STUDY OF OPTIMIZATION ALGORITHMS FOR SUPERCONDUCTING QUBIT-BASED QUANTUM COMPUTERS

By

CARLES PEDRALS MANSILLA

Bachelor's degree in Engineering Physics
Universitat Politèctinca de Catalunya
Barcelona, Spain
2024

Submitted to the Faculty of
Telecommunications Engineering of
Barcelona, Spain in fulfillment of
the requirements for
the Degree of
Bachelor's degree in Engineering Physics
June, 2024

STUDY OF OPTIMIZATION ALGORITHMS FOR SUPERCONDUCTING QUBIT-BASED QUANTUM COMPUTERS

Thesis Approved by:

Dr. Mostafa Barani

Thesis Director

Finley Alexander Quinton

Main Advisor

Dr. Rossend Rey Oriol

Co-Director

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my main advisor, Finley Alexander Quinton, whose unwavering support, insightful feedback, and countless hours of reading and re-reading have been invaluable to the completion of this thesis. Your dedication and expertise have significantly shaped this work and its format, and I am truly grateful for your guidance.

I also wish to thank my thesis director, Dr. Mostafa Barani, for scheduling our meetings and monitoring my progress throughout this journey. Your oversight ensured that I stayed on track and met all necessary milestones.

Furthermore, I am grateful to my co-director, Dr. Rossend Rey Oriol, for granting me the opportunity to undertake this thesis. Your approval was essential in making this project possible, and I appreciate your support in this capacity.

Lastly, I would like to extend my thanks to my family and friends for their constant encouragement and understanding during this challenging yet rewarding process.

Acknowledgments reflect the views of the author and are not endorsed by *Universitat Politècnica de Catalunya*.

Name: STUDY OF OPTIMIZATION ALGORITHMS FOR SUPERCONDUCTING QUBIT-BASED QUANTUM COMPUTERS

Date of Degree: JUNE, 2024

Title of Study: Bachelor's Degree in ENGINEERING PHYSICS

Major Field: PHYSICS

Abstract:

Quantum computing represents a significant advancement in computational science, offering the potential to address problems through fundamentally different computational paradigms from classical computers. One of the current prime candidates to potentially demonstrate a quantum advantage on near-term Noisy Intermediate Scale Quantum (NISQ) devices is the Quantum Approximate Optimization Algorithm (QAOA). A highly promising application of this algorithm is to tackle modern NP-hard optimization problems. In this study, the application of the QAOA algorithm is tested on an energy grid problem, aiming at the allocation of renewable energy sources, pivotal to achieving Europe's Net-Zero goal by 2050. The focus is on the Scandinavian power grid and evaluating different strategies for performing parameter optimization of QAOA ansatz circuits under various initialization strategies. This process combines fine-tuning parameters using both quantum and classical methods. Multiple extensions inspired by literature such as the Multiangle Approach, initial parameter distributions, and diverse methodologies are explored and their efficiency is discussed, highlighting Gaussian distribution as a great naive option. A new approach to improve convergence using neural networks is proposed and many techniques are applied to envision an efficient quantum algorithm capable of handling real-time demand of power grid partitioning. Error rates are discussed and the applicability of such technologies in this field is questioned rigorously.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Quantum computing is an interdisciplinary field that studies the application of quantum physics in computation. By exploiting the principles of quantum mechanics, quantum computers process information through quantum superposition and entanglement, enabling the simultaneous exploration of multiple solution spaces. This technological leap is particularly promising for optimization problems, where classical methods often encounter limitations in scalability and efficiency. The theoretical foundation of quantum computing was established by pioneers such as Richard Feynman and David Deutsch, who proposed the concept of machines capable of simulating quantum phenomena. Over the past few decades, progress in quantum hardware and algorithms has transformed these theoretical constructs into experimental platforms.

Optimization is one of the most compelling applications of quantum computing. Classical optimization algorithms often become inefficient and slow as the size and complexity of the data increase due to their sequential nature. In contrast, quantum algorithms can exploit the parallelism inherent in quantum mechanics to their advantage. This parallelism can lead to substantial decreases in computation time for certain classes of problems. Consequently, quantum computing is particularly relevant in fields where traditional methods become intractable.

1.1 Quantum Computing in the Near Term

Quantum computing, applied to optimization problems, is an area of constant research. As of now, quantum computing remains largely experimental [3], with many of its practical

applications still in the early stages. However, specific instances and models have conjectured how quantum approaches could outperform classical algorithms under certain conditions, marking significant milestones in the field.

One of the significant advancements in quantum technologies is the development of quantum computers utilizing quantum gates and quantum annealing [4] [5]. Companies, including IBM and D-Wave, have been at the forefront of this development, creating accessible systems for potential practical applications. These systems could, for specific cases [6] provide potential speedups over classical methods, which, if consistently verified, would support quantum applicability in fields currently dominated by classical methods [7]. Although the debate on whether these systems can provide a true quantum speedup over classical solutions persists, there are cases, particularly in materials science, magnetic analysis, and simulations; where quantum annealing, after technological advances, has the potential to outperform existing methods [8] [9].

Although these computers can produce valid results, it is crucial to contextualize their current capabilities. We are in the Noisy Intermediate-Scale Quantum (NISQ) era [10]. The term "noisy" indicates that our control over qubits is imperfect, meaning they are not fault-tolerant. "Intermediate-scale" refers to the number of qubits in operation, typically ranging from 50 to a few hundred, although recent advancements have surpassed the 1000-qubit mark [11]. Given these limitations, hybrid solutions that combine classical and quantum computing are currently the most effective approach.

A particularly interesting area in quantum optimization is the Quantum Approximate Optimization Algorithm (QAOA) [12]. Initially proposed as a method for solving combinatorial problems, QAOA operates by creating a parameterized quantum circuit whose parameters are optimized classically to minimize a specific cost function. This hybrid approach exploits both quantum and classical computation, making it more feasible given the current limitations of quantum computers. While QAOA is demanding in terms of quantum resources, it remains more practical than fully quantum algorithms at this stage of

technological development. Recent studies have demonstrated that QAOA has the potential to perform competitively with classical algorithms, especially for specific instances of combinatorial optimization problems [12, 13].

In experimental settings, IBM and Google have recently demonstrated that quantum processors have the potential to simulate certain quantum systems more efficiently than the world's fastest supercomputers [7]. However, these claims have spurred advancements in classical algorithms, thereby casting doubt on the validity of the so-called "quantum supremacy" by suggesting that classical methods may not yet have reached their optimal performance. While this achievement does not directly translate to solving optimization problems, it underscores the capability of quantum systems to handle exponentially large computational spaces, indicating potential future applications in optimization tasks where classical computers encounter limitations in scalability and speed.

1.2 Quantum Computing Applied to Power Distribution Grids

As society advances and the demand for electrical energy increases, particularly in localized areas, the need to prevent outages and efficiently distribute electricity becomes critical. To achieve this, the power grid must be optimized in real-time. This optimization involves aggregating the power grid into nodes that consume or produce energy. It is essential to consider a sufficient number of nodes to ensure no relevant information is lost while simultaneously limiting the number of data points to maintain computational feasibility [14].

Around the 1940s, for complex analyses, graphical methods such as nomograms were widely used [15], or analog systems where mechanical parts could be modified to simulate the network [16]. These tools allowed engineers to solve system equations visually, facilitating quicker solutions to common problems such as load flow analysis and short circuit calculations.

The advent of digital computers marked a paradigm shift in power grid calculations and numerous other research fields. Digitalization facilitated more complex mathematical mod-

els and iterative algorithms capable of handling large-scale systems. This era witnessed the development of sophisticated software for load flow studies, fault analysis, and optimal power flow, which significantly enhanced the accuracy and efficiency of grid management. The introduction, in 1967, of methods such as the Fast Decoupled Load Flow and Newton-Raphson techniques [17] revolutionized how engineers approached power distribution problems, allowing for real-time simulation and management of power grids.

Classical methods have continually improved over time; however, the rise of distributed generation, where clients also act as producers [18], has brought about profound changes in electricity distribution networks. The presence of multiple generators within the network enhances the ability to balance surpluses and deficits [19], but it necessitates much more complex and time-consuming algorithms. Similar challenges have emerged in placing charging stations in smart cities, balancing the total throughput [20].

While minor changes in network size may not seem significant, the routing and load distribution across numerous substations and transmission lines can be modeled as a graph partitioning problem, which involves dividing the network into k groups under specific constraints. This task is currently tackled using classical algorithms that grow **exponentially** more complex as the network expands increasing the number of nodes [21]. Quantum computing, however, offers a different approach. Unlike classical systems that compute power distribution scenarios sequentially in parallel processors, quantum computing employs quantum superposition to evaluate multiple configurations simultaneously in the same processor [22, 23]. This capability has demonstrated theoretical speedups in various other problem domains, suggesting that quantum algorithms could potentially achieve exponential speedups in solving optimization problems related to power distribution, compared to traditional methods. For power grid optimization, this would mean dynamically adjusting to changes in demand and supply, more effectively integrating renewable energy sources, and enhancing the overall stability and reliability of the energy infrastructure [24].

1.3 Outline

This work evaluates the performance of several configurations of the Quantum Approximate Optimization Algorithm (QAOA) on a simplified Scandinavian power grid problem. The objective is to define self-sufficient partitions capable of exchanging energy while maintaining a balance in the number of nodes per partition and respecting current power line connections. Based on a published study on clustering the German power distribution network using quantum technologies provided by D-Wave [25], this thesis adapts its approach by replacing quantum annealing with QAOA. The proposed Hamiltonian has been modified to reduce the number of necessary qubits, but the core principles remain unchanged. This research aims to measure the efficiency of QAOA in solving the partitioning problem and to explore its potential and limitations.

The structure will be the following. We will start with a brief but necessary introduction to the Laws of Quantum Mechanics and basic principles on Quantum Circuits in Chapter 2. Once acquainted with the underlying physics, algorithm theory and Quantum Algorithms are described in Chapter 3. Our main focus will be on the QAOA and additional technologies will be discussed. Employing the previous knowledge, we will apply the developed tools to a real-case scenario in Chapter 4 and finally analyze the sustainability of our project in Chapter 5.

CHAPTER 2

QUANTUM COMPUTING FUNDAMENTALS

In this chapter, the fundamental components and principles underlying quantum circuits will be explored, starting from the basics and progressing to more complex concepts. Beginning with a dive into the framework of quantum mechanics in Section 2.1 and an introduction to the fundamental unit of quantum computing, the qubit, in Section 2.2. Following this, we delve into the various operations applied to qubits in Section 2.3, and finally, we discuss how these operations impact quantum computing in Section 2.4 and their implementations in Section 2.5.

2.1 Quantum Mechanics Introduction

Quantum mechanics is a fundamental field of physics that provides a mathematical framework to explain and predict the behavior of quantum phenomena observed in experimental results. Given that the behavior of qubits is governed by the principles of quantum mechanics, a thorough understanding of this framework is crucial. This chapter begins with an introduction to the formulation of quantum mechanics, starting with the Schrödinger equation

$$i\hbar\frac{\partial\psi(\mathbf{x},t)}{\partial t} = \mathbf{H}\psi(\mathbf{x},t)\ . \tag{2.1.1}$$

Usually in the form where the Hamiltonian is expressed as

$$i\hbar\frac{\partial\psi(\mathbf{x},t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi(\mathbf{x},t) + V(\mathbf{x},t)\psi(\mathbf{x},t)\ . \tag{2.1.2}$$

Functions that are solutions of this differential equation represent possible states of the physical system under study. However, not all solutions are physically meaningful; valid functions must be normalized within a specific space, namely the Hilbert space, according to a certain norm, the L^2-norm.

2.1.1 Mathematical Background

Hilbert Space

In Equation (2.1.2), if the Hamiltonian represents a physical system, meaning it fulfills $H = H^\dagger$ (and can thus generate a basis of the space), then every isolated physical system can be associated with a separable, complex Hilbert space:

- Separable, meaning that exists a countable (infinite or not), dense subset in that space where every nonempty subset of elements contains at least one element of the sequence. For $S \subseteq H$

$$x \in H, \exists s \in S : ||x - s|| < \epsilon, \forall \epsilon > 0 \ . \tag{2.1.3}$$

  It implies that there exists a sequence of elements of the space that can approximate (with arbitrary precision) any element of the space. This fact is important because it allows us to define a basis, such that any vector of the system can be approximated infinitely close by a linear combination of the basis.

- Complex, meaning that complex coefficients are allowed.

- A Hilbert space is a vector space equipped with the structure of inner product space, which is also complete. This means it follows the rules of vector addition and scalar multiplication, and includes a scalar product. Additionally, it is complete, meaning any Cauchy sequence of elements in the space converges to an element within the space.

The scalar product is

$$||\psi|| = \sqrt{\int_{\mathbb{R}^3} \psi^* \psi d\mathbf{x}} \, . \tag{2.1.4}$$

We will only consider functions with finite norms as valid physical solutions so they conform to probability distribution rules, i.e. they must be bounded to be normalized.

One of the properties of such spaces is the capability to represent a state of the system as a vector. Say we have an orthonormal basis $\{\phi_i\}$ with n elements, then

$$\psi = \sum_i c_i \phi_i = \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} \, , \qquad \psi^* = \sum_i c_i^* \phi_i^* = \begin{pmatrix} c_0^* & \dots & c_{n-1}^* \end{pmatrix} \, . \tag{2.1.5}$$

The scalar product can be expressed as

$$\int \psi^*(x,t)\psi(x,t)d\mathbf{x} = \begin{pmatrix} c_0^* & \dots & c_{n-1}^* \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} \, . \tag{2.1.6}$$

Dirac's Notation

To ease the notation in further chapters, Dirac's notation will be used. We define the following symbols:

- Ket, an element of the solution space will be expressed as $\begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} \rightarrow |\psi\rangle$.
- Bra, an element of the dual space such that $\begin{pmatrix} c_0^* & \dots & c_{n-1}^* \end{pmatrix} \rightarrow \langle\psi|$.
- BraKet, being the definition of the scalar product $\begin{pmatrix} c_0^* & \dots & c_{n-1}^* \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1}. \end{pmatrix} \rightarrow \langle\psi|\psi\rangle$

- KetBra, the definition of the outer product $\begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0^* & \dots & c_{n-1}^* \end{pmatrix} \rightarrow |\psi\rangle \langle\psi|$.

2.1.2 Postulates of Quantum Mechanics

When describing a physical system, it is essential to define its **states**, **observables**, and **dynamics** [26]

States

Postulate I.a: *System State Postulate:*

Any state of a physical system at a given instant of time can be represented by a vector in the state space.

As it is a separable space, this also means that with a countable number of observations, we will be able to sufficiently approximate any state by adjusting the coefficients. Additionally, states that differ only by a phase factor (which is unmeasurable) are considered equivalent. Hence, our state vectors are defined not just as vectors but as rays (combinations of states differing by a phase).

When introducing composite systems:

Postulate I.b: *Composite State Postulate:* The Hilbert space of a composite system is the tensor product of the Hilbert spaces of the individual systems.

This postulate defines how to express the state of a combined system. For two systems A and B as

$$\psi_{AB} = \psi_A \otimes \psi_B \ . \tag{2.1.7}$$

It also introduces the concept of entangled states, which cannot be described as separable states and will be discussed further.

Measurements

Physical systems can be measured to determine their states. However, in quantum mechanics, measurements affect the system in the way that is explained by the following postulates

Postulate II.a: *Observing physical quantities:* Every measurable physical quantity is described by a Hermitian operator in the Hilbert state space. Being Hermitian, its eigenvectors can form an orthonormal basis, and the result of a measurement corresponds to one of these eigenvalues.

Postulate II.b: *Results of measurement:* The probability of obtaining a certain eigenvalue has the form:

- For discrete (perhaps degenerate) spectrums $P(a_n) = \sum_i^{g_n} |\langle a_n^i | \psi \rangle|^2$.
- For continuous non-degenerate space $dP(a_n) = |\langle \alpha | \psi \rangle|^2 d\alpha$.

Postulate II.c: *Effects of measurement:* After a measurement is performed, only one result is obtained, and the state is re-normalized to reflect this outcome.

Dynamics

Lastly, the evolution of the system over time is defined by

Postulate III.a: *Differential Equation:*The evolution of the system is governed by a differential equation involving the energy observable operator

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = \mathbf{H} |\psi\rangle \ . \tag{2.1.8}$$

Postulate III.b: *Time Evolution:* The evolution of the system is defined by a unitary transformation on the initial state such that

$$|\psi(t)\rangle = \mathcal{T} e^{-\frac{i}{\hbar} \int_{t_0}^{t} \mathbf{H}(t') dt'} \ . \tag{2.1.9}$$

Where $\mathcal{T}$ is the time ordering operator, accounting for non-commuting Hamiltonians, as they generally do not commute.

Density Matrix

In some cases, systems are better described by a density matrix. The previous postulates are generalized as follows

Postulate I'.a: *System State Postulate:*

Any state of a physical system at a given instant of time can be represented by a linear operator ρ called density matrix that satisfies

$$\rho = \rho^{\dagger}, Tr(\rho) = 1, \langle\psi|\rho|\psi\rangle \geq 0, \forall\psi \in H \ . \tag{2.1.10}$$

This describes the system in a probabilistic manner, as the state is a mixture of pure states.

For measurements, we have the following generalization

Postulate II'.b: *Results of measurement:* The probability of obtaining a certain eigenvalue takes the form

$$P(a_n) = Tr(\rho \left|a_n\right\rangle \left\langle a_n\right|) \ . \tag{2.1.11}$$

Postulate II'.c: *Effects of measurement:* After a measurement is performed, the state is re-normalized

$$\rho_{a_n} = \frac{1}{P(a_n)} \left|a_n\right\rangle \left\langle a_n\right| \rho \left|a_n\right\rangle \left\langle a_n\right| \ . \tag{2.1.12}$$

The rest of the postulates are applied in the same way as to pure states.

2.1.3 Allowed Hamiltonians

All Hamiltonians are required to be Hermitian Operators, which ensures that they represent **unitary evolutions**, all the eigenvalues are real and an orthonormal basis may always be found, defining by every energy a possible eigenstate or subspace of eigenstates [27]. All these properties allow for states to be normalized throughout the whole evolution, respecting the postulates of Quantum Mechanics

Ising Hamiltonians

The Ising model is a mathematical model of ferromagnetism in statistical mechanics, and its quantum version is fundamental to this thesis. The Ising Hamiltonian represents interactions between spins in a magnetic system and is used to describe and simulate quantum behavior in various materials and quantum computational models. The quantum version of the Ising Hamiltonian is slightly different from the original one and it reads as

$$H = -J \sum_{i,j} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x \ . \tag{2.1.13}$$

Where σ^z and σ^x represent Pauli matrices Z and X respectively, which have extreme importance in Quantum Physics. This Hamiltonian is the foundation of QAOA, the algorithm at the center of this thesis. The importance of the Ising Hamiltonian in QAOA stems from its ability to map computational problems onto inherently binary spin systems. That implies that all families of logistic problems with binary variables may (or not) be mapped to an Ising Hamiltonian. It also has relevance in the realm of simulating quantum systems, but it is out of the scope of this work.

2.1.4 Pauli Matrices

Pauli matrices are 2x2 complex matrices that are ubiquitous in Quantum Mechanics, and most commonly related to spin systems. Defined as

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \ , \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \ . \tag{2.1.14}$$

They are Hermitian and satisfy the following relationships:

- Commutation: $[\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k$
- Anti-commutation: $\{\sigma_i, \sigma_j\} = 2\delta_{ij}\mathbb{I}$

x	σ_x	σ_y	σ_z
σ_x	$\mathbb{I}$	$i\sigma_x$	$-i\sigma_y$
σ_y	-i σ_x	$\mathbb{I}$	$i\sigma_x$
σ_z	i σ_y	$-i\sigma_x$	$\mathbb{I}$

Where ϵ_{ijk} is the Levi-Civita, δ_{ij} Kronecker delta, and $\mathbb{I}$ is the Identity. Analogously, it can be seen in a table for in Section 2.1.4. In an attempt to simplify notation, Pauli matrices will be instantiated as X, Y, Z

Among some of their uses, their main importance is found in the following areas:

State representation

It can be proven that any state in the Bloch sphere can be represented as

$$\rho = \frac{1}{2}\left(\mathbb{I} + \vec{r}\vec{\sigma}\right) \ , \tag{2.1.15}$$

where $\vec{r}$ is the position vector on the Bloch sphere. This can be useful in certain mathematical operations allowing the decomposition of the density state into Pauli matrices.

Infinitesimal Generators of Rotations

Pauli matrices are also generators of both rotation groups $SU(2)$, Special Unitary group of degree 2, meaning all unitary operators that are 2 dimensional, and $SO(3)$, Special Orthogonal group of degree 3. The first concept allows us to parameterize any unitary operator as

$$U = a\mathbb{I} + i(b\sigma_x + c\sigma_y + d\sigma_z) \ , \tag{2.1.16}$$

where $a^2 + b^2 + c^2 + d^2 = 1$ to ensure a unitary operator. That means any system evolution can be expressed in terms of the Pauli matrices (and the Identity matrix). In particular, each Pauli matrix x, y, z, when multiplied by an angle and exponentiated, generates a rotation over that axis:

$$RX(\theta) = e^{-i\frac{\theta}{2}\sigma_x} \ , \tag{2.1.17}$$

$$RY(\theta) = e^{-i\frac{\theta}{2}\sigma_y} \ , \tag{2.1.18}$$

$$RZ(\theta) = e^{-i\frac{\theta}{2}\sigma_z} \ . \tag{2.1.19}$$

Generating the SO(3), the rotational group of the 3D Euclidian space [28].

Operator Basis

It can be proven that the Pauli matrices form a base for all Hermitian operators, and in our case, this will mean that all observables, including the Hamiltonian, can be expressed in terms of them. Any Hermitian matrix can be written as

$$H = \begin{pmatrix} x & y+iz \\ y-iz & \omega \end{pmatrix} \ , \tag{2.1.20}$$

where x, y, z, ω are all real numbers. Now it follows that

$$H = \frac{x+\omega}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + y\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} - z\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\frac{x-\omega}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.1.21}$$

$$= \begin{pmatrix} x & y+iz \\ y-iz & \omega \end{pmatrix} \ . \tag{2.1.22}$$

Generalizing to n qubits, as we perform the tensor product of the basis of each space, we obtain a basis of 4^n elements

$$\{I_i \otimes I_j, \ldots, I_i \otimes \sigma_j^z, \sigma_i^x \otimes I_j, \ldots \sigma_i^z \otimes \sigma_j^z\} \ 1 \leq i, j \leq n \ . \tag{2.1.23}$$

This decomposition is used when operators are represented as a **Sparse Pauli** operator, otherwise, the number of terms needed to compute will grow exponentially. This fact will be seen further in the study case.

Then for a Hermitian matrix,

$$\hat{H} = \sum_{k=0}^{4^n-1} \omega_k \hat{P}_k \tag{2.1.24}$$

$$\langle \hat{H} \rangle = \sum_{k=0}^{4^n-1} \omega_k \sum_{j=0}^{2^n-1} p_{kj} \lambda_{kj} \ , \tag{2.1.25}$$

where λ_{kj} is the j-th eigenvalue of the k-th operator, and as there can be as many states as 2^n, the summation over j takes into account all the range. When put into practice, both summations are done over a $polynomial(n)$ amount of times where n is the number of qubits in cases where the operator can be expressed as a Sparse Pauli Operator and some eigenvalues can be disregarded. The relevance of this decomposition is found when estimating the energy value of a Hamiltonian [29].

2.2 The Qubit

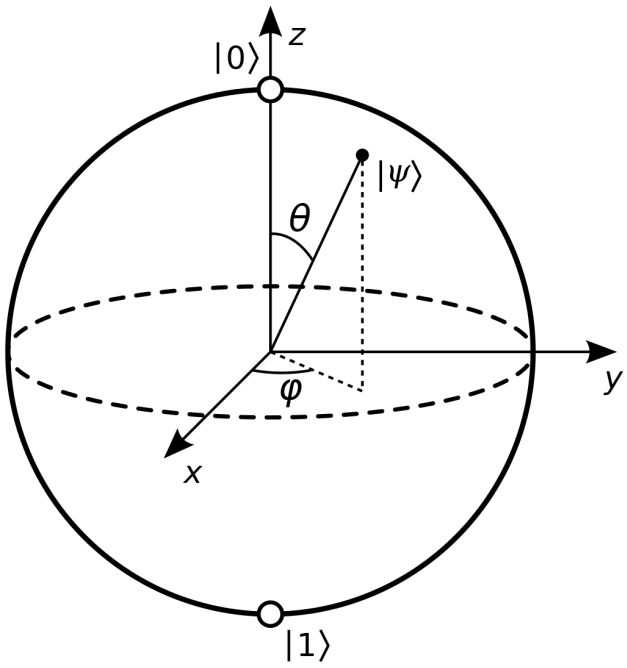


Figure 2.1: Visual representation of a qubit and its possible states. Extracted from [1]

The qubit is defined as the fundamental unit of quantum information [30], analogous to the "*bit*" in classical information theory. Unlike the classical bit, which is confined to the

binary states 0 or 1, the qubit can exist in states between them, due to a physics phenomenon called **quantum superposition** (see Figure 2.1). In mathematical terms, if the state of a system were to be defined as $f(0)$ or $f(1)$, a classical bit (cb) and a quantum bit (qb) would have the following definitions

$$cb = \{f(0), f(1)\} \tag{2.2.1}$$

$$qb = \alpha f(0) + \beta f(1) \tag{2.2.2}$$

Where α and β are complex numbers that follow certain restrictions.

This property is related to the capability of Quantum Computers to perform multiple computations at once as both states, 0 and 1, coexist at the same time. This concept will be explained in more depth and will appear in the following chapters.

When multiple qubits interact, a distinct quantum mechanical phenomenon known as quantum entanglement arises. Quantum entanglement establishes a correlation between the states of qubits such that the state of one qubit instantaneously affects the state of another, regardless of the distance between them. This phenomenon not only transforms computational processes compared to classical methods but also underpins various other fields, such as quantum encryption, where this rapid information transfer can be utilized.

Lastly, without delving too deeply into details before properly defining certain key concepts, it becomes evident that the traditional measure of information in a system, as represented by Shannon's entropy $S_{Shannon} = -\sum_i P(x_i) log_2[P(x_i)]$ requires an adaptation. This revision reflects the complex nature of quantum states where superposition and entanglement redefine how information is quantified, stored, and manipulated, necessitating a quantum-specific approach to information theory.

2.2.1 Coherence Time

Coherence time is a critical metric for any quantum system, reflecting the duration a qubit can maintain its quantum state without significant degradation. This property is vital because quantum computations depend on the superposition of states, which can be disrupted by interactions with the environment—known as decoherence. Decoherence leads to the loss of quantum information, manifesting as errors during computation. Therefore, maintaining long coherence times while minimizing decoherence and error rates is fundamental.

2.2.2 Scalability

Scalability involves the capacity to increase the number of qubits within a quantum system without excessive increases in error rates or operational complexity. A scalable quantum architecture allows for tackling more computationally intensive problems that require larger qubit arrays. Some architectures that will be mentioned in the following paragraphs will brush on the idea of introducing modular processors.

2.3 Quantum Gates

In the previous section, we found out about the quantum operators and how we can generate the most general version of them. Now we will use this theory to design more complex transformations to solve a problem. In the process, we will be defining circuits of operators that mainly have an effect on the Hamiltonian of the system and thus apply a unitary transformation.

2.3.1 Single Qubit Gates

The most common gates used, using the basis $\{|x\rangle, x = 0, 1\}$ and the corresponding matrix form in canonical basis, would be:

- Identity:

$$\mathbb{I}\,|x\rangle = |x\rangle \ , \tag{2.3.1}$$

$$\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \ , \tag{2.3.2}$$

as shown in Figure 2.2a and Figure 2.2b .

- NOT:

$$X\,|x\rangle = |1-x\rangle \ , \tag{2.3.3}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \tag{2.3.4}$$

as shown in Figure 2.2c and Figure 2.2d .

- Hadamard:

$$H\,|x\rangle = \frac{1}{\sqrt{2}}((-1)^x\,|x\rangle + |1-x\rangle) \ , \tag{2.3.5}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \ , \tag{2.3.6}$$

as shown in Figure 2.2e and Figure 2.2f.

- Rotational Gates:

  –

$$RX(\theta)\,(a\,|0\rangle + b\,|1\rangle) = \left(a \cdot cos(\frac{\theta}{2}) - b \cdot i \cdot sin(\frac{\theta}{2})\right)|0\rangle + \left(-a \cdot i \cdot sin(\frac{\theta}{2}) + b \cdot cos(\frac{\theta}{2})\right)|1\rangle \ , \tag{2.3.7}$$

$$RX(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \ , \tag{2.3.8}$$

as shown in Figure 2.3a and Figure 2.3b.

