Considering the noise-free performance scaling presented in Figure 5.7, these system sizes may already not be solvable by classical optimization methods. One must note that this discussion already holds for a scaling comparison to the worst classical algorithm, without even mentioning actual runtime comparisons between quantum and classical methods or including the resource requirements of the best classical methods, which are well below the 2^n -curve. Such analyses may paint an even darker picture of the practical feasibility of quantum computational speed-ups.

Lastly, it could be argued that these findings are based on only a limited testing setup, which does not allow for such general conclusions, as all the scaling results presented in this chapter are derived solely from the BENQO algorithm. However, the empirical

results of Chapter 4 demonstrate consistent scaling behavior across all the quantum algorithms considered, which are representative of the class of VQAs that many still regard as promising. Moreover, some of the observed behavior has already been reported in the literature for QAOA and VQE [24, 102]. For those who remain optimistic about the potential of variational quantum optimization, this may serve as a robust, empirical method to critically evaluate and potentially rule out new VQA proposals, even if they perform well in small-scale, noise-free simulations.

5.5.3 Relevant Classical Problems

If no leeway exists in improving the classical or quantum side of the variational framework, the problem formulation becomes the next logical frontier of investigation, as illustrated in Figure 3.1. In this thesis, random QUBO instances, as detailed in Section 3.1, were used in the investigations, as this formulation – equivalent to Ising problems [37, 44] – encompasses a broad range of relevant combinatorial optimization problems [46]. However, in the QUBO encoding, the problem size often directly translates to the number of binary variables to solve it, or, in the case of the Traveling Salesperson Problem (TSP), it even scales quadratically. As identified in this chapter, the solvability limits of variational quantum algorithms may be around $n \sim \mathcal{O}(10)$, while industrially relevant problem sizes are orders of magnitude larger. For example, the Concorde solver for symmetrical TSP instances [122] can find optimal solutions for routing problems involving up to 85 900 cities [123]. In the one-hot-encoding of TSP instances described in Section 2.2.2, this would correspond to a system size of $\mathcal{O}(10^9)$ – far beyond the solvability limits identified here¹⁶. However, one should note that some general classical solvers for quadratic programming problems, including QUBO, are benchmarked for problems with only up to $\mathcal{O}(100)$ binary variables [124]. Still, this underscores the considerable power of classical state-of-the-art solvers – which are already available – for industrially relevant problems. It highlights the substantial gap quantum devices would need to bridge in optimization, making the true prospects of quantum advantage using VQAs highly questionable. Chapter 7 therefore includes a brief discussion on research efforts into alternative quantum optimization approaches.

Another point of discussion concerning the problem side is the performance metric, which was used to evaluate solvability. In this thesis, the chosen measure for this was the empirically observed success rate of finding a solution within the t -vicinity of the optimal energy (see Equation (3.12)), with thresholds set at $t = 100\%$, 99% , and 95% . A natural question – raised at the end of Section 5.4 – is how practically useful solutions meeting such thresholds are, as it could be argued that the poor scaling behavior reported in this chapter might only apply to such high thresholds. After all, one might contend that approximate solutions, even those slightly further from optimality, could still be sufficient for many practical applications.

To demonstratively counter this conjecture, the TSP is used as a representative example problem. As described in Section 2.2.2 in order to find feasible solution states,

¹⁶This holds even under the noise-free limits conjectured in Table 5.5.

constraints are added as quadratic penalty terms with a corresponding penalty factor in Equation (2.14) into the QUBO formulation. Depending on the absolute size of this factor, the corresponding loss range can be enlarged substantially as the maximum loss value C_{\max} often reaches values which are orders of magnitude larger than the unconstrained loss evaluation. Consequently, $|C_{\max}| \gg |C_{\min}|$, causing the approximation ratio used in Equation (3.11) to approach 1. This shifts the energy thresholds at which feasible solutions can be identified to significantly higher levels, sometimes even exceeding the 99% threshold applied here [58, 59]. One solution for this would be choosing alternative encodings with higher ratios of feasible solutions [98].

Chapter 6

Conclusions

The potential of variational quantum algorithms (VQAs) – still widely regarded as the leading candidate for achieving near-term quantum advantage in optimization – critically depends on their ability to address the challenges posed by Noisy Intermediate-Scale Quantum (NISQ) hardware. The most pressing among these is noise, arising both from hardware imperfections and finite sampling errors, the latter of which will persist even in fault-tolerant systems. Although some studies have analytically examined the impact of noise on the performance of VQAs [16, 19–21, 24], only few have addressed how this impact scales with growing system sizes. This thesis contributes to filling this gap through a systematic numerical investigation into how classical optimizers – the main computational engine of VQAs – are affected by noise when applied to increasingly large quantum systems. Instead of theoretical conjectures or speculative hints, the findings provided clear, empirical evidence on how noise fundamentally limits the optimizability of variational quantum ansatzes, even in an idealized fault-tolerant setting.

To achieve this, the sources of error and their effects on the loss functions of three representative algorithms – the Variational Quantum Eigensolver (VQE) [62], Quantum Approximate Optimization Algorithm (QAOA) [63], and Block Encoding Quantum Optimizer (BENQO) [69, 70] – were quantitatively and qualitatively analyzed in Chapter 4. One key finding was that, under conditions anticipating continuous hardware improvements up to full error correction, the total error in a quantum loss evaluation could be effectively modeled as an additive Gaussian random variable dominated by statistical sampling errors. Given the decay in the variance of the loss functions among the algorithms studied, BENQO was found to have the most favorable loss landscape characteristics, posing it a best-case scenario for the classical optimizer studies in Chapter 5.

In these studies, a suite of state-of-the-art classical optimization methods was tested on BENQO’s loss function with varying noise levels (scaled by the standard deviation of the associated Gaussian error) and system sizes. For a fixed system size, the probability of finding high-quality solutions¹ decreased in a sigmoidal pattern relative to the logarithmic noise level. This indicated a sharp drop in performance at a critical noise threshold, which, however, differed significantly among the tested optimizers (see Section 5.3). Using tanh-fits, these thresholds – after which the optimizers’ success probability resembled that of random guessing – were extracted for system sizes $n \in [3, 10]$, which revealed a common scaling behavior across all optimizers: their noise resilience exhibited a rapid, potentially exponential, decline with increasing system size. Interest-

¹This includes both optimal solutions and those within the 99% and 95% energy vicinity of the optimum.

ingly, this drop was found to be steeper than could be explained by the corresponding decay in loss variance (barren plateaus) alone (see Section 5.4), and even in a theoretical noise-free limit², performance declined with system size, confirming that larger systems inherently present a more challenging loss landscape.

While it was discussed that current hardware error rates far exceed what would be needed to tackle classically relevant problems (see Section 5.5.1), it was also indicated that the influence of the finite sampling error imposes even stricter limitations (see Section 5.5.2). Extrapolating the observed decrease in the optimizer’s noise resilience showed that, to meet the loss accuracy requirements for system solvability, the number of shots scales exponentially with system size, leaving only minimal leeway before exceeding the resources of classical brute-force sampling, which may already be impractical. These insights raise significant questions regarding the promise of VQAs for optimization.

The contributions of this thesis are multifold. First, a general methodology was established to systematically evaluate and compare the performance of VQAs, providing a foundational framework to assess their potential and viability. Additionally, this work offers, to the best of the author’s knowledge, the first comprehensive and highly generalizable study on the scaling behavior of classical optimizers when applied to noisy variational quantum loss functions for random QUBO (quadratic unconstrained binary optimization) problem instances of increasing size. The experimental results established a clear boundary regarding noise level and system size, beyond which classical optimizers fail to find optimal or near-optimal solutions. This empirical demonstration highlighted how stochastic errors – even when caused by the finite sampling of measurements alone – fundamentally limit the practical viability of VQAs. In fact, translating the combined scaling observations into practical terms hinted toward a general unfeasibility of achieving any quantum advantage in variational quantum optimization. Alongside these significant implications, the study also provided a general performance comparison of state-of-the-art classical optimizers, which revealed the effectiveness of a newer method, normalized gradient descent with an exponentially decaying step size [69]. This algorithm, though not yet included in QISKIT’s suite, demonstrated superior noise resilience, outperforming all other optimizers tested in this context.

Despite these contributions, there may be limitations to the proposed approach and results. One could argue that the observed critical behavior is only due to unlucky choices in the experimental setup, such as specific optimizers, hyperparameter settings, initialization strategies, quantum ansatzes, error modeling, or problem formulations. However, all these choices were carefully justified in this work with mostly best-case instances, and similar findings from other studies further support their generalizability. Nevertheless, additional steps for rigorous validation of these results are outlined in Chapter 7. It should further be noted that this thesis does not address optimization on fundamentally different platforms, such as Quantum Boson Sampling [125, 126] or Quantum Annealing [48, 49]. However, given the hybrid nature of these approaches, which also rely on classical optimization routines, it is likely that the noise-related challenges identified here would impact these systems as well.

²Note that this is unachievable in practice due to finite sampling errors.

Chapter 6 Conclusions

Ultimately, this thesis serves as a framework for rigorously investigating the limitations and scaling behavior of VQAs and other quantum algorithms for solving optimization problems. It should encourage researchers – especially those skeptical of the findings’ generality – to replicate these experiments using alternative setups, whether by testing different quantum ansatzes, optimizers, problem instances, or even other hardware. If any inspired investigation of how problem solvability scales with system size and noise levels would yield qualitatively distinct results, those would be exciting insights for the community and provide promising new research directions. However, if the scaling behavior observed here proves to be universal and intrinsic to the classical optimization problem inherent to hybrid architectures, such as VQAs, it would strongly suggest that achieving quantum advantage through variational quantum algorithms in optimization may not be realistic. In fact, it seems that the challenges facing VQAs – both in the NISQ and fault-tolerant era – are much more fundamental than what can be overcome by simply improving hardware, finding better classical optimizers or tuning the ansatz.

Chapter 7

Outlook

To validate the observed scaling behavior and assess the broader applicability of this thesis’ findings, future studies could replicate the experiments with alternative quantum ansatzes. Since the loss function experiments in Chapter 4 already include analyses with both the Variational Quantum Eigensolver (VQE) and the Quantum Approximate Optimization Algorithm (QAOA) (see Section 2.3.2), it would be sufficient to repeat the optimizer studies presented in Chapter 5 using one of these algorithms to test the generality of the results. Another validation approach could involve directly using finite measurement shots rather than the simplified Gaussian error models, aiming to replicate the findings of Figure 5.10. Likewise, empirical tests on actual quantum hardware could directly confirm the performance limits outlined in Section 5.5.1. More broadly, as discussed in the last chapter, this thesis provides a flexible framework for evaluating near-term quantum optimization, with each component open to modification in the search for improved performance scaling.