–

$$RY(\theta)\left(a\left|0\right\rangle + b\left|1\right\rangle\right) = \left(a \cdot cos(\frac{\theta}{2}) - b \cdot sin(\frac{\theta}{2})\right)\left|0\right\rangle + \left(a \cdot sin(\frac{\theta}{2}) + b \cdot cos(\frac{\theta}{2})\right)\left|1\right\rangle \ , \tag{2.3.9}$$

$$RY(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \ , \tag{2.3.10}$$

as shown in Figure 2.3c and Figure 2.3d.

–

$$RZ(\theta)\left(a\left|0\right\rangle + b\left|1\right\rangle\right) = a \cdot e^{-i\frac{\theta}{2}}\left|0\right\rangle + b \cdot e^{-i\frac{\theta}{2}}\left|1\right\rangle \ , \tag{2.3.11}$$

$$RZ(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \ , \tag{2.3.12}$$

as shown in Figure 2.3e and Figure 2.3f

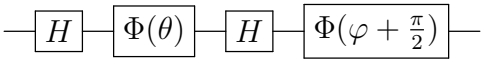
Note that the Hadamard is an operator that **transforms a single defined state into a superposition of states**. Its ability to create superposition reflects the quantum nature of particles, enabling quantum computers to explore many possibilities simultaneously.

These gates also have a matrix form in our canonic basis:

- Identity: $\mathbb{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ as shown in Figure 2.2a and Figure 2.2b .
- NOT: $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ as shown in Figure 2.2c and Figure 2.2d .
- Hadamard: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ as shown in Figure 2.2e and Figure 2.2f.
- $RX(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ as shown in Figure 2.3a and Figure 2.3b

- $RY(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ as shown in Figure 2.3c and Figure 2.3d
- $RZ(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$ as shown in Figure 2.3e and Figure 2.3f

Using only Phase shifts and Hadamards, it can be demonstrated that we can generate the most general case:



Which would result in the following set of operations to an initial state of the basis such as

$$|0\rangle \xrightarrow[H]{} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow[\Phi(\theta)]{} \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle) \xrightarrow[H]{} \frac{1}{2}((1+e^{i\theta})|0\rangle + (1-e^{i\theta})|1\rangle) \ . \tag{2.3.13}$$

Using the definition of the cosines and sines as exponential, we convert the last result to

$$e^{i\frac{\theta}{2}}(\cos\frac{\theta}{2}|0\rangle - i\sin\frac{\theta}{2}|1\rangle) \xrightarrow[\Phi(\varphi+\frac{\pi}{2})]{} e^{i\frac{\theta}{2}}(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle) \ . \tag{2.3.14}$$

So the general transformation applied is then

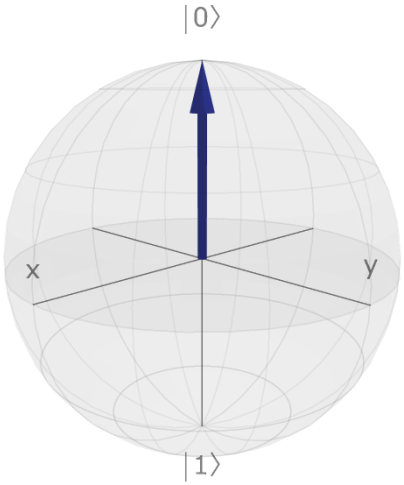
$$U = \Phi(\varphi + \frac{\pi}{2}) \cdot H \cdot \Phi(\theta) \cdot H \ . \tag{2.3.15}$$

2.3.2 Two Qubit Quantum Gates

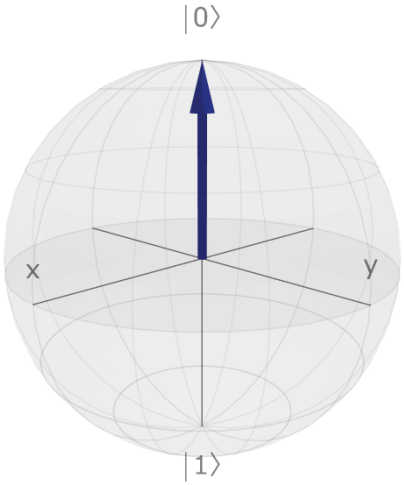
The previously discussed single-qubit gates affect individual qubits. If applied individually to each qubit, the global state can still be represented using a separation of states

$$|0\rangle \otimes |0\rangle \rightarrow_{U_1} |\psi_1\rangle \otimes |0\rangle \rightarrow_{U_2} |\psi_1\rangle \otimes |\psi_2\rangle \ . \tag{2.3.16}$$

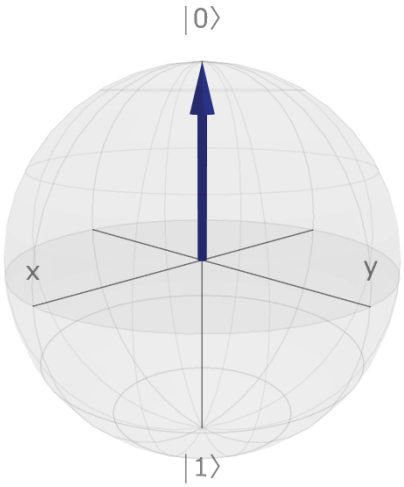
This implies that with only single-qubit gates, we cannot fully explore the new space generated by the tensor product of both Hilbert spaces. Consequently, no entanglement is



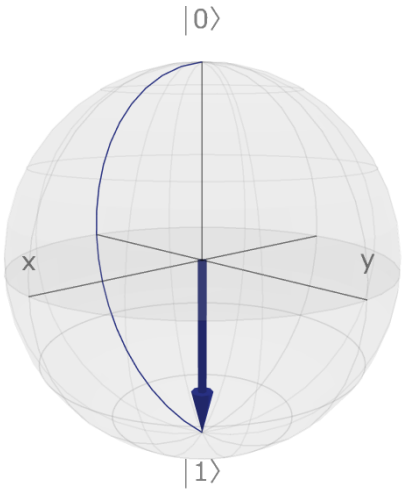
(a) Initial state before operator $\mathbb{I}$.



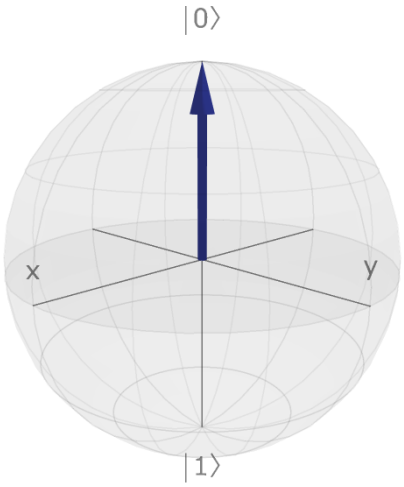
(b) The qubit is left unchanged.



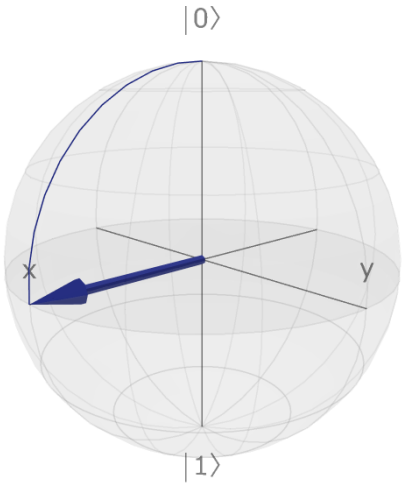
(c) Initial state before operator X.



(d) The qubit is flipped.

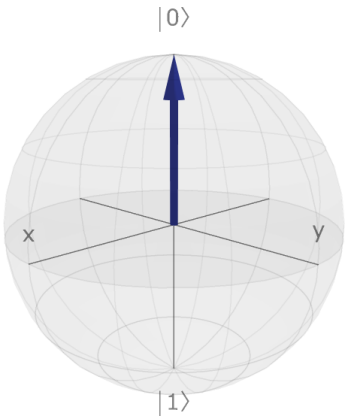


(e) Initial state before operator H.

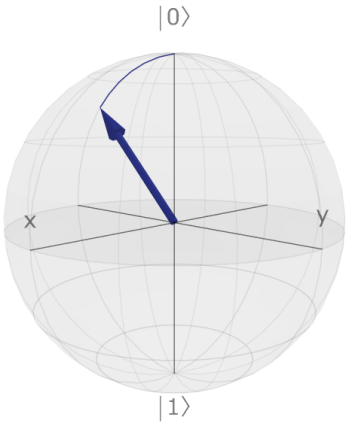


(f) Superposition has been created.

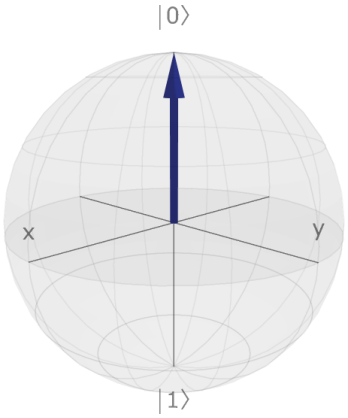
Figure 2.2: Effects of the operators $\mathbb{I}$, X and H.



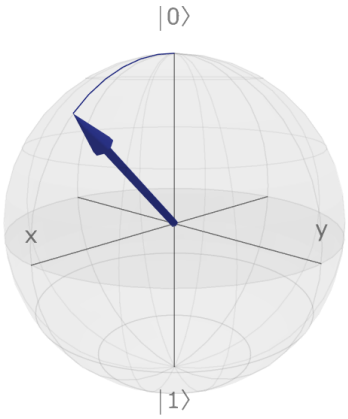
(a) Initial state before operator RX.



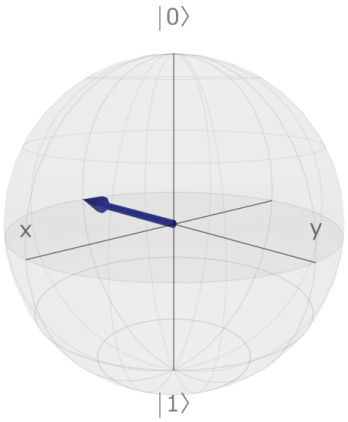
(b) Rotation around x-axis.



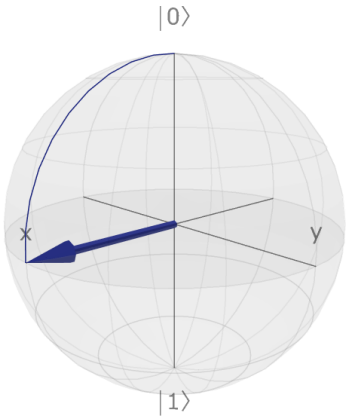
(c) Initial state before operator RY.



(d) Rotation around y-axis.



(e) Initial state before operator RZ.



(f) Equivalent to a phase change.

Figure 2.3: Effects of the operators RX, RY and RZ. Simulated with [2]

achieved. To prove this fact, let us generalize as much as possible single qubit gates. The derived expression is important because, internally, some simulators and hardware represent the previous single gate qubits using only this generalization [29].

Generalization of Single Qubit gates

Before that, we need to generalize the application of single-qubit gates to the new space we are dealing with. In the same way as with operators, that can be done with the tensor product. Let's demonstrate that, using the most general version of a unitary operator U

$$U\left|0\right\rangle = \cos\frac{\theta}{2}\left|0\right\rangle + \sin\frac{\theta}{2}e^{i\varphi}\left|1\right\rangle \ , \tag{2.3.17}$$

$$U\left|1\right\rangle = (-\sin\frac{\theta}{2}\left|0\right\rangle + \cos\frac{\theta}{2}e^{i\varphi}\left|1\right\rangle)e^{i\phi} \ . \tag{2.3.18}$$

$$U_{2D}\left|00\right\rangle = (\cos\frac{\theta_1}{2}\left|0\right\rangle + \sin\frac{\theta_1}{2}e^{i\varphi_1}\left|1\right\rangle) \otimes (\cos\frac{\theta_2}{2}\left|0\right\rangle + \sin\frac{\theta_2}{2}e^{i\varphi_2}\left|1\right\rangle) \ , \tag{2.3.19}$$

$$U_{2D}\left|01\right\rangle = (\cos\frac{\theta_1}{2}\left|0\right\rangle + \sin\frac{\theta_1}{2}e^{i\varphi_1}\left|1\right\rangle) \otimes (-\sin\frac{\theta_2}{2}\left|0\right\rangle + \cos\frac{\theta_2}{2}e^{i\varphi_2}\left|1\right\rangle)e^{i\phi_2} \ , \tag{2.3.20}$$

$$U_{2D}\left|10\right\rangle = (-\sin\frac{\theta_1}{2}\left|0\right\rangle + \cos\frac{\theta_1}{2}e^{i\varphi_1}\left|1\right\rangle)e^{i\phi_1} \otimes (\cos\frac{\theta_2}{2}\left|0\right\rangle + \sin\frac{\theta_2}{2}e^{i\varphi_2}\left|1\right\rangle) \ , \tag{2.3.21}$$

$$U_{2D}\left|11\right\rangle = (-\sin\frac{\theta_1}{2}\left|0\right\rangle + \cos\frac{\theta_1}{2}e^{i\varphi_1}\left|1\right\rangle)e^{i\phi_1} \otimes (-\sin\frac{\theta_2}{2}\left|0\right\rangle + \cos\frac{\theta_2}{2}e^{i\varphi_2}\left|1\right\rangle)e^{i\phi_2} \ . \tag{2.3.22}$$

Assuming U_1 and U_2 different operators and therefore have different parameters. This would yield the operator:

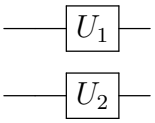
$$U_{2D} = \begin{pmatrix} \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2} & -\cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\phi_2} & -\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i\phi_1} & \sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\phi_1+\phi_2} \\ \cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\varphi_2} & \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i\varphi_2}e^{i\phi_2} & -\sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\varphi_2}e^{i\phi_1} & -\sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i\varphi_2}e^{i\phi_1+\phi_2} \\ \sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i\varphi_1} & -\sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\varphi_1}e^{i\phi_2} & \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i\varphi_1}e^{i\phi_1} & -\cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i\varphi_1}e^{i\phi_1+\phi_2} \\ \sin\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i(\varphi_1+\varphi_2)} & \sin\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i(\varphi_1+\varphi_2)}e^{i\phi_2} & \cos\frac{\theta_1}{2}\sin\frac{\theta_2}{2}e^{i(\varphi_1+\varphi_2)}e^{i\phi_1} & \cos\frac{\theta_1}{2}\cos\frac{\theta_2}{2}e^{i(\varphi_1+\varphi_2)}e^{i\phi_1+\phi_2} \end{pmatrix} . \tag{2.3.23}$$

As can be seen, it corresponds exactly to the tensor product of both operators

$$U_{2D} = \begin{pmatrix} \cos\frac{\theta_1}{2} & -\sin\frac{\theta_1}{2}e^{i\phi_1} \\ \sin\frac{\theta_1}{2}e^{i\varphi_1} & \cos\frac{\theta_1}{2}e^{i\varphi_1}e^{i\phi_1} \end{pmatrix} \otimes \begin{pmatrix} \cos\frac{\theta_2}{2} & -\sin\frac{\theta_2}{2}e^{i\phi_2} \\ \sin\frac{\theta_2}{2}e^{i\varphi_2} & \cos\frac{\theta_2}{2}e^{i\varphi_2}e^{i\phi_2} \end{pmatrix} = U_1 \otimes U_2 \ . \quad (2.3.24)$$

Then we have justified that by applying different 1 qubit transformations to each of the qubits, the resulting operator is the Kronecker product of both operators (which in the case of 2x2 matrices is easily solvable). Thus, no combination of single-qubit gates will not be able to generate entanglement.

We will express the previous transformation as:

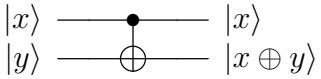


C-NOT Gate (CX)

This gate acts on 2 qubits, one acting as a control qubit and the other one as a target. If the control qubit is on 0, the target is not modified, otherwise, a NOT boolean function acts on the target

if $control = |0\rangle$ **then** $U_{target} = \mathbb{I}$

else $U_{target} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$



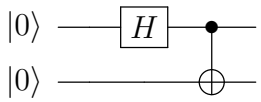
Where we can see that the control bit is unperturbed. The previous circuit can also be defined in operator form and matrix form, where $\oplus$ is the XOR operation and the basis is

$\{|x\rangle\,|y\rangle\,; x, y = 0, 1\}$

$$CNOT\,|x\rangle \otimes |y\rangle = |x\rangle \otimes |x \oplus y\rangle \ , \tag{2.3.25}$$

$$CNOT = \begin{pmatrix} \mathbb{I} & \varnothing \\ \varnothing & X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \ . \tag{2.3.26}$$

Combining this gate with a previous Hadamard gate for the control bit can yield the simplest circuit to generate entanglement:



That produces the output

$$|0\rangle \otimes |0\rangle \xrightarrow[H\otimes\mathbb{I}]{} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \xrightarrow[CNOT]{} \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle) \tag{2.3.27}$$

C-U Gate (CU)

The previous concept can be generalized to any unitary operator U, using a control (C) qubit that decides whether that operator will be applied or notations. C-NOT is a particular case of this gate

$$CU = \begin{pmatrix} \mathbb{I} & \varnothing \\ \varnothing & U \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{pmatrix} \ . \tag{2.3.28}$$

As a block diagonal matrix, the product of the matrix with its Hermitian conjugate results in the identity matrix. This type of block matrix typically arises when the transformation affects only one of the two qubits. Specifically, in this case, when the first qubit is in the $|0\rangle$

state, the transformation applied to the second qubit is the identity operation for all control gates. However, this is not generally the case for all control gates.

RZZ Gate

The RZZ gate is a two-qubit quantum gate that implements a rotation around the ZZ axis of the Bloch sphere. It is particularly significant in the context of quantum simulations of Ising models, as it directly corresponds to the Ising interaction term between spins in a quantum system. The RZZ gate is defined by the unitary operator

$$U_{RZZ}(\theta) = e^{-i\theta\sigma_z \otimes \sigma_z} \ , \tag{2.3.29}$$

where σ_z denotes the Pauli-Z matrix, and θ is the angle of rotation. The action of the RZZ gate on a two-qubit state can be described as adding a phase of θ only when both qubits are in the state $|1\rangle$. This characteristic makes it suitable for modeling interactions where the energy penalty is based on the alignment of spins in the quantum Ising model.

The matrix form of the $RZZ(\theta)$ gate is

$$RZZ(\theta) = \begin{pmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & e^{-i\theta} \end{pmatrix} \tag{2.3.30}$$

This representation shows that the RZZ gate applies a phase shift of $-\theta$ to the $|00\rangle$ and $|11\rangle$ states, and a phase shift of $+\theta$ to the $|01\rangle$ and $|10\rangle$ states, which corresponds to the conditional phase shift based on the simultaneous states of the two qubits involved.

This gate is not essential when presenting the main gates, but, the RZZ gate plays a considerable role in variational quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA), where it is used to entangle qubits and steer the quantum state towards an optimal solution of combinatorial optimization problems modeled by Ising-type interactions.

2.3.3 Gate Fidelity

Gate fidelity measures the precision with which quantum gates are implemented. High gate fidelity is essential to ensure that the quantum state manipulations conform closely to theoretical models, thus avoiding cumulative errors in computations. On the other hand, readout accuracy is crucial at the end of quantum operations to ensure that the qubit states are measured correctly. Inaccuracies in readout can lead to misinterpretation of the computational outcomes, undermining the reliability of the quantum computer.

2.3.4 Cross-talk

Connectivity refers to how effectively qubits within a quantum system can interact with one another. Enhanced connectivity reduces the need for intermediary qubits in multi-qubit operations, thereby improving the speed and efficiency of quantum gate operations. However, increased connectivity can also lead to higher cross-talk, where unintended interactions between qubits occur. Such interactions can introduce errors and disrupt the intended quantum operations.

2.4 Quantum Computing Principles

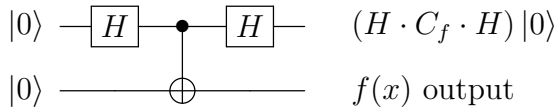
Once the building blocks have been introduced, we must address the peculiarities of quantum computing, both its strengths and weaknesses.

2.4.1 Quantum Paralelism

Quantum parallelism is the phenomenon that allows quantum computers to evaluate a function $f(x)$ for many different values of x simultaneously. This is accomplished by preparing a superposition of quantum states representing different inputs and applying a quantum operation that encodes the function.

Consider a simple function $f : \{0, 1\} \rightarrow \{0, 1\}$. It will be evaluated using a quantum

circuit for both possible inputs simultaneously. The circuit uses two qubits, one for the input and one as an ancilla (helper qubit) to store the output. $f(x)$ can be any simple operation, such as the NOT operation, where:



While this example might seem trivial, it highlights one of the unique aspects of quantum computing: the ability of superposition to facilitate massive parallel computation. The challenge then becomes designing transformations that are applicable and beneficial for our specific scenarios.

2.4.2 Quantum Entanglement

Quantum entanglement is a unique property of quantum systems where the quantum states of two or more particles become intertwined such that the state of one particle cannot be described independently of the state of the others. This phenomenon has profound implications for quantum computing.

In an entangled system, the measurement of one particle instantaneously affects the state of the other, regardless of the distance between them. This non-local characteristic is characteristic of quantum computing, enabling powerful computational techniques that are unattainable by classical means. For example, entanglement is essential for quantum algorithms like Shor's algorithm [6] for factoring large numbers and Grover's algorithm [31] for searching unsorted databases.

2.4.3 Quantum Adiabatic Theorem

In simple terms, the Quantum Adiabatic Theorem (QAT) states that, given a system in its ground state, if a change in the system conditions (change of the Hamiltonian) happens slowly enough, the system will adapt and stay in the ground state of the new conditions.

For the theorem to be true, we need to impose both ground states to be non-degenerate.

Formally, consider at any instant of time the equation

$$H(t)\,|n(t)\rangle = E_n(t)\,|n(t)\rangle \ . \tag{2.4.1}$$

For a given instant of time, if it was a constant Hamiltonian

$$|\psi(t)\rangle = \sum_n c_n(t)\phi_n(t) \qquad\qquad \phi_n(t) = \phi_n(0)e^{\frac{-iE_nt}{\hbar}} \ , \tag{2.4.2}$$

where $\{\phi_n(0)\}$ are the eigenstates that result from a constant application of the Hamiltonian defined at a time "t". This means that, for any instant, we solve the **time independent** Schrödinger Equation. In other words

$$|n(t)\rangle = |\phi_n\rangle \qquad \textit{For the n-th eigenstate of the Hamiltonian at time "t"} \ , \tag{2.4.3}$$

$$|m(t)\rangle = |\phi_m\rangle \qquad \textit{For the m-th eigenstate of the Hamiltonian at a time "t"} \ , \tag{2.4.4}$$

$$\langle m(t)|n(t)\rangle = 0 \qquad\qquad \forall n, m \tag{2.4.5}$$

It can be shown that, if no degeneracy is found in between states and the term $E_n(t) - E_m(t)$ (at any time t) is small enough compared to $H(t)$, so defining how slow we should perform the change to be adiabatic. Setting the starting point of this process to any state such that $c_n(0) = 1$ and the other states, by probability normalization, $c_m(0)\forall m, m \neq n$

$$|\Psi(t)\rangle = e^{i\gamma_n(t)}e^{i\theta_n(t)}\,|n(t)\rangle \tag{2.4.6}$$

So if the process starts in the n-th eigenstate, it finishes in the n-th eigenstate of the new system, given that the conditions of small changes are fulfilled. In particular:

$$|\Psi(t)\rangle = e^{i\gamma_0(t)}e^{i\theta_0(t)}\,|0(t)\rangle \tag{2.4.7}$$

Meaning that starting in the ground state of a known system, and performing the transition, we are in the ground state of the new system [32].

This process can be used to solve problems of a certain type. By evolving a quantum state using a Hamiltonian representing a binary problem, we can obtain the solution encoded as the ground state of the new Hamiltonian. However, for this method to be effective, certain conditions must be met: the evolution must be sufficiently **slow** to remain adiabatic, and there must be a *sufficiently large energy gap* between the ground state and excited states. The practical implementation of Quantum annealing is beyond the scope of this work [33].

This theorem is also important when explaining the inner workings of the QAOA algorithm, which will be crucial for the development of this work.

2.4.4 Error Mitigation Techniques

To ensure that all the previous properties are effectively managed, a variety of correction techniques are employed in quantum computing. These techniques are designed to enhance the overall performance of quantum systems by reducing or compensating for the errors that can accumulate during quantum operations. It is of extreme importance now that it is considered **the main problem of current quantum computing** [34].

Decoherence

Decoherence is a major challenge in maintaining the integrity of quantum information. It arises from the unwanted interaction of qubits with their environment, leading to a loss of quantum coherence. This phenomenon can be understood in circuit terms as introducing 'noise' into the circuit, which disrupts the intended operation of quantum gates and algorithms. This phenomenon limits the extent of our computations. High coherence is crucial for performing accurate quantum computations, as it ensures that superpositions and entanglements, fundamental for quantum operations, are maintained over time.

Techniques to extend coherence time include using materials with inherently low noise characteristics, implementing dynamic decoupling sequences to counteract decoherence effects, and operating quantum circuits at ultra-low temperatures to reduce thermal noise.

Dynamic decoupling involves applying a series of precisely timed pulses to the qubits during quantum operations. One of the main goals of these pulse sequences is refocusing the qubit state by effectively averaging out the effects of environmental noise over time. The idea is to apply control pulses that flip the qubits at intervals short enough to preclude the environment's influence from accumulating to a critical, coherence-destroying level. Another interpretation for these pulses can be thought of as creating a temporal filter that modulates the interaction between the qubit and its environment. By choosing appropriate timings and intervals, specific types of environmental noise can be selectively filtered out, similar to how noise-canceling headphones work.

Transpilation

As it has been mentioned, **gate fidelity** and **cross talk prevention** are factors to take into account. To prevent wrong results, our implementation needs to be aware of possible conflicts. That is where transpilation [35] comes into play, by substituting some gates with more simple, hardware-familiar ones, a careful physical layout of the qubits is applied. Applying delays to certain qubits and precise control of the timing and strength of interaction pulses has been proven to make quantum computing more **fault-tolerant** [36].

2.4.5 Logical Qubits

Logical qubits are formed by **entangling multiple physical qubits to create a redundant encoding of information**. This redundancy allows the system to detect and correct errors autonomously. For example, the surface code, one of the most promising QEC codes, uses a two-dimensional grid of qubits where error syndromes are detected by measuring the parity at the vertices of the grid. This approach allows for the correction of errors as they occur without needing to know the state of individual qubits, thereby preserving quantum information. **This seems to be one of the main approaches** looked forward to and is on IBM's roadmap [37].

2.4.6 Error Mitigation Algorithms

These algorithms are used to estimate and correct errors post-computation. Unlike QEC, which corrects errors during computation, software-level techniques adjust the outcomes based on statistical models of the errors. Techniques such as zero-noise extrapolation, probabilistic error cancellation, and virtual distillation are used to enhance the accuracy of quantum computations by extrapolating the error-free outcome from multiple noisy measurements.

2.5 Quantum Systems Representing Qubits

The physics behind each qubit implementation is well beyond the scope of this thesis, so an accurate but succinct explanation will be made. The main concern of this thesis is the **superconducting qubits**, and the other implementations will be briefly mentioned. The main comparison between implementations is **coherence time** and **error mitigation** techniques.

2.5.1 Superconducting Qubits

From when Kamerlingh Onnes observed how the conductivity of metals disappeared completely in small temperatures close to 0 degrees Kelvin, and the complete disappearance of resistance proven with persistent currents in superconducting rings, showing perfect conductivity [38]. Two types of superconducting materials, based on their reactivity to magnetic fields, can be found. Type I is formed by elemental superconductors that require temperatures close to 0 Kelvin for perfect conductivity to appear, whereas type II is made up of alloys that show these properties at higher temperatures (still below 0^{o}C).

Underlying Principles

There can exist bound electron pairs in a degenerate Fermi gas (such as those found in metals) despite the innate repulsion between them, named Cooper pairs [39]. When two superconductors are placed beside each other (insulated or not) and both are defined by wave functions differing on a phase ϕ, these pairs experiment and flow.

Since Kamerlingh Onnes first observed the phenomenon of superconductivity in 1911, where the electrical resistance of metals dropped abruptly to zero at temperatures close to absolute zero, the field of superconductivity has seen significant advancements. Onnes discovered that at these extremely low temperatures, metals not only lost their resistivity but also demonstrated the ability to maintain electrical currents indefinitely, evidenced by persistent currents in superconducting rings. This property, known as perfect conductivity, underscores the revolutionary nature of superconductors [38].

Superconducting materials are broadly classified into two types based on their response to applied magnetic fields. Type I superconductors, typically elemental metals, exhibit superconductivity at temperatures very close to 0 Kelvin and are characterized by a sharp transition from superconducting to a normal state upon reaching a critical magnetic field strength. This transition is due to their complete expulsion of magnetic fields (known as the Meissner effect) when in the superconducting state. In contrast, Type II superconductors, which are primarily made up of metallic alloys or complex oxides, operate at relatively higher temperatures (though still below 0°C). These materials allow magnetic fields to penetrate in quantized units of flux called vortices, between two critical field strengths, H_{c1} and H_{c2}, thereby entering a mixed or vortex state without losing their superconducting properties. This allows them to be applied to cases where a certain magnetic field could be present.

The concept of Cooper pairs is central to understanding the microscopic mechanism of superconductivity in Type I and Type II materials. These pairs consist of bound electron pairs within a degenerate Fermi gas, despite the natural repulsion between electrons. The pairing occurs due to an attractive interaction mediated by lattice vibrations or phonons,

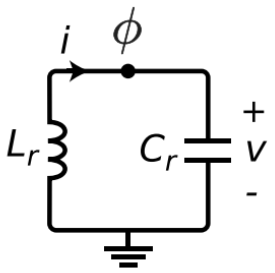


Figure 2.4: Classical LC Circuit in which the Josephson Junction is modeled, introducing a non-linear dependence on ϕ.

leading to the formation of a macroscopic quantum state that can carry electric current without resistance [39].

When two superconductors are juxtaposed, whether insulated from each other or not, and their macroscopic wave functions are characterized by a phase difference ϕ, an interesting phenomenon occurs. These Cooper pairs can tunnel between the superconductors through a Josephson junction, a process governed by the phase difference between their wave functions. This quantum tunneling leads to the Josephson effect, wherein a supercurrent (a current made entirely of Cooper pairs) flows between the superconductors without any applied voltage, depending solely on the phase difference ϕ [40].

Implementation

The implications are many, but among them, we find the unusual scenario where a microscopic effect can be seen in a macroscopic system. Suppose we relate the components of an LC circuit where to be related with kinetic energy (for the capacitor) and potential energy (for the inductor). This is a coherent approach because the solution for the time-independent Schrödinger Equation

$$|\psi(t)\rangle = e^{\frac{-i\hat{H}t}{\hbar}} |\psi(0)\rangle \tag{2.5.1}$$

Reminds us of the solution of a linear system defined by a circuit. It can be proven [41] that the energy of the circuit is

$$H = \frac{1}{2}CV^2 + \frac{1}{2}LI^2 = \frac{Q^2}{2C} + \frac{\Phi^2}{2L} \tag{2.5.2}$$

$$= 4\frac{e^2}{2C}(\frac{Q}{2e})^2 + \frac{1}{2}(\frac{h}{4\pi e})^2\frac{1}{L}\phi^2 \tag{2.5.3}$$

$$= 4E_C n^2 + \frac{1}{2}E_L\phi^2 \tag{2.5.4}$$

However, this creates a harmonic oscillator energy spectrum, with lots of unwanted, accessible, and equidistant excited states that handicap our computational basis as seen in Figure 2.5a. Adding a non-linearity dependent on the phase can be done using a Josephson Junction, which leads to

$$H = 4E_C n^2 + \frac{1}{2}E_J cos(\phi) \ , \tag{2.5.5}$$

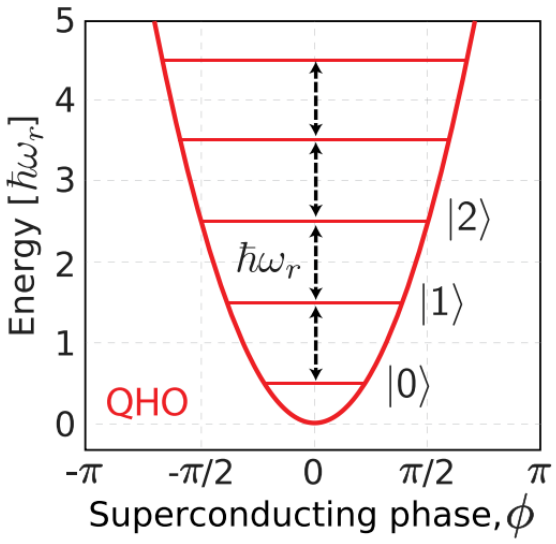
where the new E_C has been adjusted for the introduction of the junction. This leads to the energy distribution seen in Figure 2.5b, where the energy levels can be modified (by adjusting ω_{01} and ω_{12} and creating a computational basis.

In **transmon qubits**, type of superconducting qubits The ratio $\frac{E_J}{E_C}$ is typically much larger in transmon qubits than in other types of superconducting qubits. High $\frac{E_J}{E_C}$ ratios lead to energy levels that are not equidistant, reducing sensitivity to charge noise while maintaining sufficient anharmonicity to define qubit states.

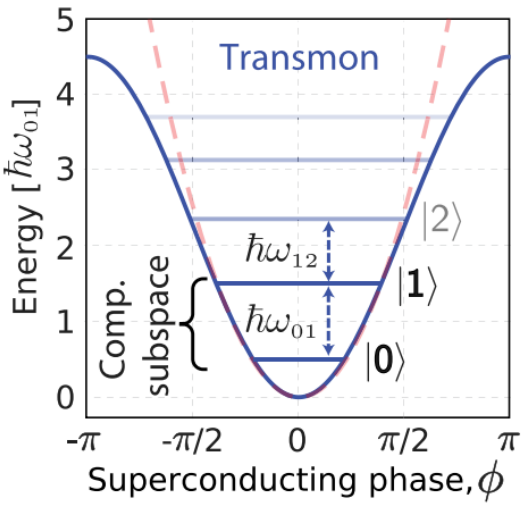
IBM

As of the current date [37], the biggest accomplishment from IBM has released the Heron, a modular processor of 133 qubits that can be coupled with others of its same kind. Said machine will be called IBM Quantum System Two and will debut in 2024. It is considered a breakthrough because it will be able to execute 5000 gates within coherence time.

Also, the release of IBM Quantum System Two [42] represents a significant advancement in quantum computing by introducing a modular architecture aimed at enhancing scalability. This new system is designed to connect multiple quantum processors into a unified



(a) Energy spectrum of the linear LC circuit with equidistant distances.



(b) Energy spectrum when Josephson Junction is applied.

Figure 2.5: Energy Spectrum comparison between different circuits. To the left, is an equidistant spectrum where the distances between each energy level cannot be independently modified. To the right, The Josephson Junction allows to separate energy states and create a computational basis. Extracted from [41].

platform through high-speed communication links, which allows for the aggregation of quantum computational power. By leveraging this modular approach, IBM aims to significantly expand computational capabilities and address the scaling challenges typically associated with increasing qubit counts.

The IBM Quantum System Two is not only about hardware advancements; it also integrates improvements in quantum software that target error correction and noise mitigation [42]. These enhancements are critical for the practical application of quantum computing, as they help to improve the reliability and performance of quantum operations, essential for executing complex algorithms and computations.

Moreover, IBM's roadmap includes plans to scale up their systems to more than 4,000 qubits by 2025 [42], further emphasizing their commitment to developing quantum-centric

supercomputing. This evolution suggests a future where quantum and classical systems are more seamlessly integrated, enhancing the overall capability and utility of quantum computing technologies. This approach is not proven to be the most effective, and regularly, papers are published attempting to IBMs developments [43].

D-Wave

D-Wave Systems specializes in quantum annealing, a quantum computing approach designed specifically for solving optimization problems. In their implementation, they also use superconducting qubits [44] proving that qubit architecture and quantum computer architecture are different aspects. Unlike general-purpose quantum computers, D-Wave's machines use a type of quantum processing that leverages quantum tunneling and energy landscape manipulation to find the minimum energy configuration of a problem, which corresponds to the optimal solution. This makes D-Wave's technology particularly suitable for applications such as logistics, finance, and materials science where optimization is key. The inner workings of its systems are unknown [45] [46], but the physics behind it is based on the QAT.

2.5.2 Other Implementations

Aside from superconducting qubits, several other technologies are being explored for implementing qubits:

- **Trapped Ion Qubits**: These use ions as qubits, trapping [47] them using electromagnetic fields and manipulating their quantum states with lasers.
- **Topological Qubits**: These are based on quasi-particles called anyons [48], whose world lines form braids in three-dimensional space, providing a way to encode information topologically protected from local perturbations.
- **Photonic Qubits**: Utilize the properties of photons to perform quantum computations [49], notably using properties such as polarization, frequency, phase, and orbital

angular momentum.

- **Quantum Dots**: Quantum dots function as artificial atoms, with discrete electron energy levels that can be finely tuned by adjusting the size and shape of the dot [50]. Quantum dot-based Kitaev chains propose a platform where Majorana modes can be simulated and observed without a true topological phase. The adaptability of quantum dots allows for precise control over the quantum state, facilitating the study of Majorana bound states through direct experimental setups [51].

Comparing all the previous technologies, we point that:

- **Coherence Time:** Superconducting qubits typically offer microseconds to milliseconds, while the latest ion trap technology extends coherence times significantly, reaching up to a minute [52].

- **Gate Fidelity:** Superconducting qubits have seen improvements in gate fidelity, but ion trap qubits generally provide higher fidelity due to their longer coherence times and more isolated quantum states.

- **Scalability:** Superconducting qubits are currently more scalable in terms of qubit count, as demonstrated by IBM's developments. However, ion trap qubits, with advancements in microfabrication and control technology, are rapidly closing the gap [53].

CHAPTER 3

VARIATIONAL QUANTUM ALGORITHMS

In this chapter, the practical applications of the theoretical principles previously discussed are explored. We examine some of the most well-known algorithmic approaches for problem-solving in quantum computing. The primary focus of this thesis is on a subgroup of quantum algorithms known as Variational Quantum Algorithms (VQAs), a subclass that leverages the power of quantum computers in conjunction with the well-established algorithms of classical computers. This focus is motivated by the current developmental stage of quantum computers, which necessitates the use of more robust and noise-tolerant algorithms. VQAs are particularly advantageous in this regard, as they reduce the number of quantum computations required, thereby increasing noise tolerance by maintaining more classical computations. Approaching problems with a combination of both quantum and classical methods has been shown to yield the best results, as suggested in numerous studies [54, 55, 56].

We address some fundamental algorithm theory in Section 3.1 to define the importance of efficient optimization algorithms in Section 3.2. Focusing on VQAs and their properties in Section 3.3 to fully define the algorithm at the center of this work, the QAOA in Section 3.4.

3.1 Algorithm Theory

To assess the efficiency of algorithms, we employ notations that describe their performance across different cases. Here are the most commonly used notations:

- **Big O Notation** (O): Describes the upper bound of the algorithm's runtime or space requirements, ensuring that the algorithm does not perform worse than this boundary

in the worst case.

- **Big Omega Notation** (Ω): Provides a lower bound, representing the best-case scenario for the algorithm's performance.
- **Big Theta Notation** (Θ): Defines both the upper and lower bounds, used when an algorithm's runtime is tightly bounded by the asymptotic function.
- **Little o Notation** (o): Similar to Big O but excludes the exact bound, indicating a non-inclusive upper bound.

These complexities help when comparing classical and quantum algorithms, particularly in highlighting the quantum speed-up achievable through algorithms that promise quantum supremacy.

3.1.1 Polynomial vs. Exponential Time Algorithms

It's crucial to understand the distinction between polynomial and exponential time algorithms. This distinction is the foundation of the computational complexity theory and highlights the potential impact of quantum computing.

Polynomial Time (P)

Algorithms that run in polynomial time, also called deterministic algorithms, have their running time growing at most as a polynomial function of the input size. Formally, an algorithm is a polynomial time if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, that is $O(n^k)$ (also seen as $O(poly(n))$ where the maximum degree would be the k-th degree)for some constant k. Problems solvable in polynomial time are considered tractable or efficient.

This category is broadened by adding asymptotical limits such as $O(1)$ (equivalent to $k = 0$ and $O(n^k log^m(n))$ as the logarithm function can be bounded by polynomials.

Nondeterministic Polynomial Time (NP)

Although there exists a whole time-hierarchy complexity theory (involving PSPACE, ...) we will base ourselves on the basic notation of P, NP, and NP-hard, NP-complete. The problems that belong to this category are those that, if a solution is given, can be verified in polynomial time, but we lack an algorithm that solves it in polynomial time. It follows that $P \subseteq NP$, and of the million dollar problems is if $P = NP$ [57]. By NP-hard problems we understand those that are the hardest to solve, and that all other NP-hard problems can be transformed into it (in a polynomial time conversion). And if a problem is NP and NP-hard means it is NP-complete. Generally, to solve NP-Hard problems we require exponential time algorithms, which have running times that grow exponentially with the input size, typically expressed as $O(2^n)$ where n is the input size. These algorithms become infeasible even for relatively small input sizes due to the dramatic increase in computations required.

Implications for Classical vs. Quantum Computing

In classical computing, many problems, especially those of high interest, belong to the NP-hard group, and currently have no known polynomial-time solutions and are only solvable in exponential time with the algorithms available today. This includes many cryptographic algorithms, optimization problems, and more.

Quantum computing introduces a paradigm shift here. Quantum algorithms, such as Shor's algorithm [6] for integer factorization, promise significant speed-ups over its best-known classical counterparts, converting an exponential difficulty problem into a polynomial one [58] [6].

Relevance

In considerably small cases, there is no visible improvement between a polynomial time algorithm with a very large constant multiplier and an exponential one. However, increases in problem size make evident the importance of asymptotic limits, such as seen in Figure 3.1.

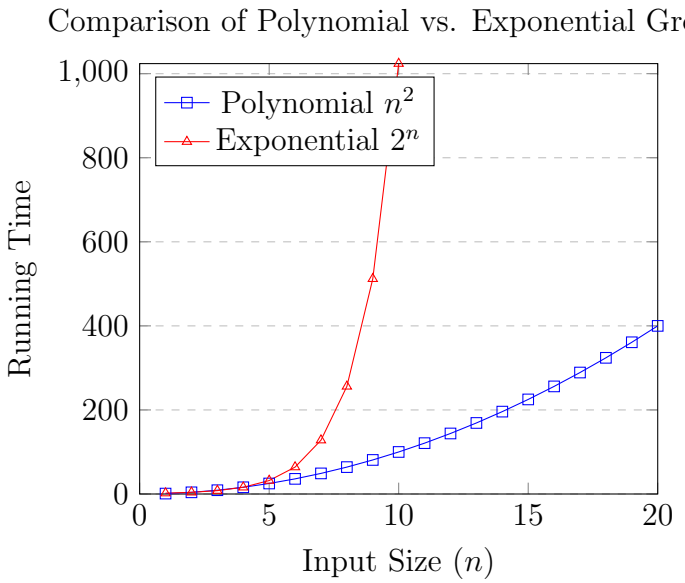


Figure 3.1: Comparison between a Polynomial time and an Exponential time algorithm. For sufficiently small cases, the coefficient of the polynomial determines which method is best. However, as the problem increases in size, the difference between both asymptotical limits becomes evident.

3.2 Optimization Problems

Optimization problems, specifically of **combinatorial** nature [59], can be characterized by the following:

- **Objective Function:** Is the mathematical expression that assigns a value to every possible solution and measures the goodness of one solution with respect to other ones. These functions can be maximized or minimized, and have **local maxima/minima**, values that are optimal in a small region, and **global maxima/minima**, the optimal value of the function over the valid solution space. Symmetries in said functions will also affect these regions [60].

- **Solution Space:** Is the space composed of all the possible solutions to the problem. All the combinatorial problems in this thesis consist of binary variables, therefore the solution space is conformed by all the possible combinations of 1s and 0s of each binary

variable. In that sense, the solution space is the set of all possible binary strings of length n, being n the number of variables.

- **Constraints:** Sometimes not all solutions are feasible, meaning that in the search space, there are combinations of 1s and 0s that go against the nature of the problem. For example, a problem where a node can have 2 possible colors, coded as $(1, 0)$ if a node has color A or $(0, 1)$ if a node has color B. Combinations $(0, 0)$ and $(1, 1)$ are technically possible, but would be invalid. Thus, a constraint is defined to avoid undesired solutions, defining then a **valid solution space** [61].

3.3 Quantum Variational Eigensolvers

This family of methods describes how problems are interpreted as operators and their interest is found in its eigenstates [62]. In our case, the problem is an energy state's definition using a Hamiltonian, and the energy of the ground state is to be found [63], with the smallest eigenvalue (energy). It can be seen as a **minimization** optimization problem where the objective function is the expectancy of the energy. To define a solution space, parameters $(\theta_1, ..., \theta_m)$ are defined and a minimization of the objective function $E(\theta)$ takes place. More formally

$$\langle\psi(\theta)|\psi(\theta)\rangle = 1 \ , \tag{3.3.1}$$

$$E(\theta) = \langle\psi(\theta)| \, H \, |\psi(\theta)\rangle \ \ . \tag{3.3.2}$$

Assuming a discrete spectrum for the energies, which will always be our case given the combinatorial nature of the selected problems, it can be proven that, for any $|\psi\rangle$, given the orthonormal basis of the operator $H \, |\phi_i\rangle = \lambda_i \, |\phi_i\rangle$ and a ground state such that $\lambda_0 \leq \lambda, \forall\lambda \in$ Spec(H)

$$E_\psi = \langle\psi|\, H\, |\psi\rangle \tag{3.3.3}$$

$$= \sum_{\lambda_1,\lambda_2 \in \mathrm{Spec(H)}} \langle\psi|\phi_{\lambda_1}\rangle \langle\phi_{\lambda_1}|\, H\, |\phi_{\lambda_2}\rangle \langle\phi_{\lambda_2}|\psi\rangle \tag{3.3.4}$$

$$= \sum_{\lambda \in \mathrm{Spec(H)}} \langle\psi|\phi_\lambda\rangle\, \lambda\, \langle\phi_\lambda|\psi\rangle \tag{3.3.5}$$

$$\geq \lambda_0 \sum_{\lambda \in \mathrm{Spec(H)}} |\, \langle\psi|\phi_\lambda\rangle\, |^2 = \lambda_0\ , \tag{3.3.6}$$

where we have defined an orthonormal basis as H is Hermitian, and $\langle\phi_i|\phi_j\rangle = \delta_{ij}$ and $\sum_{\lambda \in \mathrm{Spec(H)}} |\, \langle\psi|\phi_\lambda\rangle\, |^2$ are the components of $|\psi\rangle$, squared, in the orthonormal base, and as it is a normalized state, it yields 1.

Then we have found a lower boundary for the energy expectancy of the problem Hamiltonian, and $E(\theta) = E_0 \iff |\psi(\theta)\rangle = |\phi_0\rangle$. Then formalizing a problem as a Hamiltonian whose energy spectrum is the costs of different solutions (eigenstates) can be of interest. Here sets the foundation for optimization, establishing that a minimum exists and that finding the correct parameterization is required. As discussed in previous sections, factors such as noise and decoherence significantly impact our computations. Variational algorithms involve fewer quantum computations compared to fully quantum algorithms, making them relatively more noise-tolerant [54]. It is important to note that the solution space (ansatz space) defined by our parameterization is crucial, as it must enable us to explore the state of minimum energy. However, the general method does not provide specific guidelines on constructing the parameterization. This is where the primary method used in this thesis, the Quantum Approximate Optimization Algorithm (QAOA), becomes relevant, offering a structured approach to parameterization.

3.4 Quantum Approximate Optimization Algorithm

Regarding VQAs to solve combinatorial problems, QAOA has been proven to be the most effective among them [65]. The process of applying QAOA can be divided as follows:

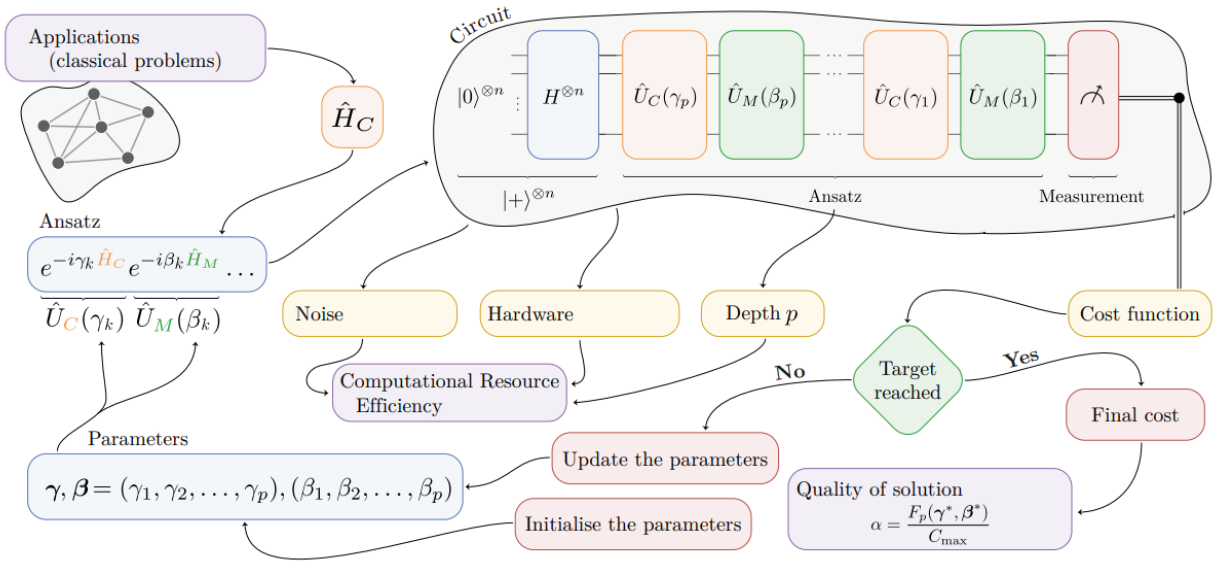


Figure 3.2: A comprehensive flowchart of the QAOA executing found in [64]. From the problem encoding of a graph to H_C, through the loop that updates the parameters optimizing them.

3.4.1 Problem Encoding

The initial step in QAOA involves encoding the combinatorial problem into a Hamiltonian H_p, whose ground state represents the optimal solution. The quality of this encoding directly impacts the algorithm's efficiency and effectiveness [66]. This Hamiltonian in binary variables represents the cost of each solution and orders them in terms of quality. The quality (cost) function is usually a quadratic function for applicability to multiple quantum implementations.

For example, proposing the cost function $f(x) = x_1 + 3x_2 + 2x_3$, where x_i are binary variables that represent if an element is in the bag (1) or not (0). It can be interpreted as presenting 3 items to be placed in a bag; each with a given value, 1, 2, and 3 respectively. This is the first part of problem encoding and it is translating the problem into a set of keys (solutions) and values (cost), and a (cost) function that gives value to each key. They must have a specific form so they can be translated into quantum problems. In this thesis, a focus on quadratic binary problem formulations is made, which can be translated into the

following forms depending on the specific solving method:

- **Quadratic Unconstrained Binary Optimization (QUBO):** This approach frames the problem in terms of binary variables, making it suitable for direct implementation into quantum hardware. This implementation will depend on the used backend. If the problem is constrained, penalization terms can be added to simulate their behavior, and then the problem is treated as unconstrained with terms weighted by some multipliers (Lagrangian Multipliers [67]). In the previous example, if only 2 objects can be placed in the bag, the constraint would be $x_1 + x_2 + x_3 = 1$, and the penalty term would be added as $\lambda(x_1 + x_2 + x_3 - 1)^2$ where λ is the Lagrange multiplier.

- **Ising Model:** Using the change of variables: $x_i = \frac{1+z_i}{2}$ we exchange values $x_i \in \{0, 1\}$ for $z_i \in \{-1, -1\}$ corresponding to the eigenvalues of the Pauli Z matrix. The reason behind this is, using the example, that now:

$$f(z) = 3 + \frac{z_1 + 3z_2 + 2z_3}{2} + \frac{\lambda}{4}(1 + z_1 + z_2 + z_3)^2 \tag{3.4.1}$$

$$H(\mathbf{Z}) = -\left(\lambda Z_1 \otimes Z_2 + \lambda Z_1 \otimes Z_3 + \lambda Z_2 \otimes Z_3\right) \tag{3.4.2}$$

$$-\left(\left(\frac{1}{2} + \lambda\right) Z_1 + \left(\frac{3}{2} + \lambda\right) Z_2 + (1 + \lambda)\, Z_3\right) \tag{3.4.3}$$

$$+ (3 + \frac{3\lambda}{4}) \tag{3.4.4}$$

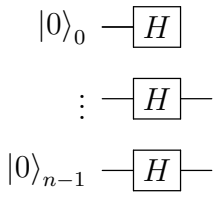
  Note that variables with possible values -1 and 1, when squared, they become constant values. The operators $Z_i \otimes Z_j$ are a simplified way to express $\mathbb{I}_1 \otimes ...\mathbb{I}_{i-1} \otimes Z_i \otimes Z_j \otimes \mathbb{I}_{j+1}...\mathbb{I}_n$. Now our problem has been translated to a quantum system with a defined Hamiltonian.

  This model, theoretically, allows for higher degrees than 2 as a generalization of the Ising form, and in QAOA, as each of these terms will correspond to Z gates on the involved $i, j, ...$ variables, it could be implemented. This is the field of HUBOs and is out of the scope of this thesis [68].

3.4.2 The algorithm

The Quantum Approximate Optimization Algorithm (QAOA)[12] is a hybrid method for solving combinatorial problems. It follows the same principle as the VQAs but the parametrization is restricted to the Hamiltonian of the problem and a mixer operator, allowing this block of operators to be repeated p times. Its principal concepts blocks are:

- **Initial State**: It can be modified and adjusted in different ways to allow a faster convergence. As a general convention and for simple cases, a state in complete superposition is provided such as

  The evolution of this state, applying unitary evolutions of Hamiltonians as circuits, is how we find the solution to the problem.

- **Variational Form**: VQAs allow for any parameterized circuit, but for the QAOA the Hamiltonian's evolution is applied. More specifically, we apply $e^{-i\gamma H_p}$ where H_p is the definition of the problem using Pauli gates. This structure is fully dependent on the problem and unique for each problem case.

  As any problem will be translated into the Ising Model in this work, the relevant terms of the objective function will be Z_i terms and $Z_i \otimes Z_j$, and their evolutions will be translated to RZ gates to the i-th qubit and RZZ gates applied to the i-th and j-th qubit, previously defined in Chapter 2.

- **Mixer Operator**: To offer a more general approach, a mixer Hamiltonian can be introduced, even parameterized, to the algorithm. This operator is said to be defined by a mixing Hamiltonian H_m and is applied as $e^{-i\beta H_m}$. In some variations of the algorithm, this mixer plays a determining part either by having a predefined form

given by a classical approximation [69], preserving constraints [70], or more complex forms [71].

With the building blocks have been defined, the algorithm is given by the following operator:

$$U(\beta, \gamma) = e^{-i\beta_p H_m} e^{-i\gamma_p H_p} \dots e^{-i\beta_p H_1} e^{-i\gamma_p H_1} \tag{3.4.5}$$

Using a total of $2p$ operators, where p is a hyperparameter, representing the **depth of the circuit**, chosen by the fitness of results and real-world NISQ-circuits depth limitations.

3.4.3 Motivation

The time evolution of a system is governed by the time evolution operator:

$$U(t) = e^{-iHt} \tag{3.4.6}$$

However, when the Hamiltonian is composed of a sum of multiple Hamiltonians that generally do not commute with each other, the Trotter-Suzuki formula provides a method to apply this sum in a manner that achieves the desired time evolution. This is accomplished using the following approach

$$U(T) = e^{-i(H_1+H_2)t} \tag{3.4.7}$$

$$\approxeq \prod_{p=1}^{p} (e^{-iH_1\Delta t} e^{-iH_2\Delta t}) \ . \tag{3.4.8}$$

Choosing an appropriate p so that $T = p\Delta t$ and p big enough so the transition is considered slow, we can justify

$$H(t) = \frac{T-t}{T} H_m + \frac{t}{T} H_p \ , \tag{3.4.9}$$

$$U(t) \approxeq \prod_{p=1}^{p} (e^{-i\frac{T-t}{T} H_m \Delta t} e^{-i\frac{t}{T} H_p \Delta t}) \tag{3.4.10}$$

$$= \prod_{p=1}^{p} (e^{-i\beta_p H_m} e^{-i\gamma_p H_p}) \ . \tag{3.4.11}$$

By this transformation, where the parameters are optimized classically, we can achieve a discretized version of the Quantum Adiabatic Theorem [32]. This is not a rigorous proof, but rather a motivation of this process.

3.4.4 Cost Function

The form of the cost function has a direct effect on the efficiency of the algorithm. Two equivalent Hamiltonians can be presented in different forms, for example, one simplified compared to the other one [72].

3.4.5 Optimizers

As the cost function is minimized classically, an optimizer is defined to compute the steps that the parameters will take [73]. The choice of this optimizer, depending on the cost function's form is a critical aspect of the algorithm, as some optimizers work well, for example, finding global minima when there are lots of local minima, and others don't.