Beyond validation, future research could also enhance the presented analyses by incorporating a runtime comparison with best-in-class classical devices, providing further insights into the practicality of these optimization algorithms. Additionally, while this thesis references several studies on noise-induced performance boundaries and scalability challenges, no existing work collectively synthesizes these findings yet. A systematic review integrating all research on the theoretical and practical limitations of variational quantum algorithms could offer conclusive insights into their true potential.

However, given the already concerning nature of current findings, future research may benefit more from exploring strategies to mitigate or possibly overcome the known limitations rather than focusing solely on further validation or expansion of these scaling analyses. If quantum advantage in optimization remains the goal – especially in the near term – efforts should be directed toward identifying solutions (or problem domains) that can circumvent the barriers demonstrated in this thesis.

One popular direction for addressing the poor scaling with system size is through improved problem encoding, which can effectively reduce the number of required qubits and, therefore, mitigate some of the adverse effects of increasing system sizes discussed here. For example, Sciorilli et al. [96] proposed a variational solver for combinatorial optimizations over $m = \mathcal{O}(n^k)$ binary variables using only n qubits for a tunable $k > 1$, leading to a super-polynomial mitigation of barren plateaus. While they experimentally achieved highly competitive approximation ratios for MaxCut problems of up to 7000 vertices using only 17 trapped-ion qubits, the scaling of the required measurement

samples for the successful application of this method remains unexplored.

Another approach, introduced by Bermejo and Orús [127], significantly reduces qubit requirements in variational quantum optimization by mapping classical variable configurations to non-orthogonal quantum states in the quantum register, with state recovery achievable via quantum state tomography. While this method offers potential efficiency gains, it remains heuristic, lacking performance guarantees and empirical validation of its performance in larger, noise-exposed scenarios.

An even larger reduction of required qubits can be achieved by the multi-level approach, proposed by Maciejewski et al. [128], which can solve large-scale QUBO problems using current quantum devices. Their strategy is inspired by classical multigrid methods, where a large target problem is iteratively condensed, and the global solution is constructed from multiple small-scale optimization runs. Combining two recent quantum methods and an additional classical preprocessing step, their method achieved competitive solution qualities on QUBO problems with up to 27 000 variables, compared to the same method using classical subsolvers. Although further analysis is needed, such problem decomposition techniques could be an interesting pathway for tackling larger-scale problems.

While these advancements suggest potential solutions to address the resource limitations of near-term quantum devices, each method still faces unresolved questions regarding their scaling, noise resilience, or practical feasibility. Moreover, given that the challenges identified in this thesis may be intrinsic to the hybrid nature of these algorithms, which rely on classical optimization, it may be more promising to explore fully alternative optimization approaches outside the variational paradigm. Recent works by Bennett et al. [129] and Sankar et al. [42] have begun exploring such alternatives, and this shift away from hybrid frameworks can be similarly observed in the quantum machine learning community [130].

Finally, rather than focusing solely on solution strategies, it may be worth reconsidering the selection of problems when targeting near-term quantum practicality. As briefly discussed in Section 2.2, instead of tackling intrinsically classical problems like Ising spin glasses, much research effort is also focussing on quantum-native optimization problems – such as simulating quantum mechanical systems [1, 4] – for which genuine exponential speed-up is expected. In fact, the real-world economic impact of these applications is almost equally prominent, as they hold significant potential in areas such as modern materials science and chemistry. These fields are highly relevant for emerging technologies such as battery development, industrial catalysis, and nitrogen fixation, all of which may hold substantial economic and environmental benefits. For insights into recent progress in quantum simulation, see reviews by Daley et al. [131] and Fauseweh [132].

Appendix A

Supplementary Material for Chapter 2

A.1 Mapping QUBO to Ising Problems

To translate the general QUBO form

$$f(x) = \mathbf{x}^\top \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n Q_{ij} x_i x_j \quad (\text{A.1})$$

into the Ising form of Equation (2.10), one can apply the mapping $x_i = \frac{z_i+1}{2}$ between the binary variables $x_i \in \{0, 1\}$ and the spin variables $z_i \in \{-1, +1\}$ as follows:

$$\begin{aligned} g(z) &= \sum_{i=1}^n \sum_{j=1}^n Q_{ij} \left(\frac{z_i+1}{2} \right) \left(\frac{z_j+1}{2} \right) \\ &= \sum_{i=1}^n \sum_{j=1}^n \frac{Q_{ij}}{4} (z_i z_j + z_i + z_j + 1) \\ &= \sum_{i=1}^n \sum_{j=1}^n \frac{Q_{ij}}{2} z_i z_j + \sum_{i=1}^n \sum_{j=1}^n \frac{Q_{ij}}{4} z_i + \sum_{i=1}^n \sum_{j=1}^n \frac{Q_{ij}}{4} z_j + \underbrace{\sum_{i=1}^n \sum_{j=1}^n \frac{Q_{ij}}{4}}_{\text{constant offset}} \\ &= \underbrace{\sum_{i=j} \frac{Q_{ij}}{4}}_{\text{constant offset}} + \sum_{i < j} \frac{Q_{ij}}{4} z_i z_j + \sum_{i=1}^n \left(\sum_{j=1}^n \frac{Q_{ij}}{4} + \sum_{j=1}^n \frac{Q_{ji}}{4} \right) z_i \end{aligned} \quad (\text{A.2})$$

Assuming that \mathbf{Q} is a symmetric matrix, and by replacing the spin variables z with Pauli- \mathbf{Z} operators, the final operator form becomes Equation (2.13). Generally, this reformulation means that the Ising weight matrix $\mathcal{C} = \{\mathcal{C}_{ij}\}$ can be retrieved from the QUBO matrix $\mathbf{Q} = \{\mathbf{Q}_{ij}\}$ as follows:

- off-diagonal elements: $\mathcal{C}_{ij} = Q_{ij}/2$
- diagonal elements: $\mathcal{C}_{ii} = \sum_{j=1}^n Q_{ij}/2$

A similar mapping between the QUBO and MaxCut formulation was derived by Barahona et al. [133]. Lastly, note that this transformation is not unique. An alternative equivalent formulation can be obtained by applying the mapping $x_i = \frac{1-z_i}{2}$, which transforms the binary variables $x \in \{0, 1\}$ into the spin variables $z \in \{+1, -1\}$. This leads

to an Ising form with a negative sign for the diagonal terms. In this formulation, the ground state configuration would have 0s where the original formulation had 1s, and vice versa. Thus, this mapping would result in an inverted ground state compared to the initial transformation.

A.2 Derivation of BENQO's Operator Representation

In this section, it is shown how to derive the unitary product representation in Equation (2.19) originally proposed by Kuete Meli et al. [69].

Due to \mathbf{C} being a diagonal matrix, the encoding matrix \mathbf{U} from Equation (2.17) might as well be understood as the 2×2 operator

$$\mathbf{U}_{2 \times 2} := \begin{bmatrix} \sin\langle\hat{\mathbf{C}}\rangle & \cos\langle\hat{\mathbf{C}}\rangle \\ \cos\langle\hat{\mathbf{C}}\rangle & -\sin\langle\hat{\mathbf{C}}\rangle \end{bmatrix} \quad (\text{A.3})$$

applied to the cost qubit in state $|0\rangle$. This allows to re-formulate the original matrix as

$$\begin{aligned} \mathbf{U}(\mathbf{C}, K) &\equiv \mathbf{U}_{2 \times 2} && \otimes \mathbf{I}^{\otimes n} \\ &\equiv \mathbf{R}_Y[2 \arccos(\sin\langle\hat{\mathbf{C}}\rangle)] \cdot \mathbf{Z} && \otimes \mathbf{I}^{\otimes n} && *1 \\ &\equiv \mathbf{R}_Y[\pi - 2\langle\hat{\mathbf{C}}\rangle] \cdot \mathbf{Z} && \otimes \mathbf{I}^{\otimes n} && *2 \\ &\equiv \mathbf{R}_Y(-2\langle\hat{\mathbf{C}}\rangle) \cdot \mathbf{X} && \otimes \mathbf{I}^{\otimes n} && *3 \end{aligned} \quad (\text{A.4})$$

using the following three identities:

$$*1: \mathbf{R}_Y(2\vartheta) \cdot \mathbf{Z} = \begin{bmatrix} \cos(\vartheta) & \sin(\vartheta) \\ \sin(\vartheta) & -\cos(\vartheta) \end{bmatrix} \text{ with } \cos(\vartheta) \stackrel{!}{=} \sin\langle\hat{\mathbf{C}}\rangle \text{ exactly when } \\ \vartheta = \arccos(\sin\langle\hat{\mathbf{C}}\rangle) \text{ up to a phase factor.}$$

$$*2: \arccos(\sin(\vartheta)) = \arccos(\cos(\frac{\pi}{2} - \vartheta)) = \frac{\pi}{2} - \vartheta \text{ for } \vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2}].$$

$$*3: \mathbf{R}_Y(\vartheta_1 + \vartheta_2) = \mathbf{R}_Y(\vartheta_1) \cdot \mathbf{R}_Y(\vartheta_2) \text{ and } \mathbf{R}_Y(\pi) \cdot \mathbf{Z} = \mathbf{X}.$$

Lastly, using that $\mathbf{R}_Y(-\vartheta) \equiv \mathbf{X} \cdot \mathbf{R}_Y(\vartheta) \cdot \mathbf{X}$, one can easily insert the cost value $\langle\mathbf{C}\rangle$ from Equation (2.11) into Equation (A.4) and arrive at the formulation (2.19) presented in Section 2.3.3.