3.4.6 Add-ons

Warm-start

Warm-start [74] is a method inspired by constraint relaxation of problems that allows for continuous variables to be defined in our binary problem that then will equate to probabilities of each qubit in the initial state. We now set a different initial state that a set of Hadamard gates applied to $|0\rangle_n$ as

$$|\phi^{(0)}\rangle = \bigotimes_{i=0}^{n-1} R_y(\theta_i)\,|0\rangle_n \ , \tag{3.4.12}$$

$$\theta_i = 2\arcsin\sqrt{\chi_i} \ , \tag{3.4.13}$$

where $R_y(\theta_i)$ is a rotation around the y-axis of the Bloch sphere by an angle θ_i. χ_i is the relaxed solution of the problem used as a warm start.

As the operator for $R_y(\theta_i)$ is

$$R_y(\theta_i) = \begin{pmatrix} \cos(\theta_i/2) & -\sin(\theta_i/2) \\ \sin(\theta_i/2) & \cos(\theta_i/2) \end{pmatrix}, \tag{3.4.14}$$

where if we now substitute our formulas for the θ_i we get

$$R_y(\theta_i) = \begin{pmatrix} \sqrt{1-\chi_i} & -\sqrt{\chi_i} \\ \sqrt{\chi_i} & \sqrt{1-\chi_i} \end{pmatrix}. \tag{3.4.15}$$

Thus the initial state after the transformation is

$$|\phi^{(0)}\rangle = \bigotimes_{i=0}^{n-1} \left(\sqrt{1-\chi_i}\,|0\rangle + \sqrt{\chi_i}\,|1\rangle\right) \ , \tag{3.4.16}$$

so the probability of obtaining a 1 in the i-th qubit is χ_i, relating in some way the solution of the relaxed problem with the initial state of the QAOA algorithm. When using this strategy, the initial state must be the ground state for the mixer Hamiltonian, as in the Adiabatic execution. Proving that the following Hamiltonian has as the ground state the initial state $|x\rangle_n = |\phi^{(0)}\rangle$

$$H_{M,i}^{(ws)} = \begin{pmatrix} 2\chi_i - 1 & -2\sqrt{\chi_i}\sqrt{1-\chi_i} \\ -2\sqrt{\chi_i}\sqrt{1-\chi_i} & 1-2\chi_i \end{pmatrix} \ . \tag{3.4.17}$$

Now we apply it to the initial state and it yields, for a single qubit:

$$H_{M,i}^{(ws)} |\phi^{(0)}\rangle = \begin{pmatrix} 2\chi_i - 1 & -2\sqrt{\chi_i}\sqrt{1-\chi_i} \\ -2\sqrt{\chi_i}\sqrt{1-\chi_i} & 1 - 2\chi_i \end{pmatrix} \begin{pmatrix} \sqrt{1-\chi_i} \\ \sqrt{\chi_i} \end{pmatrix} , \tag{3.4.18}$$

$$= \begin{pmatrix} 2\chi_i\sqrt{1-\chi_i} - \sqrt{1-\chi_i} - 2\chi_i\sqrt{1-\chi_i} \\ -2\sqrt{\chi_i} + 2\sqrt{\chi_i}\chi_i + \sqrt{\chi_i} - 2\sqrt{\chi_i}\chi_i \end{pmatrix} = - \begin{pmatrix} \sqrt{1-\chi_i} \\ \sqrt{\chi_i} \end{pmatrix} . \tag{3.4.19}$$

This state, for each qubit, represents an eigenstate of the mixer Hamiltonian with an eigenvalue of -1. Given that the determinant of a matrix is the product of its eigenvalues, we can deduce that the determinant of this matrix is 1, indicating that the other eigenstate has an eigenvalue of 1. Thus, our initial state corresponds to the ground state of the mixer Hamiltonian. It can be rewritten in its diagonal form as

$$H_{M,i}^{(ws)} = \begin{pmatrix} \sqrt{1-\chi_i} & -\sqrt{\chi_i} \\ \sqrt{\chi_i} & \sqrt{1-\chi_i} \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{1-\chi_i} & \sqrt{\chi_i} \\ -\sqrt{\chi_i} & \sqrt{1-\chi_i} \end{pmatrix} , \tag{3.4.20}$$

and expressed as an application $e^{-i\beta H_{M,i}^{(ws)}}$ yields

$$e^{-i\beta H_{M,i}^{(ws)}} = \begin{pmatrix} \sqrt{1-\chi_i} & -\sqrt{\chi_i} \\ \sqrt{\chi_i} & \sqrt{1-\chi_i} \end{pmatrix} \begin{pmatrix} e^{i\beta} & 0 \\ 0 & e^{-i\beta} \end{pmatrix} \begin{pmatrix} \sqrt{1-\chi_i} & \sqrt{\chi_i} \\ -\sqrt{\chi_i} & \sqrt{1-\chi_i} \end{pmatrix} \tag{3.4.21}$$

$$= R_y(\theta_i) \begin{pmatrix} e^{i\beta} & 0 \\ 0 & e^{-i\beta} \end{pmatrix} R_y(-\theta_i) , \tag{3.4.22}$$

where substituting

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} , \tag{3.4.23}$$

yields the final expression for the mixer Hamiltonian

$$H_{M,i}^{(ws)} = R_y(\theta_i) R_z(-2\beta) R_y(-\theta_i) . \tag{3.4.24}$$

An important requirement for this method is that the cost and mixer Hamiltonians must not commute, meaning that both have the same basis of eigenstates. If that is to happen, the algorithm tends to get stuck into less optimal solutions.

Parametrization of Gammas and Betas

A parallel approach to improve the algorithm is to propose a parametrization of the gammas and betas aiming to improve the QAOA ansatz generated. The optimized parameters are then "u"s and "v"s which define a possible subset of the possible values and, as seen in [72], can improve results.

We introduce the Fourier parametrization method described in [72]. This approach aims to improve the parameterization by using a Fourier series representation for the parameters. The cost function for the QAOA, which estimates the energy of a given solution, is defined as

$$E(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) | \hat{H} | \psi(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle \ , \tag{3.4.25}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are the parameters of the ansatz, and $\hat{H}$ is the Hamiltonian of the system. The Fourier parametrization is introduced to redefine these parameters using a Fourier series as follows

$$\gamma_i = \sum_{k=1}^{q} u_k \sin\left(\left(k - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right), \quad i = 1, 2, \ldots, p \ , \tag{3.4.26}$$

$$\beta_i = \sum_{k=1}^{q} v_k \cos\left(\left(k - \frac{1}{2}\right)\left(i - \frac{1}{2}\right)\frac{\pi}{p}\right), \quad i = 1, 2, \ldots, p \ , \tag{3.4.27}$$

where u_k and v_k are the Fourier coefficients, q is the number of these coefficients, and p is the number of repetitions (depth) of the QAOA circuit.

With this extra step of computing the value that we desire to optimize we leverage the periodic nature of the trigonometric functions to provide a more compact and potentially more expressive parameter space for the QAOA, which can enhance the algorithm's performance in finding optimal solutions. It is also seen in [72] that as p increases, the amplitudes

of the Fourier series diminish, justifying the possible use of even fewer parameters (lower frequencies) with the same accuracy.

Multiangle Approach

The Multiangle variation assigns individual parameters to each term of the cost and mixer Hamiltonians, unlike the standard QAOA where single parameters are applied to the entire cost and mixer Hamiltonians. The cost function for the QAOA using the Multiangle parametrization is defined as

$$E(\beta_{ij}, \gamma_{ij}) = \langle \psi(\beta_{ij}, \gamma_{ij}) | \hat{H} | \psi(\beta_{ij}, \gamma_{ij}) \rangle, \tag{3.4.28}$$

where $\{\beta_{ij}\}$ and $\{\gamma_{ij}\}$ are sets of parameters applied to each term in the mixer and cost Hamiltonians, respectively. The Hamiltonian $\hat{H}$ is typically expressed as a sum of Pauli operators. Now we consider the cost Hamiltonian $\hat{H}_C$ and the mixer Hamiltonian $\hat{H}_M$ decomposed into their constituent terms

$$\hat{H}_C = \sum_j C_j \hat{P}_j \ , \tag{3.4.29}$$

$$\hat{H}_M = \sum_k M_k \hat{Q}_k \ , \tag{3.4.30}$$

where $\hat{P}_j$ are, in our case, Z or ZZ operators, and $\hat{Q}_k$ are Pauli operators by default chosen as X operators. C_j and M_k are complex coefficients. Applying Equation (3.4.30) yields

$$U(\gamma_{ij}, \beta_{ij}) = \prod_{l=1}^{p} \left(\prod_j e^{-i\gamma_{lj} C_j \hat{P}_j} \prod_k e^{-i\beta_{lk} M_k \hat{Q}_k} \right) \ , \tag{3.4.31}$$

where p is the number of QAOA layers (repetitions), and γ_{lj} and β_{lk} are the parameters applied to each term $\hat{P}_j$ and $\hat{Q}_k$ in the l-th layer.

This approach provides more fine-grained control over the parameterization of the QAOA, promising to lead to better optimization results by exploiting the structure of the Hamiltonians more effectively. Although it presents itself as a more robust general method, the greater amount of parameters emphasizes the importance of a fit optimization algorithm and introduces more numerical errors to take into account regarding precision.

CHAPTER 4

CASE STUDY

Our study examines an analogous problem to the one seen in power distribution networks [22] applied to the Scandinavian energy grid. This problem is a variation of the k-Max-Cut problem, a well-known NP-hard problem, with some additional cost and constraint terms.

This chapter begins by explaining how a problem is encoded in a framework using a simpler example in Section 4.1 and formalizing our problem definition in Section 4.2, justifying as well its form in Section 4.3. Next, we apply our problem definition to a real-case scenario in Section 4.4 and present the results of executions for 8 nodes in Section 4.5, 11 nodes in Section 4.6, and real execution data from a quantum computer in Section 4.7. Finally, we conclude this work in Section 4.8.

4.1 Problem Encoding

To illustrate the problem encoding used in this research, we present the weighted Max-Cut problem as a foundation for understanding how to approach more complex problems. The formal definition of the problem is the following:

- We have an n-node undirected graph $G = (V, E)$ where $|V| = n$.
- Each edge has a weight $\omega_{ij} > 0$, and $\omega_{ij} = \omega_{ji}$ for all $(i, j) \in E$.
- Each node has a weight $v_i > 0$ for each $i \in V$.
- A cut is defined as partitioning the original graph into two subsets:
  - $G_0 = (V_0, E_0)$ where $E_0 = \{(i, j) \,|\, i, j \in V_0\}$

– G_1 (similarly defined)

- This partitioning results in some edges not belonging to either subset E_0 or E_1. As illustrated in Figure 4.1, V_0 could represent the black vertices, V_1 the white vertices, and E_0 would consist of only one black edge, with no edges in G_1. A partitioning can also be represented by a cut that crosses all edges with nodes on different subsets.

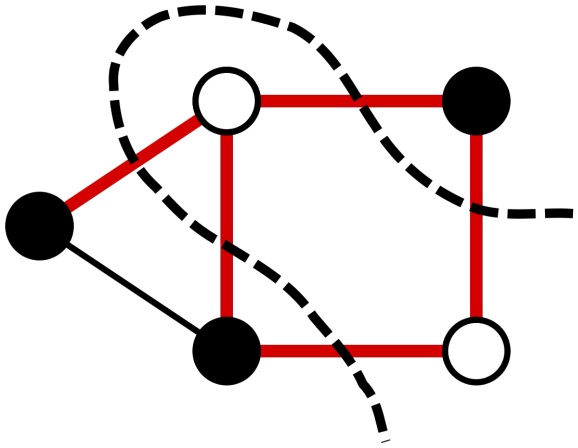


Figure 4.1: A simple graphical example of a cut that partitions a graph in two. The edges crossed by the dashed line are those whose weights will be counted in the cost function.

In optimization problems, a cost function is defined to evaluate the quality of candidate solutions. In the Max-Cut problem, the cost function is the sum of all weights of the edges crossed by a cut, which are the edges not belonging to E_0 nor E_1. It can be understood as if the crossed edges represent the influence ω_{ij} of customer j if customer i receives a free sample, quantifying how much j will purchase if recommended by i, who received the sample for free (with i and j being interchangeable in undirected graphs). The individual weights v_i represent the multiplying factor for the customer who is given a free sample ($x_i = 1$) and the probability he decides to buy again.

To translate this problem into a binary representation suitable for quantum computation, we define x_i for each node i where $x_i = 0$ if the node is assigned to G_0, and $x_i = 1$ if assigned to G_1. Then the cost becomes

$$C(x) = \sum_{i,j}^{n-1} \omega_{ij} x_i (1 - x_j) + \sum_{i}^{n-1} v_i x_i \ . \tag{4.1.1}$$

Imposing $i < j$ in the summation ensures that each edge is considered only once, avoiding redundant calculations while not affecting the symmetrical nature of solutions (e.g., "0101" being equivalent to "1010"). Nevertheless, it is not technically needed.

Although Equation (4.1.1) is intended to be maximized, firstly must be translated into the quantum realm. Consider a solution to the problem is $|010...101\rangle$, a quantum state defined by a string of 1s and 0s of length n. Applying the operator

$$\frac{1 - Z_i}{2} = 0.5 \begin{pmatrix} 1-1 & 0 \\ 0 & 1-(-1) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \ , \tag{4.1.2}$$

where Z_i is defined in Chapter 2. This operator has eigenvalues 0 and 1 for the eigenstates $|0\rangle$ and $|1\rangle$ respectively. Then the cost function becomes

$$C(Z) = \sum_{ij} \frac{\omega_{ij}}{4}(1 - Z_i)(1 + Z_j) + \sum_{i} \frac{v_i}{2}(1 - Z_i) \tag{4.1.3}$$

$$= \sum_{ij} \frac{\omega_{ij}}{4}(Z_j - Z_i - Z_i Z_j) + \sum_{i} -\frac{v_i}{2} Z_i + \sum_{ij} \frac{\omega_{ij}}{4} + \sum_{i} \frac{v_i}{2} \tag{4.1.4}$$

$$= -\frac{1}{2}(\sum_{i<j} \omega_{ij} Z_i Z_j + \sum_{i} v_i Z_i) + K \ . \tag{4.1.5}$$

Applying that $\sum_{ij} \frac{\omega_{ij}}{4}(Z_j - Z_i) = 0$, as terms cancel each other when the pairs (i, j) are permuted to (j, i), and $\sum_{ij} \frac{\omega_{ij}}{4}(-Z_i Z_j) = \sum_{i<j} \frac{\omega_{ij}}{2}(-Z_i Z_j)$ to collapse permutations. The constant K is irrelevant and can be excluded in the optimization and the expression in Equation (4.1.5) can be seen as an Ising Hamiltonian [75]

$$H = \sum_{i<j} \omega_{ij} Z_i Z_j + \sum_{i} v_i Z_i \ , \tag{4.1.6}$$

whose ground state, the eigenstate with minimal energy, is the solution to the problem.

For every pair of edges x_i, x_j, there is a $Z_i \otimes Z_j$ term in the Hamiltonian of the problem. It is important to note that in Equation (4.1.6), the summation terms have been simplified for readability purposes. Any Z_i term is an operator $\mathbb{I} \otimes ... \otimes Z_i \otimes ... \otimes \mathbb{I}$, as the tensor product of operators from different qubits generates operators that act on a larger system. A linear combination of operators in the previous format defines a sparse Pauli Operator.

The expression

$$H = \frac{1}{2}\left(\sum_{i<j} \omega_{ij} Z_i Z_j + \sum_i v_i Z_i\right) - \frac{1}{2}\left(\sum_{i<j} \omega_{ij} + \sum_i v_i\right) \tag{4.1.7}$$

allows us to easily compute the real cost if the problem at stake requires it and the offset must be considered.

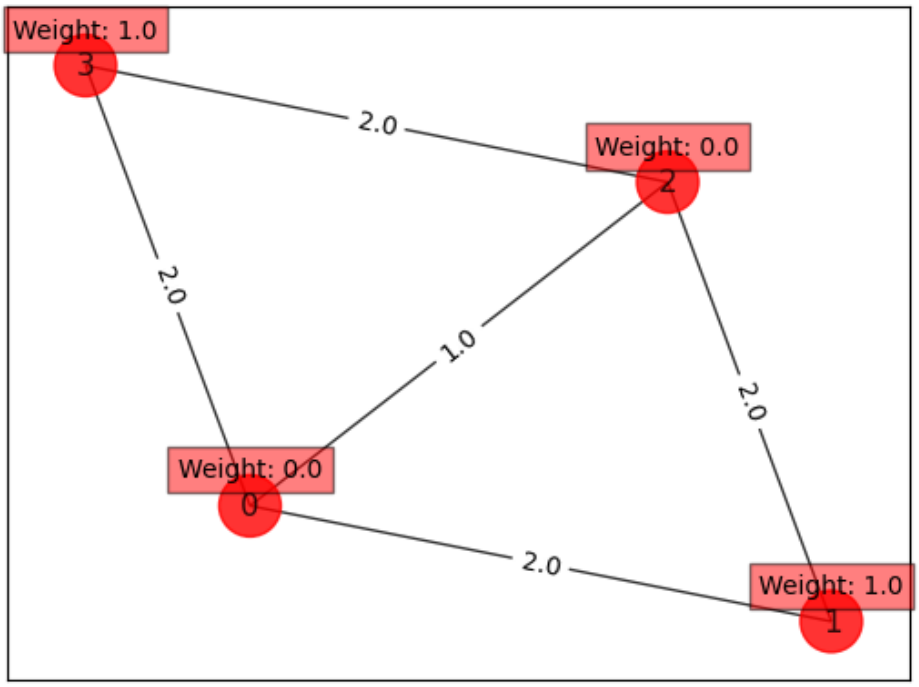


Figure 4.2: Simple example case of a 4-node undirected graph with weighted nodes and edges. As the problem is simple enough, the best solution can be found by computing all possible scenarios.

Following the example in Figure 4.2, the Hamiltonian is

$$H = \frac{1}{2}\Big((2.0)IIZZ + (1.0)IZIZ + (2.0)ZIIZ + (2.0)IZZI + (2.0)ZZII\Big) \quad \left(\sum_{i<j} \omega_{ij} Z_i Z_j\right) \tag{4.1.8}$$

$$+ \frac{1}{2}\Big((0.5)IIZI + (0.5)ZIII\Big) \qquad \left(\sum_{i} v_i Z_i\right) \tag{4.1.9}$$

$$- \frac{1}{2}\Big(-2.0 - 1.0 - 2.0 - 2.0 - 2.0\Big) \qquad \left(\sum_{i<j} \omega_{ij}\right) \tag{4.1.10}$$

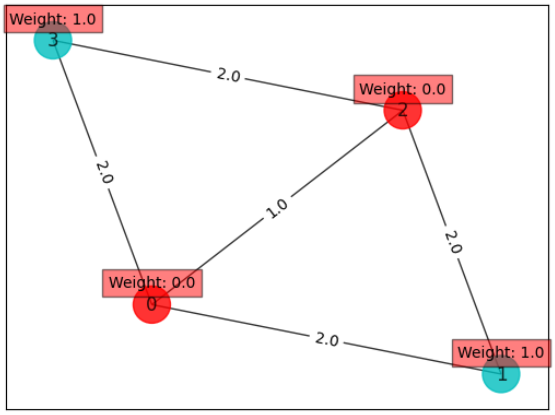
$$- \frac{1}{2}\Big(-1.0 - 1.0\Big) \qquad \left(\sum_{i} v_i\right) . \tag{4.1.11}$$

This problem can be brute-forced by trying all the possible combinations of 1 and 0 in a bit string of length n, so $n = 4$ yields $2^n = 16$ different solutions as shown in Figure 4.3a

```
(0 => 0000) case = [0, 0, 0, 0] cost = 0.0
(1 => 0001) case = [1, 0, 0, 0] cost = 5.0
(2 => 0010) case = [0, 1, 0, 0] cost = 5.0
(3 => 0011) case = [1, 1, 0, 0] cost = 6.0
(4 => 0100) case = [0, 0, 1, 0] cost = 5.0
(5 => 0101) case = [1, 0, 1, 0] cost = 8.0
(6 => 0110) case = [0, 1, 1, 0] cost = 6.0
(7 => 0111) case = [1, 1, 1, 0] cost = 5.0
(8 => 1000) case = [0, 0, 0, 1] cost = 5.0
(9 => 1001) case = [1, 0, 0, 1] cost = 6.0
(10 => 1010) case = [0, 1, 0, 1] cost = 10.0
(11 => 1011) case = [1, 1, 0, 1] cost = 7.0
(12 => 1100) case = [0, 0, 1, 1] cost = 6.0
(13 => 1101) case = [1, 0, 1, 1] cost = 5.0
(14 => 1110) case = [0, 1, 1, 1] cost = 7.0
(15 => 1111) case = [1, 1, 1, 1] cost = 2.0

Best solution = [0, 1, 0, 1] cost = 10.0
```

(a) Example of brute-forcing the solution. For each possible case, equivalent to a binary string, we can compute the partitioning of nodes and its total cost. Taking the minimum of such expression yields the result of the Max-Cut problem.



(b) Visual representation of the best solution seen in Figure 4.3a, where nodes forwhich $x_i = 1$ are colored in cian.

Figure 4.3: A visual representation of the brute-force method and how solutions are represented in the underlying problem graph.

4.2 Formal Definition

Sharing similarities with the previous problem, this work proposes a more general formulation to efficiently partition a graph based on conditions such as balancing the number of nodes, leveling surpluses of nodes per partition, and rewarding the presence of edges inside a partition as existing power line connections. Considering an n-node undirected graph $G = (V, E)$ where $|V| = n$, node weights w_i for $i \in V$ indicate the surplus of a certain geographic location. These nodes are carefully chosen to be representative aggregations of real, smaller regions. Partitions in the set of nodes are defined by introducing V_p for $p = 1, ..., P$ with P being the total number of partitions and each V_p a subset of V.

Cost Terms

The cost function to be optimized includes the following terms:

1. Partitions must be balanced in the number of nodes, as real-world applications experience that a solution with unbalanced partitions requires the addition of extra power lines at a high expense. This concept is accounted for by defining

$$C_{balanced}(\mathbf{x}) = \sum_{p}^{P} \left(\sum_{n \in V} x_{i,p} \right)^2 , \tag{4.2.1}$$

   where $x_{i,p}$ is a binary variable that is 1 if node i is in partition p and 0 otherwise. And V_p is the set of nodes in partition p.

2. Using existing edges to communicate nodes in the same partition reduces the cost of a solution. This concept can be expressed as

$$C_{discount}(\mathbf{x}) = \sum_{p}^{P} \left(|E| - \sum_{(i,j) \in E} x_{i,p} x_{j,p} \right) , \tag{4.2.2}$$

   where $|E|$ is the total number of edges in the graph. This term is definite positive, as the number of edges in the graph will always be greater than the number of edges in the partitions.

Constraint Terms

To restrict the number of solutions to feasible ones, the following constraints must be imposed:

1. Imposing a node can only be part of a single partition is achieved by the expression

$$\sum_{p}^{P} x_{i,p} = 1 \qquad\qquad n = 1, ..., N \ . \tag{4.2.3}$$

2. The mean surplus of a partition must be below a threshold value, referred to as the "self-sufficiency" clause, indicating that the surplus in the area need not be higher to achieve self-sufficiency. This is expressed as

$$\sum_{n=1}^{N} w_n x_{i,p} - k \sum_{n=1}^{N} x_{i,p} \leq 0 \qquad\qquad p = 1, ..., P \ , \tag{4.2.4}$$

where k is chosen as the average surplus among all nodes [22]. It serves as a balancing constraint for the partitions and the surplus of the nodes. In our case, we do not have energy sharing (flow function inside a partition $F_{pq}(x)$), which would modify this term as seen in [22].

4.3 Analysis of the Objective Function

The previous terms have been carefully tailored to generate optimal solutions according to the conditions each imposes. Seeking completeness, they will be properly justified.

Justification of Cost 1

The term Equation (4.2.1) can be interpreted as $f(x) = \sum p^P x_p^2$, where x_p represents the number of nodes in partition p. We perform a minimization on this function with the constraint (given by the problem itself) that the total number of nodes is constant so that

$$g(x) = \sum_{p}^{P} x_p - N = 0 \ . \tag{4.3.1}$$

Using the Lagrange Multiplier method, the minimum of the function $f(x)$ conditioned by $g(x)$ can be found by computing

$$L(x, \lambda) = f(x) - \lambda g(x) \ , \tag{4.3.2}$$

where λ is the Lagrange Multiplier, and derivating with respect to x_p and λ, setting them both to zero:

$$\frac{\partial L}{\partial x_p} = 2x_p - \lambda = 0 \tag{4.3.3}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{p}^{P} x_p - N = 0 \tag{4.3.4}$$

Solving the first equation for λ and substituting it into the second equation

$$x_p = \frac{N}{P} \qquad\qquad p = 1, ..., P \ , \tag{4.3.5}$$

yields the minimum proven unique by the Hessian matrix being constant and positive definite.

$$H = \begin{bmatrix} 2 & 0 & \cdots & 0 \\ 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \end{bmatrix} \ . \tag{4.3.6}$$

This means the function $f(x)$ is minimized when the partitions are balanced and can be seen using a geometric representation Figure 4.4.

Justification of Cost 2

If Equation (4.2.2) is not present, partitions are done arbitrarily without considering the edges already present in the graph. This function is analogous to the Max-Cut Hamiltonian's problem in Equation (4.1.1) when all edge weights are equal to 1 and node weights are excluded.

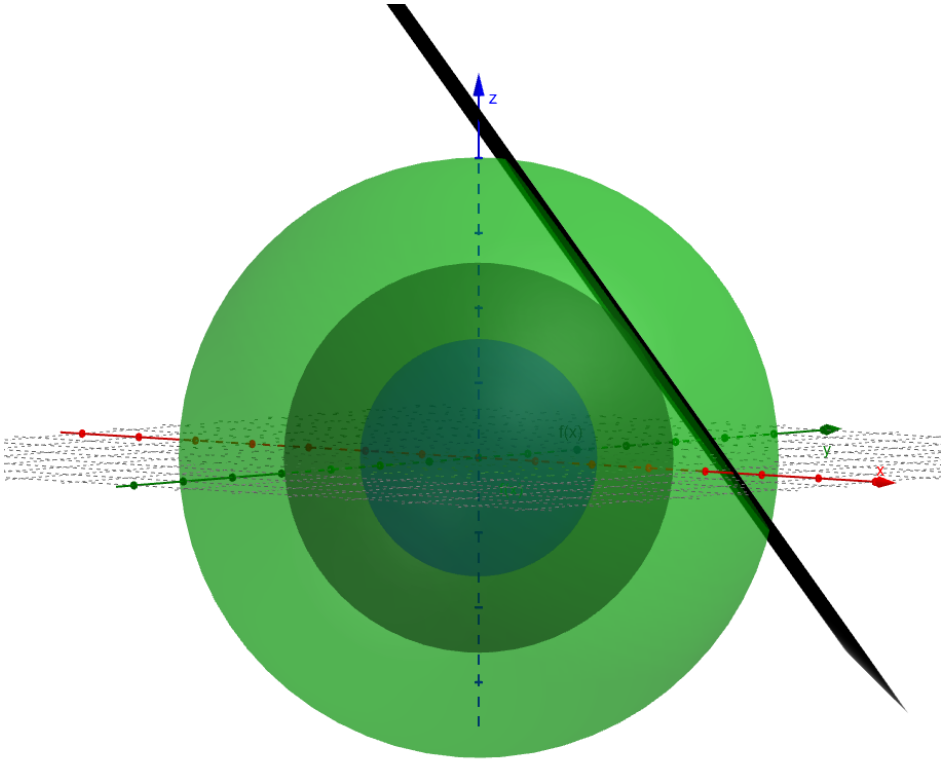


Figure 4.4: Justification of Cost 1: Each concentric sphere is the K surface level of $f(x) = x^2 + y^2 + z^2$ for a given K. Then $g(x) = x + y + z - N$, being N the number of nodes, defines the (hyper)plane constraining the solution. The intersection of both figures that generates the minimal value of $f(x)$ is when $g(x)$ is the plane tangent to the point $(\frac{x}{3}, \frac{y}{3}, \frac{z}{3})$. This plot has been generated with the tool in [76].

Justification of Constraint 1

The translation of the constraint Equation (4.2.3), also known as the One-Hot Encoding constraint, is done using the penalty method so that

$$Penalty = \sum_{p}^{P} \left(\sum_{n \in V} x_{i,p} - 1 \right)^2 . \tag{4.3.7}$$

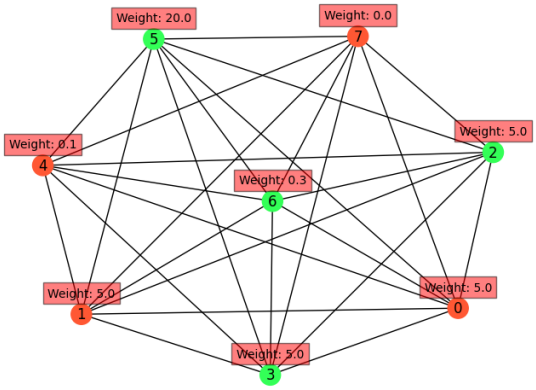
Any result that does not have a unique $x_{i,p} = 1$ for a given node i will result in a positively valued penalty, increasing the energy of that unfeasible eigenstate. For instance, encoding two partitions ($P = 2$), "10" and "01" are valid, and "00" or "11" are penalized. This constraint is imposed to prevent a node from being part of more than one partition, ensuring the same node is not counted multiple times in the cost function.