A.3 Algorithmic Details of Selected Optimizers

In Section 2.3.4, the main working principles of various gradient-based and gradient-free optimizers were outlined. Here, the concept of the algorithmic details is provided for some relevant optimizers investigated in Chapter 5:

Algorithm 1 BFGS (Broyden-Fletcher-Goldfarb-Shanno) method

Initialize: $\boldsymbol{\theta}_0, \mathbf{H}_0$
Iterate until convergence:

$$\begin{aligned} \mathbf{d}_k &= -\mathbf{H}_k \nabla_k \mathcal{L} && \triangleright \text{Compute search direction} \\ \boldsymbol{\theta}_{k+1} &\leftarrow \boldsymbol{\theta}_k + \alpha_k \mathbf{d}_k && \triangleright \text{Update parameters (with } \alpha_k \text{ from line search)} \\ \mathbf{s}_k &= \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k \\ \mathbf{y}_k &= \nabla_{k+1} \mathcal{L} - \nabla_k \mathcal{L} \\ \rho_k &= 1/\mathbf{y}_k^\top \mathbf{s}_k && \triangleright \text{Define scaling factor} \\ \mathbf{H}_{k+1} &\leftarrow (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^\top) \mathbf{H}_k (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^\top) + \rho_k \mathbf{s}_k \mathbf{s}_k^\top && \triangleright \text{Update inverse Hessian approx.} \end{aligned}$$

Algorithm 2 CG (Conjugate Gradient) method

Initialize: $\boldsymbol{\theta}_0$

$$\begin{aligned} \mathbf{s}_0 &= -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_0) && \triangleright \text{Set first conjugate direction} \\ \alpha_0 &= \operatorname{argmin}_{\alpha} \mathcal{L}(\boldsymbol{\theta}_0 + \alpha \mathbf{s}_0) && \triangleright \text{Line search for first step length} \\ \boldsymbol{\theta}_1 &= \boldsymbol{\theta}_0 + \alpha_0 \mathbf{s}_0 && \triangleright \text{Update parameters} \\ \text{Iterate until convergence:} \\ \mathbf{y}_k &= -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k) && \triangleright \text{Compute steepest direction} \\ \beta_k &= \frac{\mathbf{y}_k^\top (\mathbf{y}_k - \mathbf{y}_{k-1})}{\mathbf{y}_{k-1}^\top \mathbf{y}_{k-1}} && \triangleright \text{Compute } \textit{Polak-Ribiere} \text{ length} \\ \mathbf{s}_k &\leftarrow \mathbf{y}_k + \beta_k \mathbf{s}_{k-1} && \triangleright \text{Update conjugate direction} \\ \alpha_k &= \operatorname{argmin}_{\alpha} \mathcal{L}(\boldsymbol{\theta}_k + \alpha \mathbf{s}_k) && \triangleright \text{Line search for step length} \\ \boldsymbol{\theta}_{k+1} &\leftarrow \boldsymbol{\theta}_k + \alpha_k \mathbf{s}_k && \triangleright \text{Update parameters} \end{aligned}$$

Algorithm 3 Powell (Conjugate Direction) method

Initialize: $\boldsymbol{\theta}_0$

$$\begin{aligned} \{\mathbf{u}_i\}_{i=1,\dots,n} &= \{\mathbf{e}_i\}_{i=1,\dots,n} && \triangleright \text{Initialize set of directions to } n \text{ basis vectors} \\ \text{Iterate until convergence:} \\ \text{For } i &= 1, \dots, n \text{ do:} \\ &\lambda_i = \operatorname{argmin}_{\lambda} \mathcal{L}(\boldsymbol{\theta}_{i-1} + \lambda \mathbf{u}_i) \\ &\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{i-1} + \lambda_i \mathbf{u}_i \\ \mathbf{u}_i &\leftarrow \mathbf{u}_{i+1} \text{ (for } i = 1, \dots, n-1) && \triangleright \text{Update conjugate directions} \\ \mathbf{u}_n &= \boldsymbol{\theta}_n - \boldsymbol{\theta}_0 \\ \lambda_k &= \operatorname{argmin}_{\lambda} \mathcal{L}(\boldsymbol{\theta}_n + \lambda \mathbf{u}_n) && \triangleright \text{Line search for step length} \\ \boldsymbol{\theta}_0 &\leftarrow \boldsymbol{\theta}_n + \lambda \mathbf{u}_n && \triangleright \text{Update starting position} \end{aligned}$$

Algorithm 4 NFT (Nakashini-Fuji-Todo) method

Initialize: $\boldsymbol{\theta}_0$
Iterate until convergence:

$$\begin{aligned} &\text{Randomly (or sequentially) choose an index } j_k \in 1, \dots, n \\ &\text{Compute } \mathcal{L}(\dots, \theta_{j_k} \pm \frac{\pi}{2}, \dots) \text{ with a quantum device} \\ &\text{Define } a_{l,k} = \mathcal{L}(\dots, \theta_{j_k} + \lambda_l, \dots) \text{ for } \lambda_1 = 0, \text{ and } \lambda_{2,3} = \pm \frac{\pi}{2} \\ \theta_{j_k} &\leftarrow \operatorname{argmin}_{\theta_j} a_{1,k} \cos(\theta_j - a_{2,k}) + a_{3,k} && \triangleright \text{Update } j_k\text{-th parameter} \\ \theta_j &\leftarrow \theta_j \text{ for } j \neq j_k && \triangleright \text{Keep all other parameters} \end{aligned}$$

Appendix B

Supplementary Material for Chapter 4

B.1 IBM Quantum Backends

In the following, a short summary of the device specifications of the used IBM backends is given. IBM offers cloud access to an ever-growing pool of quantum processing units through their SDK QISKIT [32]. Three of their backends, all based on the “Eagle”¹ processor type, were chosen for the analysis in Section 4.2: `ibm_brisbane`, `ibm_kyoto`, and `ibm_sherbrooke`.

	<code>ibm_brisbane</code>	<code>ibm_kyoto</code>	<code>ibm_sherbrooke</code>
ECR error	7.943×10^{-3}	9.971×10^{-3}	7.095×10^{-3}
SX error	2.540×10^{-4}	3.480×10^{-4}	2.184×10^{-4}
readout error	1.390×10^{-2}	1.700×10^{-2}	1.270×10^{-2}
t_1	$223.77 \mu\text{s}$	$217.84 \mu\text{s}$	$280.82 \mu\text{s}$
t_2	$136.4 \mu\text{s}$	$81.87 \mu\text{s}$	$204.19 \mu\text{s}$

Table B.1: Specifications of three IBM Quantum Processing Units. All error probabilities and times represent median values. For definitions of the gate error values and the t_1 and t_2 times, refer to Section 2.1.2.

B.2 Distribution of Modelled Hardware Noise

In Section 4.2, it was noted that even though the noise on the loss value at each point in parameter space – resulting from calculations using a simulated hardware model – is biased and not centered around zero, the accumulated errors across the full landscape have an approximate zero mean. Figure B.1 presents the distribution of total (normalized) errors, $\delta\mathcal{L} = \hat{\mathcal{L}} - \tilde{\mathcal{L}}$, for 1000 randomly selected points on the loss landscape of an arbitrary Ising problem of size $n = 6$. For all algorithms and modeled backends under consideration, these noise distributions have their means at zero. However, the density histograms of BENQO and VQE exhibit a clear right-skew. This indicates that while negative values (representing noisy loss values higher than their exact counterparts) are

¹The Eagle processor family corresponds to a 127-qubit system by IBM that utilizes advanced multi-layer chip technology to enable high-density I/O without compromising performance, having a quantum volume of 128. Its native gates are: ECR, ID, RZ, SX, X (cf. Section 2.1.1).

more probable, positive outliers are more frequent, as evidenced by the longer tail. This can be explained by the distribution of loss values within their value range, and the fixed point of the respective error channels used in the hardware noise model. For example, the depolarizing error makes any quantum state converge to a totally mixed state, whose loss value corresponds to zero (see Equation (2.10)) – the mean of all possible solution states. Due to the asymmetric distribution of loss values, which is generally skewed toward negative values (cf. Figure 5.8), however, the fixed point of this error channel produces loss values which are larger than the majority of points, likewise causing an asymmetry in the error distribution.

In contrast, QAOA shows no visible skewness but is markedly leptokurtic compared to the corresponding Gaussian. This implies that the distribution of errors is more tightly centered around zero, with fewer extreme deviations in both directions than a normal distribution would exhibit. In this case, the Gaussian serves as a broader reference for errors occurring in a noisy QAOA.

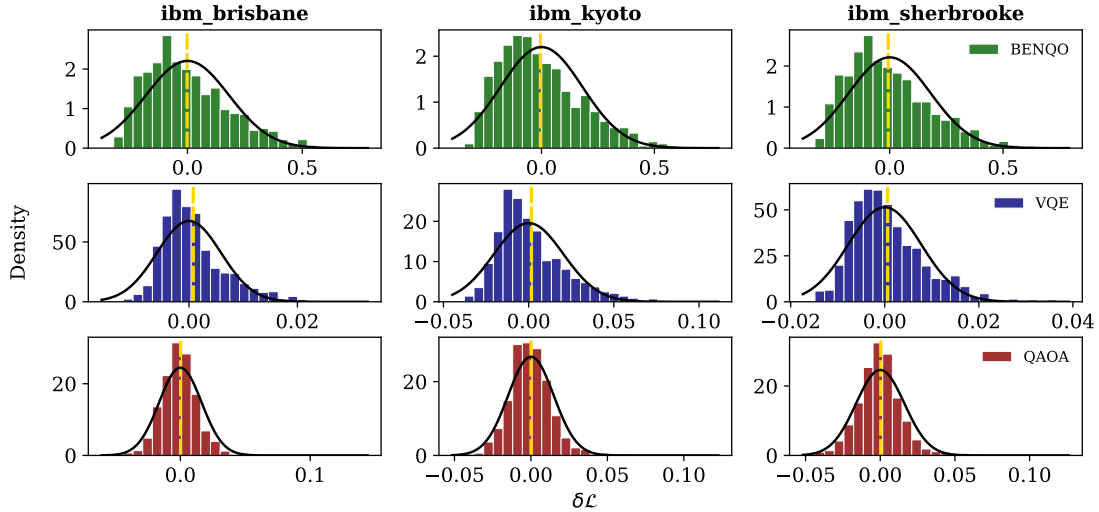


Figure B.1: Distribution of Accumulated Hardware-Induced Errors Across Full Loss Landscape. Density histograms of the errors of 1000 (normalized) loss evaluations, done with simulated noise models from three representative quantum backends by IBM, randomly sampled across the loss landscape of a $n = 6$ problem instance are shown, with their mean indicated as vertical dashed lines. The bold lines represent normal distributions with mean zero and the sample standard deviations of these errors.

B.3 Derivations for Statistical Error

The standard error of a measured proportion is given in Equation (4.1). BENQO’s loss estimation can be written fully in terms of the measured counts $\hat{p}_0 = k_0/N$ via

$$\mathbb{E}_N[\mathbf{C}] = K \arcsin(2\hat{p}_0 - 1). \quad (\text{B.1})$$

The linear approximation for the propagation of uncertainties gives

$$\begin{aligned}
 \sigma(\mathbb{E}_N[C]) &\approx \left. \frac{d\mathbb{E}_N[C]}{d\hat{p}_0} \right|_{\hat{p}_0=p_0} \sigma(\hat{p}_0) \\
 &= \frac{K}{\sqrt{1 - (2\hat{p}_0 - 1)^2}} \sqrt{\frac{p_0(1 - p_0)}{N}} \\
 &= \frac{K}{\sqrt{4p_0(1 - p_0)}} \sqrt{\frac{p_0(1 - p_0)}{N}} = \frac{K}{2\sqrt{N}}.
 \end{aligned} \tag{B.2}$$

For the loss evaluation of VQE and QAOA, the full quantum state $|\psi(\theta)\rangle$ of the n -qubit system has to be measured. The estimated loss value derived from the measurement of m individual Pauli observables \mathbf{P}_i can be written as

$$\mathbb{E}_N[\mathbf{C}] = \sum_{i=1}^m \alpha_i \langle \psi(\theta) | \mathbf{P}_i | \psi(\theta) \rangle = \sum_{i=1}^m \alpha_i \left(\sum_{j=1}^{2^n} \hat{p}_j \langle q_j | \mathbf{P}_i | q_j \rangle \right), \tag{B.3}$$

where \hat{p}_j are the estimates of the overlap $|\langle \psi(\theta) | q_j \rangle|^2$, corresponding to the probability of “finding” the quantum system in the basis state $|q_j\rangle$. The Ising cost function defined by Equation (2.10) includes only Pauli strings that consist of \mathbf{Z} and identity operators, so the expectation value $\langle q_j | \mathbf{P}_i | q_j \rangle$ will be either $+1$ or -1 . By applying the linear approximation from above and using the fact that the variances of multiple independent² variables combine in quadrature, one can derive the following expression for the statistical uncertainty of the estimated expectation value:

$$\begin{aligned}
 \sigma(\mathbb{E}_N[C]) &\approx \sqrt{\sum_{j=1}^{2^n} \left(\left. \frac{d\mathbb{E}_N[C]}{d\hat{p}_j} \right|_{\hat{p}_j=p_j} \sigma(\hat{p}_j) \right)^2} \\
 &= \sqrt{\sum_{j=1}^{2^n} \underbrace{\left(\sum_{i=1}^m \alpha_i \langle q_j | \mathbf{P}_i | q_j \rangle \right)^2}_{=: S_j} \sigma(\hat{p}_j)^2} \\
 &= \sqrt{\sum_{j=1}^{2^n} S_j^2 \frac{p_j(1 - p_j)}{N}}.
 \end{aligned} \tag{B.4}$$

Note that the sum, denoted as S_j , corresponds to the loss value evaluated at the j -th basis state, which can be represented as a binary string of length n .

²While this is not true for the individual proportions \hat{p}_j for $j = 1, \dots, 2^n$, it provides a sufficiently accurate approximation for large N .

Appendix C

Supplementary Material for Chapter 5

C.1 Local Qiskit Optimizers

Table C.1 presents a summary of state-of-the-art QISKIT optimizers used in the initial screening of the classical optimizer studies. Out of those, eight with “promising” performance were picked for the final evaluations based on their success probability on a trial Ising problem and their required number of function evaluations.

Name	Description	Reference
<i>gradient-based</i>		
ADAM	Adaptive Moment Estimation	Kingma and Ba [79]
AMSGrad	ADAM improved for better convergence	Reddi et al. [80]
AQGD	Analytic Quantum Gradient Descent method	Mitarai et al. [74]
BFGS	Broyden-Fletcher-Goldfarb-Shanno optimizer	Byrd et al. [81]
CG	Conjugate Gradient method	Fletcher [82]
SLSQP	Sequential Least Squares Programming	Kraft [84]
SPSA ¹	Simultaneous Perturbation Stochastic Approx.	Spall [85, 115]
TNC	Truncated Newton Optimizer	Nash [83]
<i>gradient-free</i>		
COBYLA	Constrained Optimization By Linear Approx.	Powell [73, 88]
Nelder-Mead	simplex algorithm	Nelder and Mead [87]
NFT	Nakanishi-Fuji-Todo algorithm	Nakanishi et al. [72]
Powell	conjugate direction method	Powell [86]
UMDA	Univariate Marginal Distribution Algorithm	Soloviev et al. [89]

Table C.1: Summary of local Qiskit Optimizers. The whole suite of gradient-based (upper part) and gradient-free (lower-part) optimizers used for the experiments described in Chapter 5 is listed along with the corresponding references.

¹Note that the SPSA optimizer is sometimes listed as a gradient-free method since it does not rely on exact gradient calculations. However, since it approximates the gradient, it is classified as a gradient-based method here.

C.2 Derivation of the Lower Confidence Limit

The quantum state solution of the BENQO algorithm is

$$|\psi^*\rangle = |\psi(\theta)\rangle = \bigotimes_{i=1}^n \mathbf{R}_Y(\vartheta_i^*)|0\rangle = \bigotimes_{i=1}^n \left(\cos \frac{\vartheta_i^*}{2} |0\rangle + \sin \frac{\vartheta_i^*}{2} |1\rangle \right), \quad (\text{C.1})$$

where $\theta^* = (\vartheta_1^*, \dots, \vartheta_n^*)^\top$ are the classically optimized parameters, and \mathbf{R}_Y is the mathematical operator of a RY gate (cf. Section 2.1.1). The overlap of an individual basis state $|q\rangle = (q_1, \dots, q_n)^\top$ with the full quantum state $|\psi\rangle$ is given by

$$P_\theta(|q\rangle) = |\langle q|\psi(\theta)\rangle|^2 = \prod_{i=1}^n \left(\cos \left(\frac{\vartheta_i}{2} \right) \delta_{q_i,0} + \sin \left(\frac{\vartheta_i}{2} \right) \delta_{q_i,1} \right)^2 \quad (\text{C.2})$$

with the binary variables $q_i \in \{0, 1\}$. This formulation corresponds to the expected proportion of finding the variational quantum system in state $|q\rangle$ after measurement. Relevant for classical optimization problems is the basis state $|\psi_{\max}\rangle$ for which $P_\theta(|\psi_{\max}\rangle)$ is maximal. The so-called confidence defined in Equation (3.10) is the maximum probability

$$\begin{aligned} c(|\psi_{\max}\rangle) &= \max_{|q\rangle} P_\theta(|q\rangle) \\ &= \prod_{i=1}^n \max_{q_i} \left(\cos \left(\frac{\vartheta_i}{2} \right) \delta_{q_i,0} + \sin \left(\frac{\vartheta_i}{2} \right) \delta_{q_i,1} \right)^2 \\ &= \prod_{i=1}^n \begin{cases} \cos^2 \left(\frac{\vartheta_i}{2} \right) & \text{for } \vartheta_i \in [0, \frac{\pi}{2}] \cup [\frac{3\pi}{2}, 2\pi] \\ \sin^2 \left(\frac{\vartheta_i}{2} \right) & \text{for } \vartheta_i \in [\frac{\pi}{2}, \frac{3\pi}{2}] \end{cases} \end{aligned} \quad (\text{C.3})$$

still depending on the specific parameters θ . Finally, one can derive how the mean confidence over a uniform distribution of the angles in the parameter space $[0, 2\pi]^n$ behaves for varying system sizes n (to explain the empirical findings of Section 5.3) like

$$\begin{aligned} \bar{c}(|\psi_{\max}\rangle) &= E_\theta [c(|\psi_{\max}\rangle)] \\ &= \left(\frac{1}{2\pi} \right)^n \int_0^{2\pi} \dots \int_0^{2\pi} d\vartheta_1 \dots d\vartheta_n c(|\psi_{\max}\rangle) \\ &= \frac{1}{(2\pi)^n} \left[\underbrace{\int_0^{\pi/2} d\vartheta \cos^2 \left(\frac{\vartheta}{2} \right)}_{\frac{1}{4}(2+\pi)} + \underbrace{\int_{\pi/2}^{3\pi/2} d\vartheta \sin^2 \left(\frac{\vartheta}{2} \right)}_{\frac{1}{2}(2+\pi)} + \underbrace{\int_{3\pi/2}^{2\pi} d\vartheta \cos^2 \left(\frac{\vartheta}{2} \right)}_{\frac{1}{4}(2+\pi)} \right]^n \\ &= \left(\frac{2+\pi}{2\pi} \right)^n. \end{aligned} \quad (\text{C.4})$$

Figure C.1 visually summarizes these findings, displaying the experimentally found probability distribution of $c(|\psi_{\max}\rangle)$ and the behavior of $\bar{c}(|\psi_{\max}\rangle)$ for increasing system sizes n (as derived in Equation (C.4)).

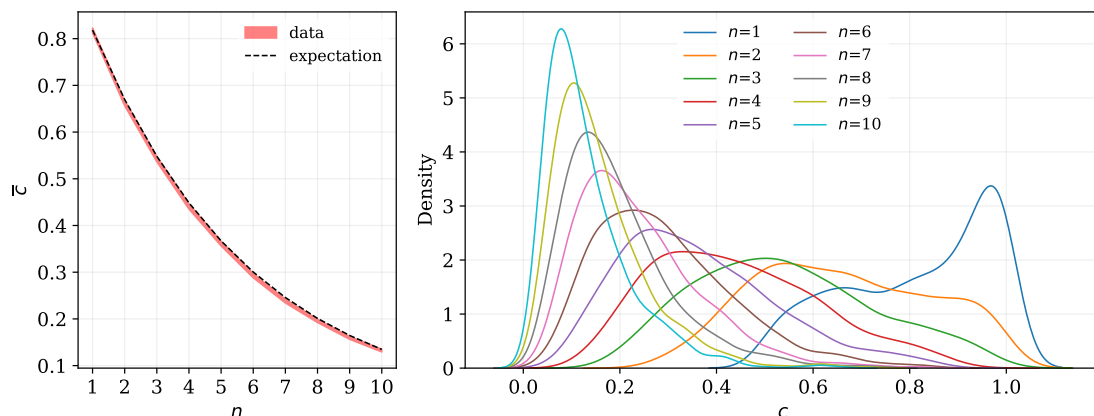


Figure C.1: Distribution of Confidence Values over System Size. The left plot shows the range of data when taking the average of $c(|\psi_{\max}\rangle)$ over 1000 uniformly sampled values of $\theta \in [0, 2\pi]^n$ for various system sizes n , including the theory function derived in Equation (C.4). Both curves are perfectly matching. On the right side, the kernel distribution estimates are plotted for $c(|\psi_{\max}\rangle)$ across the full parameter space.