Justification of Constraint 2

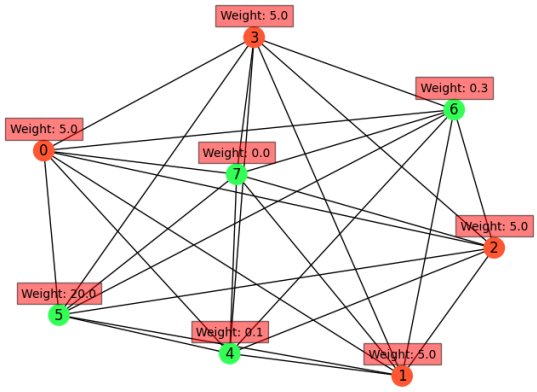
The formula for this constraint differs from the one proposed in [22], as the implementation requires a considerable amount of extra qubits. Alternatively, we propose

$$P_{\text{surplus}} = \sum_{p}^{P} \left(\sum_{i=1}^{n} x_{i,p} \left(\text{weights}_i - k \right) \right)^2 \tag{4.3.8}$$

It has the same structure as the previous penalty and penalizes solutions with partitions whose balances are out of the range $(0, k)$, meaning partitions that are not self-sufficient are penalized, as well as partitions exceeding k, which defines the threshold of self-sufficiency. By convention [22], k is chosen as the mean of all the node weights. Applying this penalty allows for an optimal solution such as Figure 4.5b instead of Figure 4.5a, and avoids the degeneracy of energy levels, as it prevents the existence of multiple optimal solutions that are permutations of each other. It is a direct effect of cost in Equation (4.2.4) being trivial in fully connected graphs, and multiple distinct partitions balanced in the number of nodes can be made.



(a) Unbalanced Surplus: **One of the optimal states** when no surplus constraint is applied. Causes a degeneracy that allows, in fully connected graphs, many solutions with the same optimal value.



(b) Balanced Surplus: **The optimal state** ensures a balanced, self-sufficient energy surplus when the surplus constraint is applied.

Figure 4.5: Comparison on the impact of Constraint 2: For a fully connected 8-node graph, to the left, the constraint is not applied, and the sum of weights in the red partition is 10.1, meanwhile in the green partition is 30.3. On the other hand, when the constraint is applied, we obtain a balance of 20.0 and 20.4, significantly closer to a surplus balance between partitions.

4.4 Real-Case Scenario

Once the objective function has been defined, we present a case study where it can be applied by following a parallel approach to the one considered in [22]. The Scandinavian power distribution network is aggregated in an 8-node graph representing the different regions of Scandinavia, using uniform randomly generated node weights to create randomized cases such as in Figure 4.6.

In Chapter 3, the Quantum Approximate Optimization Algorithm (QAOA) was defined along with some of its principal "add-ons." These will be combined into 8 different proposed solving methods for later comparison. Defining the four following procedures:

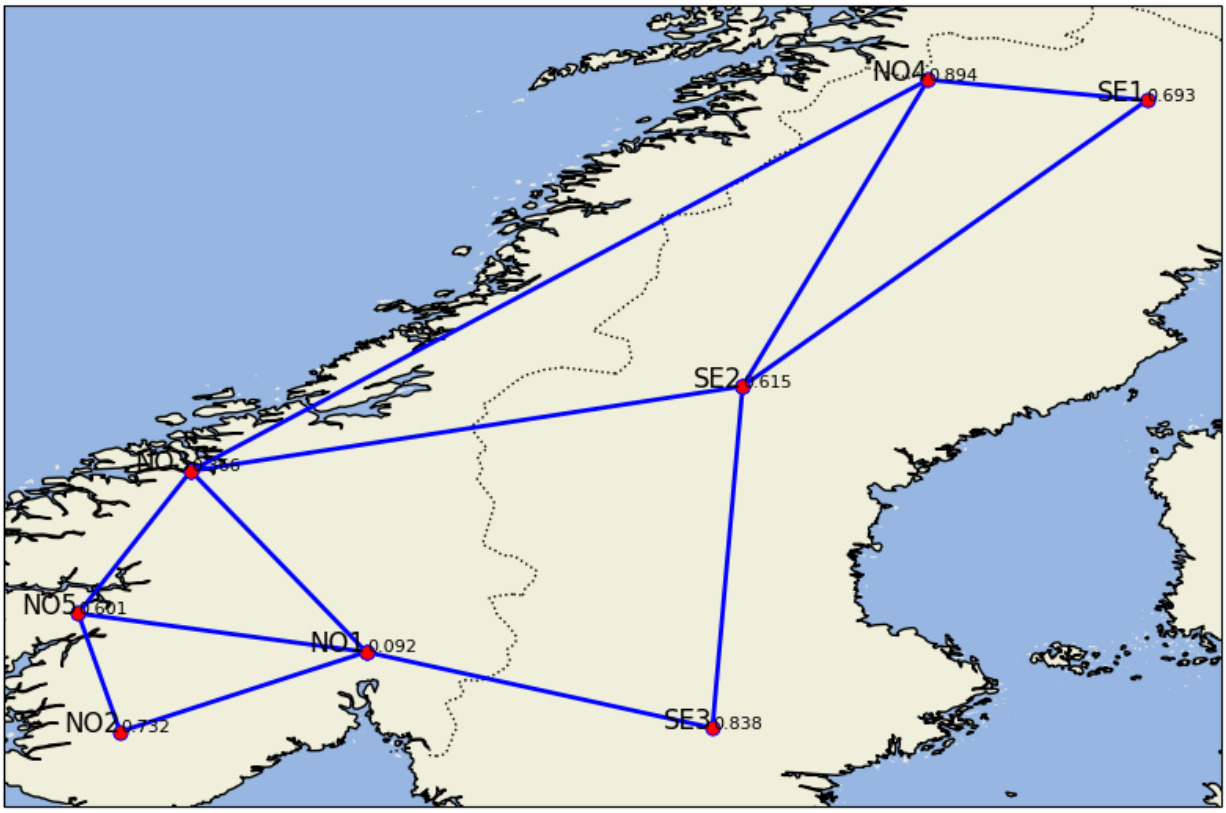


Figure 4.6: Example of the Network for $n = 8$. In this case, only points from Norway and Sweden are used from geographic locations that are representative of the whole network.

- **Straight Minimization**: Given a circuit of max depth p, the minimization is performed using the maximum amount of parameters possible, without minimizing at each p increment.

- **Straight Minimization Multiangle**: Similar to the previous method, but allowing for different parameters for each term of the cost Hamiltonian.

- **Optimize**: Instead of directly approaching the entire circuit for a given p, intermediate p circuits are also computed to find optimal solutions, and these results are used in the next iteration of minimization $p \to p + 1$.

- **Optimize Fourier**: Builds on the previous approach by adding Fourier parametrization for γ_i and β_i.

Each will be applied with and without the warm-start technique. The distribution from which initial random parameters are extracted plays a crucial role in the performance of optimization algorithms as initial parameters are starting points in the algorithm's search for optimal solutions. The choice of distribution can affect the convergence speed, the likelihood of finding global versus local optima, and the overall efficiency of the optimization process. Research has shown that specific distributions, such as *Xavier* or *Glorot* initialization [77], *He* initialization [78], and others tailored for specific problems, can significantly improve the performance of optimization algorithms [79, 80]. Thus, we will work with three different distributions that have been proven to work well:

- **Uniform Distribution**: Parameters are initialized with values drawn from a uniform distribution within the range $(0, 2\pi)$. This approach ensures that all initial values are equally likely, promoting a broad exploration of the solution space. It is a good baseline from which to start.

- **Perturbed Distribution**: Involves starting with a baseline value drawn from a uniform distribution and adding small random perturbations, drawn from a normal distribution. This approach is useful when there is prior knowledge about the baseline parameters close to optimal values.

- **Gaussian Distribution**: Parameters are drawn from a Gaussian distribution with a mean equal to π and a standard deviation of 0.1. This method tends to place more initial values near the mean, which can be advantageous for problems where central values are more likely to be near the optimal solution.

Regarding the choice of the optimizer, COBYLA [81] has shown the fastest convergence time. Other optimizers such as BFGS do not converge in some cases and will not be considered. Moreover, COBYLA is an optimizer efficient in searching global minima when multiple local minima are present. For further reference, all code definitions can be found in [82] and the cluster specifications in [83].

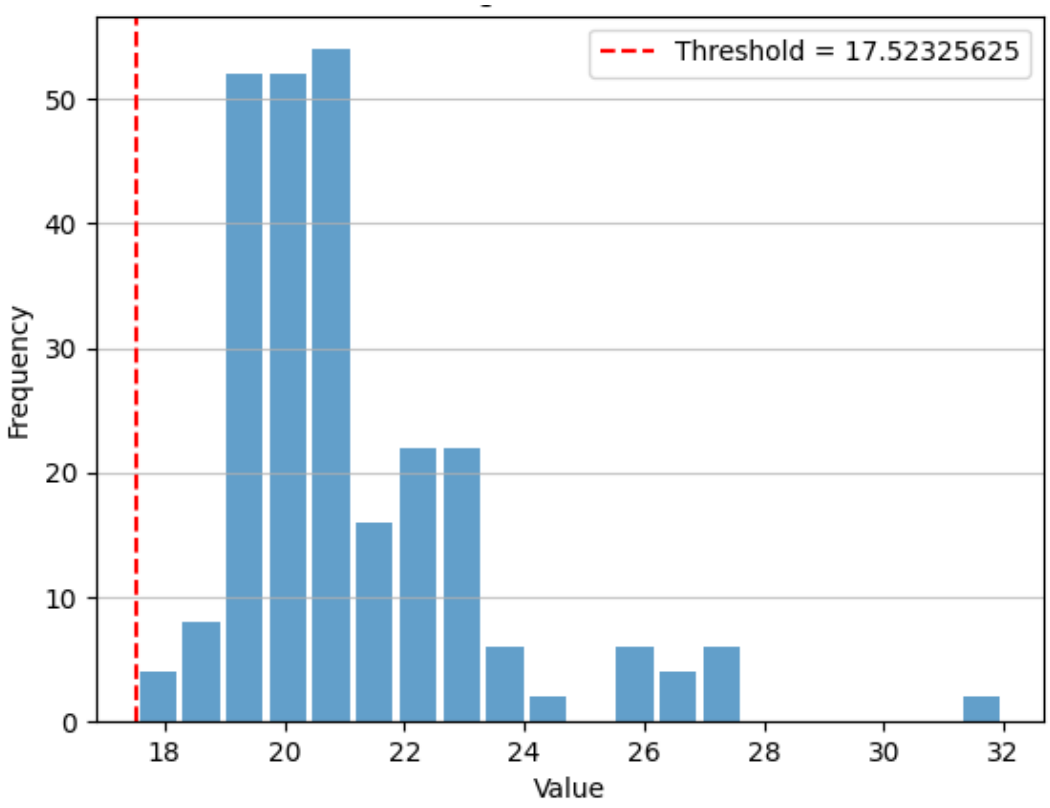


Figure 4.7: Brute Force Histogram: For each energy range, the amount of states in the range is shown. Each bin represents a range of close energy values, and, in red, the lowest energy value found for the instance in Figure 4.6.

4.5 8 Node Execution

Using the same example as in Figure 4.6, we can brute-force the solution and plot the histogram of the energy value for each solution to obtain Figure 4.7.

Executing the instanced problem 100 times for each of the 24 configurations (4 methods, with or without warm-start, and initial parameters strategy) and graphically representing the energy distributions, we study for a specific case, how our methodologies perform. This data contains information on which method is more likely to converge into the correct solution. As both "optimize" methods go through intermediate steps, these results are also stored for comparison.

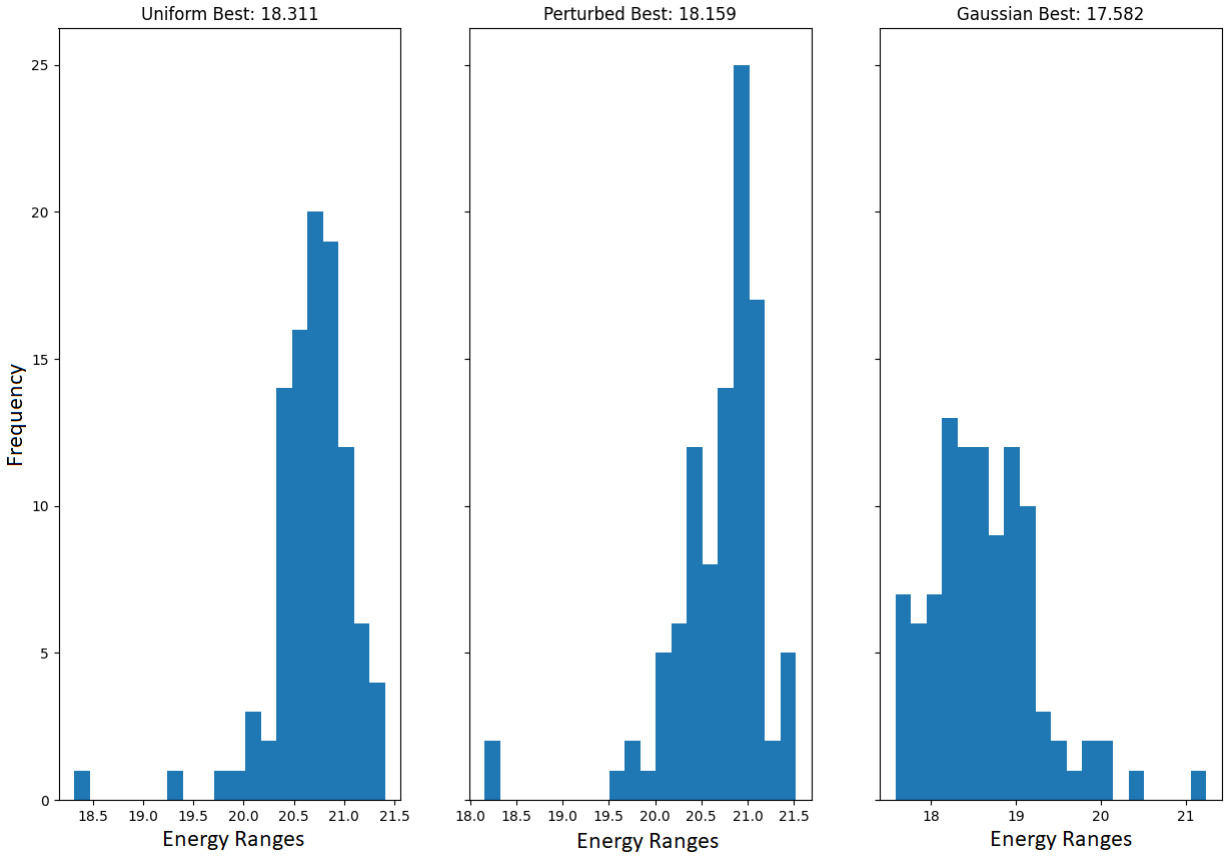


Figure 4.8: Straight Minimization method, for $p = 15$: The Gaussian achieves the best convergence to the optimal solution between all the initial point strategies, and its distribution is leaning more towards the optimal value than the other strategies, proving to be more robust.

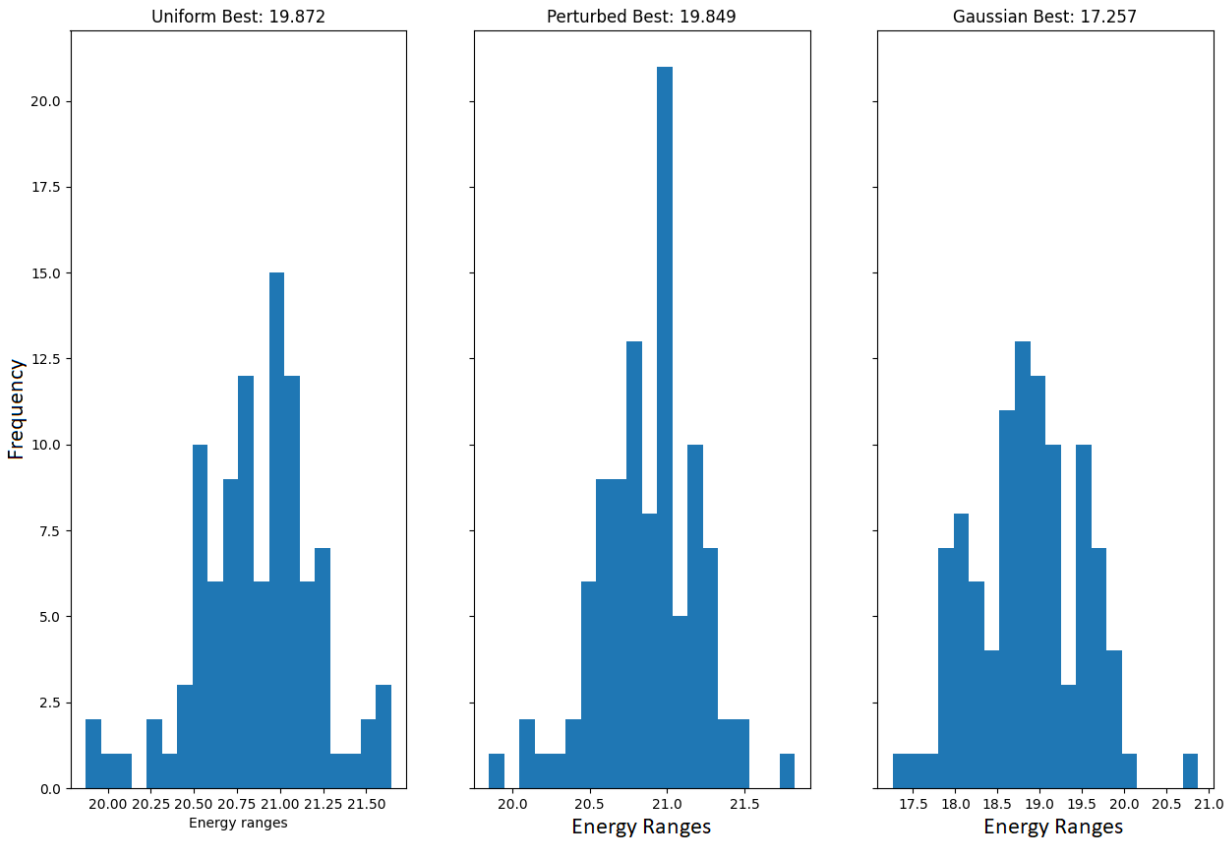


Figure 4.9: Straight Minimization method, warm-started, for $p = 15$: The Gaussian achieves the best convergence to the optimal solution between all the initial point strategies, surpassing the classical best solution in a 1%, which can be attributed to numerical errors. The other distributions are far from the optimal solution and converge to other local minima. Comparing it to the histogram in Figure 4.7, the perturbed and uniform strategies do not yield near-optimal results, as there is a large number of solutions in the range they converge.

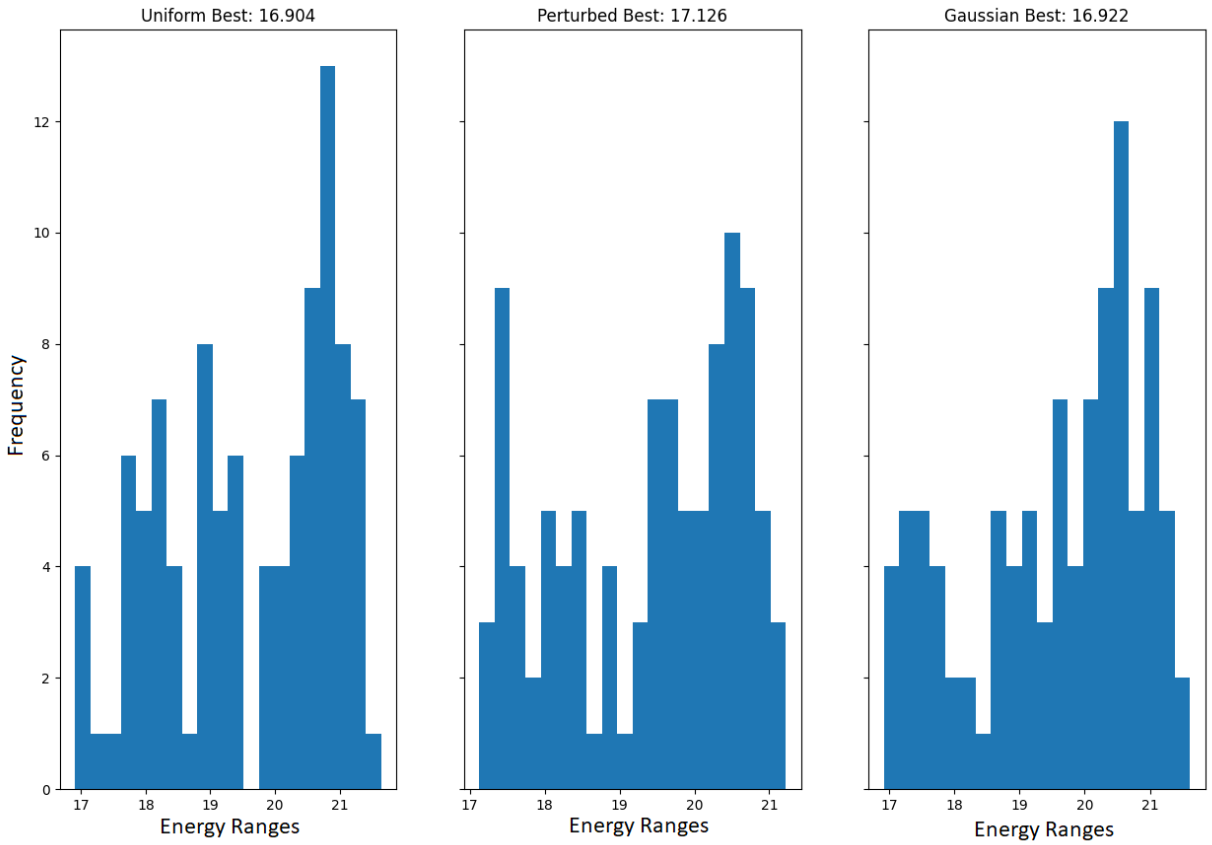


Figure 4.10: Optimize method, for $p = 15$, filtered for only results with $p = 15$: The three strategies seem to behave quite similarly, but all of them show results better than the optimal solution, which is not possible in the range of feasibility.

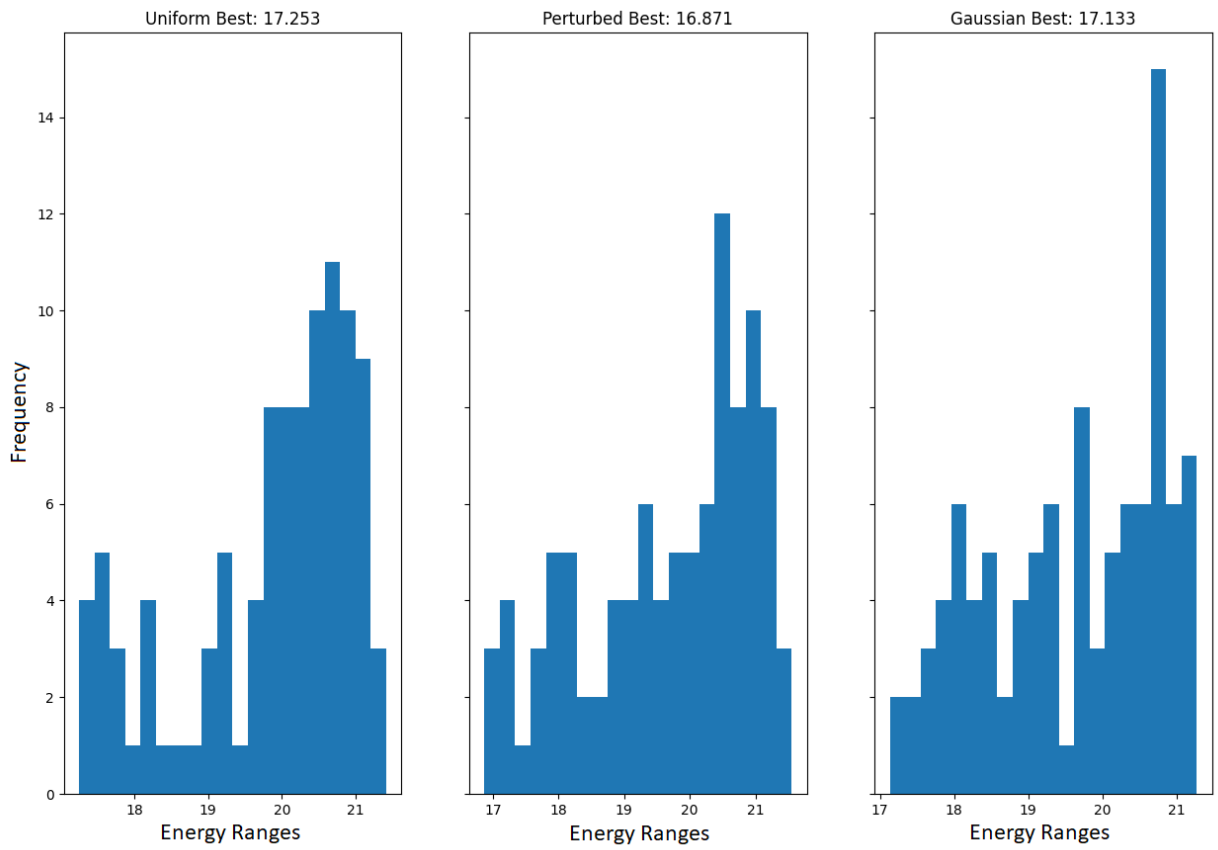


Figure 4.11: Optimize method, warm-started, for $p = 15$, filtered for only results with $p = 15$: The three strategies seem to behave quite similarly, but all of them show results better than the optimal solution, which is not possible in the range of feasibility.

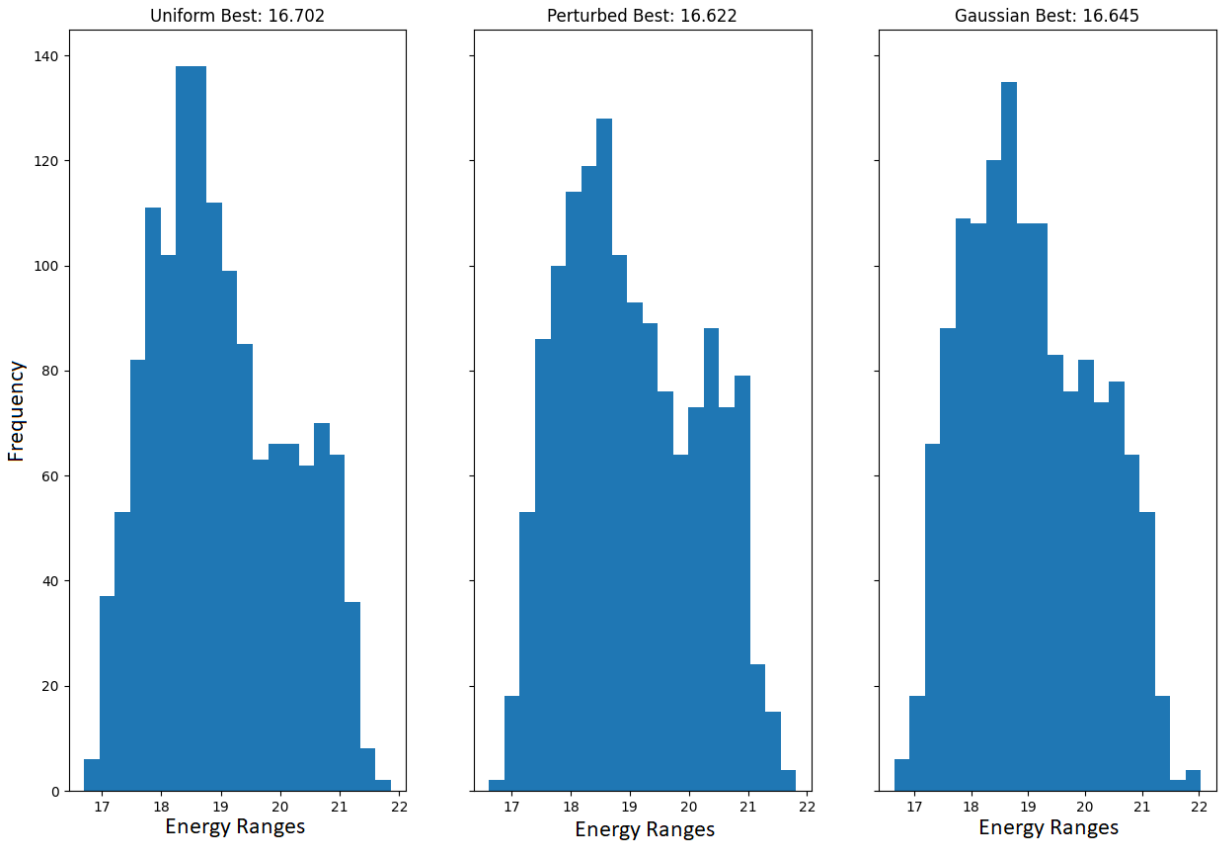


Figure 4.12: Optimize method, for $p = 15$, allowing intermediate results: When intermediate results are allowed, the distribution achieves values closer to the classical optimal solution, showing that the number of parameters could be shortened yielding similar results. All strategies have values surpassing the classically obtained minima.