C.3 Ranking of Optimizers

Inspired by the work of Singh et al. [109], a summary of the optimizers' performance rankings is presented in Figure C.2. To provide a meaningful comparison with related studies on classical optimizers, however in the context of the VQE algorithm, the results from Singh et al. [109] are also included. Despite some differences in the details, several common trends emerge: SPSA, Powell and NFT rank among the most noise-resilient optimizers in both studies, whereas CG, SLSQP, and BFGS show poor performance under noise. Furthermore, as noted in Section 5.4, there appears to be a trend along the anti-diagonal in the left plot of Figure C.2, suggesting a trade-off between an optimizer's performance in noisy and noise-free settings. Methods that perform well in the idealized scenario were the most susceptible to noise, likely because they rely on exact gradient information, which becomes unreliable in noisy environments. In contrast, the noise-resilient optimizers don't rely on precise or any gradient calculations: The SPSA method, for example, uses stochastic parameter perturbations to approximate a loss-minimizing direction [85]. Powell's method iteratively performs one-dimensional line searches along conjugate directions [86], resulting in only small dependencies on the outcome of individual function evaluations. Likewise, the NFT method minimizes the loss function with respect to one parameter at a time such that the impact of one noisy loss evaluation may not be substantial [72]. Overall, this indicates that while gradient-based optimizers excel in ideal conditions, they may not be well-suited for noisy settings, where alternative strategies prove more robust – a finding also supported by other studies [108, 109].

Furthermore, the right side of Figure C.2 investigates the relationship between solver efficiency and noise resilience, with efficiency measured by comparing the mean runtimes

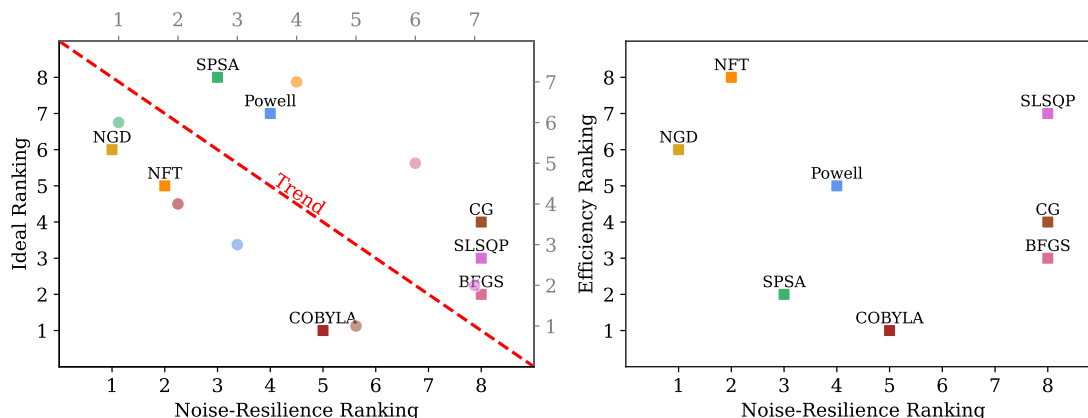


Figure C.2: Performance Ranking of Classical Optimizers Under Consideration.

The left plot shows the rank of each optimizer’s performance in an ideal (noise-free) setting (from Figure 5.7) versus in a noisy setting (from Figure 5.6). Transparent dots represent the results reported by Singh et al. [109], plotted along the gray axes for better comparability. The dashed diagonal line indicates a potential trend between both sets of results. The right plot illustrated the relationship between the noise resilience of each optimizer and its efficiency ranking, based on measurements of mean runtimes in the same noisy experiments. No comparative studies were available for inclusion here.

across all conducted experiments, which are directly tied to the number of quantum function calls. While no clear correlation or trend is immediately apparent from the plot, one observation stands out: the most noise-resilient optimizers, NGD and NFT, tend to be the least efficient in terms of runtime. On the other hand, optimizers such as SPSA and COBYLA strike a better balance between efficiency and noise resilience, emerging as the best overall choices in terms of this trade-off.

Lastly note, that this form of numerical presentation inspired by Singh et al. [109] – considering discrete ranks instead of the actual continuous measures of performance – may not be the most representative, as it does not capture the actual quantitative performance differences between individual optimizers.

C.4 Investigation of Decay Type via Fitting Experiment

In an attempt to determine the type of decay of the noise resilience (representatively given by the point of steepest descent of the performance σ^*) with increasing system size n , several functions f_{fit} were fitted to the experimental data in Section 5.4. The study of the fit residuals $\sigma^*(n) - f_{\text{fit}}(n)$ and their MSE was used to derive the best fitting curve. However, no curve had significantly smaller residuals, which is why the exact type of decay remains a speculation, which might be cleared up with more experimental data. These results are graphically summarized in Figure C.3.

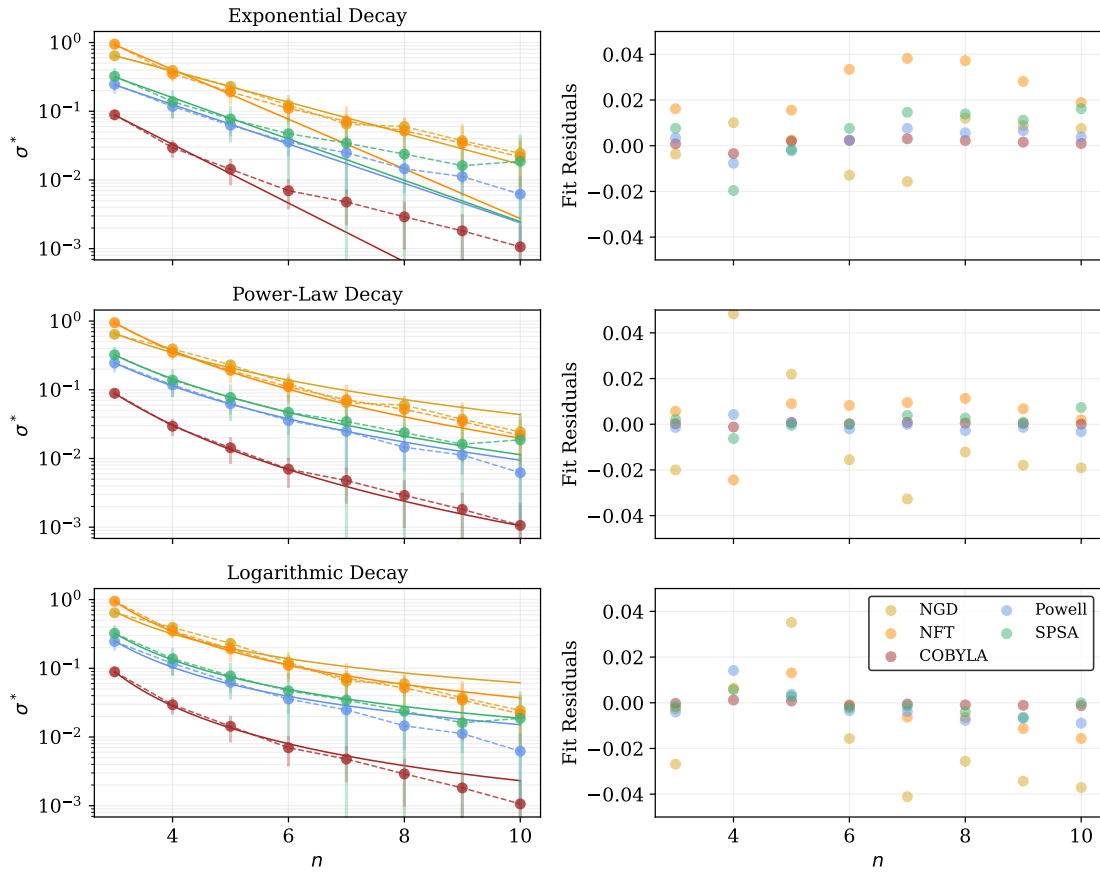


Figure C.3: Comparison of Different Fitting Functions. The decrease in noise resilience σ^* with system size n of the classical optimizers (described in Section 5.4) is fitted to the experimental data. When comparing the final fit residuals (right column), it seems that one of the three decay types stands out as the most suitable.

Bibliography

- [1] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6–7):467–488, June 1982. ISSN 1572-9575. doi: 10.1007/bf02650179.
- [2] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, January 1999. ISSN 1095-7200. doi: 10.1137/s0036144598347011.
- [3] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, page 212–219. ACM Press, 1996. doi: 10.1145/237814.237866.
- [4] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, August 1996. ISSN 1095-9203. doi: 10.1126/science.273.5278.1073.
- [5] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), October 2009. ISSN 1079-7114. doi: 10.1103/physrevlett.103.150502.
- [6] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779): 505–510, October 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1666-5.