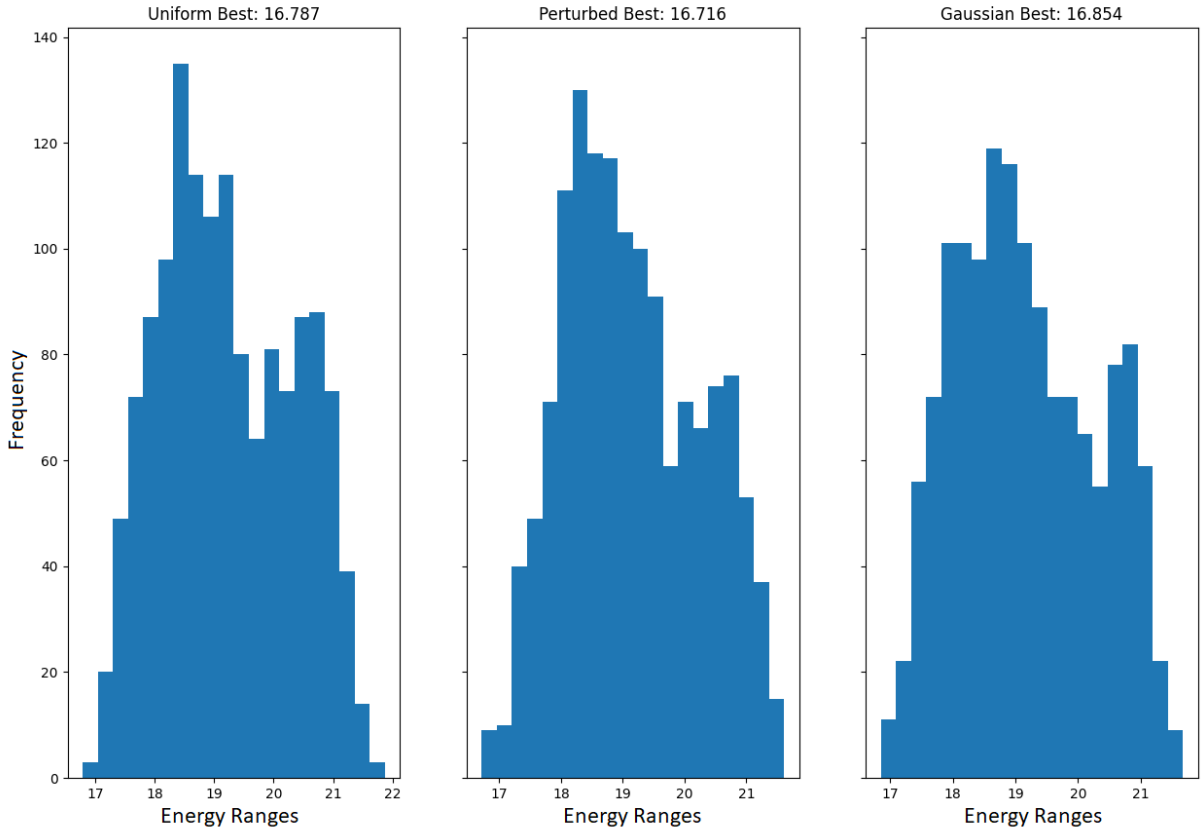


Figure 4.13: Optimize method, warm-started, for $p = 15$, allowing intermediate results: The solution does not differ much from its counterpart in Figure 4.12, with the same concerns about numerical errors.

A preliminary result comparison between warm-started and non-warm-started methods shows the Straight Minimization case does not improve significantly with the warm-start add-on. Between Figure 4.8 and Figure 4.9, there is a decrease in the number of solutions close to the global minima. The only strategy that does not seem to be hindered by the add-on is the Gaussian, which approximates nearer to the optimal solution but loses robustness. The results for the Straight Minimization Multiangle share the same behavior and have been omitted for brevity.

Regarding the Optimize Methods, no improvement is perceived between Figure 4.10 and Figure 4.11. Allowing possible optimal results for $p < 15$ leads to no improvement in the

warm-start method, as seen between Figure 4.12 and Figure 4.13.

In the Optimize methods, some results surpass the classically obtained best result. This is due to the specific problem, with the parameters proposed, having a relatively high variance in the energy estimation, which increases with the addition of the warm-started methods as they modify the mixer operator and transform it into a more complex operator to evaluate. We can conclude that some problems, such as the one presented now, can be ill-conditioned to solve with our methodology, and it is a significant factor when evaluating a certain graph.

A comparison of all these results can be presented to find which method approaches better values for the energy.

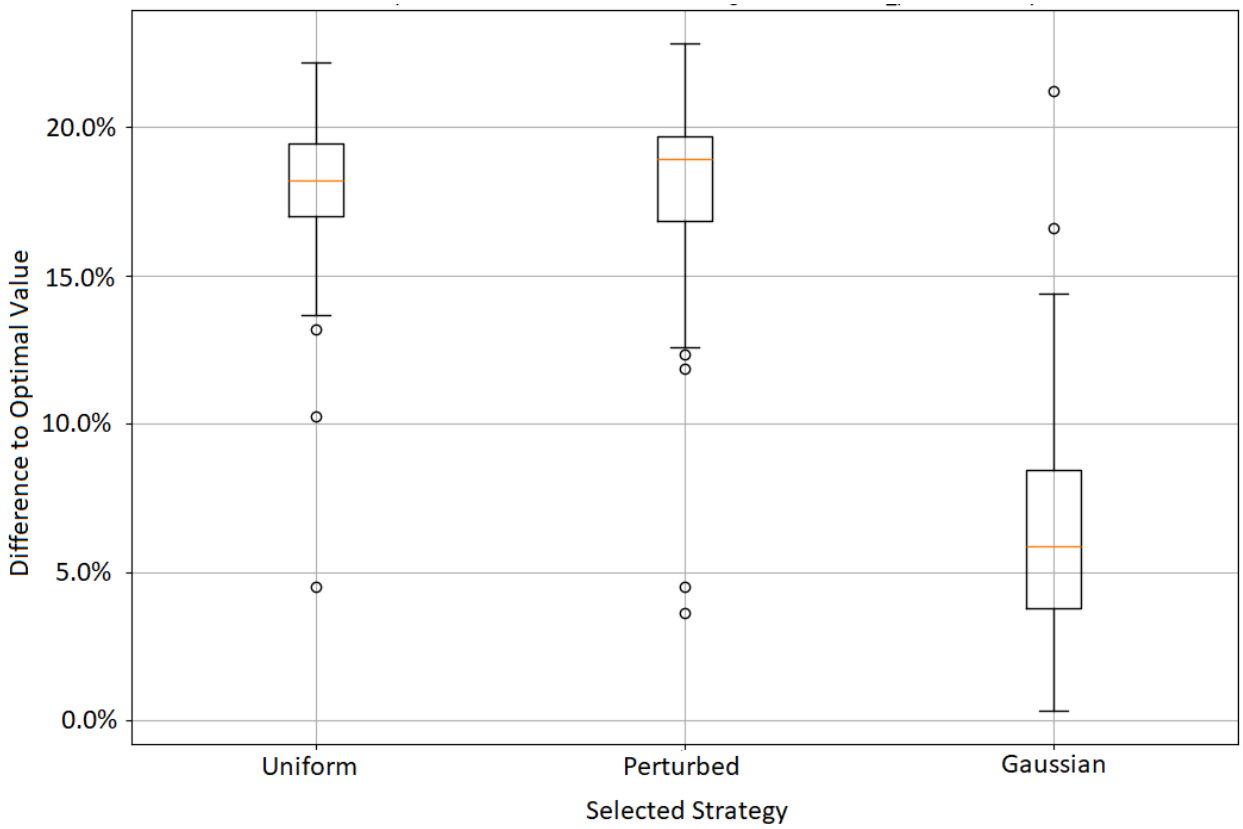


Figure 4.14: Boxplot for Straight Minimization, $p = 15$: The Gaussian strategy is closer to the classical optimal result, and it is proven that the whole distribution in Figure 4.8 is shifted towards better values.

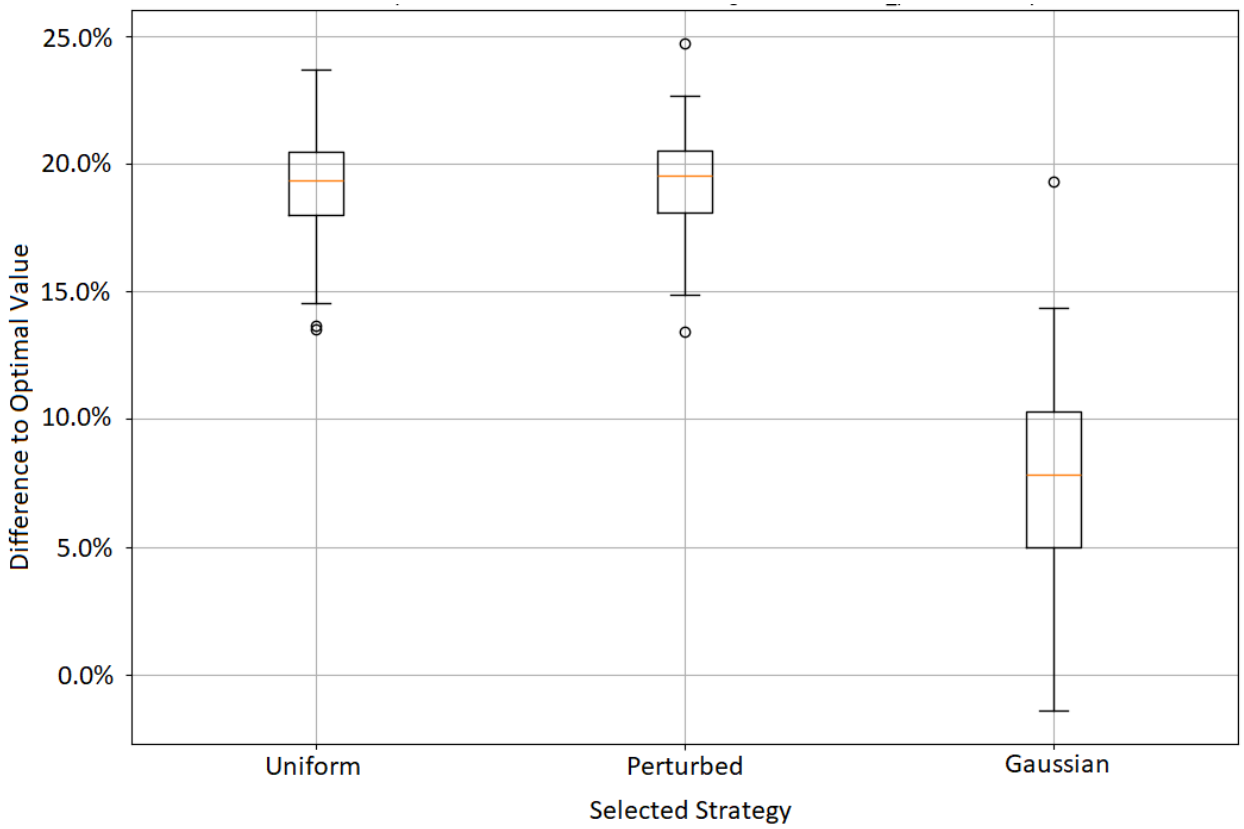


Figure 4.15: Boxplot for Straight Minimization, warm-started, $p = 15$: The Gaussian strategy is closer to the classical optimal result, but some numerical errors start to appear when our results are lower than the optimal solution, as shown in Figure 4.9.

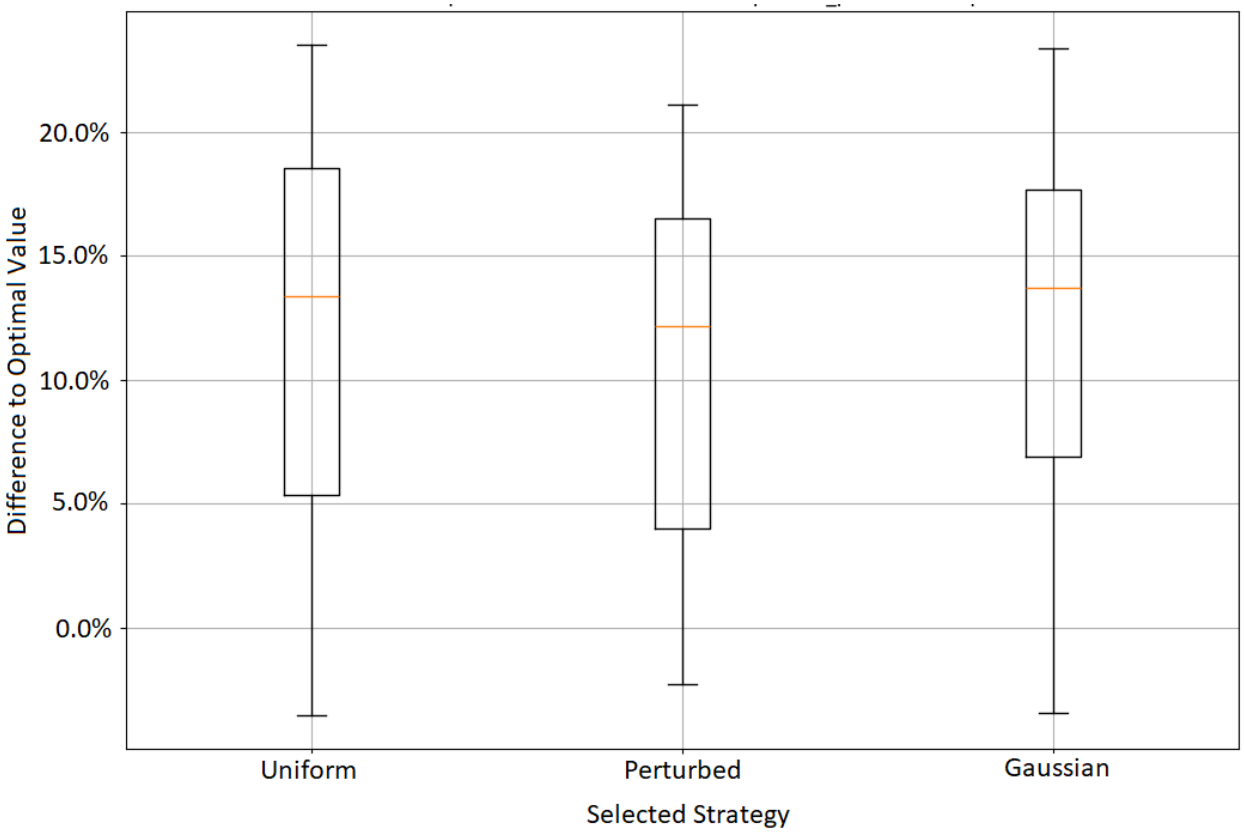


Figure 4.16: Boxplot for Optimize, $p = 15$: This method introduces a higher variance in the results, as the distribution loses quality and the mean of the results is close to 15% away from the optimal.

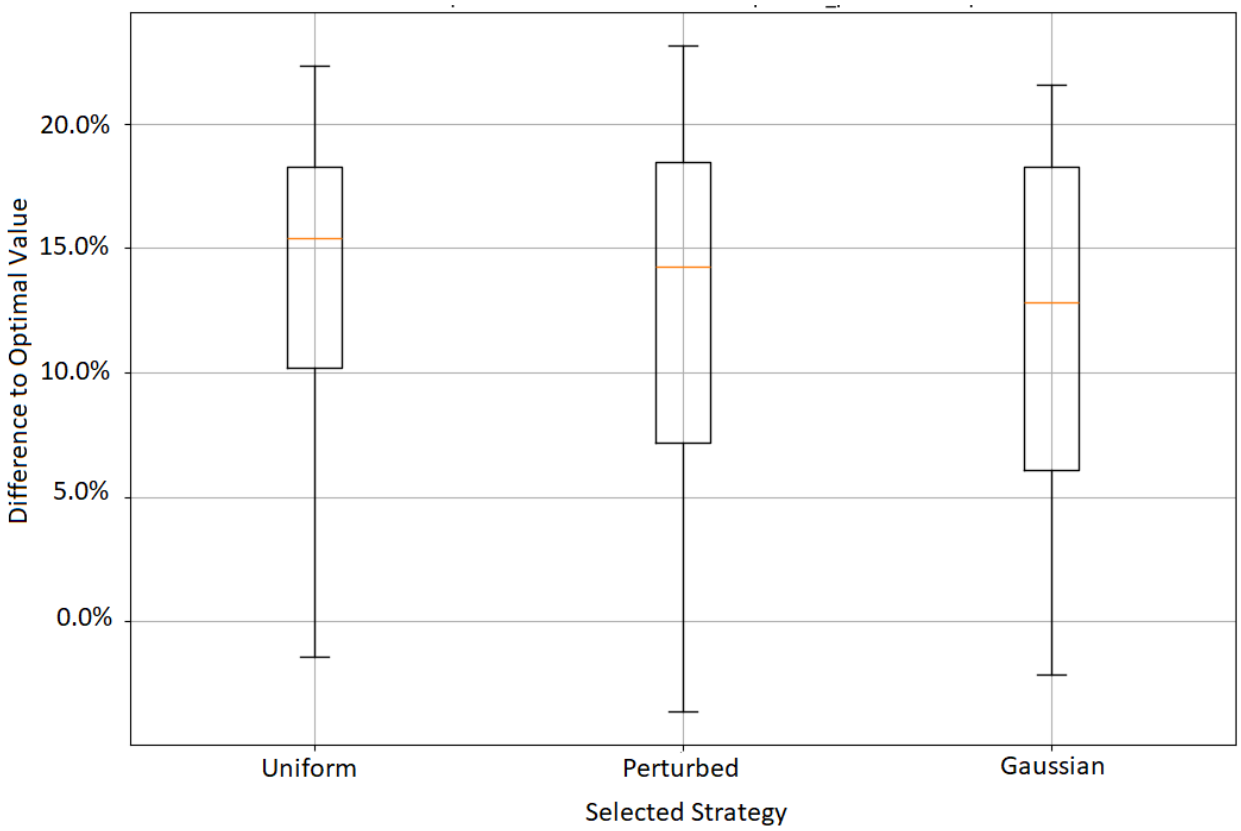


Figure 4.17: Boxplot for Optimize, warm-started, $p = 15$: As well as higher variance in results, this method yields poorer quality in the relative error with respect to Figure 4.16. The whiskers show that some closer-to-optimal results are reached but the numerical error is considerable.

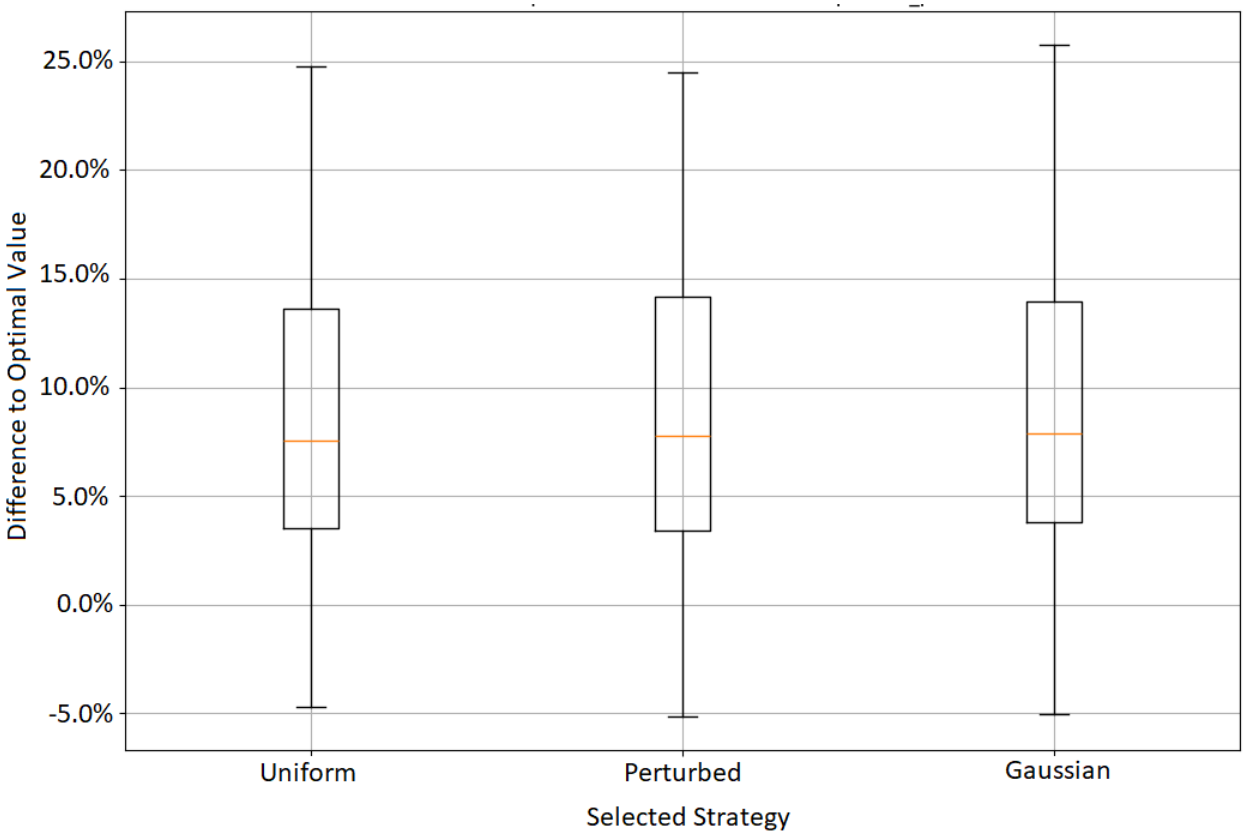


Figure 4.18: Boxplot for Optimize, allowing intermediate results, $p = 15$: allowing for intermediate results increases the variance even more than in Figure 4.16, as some added results are performed with simpler circuits which do not have the capability of getting closer to the global minima.

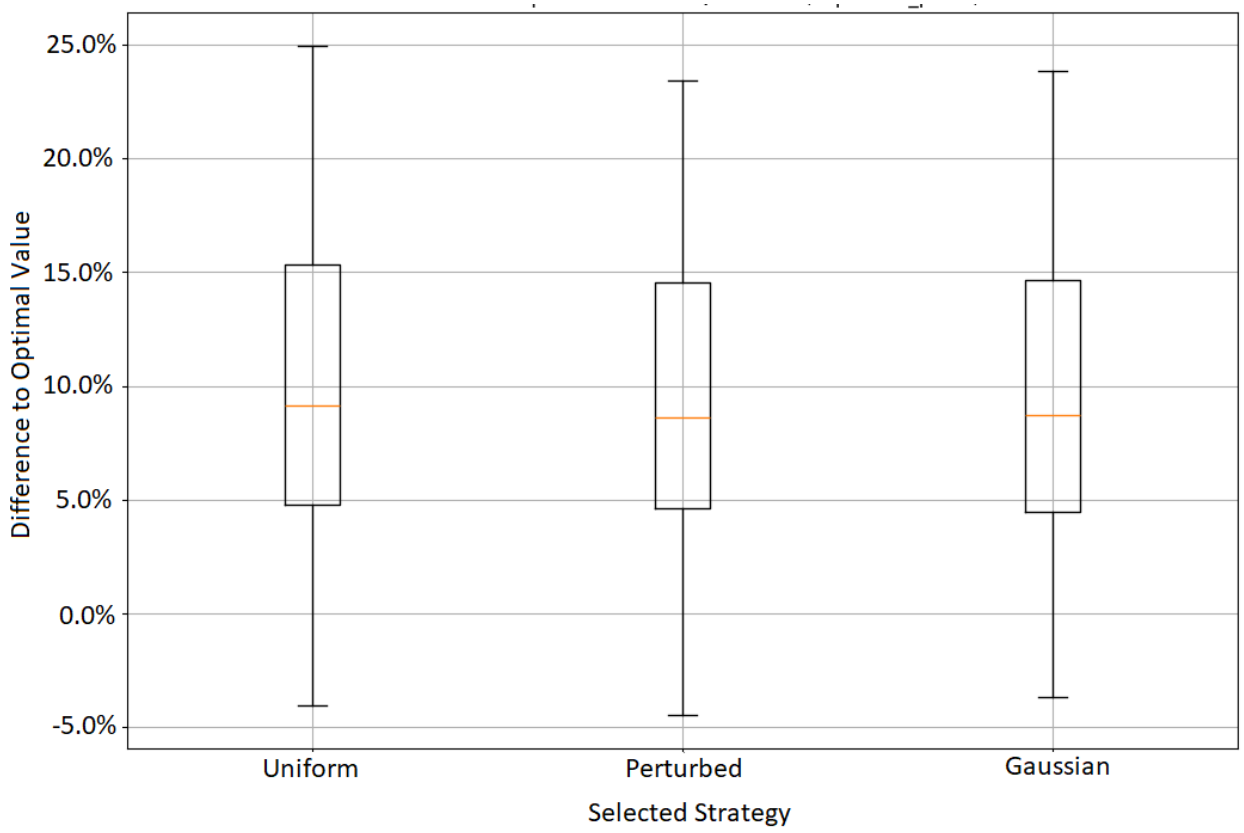


Figure 4.19: Boxplot for Optimize, warm-started, allowing intermediate results, $p = 15$: Once again, no significant improvement is obtained from applying the warm-start add-on. Wide distributions with high error percentages.

This first experiment is a guideline to showcase the methods used and their effect on a specific case. One key concept is the Gaussian strategy being the most robust among the other two across all the methods. Regarding warm-start, no significant improvement is obtained from it, but it can be attributed to the specific case, as the preliminary initial solution given by CPLEX is close to the superposition solution, so the initial state and, by default, the mixer operator will remain unchanged. Another key concept is the errors seen in the optimization method, which for this specific problem, leads to the minimization of a circuit with high variance. Results from $p-1$ being used to start minimizing at p condition the sequence of parameters the optimizer iterates, and converge into an ill-conditioned part of the cost function, generating larger errors.

4.6 11 Node Execution with Pseudo-random Graphs

The previous experiment was performed using only one case of the infinite amount of problems that can be generated. As a single case comparison lacks insight into the behavior of our methods, the following experiment is proposed:

A thousand different graphs such as the one in Figure 4.20 with the Scandinavian network structure are generated, and their node weights are randomized. Their purpose is to simulate a tool that, at each timestep, can infer the best solution to the problem and generate partitions that share energy surpluses. Across all the experiments, $p = 15$ as a maximum will be used, as it is a value after which no improvement is seen.

Executing each methodology on a thousand graphs for ten runs and averaging the result, yields a real idea of the applicability of each method is obtained. All executions were performed on a simulator and results were also computed using our warm-start technique, but they did not show any class of improvement so they have been omitted.

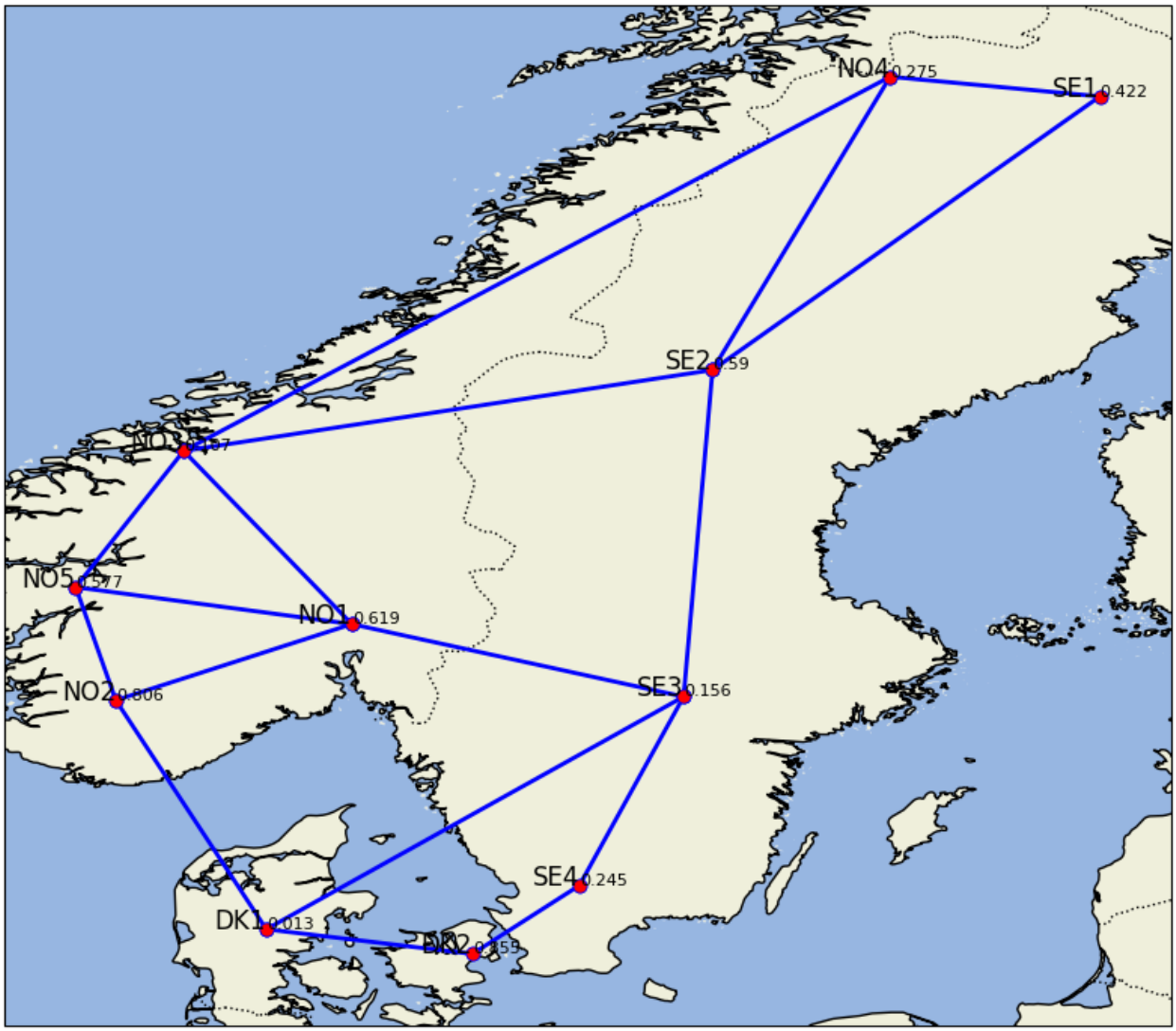


Figure 4.20: Scandinavian Power Grid for $n = 11$, the full extent of the proposed problem, involving the regions of Norway, Sweden, and Denmark.

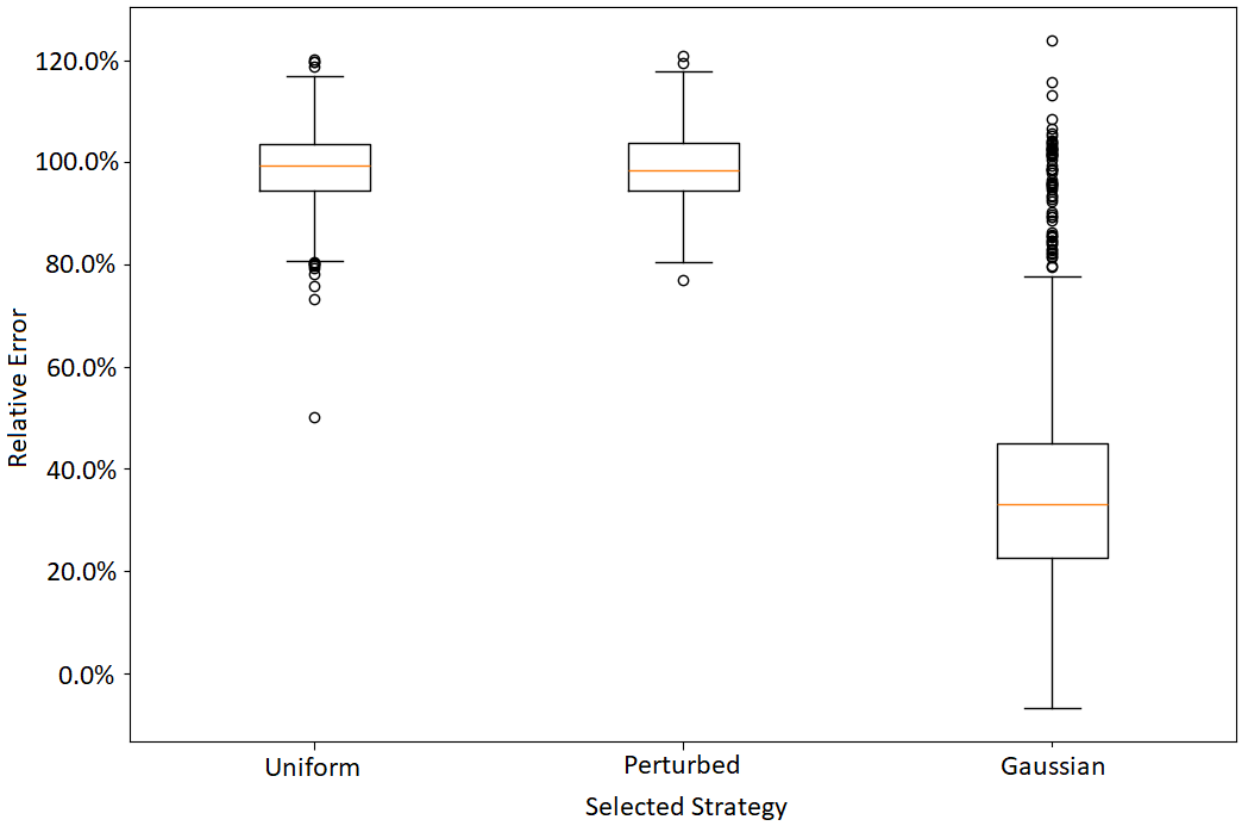


Figure 4.21: Relative error between the three strategies applied with the Straight Minimization method, in boxplot format. Only the Gaussian strategy proves capable of approximating the optimal solution in some graph instances.

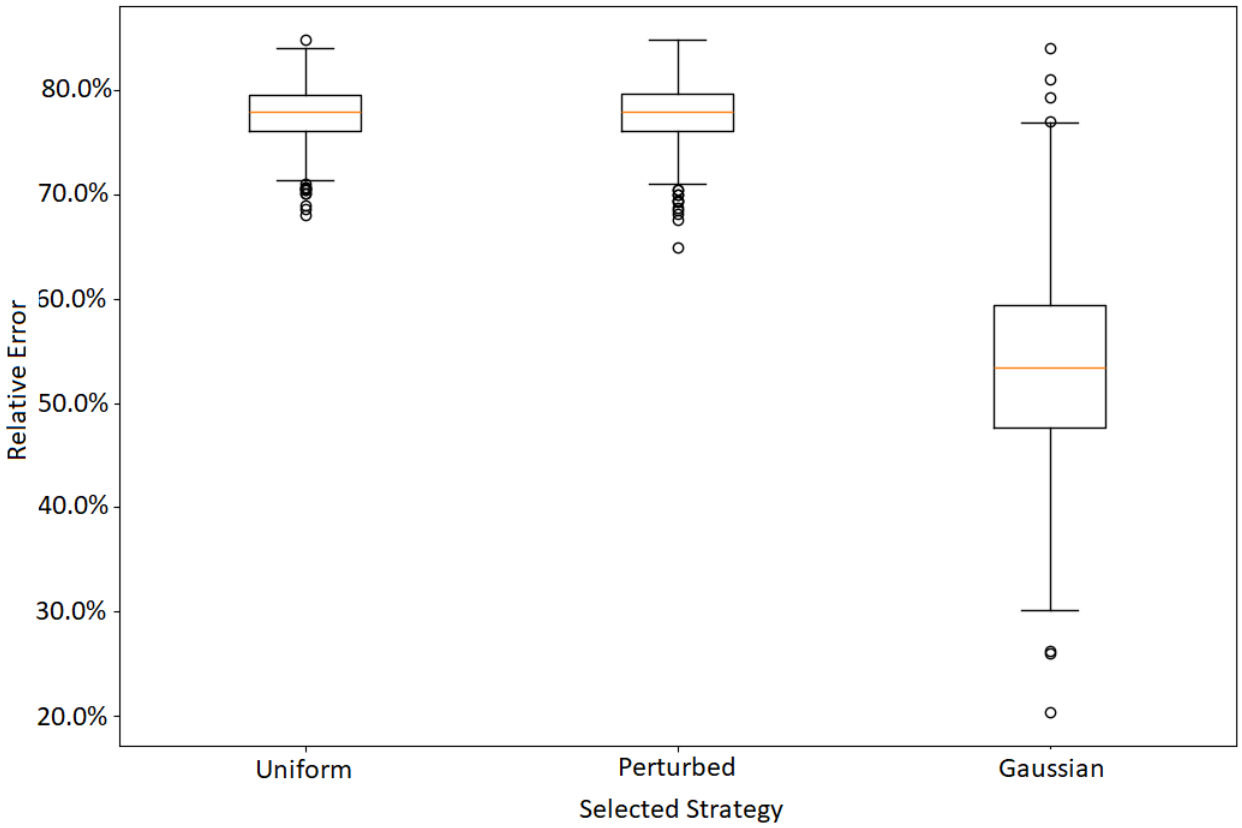


Figure 4.22: Relative error between the three strategies applied with the Straight Minimization Multiangle method. The only strategy that performs well with this method is the Gaussian.

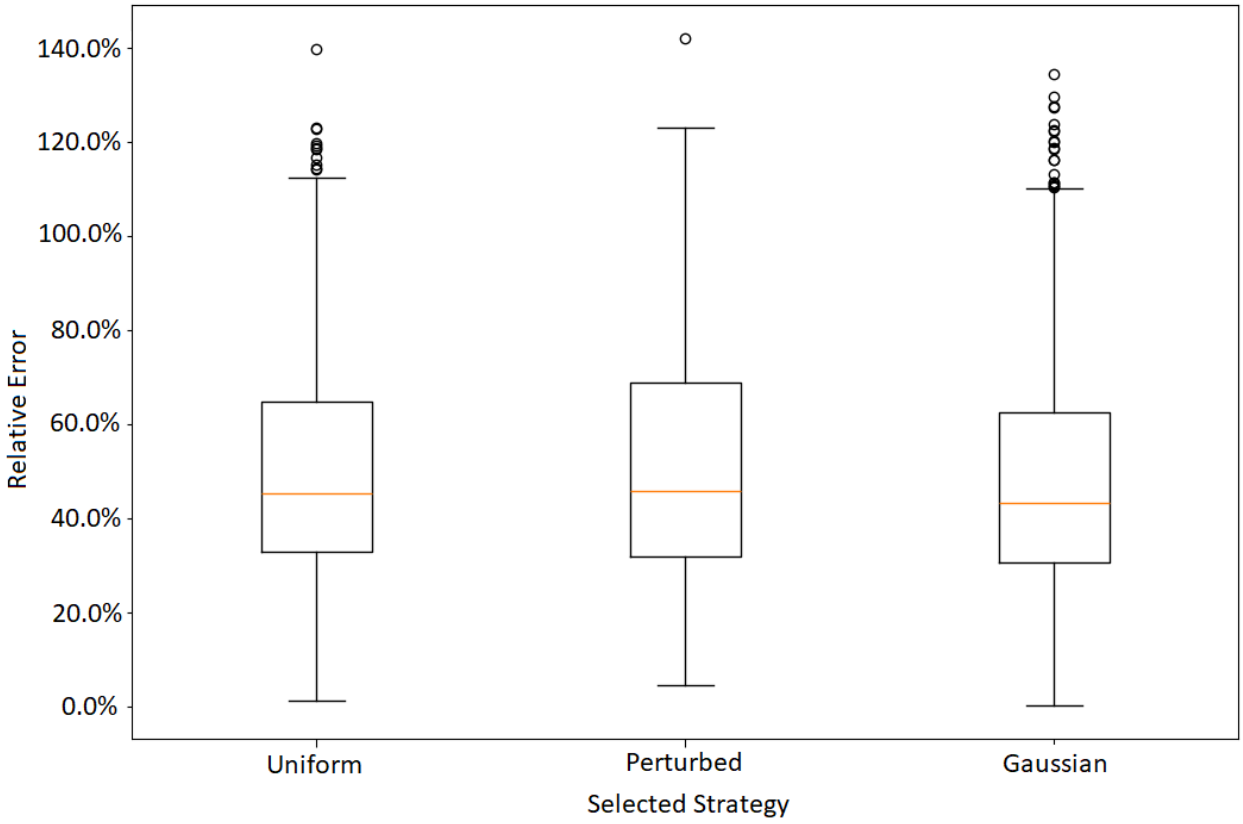


Figure 4.23: Relative error between the three strategies applied with the Optimize method. All strategies perform similarly and, on average successfully.

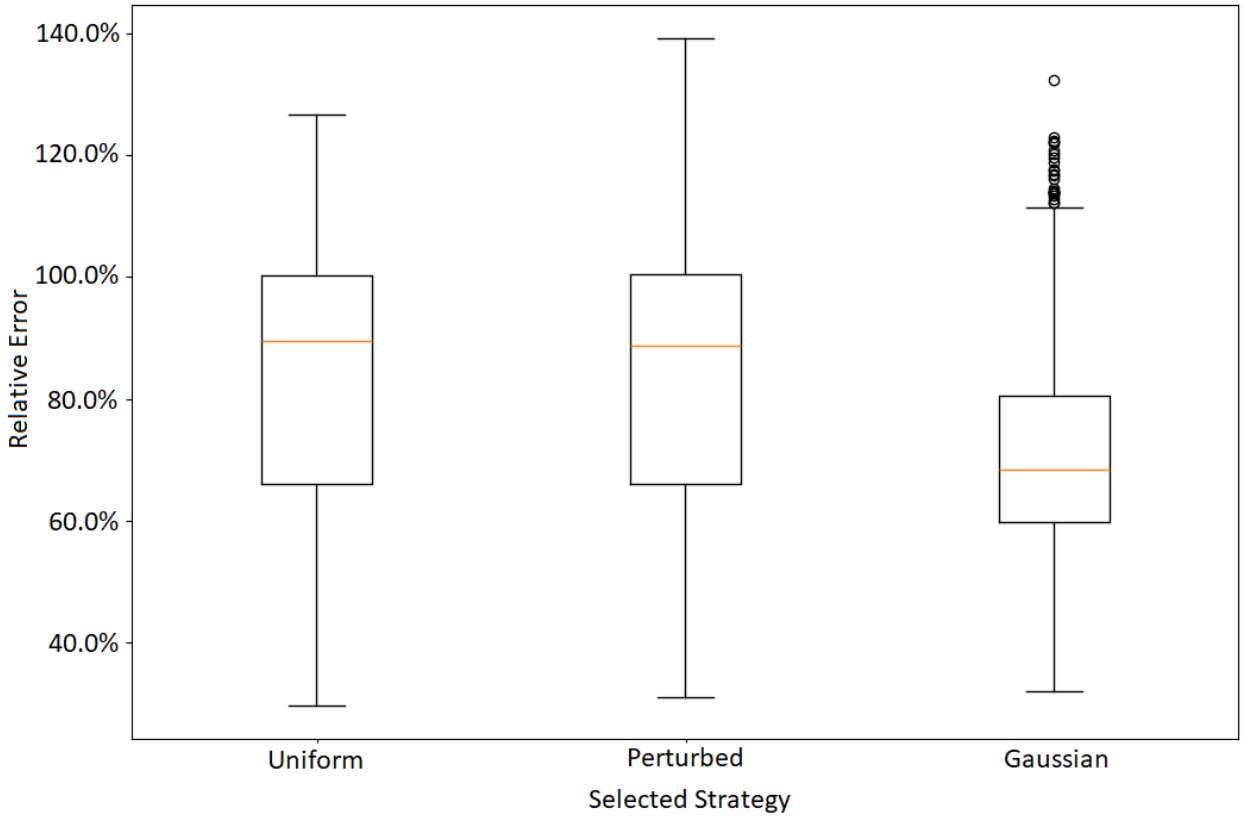


Figure 4.24: Relative error between the three strategies applied with the Optimize method. This method incurs many errors in the computation and yields suboptimal solutions.

The initialization point given to the straight minimization algorithms is crucial, as seen in Figure 4.21 and Figure 4.22. On the other hand, optimizing methods in Figure 4.23 and Figure 4.24 seem more unaffected by initialization. All these results are reflected in Table 4.1. If we look at the best case for each method, assuming the Euclidean norm of the value differences as goodness criteria, the result in Figure 4.25. Comparing between all the norms of each method (for each, the best strategy), yields Table 4.2.

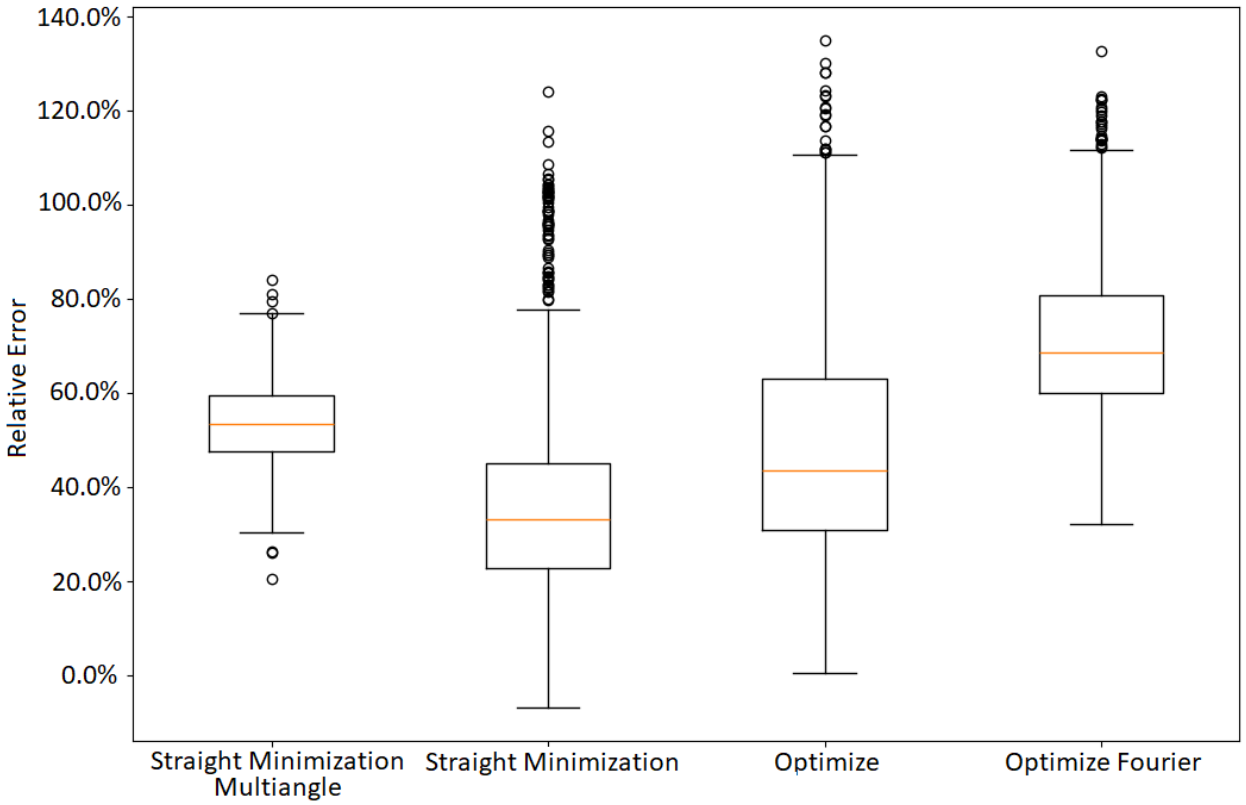


Figure 4.25: The following comparison is achieved given the best strategy for each method. Only the methods of Straight Minimization and Optimization can reach enough degree of convergence. The rest of the methods converge to less optimal solutions in a lower percentile. It is important to note that numerical errors can be found in the solutions that exceed the classical optimal solution.

Method	**Strategy**	**Norm of the Difference**
Straight Minimization	Uniform	150.455
	Perturbed	150.177
	Gaussian	63.727
Straight Minimization Multiangle	Uniform	117.827
	Perturbed	117.886
	Gaussian	82.325
Optimize	Uniform	88.563
	Perturbed	89.311
	Gaussian	86.453
Optimize Fourier	Uniform	129.985
	Perturbed	129.201
	Gaussian	111.438

Table 4.1: A full comparison between norms of different methods and strategies, showing how the Gaussian strategy performs best and initialization is very relevant in the Optimizing methods.

Method	**Best Strategy**	**Norm of the Difference**
Straight Minimization	Gaussian	63.727
Straight Minimization Multiangle	Gaussian	82.325
Optimize	Gaussian	86.453
Optimize Fourier	Gaussian	111.438

Table 4.2: A comparison between the Euclidean norms of different strategies, portraying how the Straight Minimization method can perform better than the other methods.

For each graph, we compare the results from all the methods and select the best-

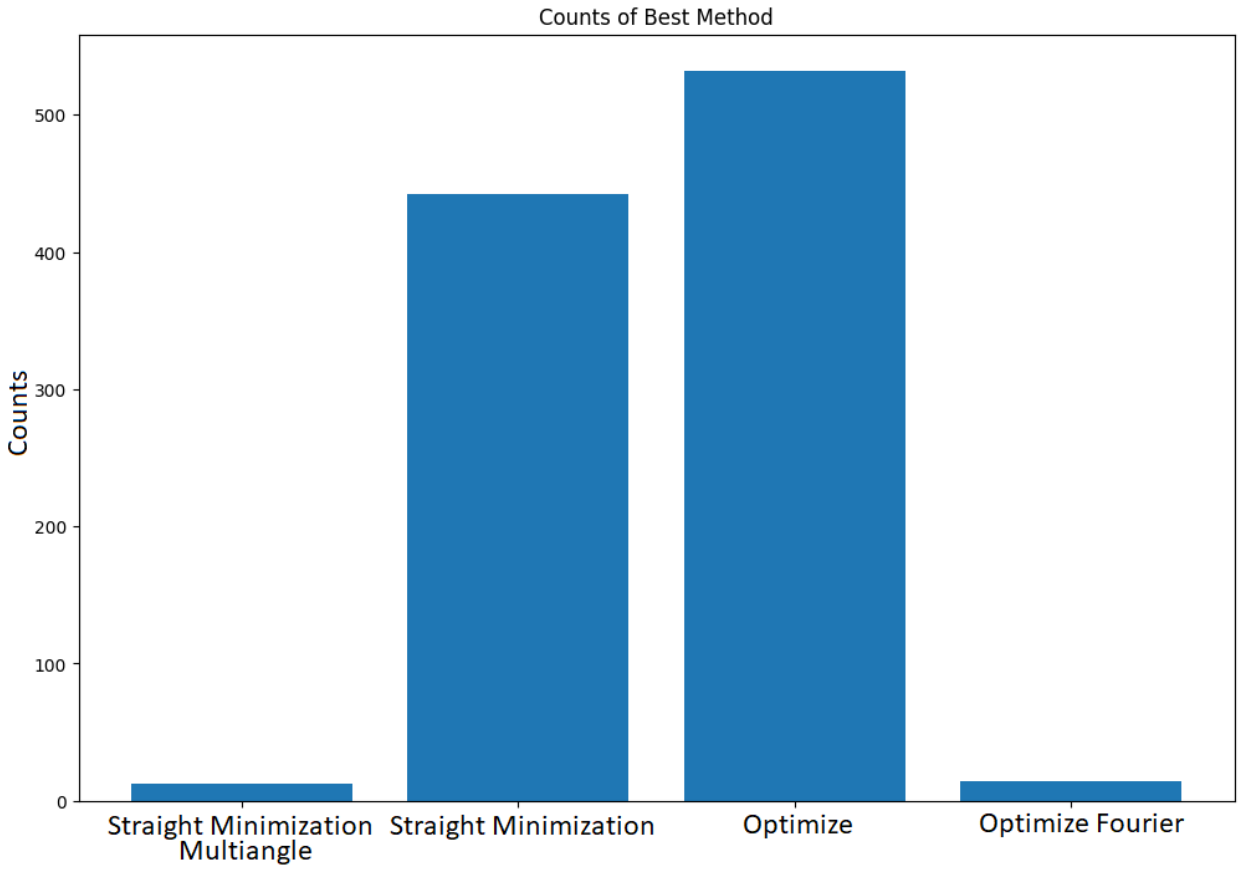


Figure 4.26: Histogram comparison where we count, for each method, how many times it was the best at its task. Concerning Figure 4.25, we can see coherence in obtained results, as these are the only methods that have gotten close enough to an optimal solution.

performing one in the specific instance. Thus, we receive information about, on average, the technique that yields better results consistently across scenarios.

We have seen how susceptible "Minimize" methods are to parameter initialization, and how the Optimize method, despite the numerical error, was the best in some of the previous cases (Figure 4.26). Analyzing cases where the best method was "Optimize" and extracting their execution data highlights instances only requiring 4 parameters to fit the problem. We propose to add to the solution pipeline in Chapter 3 an extra step where a neural network is the one that computes the random initial parameter. A simple network with one intermediate layer of 16 nodes is enough to capture the relationship between previous optimal parameters

and their surplus weights, allowing for shallower quantum circuits and decreasing the error rates. The actual design of the neural network and optimization of its parameters and hyper-parameters is out of the scope of this thesis and the results of this process can be seen in Figure 4.27.

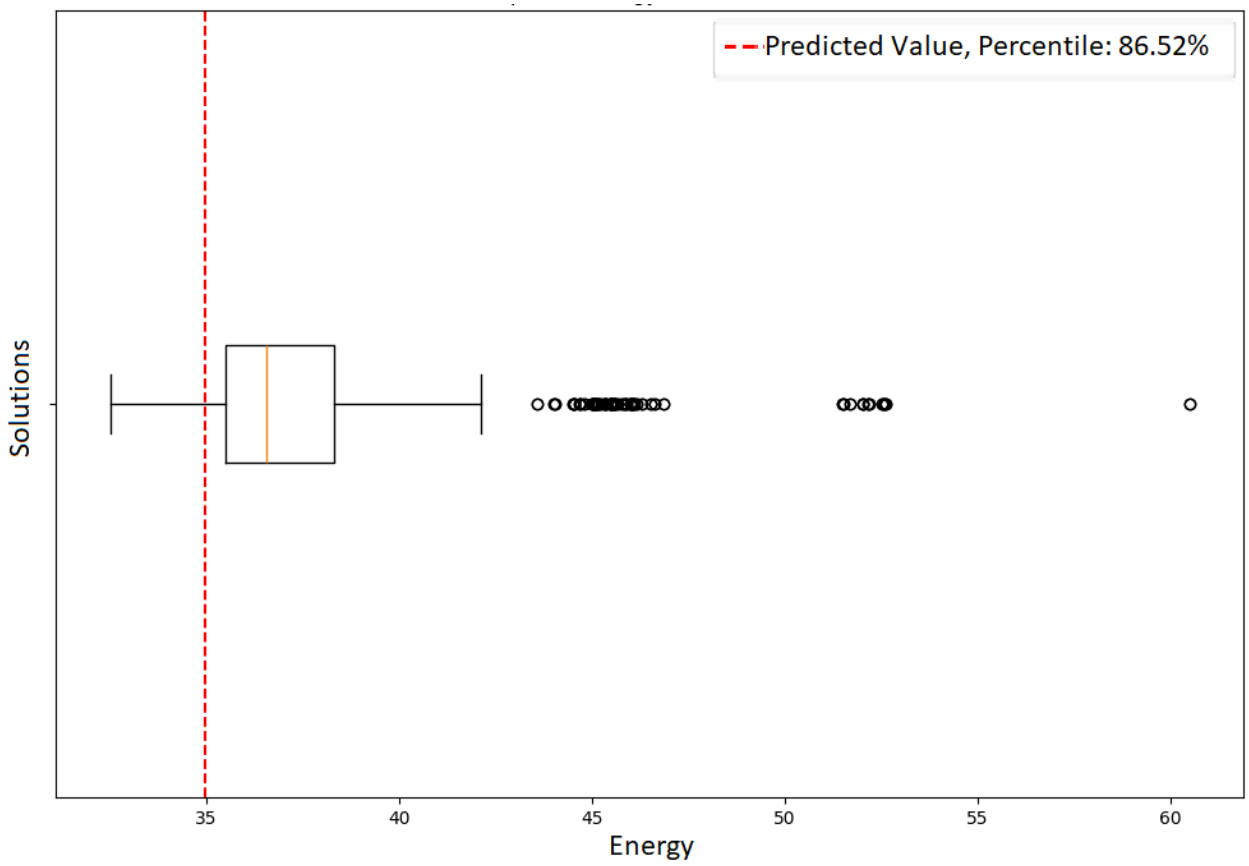


Figure 4.27: Results of the complete pipeline with a neural network seeding the initial parameters for the circuit with $p = 2$. 100 executions averaged showcase this method as robust and more efficient than executing $p = 15$, with similar suboptimal solutions in total.