Bibliography

- [7] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), May 2019. ISSN 1931-9401. doi: 10.1063/1.5088164.
- [8] Ivan H. Deutsch, Gavin K. Brennen, and Poul S. Jessen. Quantum computing with neutral atoms in an optical lattice. *Fortschritte der Physik*, 48(9–11):925–943, September 2000. ISSN 1521-3978. doi: 10.1002/1521-3978(200009)48:9/11<925::aid-prop925>3.0.co;2-a.
- [9] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, December 2020. ISSN 1095-9203. doi: 10.1126/science.abe8770.
- [10] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, August 2018. ISSN 2521-327X. doi: 10.22331/q-2018-08-06-79.
- [11] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bäertschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, Nicola Franco, Filippo Fratini, Bryce Fuller, Julien Gacon, Constantin Gconciulea, Sander Gribling, Swati Gupta, Stuart Hadfield, Raoul Heese, Gerhard Kircher, Thomas Kleinert, Thorsten Koch, Georgios Korpas, Steve Lenk, Jakub Marecek, Vanio Markov, Guglielmo Mazzola, Stefano Mensa, Naeimeh Mohseni, Giacomo Nannicini, Corey O’Meara, Elena Peña Tapia, Sebastian Pokutta, Manuel Proissl, Patrick Rebentrost, Emre Sahin, Benjamin C. B. Symons, Sabine Tornow, Víctor Valls, Stefan Woerner, Mira L. Wolf-Bauwens, Jon Yard, Sheir Yarkoni, Dirk Zechiel, Sergiy Zhuk, and Christa Zoufal. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, October 2024. ISSN 2522-5820. doi: 10.1038/s42254-024-00770-9.
- [12] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, February 2016. ISSN 1367-2630. doi: 10.1088/1367-2630/18/2/023023.
- [13] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9): 625–644, August 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00348-9.
- [14] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of*

Bibliography

- Modern Physics*, 94(1), February 2022. ISSN 1539-0756. doi: 10.1103/revmodphys.94.015004.
- [15] Wim Lavrijsen, Ana Tudor, Juliane Muller, Costin Iancu, and Wibe de Jong. Classical optimizers for noisy intermediate-scale quantum devices. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, page 267–277. IEEE, October 2020. doi: 10.1109/qce49297.2020.00041.
- [16] Enrico Fontana, Nathan Fitzpatrick, David Muñoz Ramo, Ross Duncan, and Ivan Rungger. Evaluating the noise resilience of variational quantum algorithms. *Physical Review A*, 104(2), August 2021. ISSN 2469-9934. doi: 10.1103/physreva.104.022403.
- [17] Torsten Hoeffler, Thomas Häner, and Matthias Troyer. Disentangling hype from practicality: On realistically achieving quantum advantage. *Communications of the ACM*, 66(5):82–87, April 2023. ISSN 1557-7317. doi: 10.1145/3571725.
- [18] Olivier Ezratty. Where are we heading with NISQ?, 2023. URL <https://arxiv.org/abs/2305.09518>.
- [19] Daniel Stilck França and Raul García-Patrón. Limitations of optimization algorithms on noisy quantum devices. *Nature Physics*, 17(11):1221–1227, October 2021. ISSN 1745-2481. doi: 10.1038/s41567-021-01356-3.
- [20] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1), November 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-27045-6.
- [21] Guillermo González-García, Rahul Trivedi, and J. Ignacio Cirac. Error propagation in NISQ devices for solving classical optimization problems. *PRX Quantum*, 3(4), December 2022. ISSN 2691-3399. doi: 10.1103/prxquantum.3.040326.
- [22] Yuguo Shao, Fuchuan Wei, Song Cheng, and Zhengwei Liu. Simulating noisy variational quantum algorithms: A polynomial approach. *Physical Review Letters*, 133(12), September 2024. ISSN 1079-7114. doi: 10.1103/physrevlett.133.120603.
- [23] E. M. Stoudenmire and Xavier Waintal. Opening the black box inside grover’s algorithm. *Physical Review X*, 14(4), November 2024. ISSN 2160-3308. doi: 10.1103/physrevx.14.041029.
- [24] Giuseppe Scrivera, Nikita Astrakhantsev, Sebastiano Pilati, and Guglielmo Mazzola. Challenges of variational quantum optimization with measurement shot noise. *Physical Review A*, 109(3), March 2024. ISSN 2469-9934. doi: 10.1103/physreva.109.032408.
- [25] Frank Leymann and Johanna Barzen. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology*, 5(4):044007, September 2020. ISSN 2058-9565. doi: 10.1088/2058-9565/abae7d.

Bibliography

- [26] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is np-hard. *Physical Review Letters*, 127(12), September 2021. ISSN 1079-7114. doi: 10.1103/physrevlett.127.120502.
- [27] Eric R. Anschuetz and Bobak T. Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, 13(1), December 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-35364-5.
- [28] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. The complexity of NISQ. *Nature Communications*, 14(1), September 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-41217-6.
- [29] Vyacheslav Kungurtsev, Georgios Korpas, Jakub Marecek, and Elton Yechao Zhu. Iteration complexity of variational quantum algorithms. *Quantum*, 8:1495, October 2024. ISSN 2521-327X. doi: 10.22331/q-2024-10-10-1495.
- [30] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, STOC '93*, page 11–20. ACM Press, 1993. doi: 10.1145/167088.167097.
- [31] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, 2016. ISBN 978-1-107-00217-3.
- [32] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with qiskit, 2024. URL <https://arxiv.org/abs/2405.08810>.
- [33] Emily Grumbling and Mark Horowitz, editors. *Quantum Computing: Progress and Prospects*. National Academies Press, March 2019. ISBN 9780309479691. doi: 10.17226/25196.
- [34] McKinsey & Company. Quantum Technology Monitor. Technical report, April 2023.
- [35] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Berlin Heidelberg, 2018. ISBN 9783662560396. doi: 10.1007/978-3-662-56039-6.
- [36] Giorgio Ausiello, Alberto Marchetti-Spaccamela, Pierluigi Crescenzi, Giorgio Gambosi, Marco Protasi, and Viggo Kann. *Complexity and Approximation*. Springer Berlin Heidelberg, 1999. ISBN 9783642584121. doi: 10.1007/978-3-642-58412-1.
- [37] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014. ISSN 2296-424X. doi: 10.3389/fphy.2014.00005.

Bibliography

- [38] G. G. Guerreschi and A. Y. Matsuura. Qaoa for max-cut requires hundreds of qubits for quantum speed-up. *Scientific Reports*, 9(1), May 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-43176-9.
- [39] Alexey Bochkarev, Raoul Heese, Sven Jäger, Philine Schiewe, and Anita Schöbel. Quantum computing for discrete optimization: A highlight of three technologies, 2024. URL <https://arxiv.org/abs/2409.01373>.
- [40] Sovanmonynuth Heng, Dongmin Kim, Taekyung Kim, and Youngsun Han. How to solve combinatorial optimization problems using real quantum machines: A recent survey. *IEEE Access*, 10:120106–120121, 2022. ISSN 2169-3536. doi: 10.1109/access.2022.3218908.
- [41] Ehsan Zahedinejad and Arman Zaribafiyani. Combinatorial optimization on gate model quantum computers: A survey, 2017. URL <https://arxiv.org/abs/1708.05294>.
- [42] Krishanu Sankar, Artur Scherer, Satoshi Kako, Sam Reifenstein, Navid Ghadermarzy, Willem B. Krayenhoff, Yoshitaka Inui, Edwin Ng, Tatsuhiko Onodera, Pooya Ronagh, and Yoshihisa Yamamoto. A benchmarking study of quantum algorithms for combinatorial optimization. *npj Quantum Information*, 10(1), June 2024. ISSN 2056-6387. doi: 10.1038/s41534-024-00856-3.
- [43] F Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, October 1982. ISSN 1361-6447. doi: 10.1088/0305-4470/15/10/028.
- [44] Y Fu and P W Anderson. Application of statistical mechanics to np-complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General*, 19(9):1605–1620, June 1986. ISSN 1361-6447. doi: 10.1088/0305-4470/19/9/033.
- [45] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28(1):58–81, April 2014. ISSN 1573-2886. doi: 10.1007/s10878-014-9734-0.
- [46] Fred Glover, Gary Kochenberger, Rick Hennig, and Yu Du. Quantum bridge analytics i: a tutorial on formulating and using qubo models. *Annals of Operations Research*, 314(1):141–183, April 2022. ISSN 1572-9338. doi: 10.1007/s10479-022-04634-2.
- [47] Fred Glover, Gary Kochenberger, and Yu Du. *Applications and Computational Advances for Solving the QUBO Model*, pages 39–56. Springer International Publishing, Cham, 2022. ISBN 978-3-031-04520-2. doi: 10.1007/978-3-031-04520-2.2.

Bibliography

- [48] A.B. Finnila, M.A. Gomez, C. Sebenik, C. Stenson, and J.D. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letters*, 219(5–6):343–348, March 1994. ISSN 0009-2614. doi: 10.1016/009-2614(94)00117-0.
- [49] Sheir Yarkoni, Elena Raponi, Thomas Bäck, and Sebastian Schmitt. Quantum annealing for industry applications: introduction and review. *Reports on Progress in Physics*, 85(10):104001, September 2022. ISSN 1361-6633. doi: 10.1088/1361-6633/ac8c54.
- [50] Richard M. Karp. *Reducibility among Combinatorial Problems*, page 85–103. Springer US, 1972. ISBN 9781468420012. doi: 10.1007/978-1-4684-2001-2_9.
- [51] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, September 1998. ISSN 1557-735X. doi: 10.1145/290179.290180.
- [52] Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8):1665–1677, December 2015. ISSN 0022-0000. doi: 10.1016/j.jcss.2015.06.003.
- [53] Gilbert Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, June 1992. ISSN 0377-2217. doi: 10.1016/0377-2217(92)90138-y.
- [54] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '21. ACM, June 2021. doi: 10.1145/3406325.3451009.
- [55] Michael R. Garey, David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A Series of Books in the Mathematical Sciences. W.H. Freeman, New York, 1979. ISBN 0-7167-1044-7.
- [56] Yue Ruan, Samuel Marsh, Xilin Xue, Zhihao Liu, and Jingbo Wang. The quantum approximate algorithm for solving traveling salesman problem. *Computers, Materials amp; Continua*, 63(3):1237–1247, 2020. ISSN 1546-2226. doi: 10.32604/cmc.2020.010001.
- [57] Maxine T. Khumalo, Hazel A. Chieza, Krupa Prag, and Matthew Woolway. An investigation of IBM quantum computing device performance on combinatorial optimisation problems, 2022. URL <https://arxiv.org/abs/2107.03638>.
- [58] Özlem Salehi, Adam Glos, and Jarosław Adam Mischczak. Unconstrained binary models of the travelling salesman problem variants for quantum optimization. *Quantum Information Processing*, 21(2), January 2022. ISSN 1573-1332. doi: 10.1007/s11128-021-03405-5.

Bibliography

- [59] Lilly Palackal, Benedikt Poggel, Matthias Wulff, Hans Ehm, Jeanette Miriam Lorenz, and Christian B. Mendl. Quantum-assisted solution paths for the capacitated vehicle routing problem. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, September 2023. doi: 10.1109/qce57702.2023.00080.
- [60] Wenyang Qian, Robert A. M. Basili, Mary Mehrnoosh Eshaghian-Wilner, Ashfaq Khokhar, Glenn Luecke, and James P. Vary. Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem. *Entropy*, 25(8):1238, August 2023. ISSN 1099-4300. doi: 10.3390/e25081238.
- [61] Daniel Goldsmith and Joe Day-Evans. Beyond qubo and hobo formulations, solving the travelling salesman problem on a quantum boson sampler, 2024. URL <https://arxiv.org/abs/2406.14252>.
- [62] 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), July 2014. ISSN 2041-1723. doi: 10.1038/ncomms5213.
- [63] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. URL <https://arxiv.org/abs/1411.4028>.
- [64] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, June 2018. ISSN 2058-9565. doi: 10.1088/2058-9565/aab822.
- [65] Giacomo Nannicini. Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. *Physical Review E*, 99(1), January 2019. ISSN 2470-0053. doi: 10.1103/physreve.99.013304.
- [66] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, November 2022. ISSN 0370-1573. doi: 10.1016/j.physrep.2022.08.003.
- [67] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 1068:1–66, June 2024. ISSN 0370-1573. doi: 10.1016/j.physrep.2024.03.002.
- [68] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. URL <https://arxiv.org/abs/quant-ph/0001106>.

Bibliography

- [69] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. A universal quantum algorithm for weighted maximum cut and ising problems. *Quantum Information Processing*, 22(7), July 2023. ISSN 1573-1332. doi: 10.1007/s11128-023-04025-x.
- [70] Adelina Bärligea, Benedikt Poggel, and Jeanette Miriam Lorenz. Solving combinatorial optimization problems with a block encoding quantum optimizer, 2024. URL <https://arxiv.org/abs/2404.14054>.
- [71] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4), December 2021. ISSN 2691-3399. doi: 10.1103/prxquantum.2.040203.
- [72] Ken M. Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4), October 2020. ISSN 2643-1564. doi: 10.1103/physrevresearch.2.043158.
- [73] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, January 1998. ISSN 1474-0508. doi: 10.1017/s0962492900002841.
- [74] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3), September 2018. ISSN 2469-9934. doi: 10.1103/physreva.98.032309.
- [75] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), March 2019. ISSN 2469-9934. doi: 10.1103/physreva.99.032331.
- [76] Ryan Murray, Brian Swenson, and Soumya Kar. Revisiting normalized gradient descent: Fast evasion of saddle points. *IEEE Transactions on Automatic Control*, 64(11):4818–4824, November 2019. ISSN 2334-3303. doi: 10.1109/tac.2019.2914998.
- [77] 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)*, page 1–9. IEEE, October 2021. doi: 10.1109/qce52317.2021.00015.
- [78] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel

Bibliography

- Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, February 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2.
- [79] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL <https://arxiv.org/abs/1412.6980>.
- [80] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond, 2019. URL <https://arxiv.org/abs/1904.09237>.
- [81] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, September 1995. ISSN 1095-7197. doi: 10.1137/0916069.
- [82] R. Fletcher. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, February 1964. ISSN 1460-2067. doi: 10.1093/comjnl/7.2.149.
- [83] Stephen G. Nash. Newton-type minimization via the lanczos method. *SIAM Journal on Numerical Analysis*, 21(4):770–788, August 1984. ISSN 1095-7170. doi: 10.1137/0721052.
- [84] Dieter Kraft. A software package for sequential quadratic programming, July 1988.
- [85] James C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.
- [86] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2): 155–162, February 1964. ISSN 1460-2067. doi: 10.1093/comjnl/7.2.155.
- [87] J. A. Nelder and R. Mead. A simplex method for function minimization: Nelder_mead. *The Computer Journal*, 7(4):308–313, 1965. ISSN 0010-4620. doi: [10.1093/comjnl/7.4.308](https://doi.org/10.1093/comjnl/7.4.308).

Bibliography

- [88] M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, page 51–67. Springer Netherlands, 1994. ISBN 9789401583305. doi: 10.1007/978-94-015-8330-5_4.
- [89] Vicente P. Soloviev, Pedro Larrañaga, and Concha Bielza. Quantum parametric circuit optimization with estimation of distribution algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, page 2247–2250. ACM, July 2022. doi: 10.1145/3520304.3533963.
- [90] Ricard Puig, Marc Drudis, Supanut Thanasilp, and Zoë Holmes. Variational quantum simulation: a case study for understanding warm starts, 2024. URL <https://arxiv.org/abs/2404.10044>.
- [91] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo. A review of barren plateaus in variational quantum computing, 2024. URL <https://arxiv.org/abs/2405.00781>.
- [92] M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana, Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë Holmes. Does provable absence of barren plateaus imply classical simulability? or, why we need to rethink variational quantum computing, 2024. URL <https://arxiv.org/abs/2312.09121>.
- [93] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), November 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-07090-4.
- [94] Federico Dominguez, Josua Unger, Matthias Traube, Barry Mant, Christian Ertler, and Wolfgang Lechner. Encoding-independent optimization problem formulation for quantum computing. *Frontiers in Quantum Science and Technology*, 2, September 2023. ISSN 2813-2181. doi: 10.3389/frqst.2023.1229471.
- [95] Einar Gabbassov, Gili Rosenberg, and Artur Scherer. Lagrangian duality in quantum optimization: Overcoming qubo limitations for constrained problems, 2023.
- [96] Marco Sciorilli, Lucas Borges, Taylor L. Patti, Diego García-Martín, Giancarlo Camilo, Anima Anandkumar, and Leandro Aolita. Towards large-scale quantum optimization solvers with few qubits, 2024. URL <https://arxiv.org/abs/2401.09421>.
- [97] Dario De Santis, Salvatore Tirone, Stefano Marmi, and Vittorio Giovannetti. Optimized qubo formulation methods for quantum computing, 2024. URL <https://arxiv.org/abs/2406.07681>.

Bibliography

- [98] Manuel Schnaus, Lilly Palackal, Benedikt Poggel, Xiomara Runge, Hans Ehm, Jeanette Miriam Lorenz, and Christian B. Mendl. Efficient encodings of the travelling salesperson problem for variational quantum algorithms, 2024. URL <https://arxiv.org/abs/2404.05448>.
- [99] Samson Wang, Piotr Czarnik, Andrew Arrasmith, M. Cerezo, Lukasz Cincio, and Patrick J. Coles. Can error mitigation improve trainability of noisy variational quantum algorithms? *Quantum*, 8:1287, March 2024. ISSN 2521-327X. doi: 10.22331/q-2024-03-14-1287.
- [100] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J. Coles, and M. Cerezo. Diagnosing barren plateaus with tools from quantum optimal control. *Quantum*, 6:824, September 2022. ISSN 2521-327X. doi: 10.22331/q-2022-09-29-824.
- [101] Eric Benhamou. A few properties of sample variance, 2018. URL <https://arxiv.org/abs/1809.03774>.
- [102] Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. Using models to improve optimizers for variational quantum algorithms. *Quantum Science and Technology*, 5(4):044008, September 2020. ISSN 2058-9565. doi: 10.1088/2058-9565/abb6d9.
- [103] Hans Fischer. *A History of the Central Limit Theorem: From Classical to Modern Probability Theory*. Springer New York, 2011. ISBN 9780387878577. doi: 10.1007/978-0-387-87857-7.
- [104] Joonho Kim and Yaron Oz. Quantum energy landscape and circuit optimization. *Physical Review A*, 106(5), November 2022. ISSN 2469-9934. doi: 10.1103/physreva.106.052424.
- [105] Joel Rajakumar, John Golden, Andreas Bärtzchi, and Stephan Eidenbenz. Trainability barriers in low-depth qaoa landscapes. In *Proceedings of the 21st ACM International Conference on Computing Frontiers, CF '24*. ACM, May 2024. doi: 10.1145/3649153.3649204.
- [106] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18*, page 6391–6401, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [107] Peter Gleißner, Georg Kruse, and Andreas Roßkopf. Restricted global optimization for QAOA. *APL Quantum*, 1(2), April 2024. ISSN 2835-0103. doi: 10.1063/5.0189374.
- [108] Aidan Pellow-Jarman, Ilya Sinayskiy, Anban Pillay, and Francesco Petruccione. A comparison of various classical optimizers for a variational quantum linear solver.

Bibliography

- Quantum Information Processing*, 20(6), June 2021. ISSN 1573-1332. doi: 10.1007/s11128-021-03140-x.
- [109] Harshdeep Singh, Sonjoy Majumder, and Sabyashachi Mishra. Benchmarking of different optimizers in the variational quantum algorithms for applications in quantum chemistry. *The Journal of Chemical Physics*, 159(4), July 2023. ISSN 1089-7690. doi: 10.1063/5.0161057.
- [110] Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E. O’Brien. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A*, 107(3), March 2023. ISSN 2469-9934. doi: 10.1103/physreva.107.032407.
- [111] Aidan Pellow-Jarman, Shane McFarthing, Ilya Sinayskiy, Daniel K. Park, Anban Pillay, and Francesco Petruccione. The effect of classical optimizers and ansatz depth on qaoa performance in noisy devices. *Scientific Reports*, 14(1), July 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-66625-6.
- [112] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, September 1956. ISSN 0003-4851. doi: 10.1214/aoms/1177728190.
- [113] Jürgen Branke and Christian Schmidt. *Selection in the Presence of Noise*, page 766–777. Springer Berlin Heidelberg, 2003. ISBN 9783540451051. doi: 10.1007/3-540-45105-6_91.
- [114] Sebastian Stich and Harsh Harshvardhan. Escaping local minima with stochastic noise. In *OPT2021: 13th Annual Workshop on Optimization for Machine Learning*, 2021.
- [115] James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, March 1992. ISSN 0018-9286. doi: 10.1109/9.119632.
- [116] Enrico Fontana, M. Cerezo, Andrew Arrasmith, Ivan Rungger, and Patrick J. Coles. Non-trivial symmetries in quantum landscapes and their resilience to quantum noise. *Quantum*, 6:804, September 2022. ISSN 2521-327X. doi: 10.22331/q-2022-09-15-804.
- [117] Pablo Díez-Valle, Diego Porras, and Juan José García-Ripoll. Quantum variational optimization: The role of entanglement and problem hardness. *Physical Review A*, 104(6), December 2021. ISSN 2469-9934. doi: 10.1103/physreva.104.062426.
- [118] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte. Reachability deficits in quantum approximate optimization. *Physical Review Letters*, 124(9), March 2020. ISSN 1079-7114. doi: 10.1103/physrevlett.124.090504.

Bibliography

- [119] IBM Quantum. Development Innovation Roadmap, 2024. URL https://www.ibm.com/quantum/assets/IBM_Quantum_Development_&_Innovation_Roadmap.pdf. [Accessed on 2024-10-16].
- [120] Quantinuum. Quantinuum Unveils Accelerated Roadmap to Achieve Universal, Fully Fault-Tolerant Quantum Computing by 2030, 2024. URL <https://www.quantinuum.com/press-releases/quantinuum-unveils-accelerated-roadmap-to-achieve-universal-fault-tolerant-quantum-computing-by-2030>. [Accessed on 2024-10-16].
- [121] Guglielmo Mazzola and Giuseppe Carleo. Exponential challenges in unbiasing quantum monte carlo algorithms with quantum computers, 2022. URL <https://arxiv.org/abs/2205.09203>.
- [122] D. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. Concorde TSP Solver, 2003. URL <https://www.math.uwaterloo.ca/tsp/concorde/index.html>. [Accessed on 2024-10-21].
- [123] David L. Applegate, Robert E. Bixby, Vasek Chvatal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, USA, January 2007. ISBN 0691129932.
- [124] Fabio Furini, Emiliano Traversi, Pietro Belotti, Antonio Frangioni, Ambros Gleixner, Nick Gould, Leo Liberti, Andrea Lodi, Ruth Misener, Hans Mittelmann, Nikolaos V. Sahinidis, Stefan Vigerske, and Angelika Wiegele. QPLIB: a library of quadratic programming instances. *Mathematical Programming Computation*, 11(2):237–265, September 2018. ISSN 1867-2957. doi: 10.1007/s12532-018-0147-4.
- [125] Craig S. Hamilton, Regina Kruse, Linda Sansoni, Sonja Barkhofen, Christine Silberhorn, and Igor Jex. Gaussian boson sampling. *Physical Review Letters*, 119(17), October 2017. ISSN 1079-7114. doi: 10.1103/physrevlett.119.170501.
- [126] Mateusz Słysz, Krzysztof Kurowski, and Grzegorz Waligóra. Solving combinatorial optimization problems on a photonic quantum computer, 2024. URL <https://arxiv.org/abs/2409.13781>.
- [127] Pablo Bermejo and Román Orús. Variational quantum non-orthogonal optimization. *Scientific Reports*, 13(1), June 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-37068-2.
- [128] Filip B. Maciejewski, Bao Gia Bach, Maxime Dupont, P. Aaron Lott, Bhuvanesh Sundar, David E. Bernal Neira, Ilya Safro, and Davide Venturelli. A multilevel approach for solving large-scale qubo problems with noisy hybrid quantum approximate optimization, 2024. URL <https://arxiv.org/abs/2408.07793>.
- [129] Tavis Bennett, Lyle Noakes, and Jingbo Wang. Non-variational quantum combinatorial optimisation, 2024. URL <https://arxiv.org/abs/2404.03167>.

Bibliography

- [130] Po-Wei Huang and Patrick Rebentrost. Post-variational quantum neural networks, 2024. URL <https://arxiv.org/abs/2307.10560>.
- [131] Andrew J. Daley, Immanuel Bloch, Christian Kokail, Stuart Flannigan, Natalie Pearson, Matthias Troyer, and Peter Zoller. Practical quantum advantage in quantum simulation. *Nature*, 607(7920):667–676, July 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04940-6.
- [132] Benedikt Fauseweh. Quantum many-body simulations on digital quantum computers: State-of-the-art and future challenges. *Nature Communications*, 15(1), March 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-46402-9.
- [133] F. Barahona, M. Jünger, and G. Reinelt. Experiments in quadratic 0–1 programming. *Mathematical Programming*, 44(1–3):127–137, May 1989. ISSN 1436-4646. doi: 10.1007/bf01587084.

Abbreviations

- AQC** Adiabatic Quantum Computing. 11
- AR** approximation ratio. 22, 23, 48
- BENQO** Block Encoding Quantum Optimizer. 12, 14, 19, 25–28, 30, 31, 33, 34, 36, 38–40, 42, 52, 57–59, 62, 70, 71, 74, 93
- BP** Barren Plateau. 16, 17, 36, 37
- MAD** mean absolute deviation. 21, 32, 36–38, 57, 94
- MaxCut** Maximum Cut Problem. 8, 41, 65, 67
- MSE** mean squared error. 21, 37, 38, 51, 52, 76
- NAR** normalized approximation ratio. 23, 48
- NGD** Normalized Gradient Descent. 15, 56–59
- NISQ** Noisy Intermediate-Scale Quantum. i, 1, 5, 12, 16–19, 25, 28, 55, 62, 64
- PCC** Pearson correlation coefficient. 22, 28, 29
- PTAS** Polynomial-Time Approximation Scheme. 8
- QAOA** Quantum Approximate Optimization Algorithm. 11, 12, 16, 19, 25–28, 31–35, 37, 38, 41, 42, 60, 62, 65, 71, 72, 93
- QUBO** Quadratic Unconstrained Binary Optimization. i, 8, 9, 20, 21, 25, 26, 42, 59–61, 63, 66, 67, 93
- RAE** relative absolute error. 22, 28, 29, 32, 52, 56–58
- TSP** Traveling Salesperson Problem. 8, 9, 20, 41, 60
- VQA** Variational Quantum Algorithm. i, 1, 2, 9–12, 16–19, 22, 23, 25–29, 31–35, 37–42, 47, 51, 52, 55, 58, 60, 62–64, 93, 94
- VQE** Variational Quantum Eigensolver. 11, 16, 19, 25–28, 31–34, 40–42, 53, 60, 62, 65, 70, 72, 75, 93
- VQLS** Variational Quantum Linear Solver. 41, 42

List of Figures

2.1	Representative Illustration of the VQA workflow.	10
2.2	Circuit Representation of a Two-Local Ansatz for VQE	11
2.3	Circuit Representation of the QAOA Ansatz	12
2.4	BENQO’s Circuit Representation of \mathbf{U}	14
2.5	Hadamard Test for Measuring $\langle \mathbf{U} \rangle$	14
3.1	Components of the Performance Assessment of a VQA	19
3.2	Example of a General QUBO Problem	21
4.1	Comparison of the Circuit Depths of the three VQAs	27
4.2	Comparison of the Hardware-Induced Errors in the Loss Functions of the three VQAs.	29
4.3	Distribution of the Hardware-Induced Error Across the Loss Landscape	30
4.4	Comparison of the Finite Sampling Errors of the three VQAs	32
4.5	Distribution of the Finite Sampling Errors Across the Loss Landscape	33
4.6	Illustrative Loss Landscapes of the three VQAs	35
4.7	Illustrative Loss Landscape with Two Representations of Noise	36
4.8	Mean Absolute Deviation of Loss Values over System Size	37
5.1	Scaling of the Success Probability with the Noise Level.	45
5.2	Graphical Goodness-of-Fit Analysis.	45
5.3	Visualization of Important Fit-Parameters	46
5.4	Mean Approximation Ratio and Confidence over Error Level	48
5.5	Success Probability across various Noise Levels and System Sizes	50
5.6	Noise Resilience over System Size	51
5.7	Performance Limits over System Size	53
5.8	Distribution of the Optimality of Solution States	54
5.9	Exemplary Hardware Error and Regions of Problem Solvability	56
5.10	Number of Required Shots for Reliably Solving a QUBO Problem of Size n	59
B.1	Distribution of the Accumulated Hardware-Induced Errors Across the Full Loss Landscape	71
C.1	Distribution of Confidence Values over System Size	75
C.2	Performance Ranking of Classical Optimizers Under Consideration	76
C.3	Comparison of Different Fitting Functions	77

List of Tables

2.1	Circuit Diagrams and Matrix Representations of Selected Quantum Gates	5
4.1	Comparison of the Circuit Resources of the three VQAs	26
4.2	Mean Squared Error of the Fitted MAD Decay Curves	38
4.3	MAD Decay Parameters of Different Theory Curves	38
5.1	Summary of Related Work	41
5.2	Overview of Tested Optimizers	43
5.3	Relevant Fit Quantities for \hat{p}_{opt}	47
5.4	Mean Squared Error from the Fit of Decay Curves	52
5.5	Linear Fit Parameters of No-Noise Solvability with System Size	54
B.1	Specifications of IBM Quantum Processing Units	70
C.1	Summary of QISKIT Optimizers	73

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my official thesis advisor, Prof. Frank Pollmann at TUM, for his invaluable guidance, constructive feedback, and support throughout this year. His insights and thoughtful discussions were pivotal in shaping the direction of this research, and I am profoundly grateful for his mentorship.

I am equally indebted to my direct supervisor, Bendikt Poggel, at Fraunhofer IKS, whose consistent willingness to tackle challenges together and openness to brainstorming ideas was indispensable to completing this thesis. His encouragement and insightful input made this journey both smoother and more enjoyable.

I am also deeply thankful to our department head, Dr. habil. Jeanette Miriam Lorenz, at Fraunhofer IKS, for giving me the opportunity to work within such an engaging and collaborative environment. Her leadership and vision provided the foundation for my research project and were key to this enriching experience.

Special thanks go to my incredible team at Fraunhofer IKS, especially Paul Luger, Sebastian Egginger, Marita Oliv, Amine Bentellis, Nicola Franco, and Florian Geissler. Their sense of team spirit, unwavering support, and collective enthusiasm created a welcoming atmosphere that fostered productivity and creativity and was a constant source of motivation.

I further want to extend my appreciation to everyone who showed genuine interest in my research, asked thought-provoking questions, or encouraged me to communicate my findings. Special thanks to David Philipps, Julia Häusele, Bastian Heinrich, my sister Mirella, and my parents, Natascha and Valentin, for their unwavering support and belief in me. Their curiosity and encouragement helped me refine my ideas and reminded me of the broader significance of this work.

Lastly, I want to express my gratitude to the Hanns-Seidel-Foundation for financially supporting me throughout my studies at TUM by granting me a scholarship.

To all of you, thank you for making this thesis possible.