4.7 Execution Time as n Increases

One of the main promises of quantum computing has been its ability to scale as problems become larger [7]. In the next experiment setup, we solve problems in a quantum computer provided by IBM [84] and obtain real execution times using the previous pipeline. The system specifications of the used machine are found in Table 4.3.

Attribute	**Details**	**Attribute**	**Details**
Qubits	127	Status	Online
EPLG	3.9%	System region	us-east
CLOPS	5K	Total pending jobs	2 jobs
Processor type	Eagle r3	Basis gates	ECR, ID, RZ, SX, X
Version	1.2.38	Instance usage	93 jobs
Median ECR error	9.485e-3	Median SX error	2.997e-4
Median readout error	1.690e-2	Median T1	229.32 µs
Median T2	99.07 µs		

Table 4.3: System Information of *ibm_kyoto* from [84]. Some of the most relevant information to us is the number of qubits, which affects the size of the problem. Also, ECR (Echoed Cross-Resonance) gates are the native implementation of our RZZ gates and will determine their error rate. SX is only relevant for the initial state if H gates are used, and RZ gates are considered 0 error gates [85].

Execution time does not appear to be directly correlated with the number of qubits used (Figure 4.28), as increasing the problem size yields similar results. Compared to how a classical brute-force method behaves, a clear potential in providing polynomial execution time to current exponential time problems can be seen. Nevertheless, as p increases, we can see a significant increase in execution time as shown in Figure 4.29. As a problem increases its size, so will the Hamiltonian. The proposed Hamiltonian exemplifies this case as a size increase leads to more edges and more nodes, both adding terms to the energy operator. With complexity increase, comes a readjustment of p value, as the adiabatic process that simulates the QAOA needs more fine-grained steps and parameters to adjust. Thus, as p must be increased for larger n values, the execution time will consequently increase by the ratio shown in Figure 4.29.

In an ideal scenario, all the previous experiments would have been executed in one of the

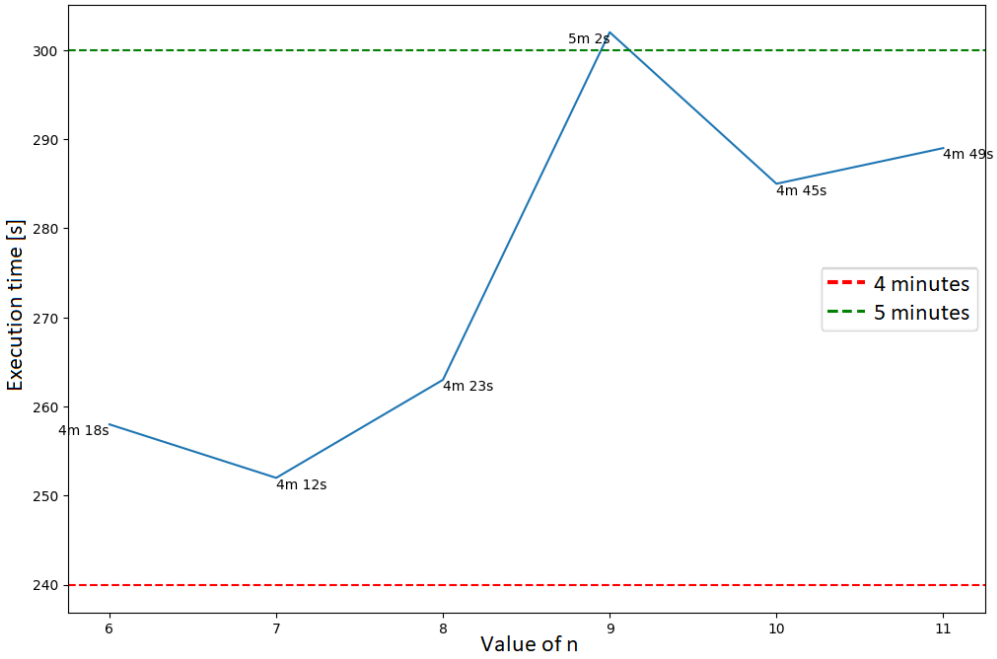


Figure 4.28: Relationship between the execution time and the size of the problem. It is important to note that all of these results are taken with $p = 2$, and the execution time for each n value is also dependent on the number of shots of the circuit, which in all our cases 100 shots are taken and averaged; and the number of optimization rounds needed to converge. These 100 shots took 6 seconds in all the different cases, and differ from the exponential growth of classical methods.

provided quantum computers by IBM, but these services are charged at 1.6$ per quantum second, an unfeasible expense for our budget. Therefore, experiments were performed in a simulator that ran in a cluster with execution time and RAM limiting the performance. Therefore, the economic cost of the solution must also be considered and it will be mentioned in the next chapter.

Finally, $p = 15$ was assumed throughout all experiments as a correct value of p to avoid error, and a noise-free simulator was used in the previous experiments. In IBM's documentation [29], a simple example is provided and used with $p = 2$. Using the data found in Table 4.3 and computing the total number of single qubit gates and coupling

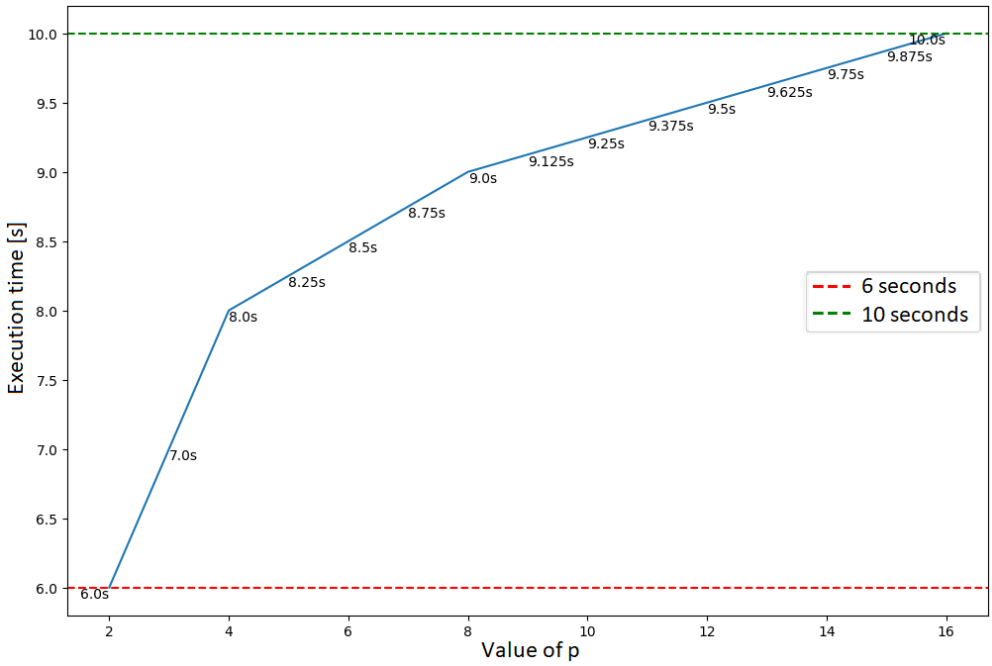


Figure 4.29: For $n = 11$, we vary the depth of the circuit and observe how much time it takes to compute 100 shots from it. Here an increase in computation time is noticed. For larger n, the curve should be reevaluated.

qubit gates, a 15% error rate can be found just in gate fidelity, without taking into account decoherence errors. For our case scenario, given $p = 2$, **the error rate is** 105%, meaning that our problem complexity is far greater than the current machines in IBM can handle as far as gate fidelity concerns.

4.8 Conclusion and Outlook

In this study, we provide several insights into the performance and limitations of different quantum optimization methods, particularly in the context of solving complex partitioning problems on quantum hardware. For this purpose, we study the distribution of energy surpluses in the Scandinavian power grid as a custom graph partitioning problem. Strategies employed include iterative minimization with increasing p and using intermediate results (Optimize methods), deriving a multiangle implementation, warm-starting with classically

obtained solutions of the relaxed problem, and using reparametrizations.

Straight Minimization methods, including both the standard and Multiangle approaches, exhibited varying degrees of effectiveness. The standard Straight Minimization method, while straightforward in its implementation, struggled with convergence to optimal solutions when initialization was not ideal. The Gaussian strategy consistently outperformed the uniform and perturbed strategies, emphasizing the non-trivial role of biased parameter initialization. In contrast, the multiangle approach, although offering greater flexibility by allowing different parameters for each term of the cost Hamiltonian, showed improved performance but increased sensitivity to errors. The balance between fine-grained control and error susceptibility needs careful consideration in practical applications. The latter approach demonstrates the potential for improved optimization by introducing more fine-grained control of the ansatz. However, this extra control is more error-susceptible and underscores the importance of problem-specific parameter initialization methods as quantum circuits become more complex.

Optimize methods, both standard and Fourier-based, showed results achieving near-optimal solutions. These methods exhibited a higher tolerance to initial parameter variations, making them more robust in practical scenarios. The standard Optimize method, in particular, demonstrated a good balance between convergence speed and the practical use of a solution. However, the error rates presented, even in a noise-free simulator, disqualify it as a potential solution by itself without any other tools supporting it. The Fourier-based method, while introducing more complex parameter dependencies, also incurred higher computational overhead and did not consistently outperform the standard Optimize method. This indicates that while Fourier-based parameterization has potential, it requires further refinement to be practically applicable.

Additionally, our study explores the importance of warm-start techniques in optimization. Despite initial expectations, warm-starting did not consistently improve performance across all methods. This suggests that the benefits of warm-starting are highly dependent on

the specific problem instance and the quality of the initial solution provided. In our case, initial solutions seemed to always lead to the default superposition state or some more ill-conditioned derivation of it.

In a certain number of cases, we could reduce the circuit depth required to lower p values. The use of a neural network could potentially mitigate some of the initialization challenges observed with the Multiangle and other methods. Employing past optimal parameters conditioned on the surplus weights selection allows us to infer a fitting initial parameter set that can potentially reduce circuit complexity while maintaining solution quality.

Another key result of the performed experiments is the relationship between the depth of the circuit and the computational time, but the apparent independence between n and the former. However, as n increases, the complexity of the Hamiltonian grows accordingly, requiring higher p values to achieve accurate solutions. This increase in p directly translates to longer execution times and higher error rates, but it does not show an exponential increase as classical brute-forcing methods do.

Lastly, our findings emphasize the need for ongoing development in both quantum algorithms and hardware. The constraints imposed by gate errors and decoherence times are significant barriers that must be addressed to fully achieve the potential of quantum optimization. Future work should focus on error correction, more fault-tolerant systems, the introduction of some error mitigation techniques such as the ones explained in this work, and hardware improvements to support larger and more complex quantum circuits. In our case study, more than two partitions could not be considered due to the lack of computational capabilities of the simulator, as the increase in the number of qubits multiple partitions require was not feasible. Moreover, a proper comparison with the Quantum annealing version of the problem was left out of the scope of this work and could be an interesting continuation, as our problem presents eigenstates with energy levels relatively close to each other, a characteristic Quantum annealing struggles with [86].

In conclusion, quantum computing remains an expensive methodology both in terms of

computational and economic resources, and in time execution, compared to the current best classical methods. While quantum optimization shows great potential for solving NP-hard problems like the k-Max-Cut, current hardware limitations necessitate careful consideration of circuit design, initialization strategies, and error mitigation techniques. According to predictions from other works [7], we are not yet close to substituting supercomputers with quantum computers for logistical applications. The full capacity of universal gate computing remains uncertain, but continued advancements in these areas are crucial for achieving practical and reliable quantum computation. An important aspect to reconsider is the chosen technology, as there is a constant increase in promising new quantum computer implementations. Superconducting qubits, for instance, is a technology still in its early stages, and the coming years promise significant developments. These advancements, coupled with cutting-edge quantum computer implementations based on different physical systems, and substantial investments in the field [87], hold the potential to transform the field of quantum computing.

CHAPTER 5

SUSTAINABILITY STUDY

In this chapter, the sustainability of this work is presented, analyzing some key general concepts, ethical implications, and their relationship to the *Sustainable Development Goals* [88]. We study the sustainability matrix in Section 5.1 and the ethical implications involved in Section 5.2. Finally, we match our development with the 2030 Agenda's goals in Section 5.3.

5.1 Sustainability Matrix

This section explores the three key aspects of sustainability analysis and their connection to the experiments performed.

5.1.1 Environmental Impact

The Quantum Approximate Optimization Algorithm (QAOA), together with the strategies and add-ons proposed, offers a novel approach to partitioning the Scandinavian power grid into self-sufficient communities. This contributes to sustainability by optimizing energy distribution, reducing waste, and enhancing the efficiency of renewable energy utilization. In Scandinavian power grids, when energy demands are not met, flexibility providers [89] must be contacted and connected to the network, using highly contaminating methods to respond to the demand. Otherwise, the network would black out. By enabling real-time adjustments to energy demand and supply, QAOA supports the distribution of clean energy on the grid, as it is known that almost all energy production in Scandinavian countries comes from renewable sources [90].

Implementing QAOA in real-time energy demand partitioning has the potential to signif-

icantly reduce the environmental footprint of power grids. The optimized partitioning leads to improved energy efficiency and lower transmission losses. However, the environmental benefits are contingent on overcoming current limitations in quantum computing, such as high error rates and significant energy consumption during computations. Continuous advancements in quantum hardware efficiency are crucial to maximizing these environmental benefits.

The high error rates and current limitations of quantum computing pose challenges to the sustainability of the proposed QAOA solution. Frequent computational errors necessitate repeated calculations, which can offset the energy savings achieved through optimization. Moreover, the error rates impact the accuracy of energy demand partitioning, potentially leading to suboptimal decisions. Developing more error-tolerant algorithms and improving quantum error correction methods are essential for enhancing the sustainability and reliability of QAOA applications [91].

Quantum computers, particularly in their current early stages, consume considerable energy compared to classical computers. The cooling systems required for maintaining qubits at extremely low temperatures contribute significantly to this energy demand. However, as quantum computing technology matures, the energy efficiency of quantum processors is expected to improve. Quantum computing has the potential to lower CO2 emissions significantly, especially if advances reduce the need for extensive refrigeration systems. Given that energy generation accounts for approximately 40% of global CO2 emissions [92], optimizing this sector using efficient quantum algorithms could have substantial environmental benefits [93].

The neural network-based initialization strategy enhances the sustainability of the QAOA approach by improving convergence rates and reducing the computational overhead associated with finding optimal parameters. By leveraging past optimal parameters, the neural network can provide a better starting point for the optimization process, thereby decreasing the number of iterations required and the overall energy consumption. This approach

not only improves the efficiency of quantum computations but also aligns with sustainable computing practices.

Currently, the highest expense of quantum computers is the refrigeration system that keeps the chip at superconducting conditions. Achieving it requires a high energy consumption. Therefore, in this project, we have only resorted to quantum execution when results have been performed on the simulator, and just the best cases were executed in a quantum computer.

At the end of the QAOA project's life-cycle, the disposal of quantum hardware poses an environmental challenge. To mitigate this, strategies for recycling and repurposing quantum components should be developed. Implementing take-back programs and designing hardware with modularity in mind can facilitate easier upgrades and reduce electronic waste .

5.1.2 Economic Impact

Adopting quantum computing for energy grid optimization has significant economic implications. It can lead to cost savings through more efficient energy distribution and reduced reliance on fossil fuels. However, the high initial investment and operational costs of quantum computing infrastructure need to be justified by long-term benefits and widespread adoption.

Ongoing advancements in quantum computing hardware and error mitigation techniques are necessary for the long-term sustainability of the proposed solution. Improvements in qubit coherence times, gate fidelities, and error correction methods will directly impact the viability of QAOA for large-scale energy optimization. Furthermore, developments in quantum algorithms tailored for specific applications will enhance the practicality and effectiveness of quantum computing in addressing complex optimization problems.

As of now, using the *pay-as-you-go* version offered by IBM [94]. Each second of quantum execution is valued at 1.6$. Using the results found in Chapter 4, we can find an upper bound for the expense of a single run at 300$ (prices are given in dollars as the exchange rate may

vary from the time this work is written. As technology advances, this price is expected to be significantly reduced, and so its execution time is estimated to be 5 minutes on average. This value uniquely determines the expense per run, and, assuming a system that computes redistribution every half an hour, equates to 14400$. Prices for specific flexibility providers have not been found, but they could very well exceed this number.

In addition, more applications of this strategy could be found that reduce computation times in other logistic problems, increasing its economic impact through the reduction of immense computation times and computational power spent by super-computers. The deployment and operational phases of the QAOA project will necessitate substantial computational resources, primarily quantum hardware and cooling systems. The environmental impact of these resources includes energy consumption and the associated carbon footprint. While the initial resource investment is high, the long-term benefits of optimized energy distribution could offset these costs. The reduction in transmission losses and improved integration of renewable energy sources are expected to lower the overall environmental impact over the project's life cycle.

5.1.3 Social Impact

Socially, the enhanced reliability and stability of power grids contribute to energy security and resilience, benefiting communities by ensuring consistent access to clean energy.

The proposed QAOA solution has the potential to be scaled to other regions and sectors beyond the Scandinavian power grid. Its application can extend to various optimization problems in logistics, finance, and other industries where efficient resource allocation is crucial. The adaptability of QAOA to different types of graphs and constraints makes it a versatile tool for addressing a wide range of sustainability challenges. However, scalability will depend on the continued evolution of quantum hardware and software capabilities.

5.2 Ethical Implications

The integration of the Quantum Approximate Optimization Algorithm (QAOA) in optimizing the Scandinavian power grid carries several ethical implications. One primary concern is data privacy, as the algorithm requires extensive data on energy usage patterns. Ensuring that this data is collected, stored, and processed securely is paramount to protect user privacy. Robust encryption methods, which could also utilize quantum technologies and strict access controls must be implemented to safeguard sensitive information against unauthorized access and potential breaches. Furthermore, transparency in data handling practices is essential to maintain public trust and ensure compliance with data protection regulations.

A future deployment must also consider the digital divide. There is a risk of exacerbating inequalities if only certain regions or groups have access to these advanced technologies. To mitigate this, it is crucial to promote equitable access to quantum computing resources and ensure that the benefits of technological advancements are distributed fairly across all segments of society. This includes investing in education and training programs to build quantum literacy and skills among diverse populations.

Environmental ethics are another crucial aspect. While QAOA aims to reduce the environmental impact of energy grids, the production and disposal of quantum hardware present significant environmental challenges. Ethical considerations extend to the entire life-cycle of quantum hardware, from the sourcing of raw materials to the end-of-life disposal of components. Responsible sourcing practices, fair labor conditions in the supply chain, and the development of sustainable recycling and disposal methods are essential to minimize the environmental footprint of quantum computing.

In addition, the potential societal impacts of widespread quantum computing adoption must be considered. This includes addressing concerns about job displacement due to automation and ensuring that technological advancements lead to broad economic benefits. Engaging stakeholders, including communities, policymakers, and industry leaders, in dis-

cussions about the ethical implications of quantum computing, is crucial for developing policies and practices that promote social justice and environmental sustainability.

5.3 The 2030 Agenda for Sustainable Development

The implementation of QAOA for optimizing energy grids aligns with several goals of the 2030 Agenda [88] for Sustainable Development. Primarily, it supports Goal 7 (Affordable and Clean Energy) by enhancing the efficiency and integration of renewable energy sources, thereby contributing to a more sustainable energy future. By optimizing energy distribution and reducing transmission losses, QAOA can make clean energy more accessible and affordable for all, promoting energy equity and security.

The algorithm's potential to reduce energy waste and optimize resource use also correlates with Goal 12 (Responsible Consumption and Production). By ensuring that energy resources are used efficiently and sustainably, QAOA helps to minimize the environmental impact of energy production and consumption. This contributes to the reduction of waste and promotes the sustainable management of natural resources.

By minimizing reliance on fossil fuels and supporting renewable energy, QAOA contributes to Goal 13 (Climate Action), aiming to reduce greenhouse gas emissions. The enhanced efficiency of energy grids facilitated by QAOA can lead to significant reductions in CO2 emissions, mitigating the effects of climate change and supporting global efforts to achieve carbon neutrality.

Additionally, the equitable distribution of energy resources and improved energy security address aspects of Goal 10 (Reduced Inequalities) and Goal 11 (Sustainable Cities and Communities). By ensuring that all communities have access to reliable and sustainable energy, QAOA supports inclusive and resilient urban development. This can improve the quality of life in urban areas, enhance economic opportunities, and promote social inclusion.

The adaptability of QAOA to different types of graphs and constraints makes it a versatile tool for addressing a wide range of sustainability challenges. However, scalability will

depend on the continued evolution of quantum hardware and software capabilities. Ongoing advancements in quantum computing technology will be essential to fully realize the potential of QAOA in contributing to the Sustainable Development Goals (SDGs).

REFERENCES

[1] A. Doinikov, "Geometry of the system under study". https://www.researchgate.net/figure/Geometry-of-the-system-under-study-In-our-calculations-we-use-the-spherical-coordinate_fig1_338610672, 2024. Accessed: 2024-06-25.

[2] K. Herb, "Bloch Sphere Visualizer". https://bloch.kherb.io/, 2023.

[3] A. D. Córcoles, A. Kandala, A. Javadi-Abhari et al., "Challenges and opportunities of near-term quantum computing systems", *Proceedings of the IEEE* **108** (2019), no. 8, 1338–1352.

[4] R. P. Feynman, "Simulating physics with computers", in *Feynman and computation*, pp. 133–153. CRC Press, 2018.

[5] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model", *Phys. Rev. E* **58** (Nov, 1998) 5355–5363, doi:10.1103/PhysRevE.58.5355.

[6] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM review* **41** (1999), no. 2, 303–332.

[7] F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor", *Nature* **574** (2019), no. 7779, 505–510.

[8] K. Utimula, T. Ichibha, G. I. Prayogo et al., "A quantum annealing approach to ionic diffusion in solids", *Scientific Reports* **11** (2021), no. 1, 7261.

[9] Authors, "Quantum Annealing Optimization Method for the Design of Barrier Materials", *Physical Review Letters* (2024).

[10] J. Preskill, "Quantum Computing in the NISQ era and beyond", *Quantum* **2** (August, 2018) 79, doi:10.22331/q-2018-08-06-79.

[11] Atom Computing, "Quantum Startup Atom Computing First to Exceed 1000 Qubits". https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/, 2023. Accessed: 2024-05-07.

[12] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm", *arXiv preprint arXiv:1411.4028* (2014).

[13] J. S. Otterbach, R. Manenti, N. Alidoust et al., "Unsupervised machine learning on a hybrid quantum computer", *arXiv preprint arXiv:1712.05771* (2017).

[14] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs", in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, p. 1222–1230. Association for Computing Machinery, New York, NY, USA, 2012. doi:10.1145/2339530.2339722.

[15] J. J. Grainger and W. D. Stevenson, "Power System Analysis". McGraw-Hill, Inc., New York, 1994.

[16] W. W. Soroka, "Analog Methods in Computation and Simulation". McGraw-Hill, New York, 1954. Available online at HathiTrust: http://hdl.handle.net/2027/mdp.39015001338428.

[17] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method", *IEEE Transactions on Power Apparatus and Systems* **PAS-86** (1967), no. 11, 1449–1460, doi:10.1109/TPAS.1967.291823.

[18] C. De Jonghe, B. F. Hobbs, and R. Belmans, "Optimal Generation Mix With Short-Term Demand Response and Wind Penetration", *IEEE Transactions on Power Systems* **27** (2012), no. 2, 830–839, doi:10.1109/TPWRS.2011.2174257.

[19] D. Q. Hung and N. Mithulananthan, "Multiple Distributed Generator Placement in Primary Distribution Networks for Loss Reduction", *IEEE Transactions on Industrial Electronics* **60** (2013), no. 4, 1700–1708, doi:10.1109/TIE.2011.2112316.

[20] A. Awasthi, K. Venkitusamy, S. Padmanaban et al., "Optimal planning of electric vehicle charging station at the distribution system using hybrid optimization algorithm", *Energy* **133** (2017) 70–78, doi:https://doi.org/10.1016/j.energy.2017.05.094.

[21] D. Bienstock and A. Verma, "Strong NP-hardness of AC power flows feasibility", *Operations Research Letters* **47** (2019), no. 6, 494–501, doi:https://doi.org/10.1016/j.orl.2019.08.009.

[22] Y. Zhou, Z. Tang, N. Nikmehr et al., "Quantum computing in power systems", *iEnergy* **1** (2022), no. 2, 170–187, doi:10.23919/IEN.2022.0021.

[23] M. H. Ullah, R. Eskandarpour, H. Zheng et al., "Quantum computing for smart grid applications", *IET Generation, Transmission & Distribution* **16** (2022), no. 21, 4239–4257.

[24] A. Ajagekar and F. You, "Quantum computing for energy systems optimization: Challenges and opportunities", *Energy* **179** (2019) 76–89, doi:https://doi.org/10.1016/j.energy.2019.04.186.

[25] G. Colucci, S. van der Linde, and F. Phillipson, "Power network optimization: a quantum approach", *IEEE Access* (2023).

[26] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information". Cambridge University Press, 2002.

[27] P. R. Halmos, "What does the spectral theorem say?", *The American Mathematical Monthly* **70** (1963), no. 3, 241–247.

[28] J. C. Familton, "Quaternions: A history of complex noncommutative rotation groups in theoretical physics". Columbia University, 2015.

[29] IBM, "Building Quantum Circuits". https://docs.quantum.ibm.com/build/circuit-construction, 2023. Accessed: date-of-access.

[30] B. Schumacher, "Quantum coding", *Phys. Rev. A* **51** (Apr, 1995) 2738–2747, doi:10.1103/PhysRevA.51.2738.

[31] L. K. Grover, "A fast quantum mechanical algorithm for database search", 1996.

[32] A. Ambainis and O. Regev, "An elementary proof of the quantum adiabatic theorem", *arXiv preprint quant-ph/0411152* (2004).

[33] S. Morita and H. Nishimori, "Mathematical foundation of quantum annealing", *Journal of Mathematical Physics* **49** (2008), no. 12,.

[34] G. T. Byrd, "Current State of Quantum Computing". https://www.computer.org/publications/tech-news/research/current-state-of-quantum-computing, 2024. Accessed: 2024-05-08.

[35] M. Pivoluska and M. Plesch, "Implementation of quantum compression on IBM quantum computers", *Scientific Reports* **12** (2022), no. 1, 5841.

[36] N. Earnest, C. Tornow, and D. J. Egger, "Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware", *Physical Review Research* **3** (2021), no. 4, 043088.

[37] IBM Corporation, "IBM Quantum Development & Innovation Roadmap", 2024. PDF document.

[38] D. van Delft and P. Kes, "The discovery of superconductivity", *Physics Today* **63** (09, 2010) 38–43, doi:10.1063/1.3490499.

[39] L. N. Cooper, "Bound Electron Pairs in a Degenerate Fermi Gas", *Phys. Rev.* **104** (Nov, 1956) 1189–1190, doi:10.1103/PhysRev.104.1189.

[40] B. Josephson, "Coupled superconductors", *Reviews of Modern Physics* **36** (1964), no. 1, 216.

[41] P. Krantz, M. Kjaergaard, F. Yan et al., "A quantum engineer's guide to superconducting qubits", *Applied Physics Reviews* **6** (06, 2019) 021318, doi:10.1063/1.5089550.

[42] IBM, "IBM Debuts Next-Generation Quantum Processor, IBM Quantum System Two, Extends Roadmap to Advance Era of Quantum Utility", December, 2023. Accessed: 2023-12-04.

[43] J. Tindall, M. Fishman, E. M. Stoudenmire et al., "Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment", *PRX Quantum* **5** (Jan, 2024) 010308, doi:10.1103/PRXQuantum.5.010308.

[44] Y. Tanaka and S. Kashiwaya, "Theory of tunneling spectroscopy of d-wave superconductors", *Physical review letters* **74** (1995), no. 17, 3451.

[45] A. S. Koshikawa, M. Ohzeki, T. Kadowaki et al., "Benchmark test of black-box optimization using d-wave quantum annealer", *Journal of the Physical Society of Japan* **90** (2021), no. 6, 064001.

[46] S. W. Shin, G. Smith, J. A. Smolin et al., "How" quantum" is the D-Wave machine?", *arXiv preprint arXiv:1401.7087* (2014).

[47] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer", *Nature* **417** (2002), no. 6890, 709–711.

[48] R. Ainsworth and J. Slingerland, "Topological qubit design and leakage", *New Journal of Physics* **13** (2011), no. 6, 065030.

[49] P. Kok, W. J. Munro, K. Nemoto et al., "Linear optical quantum computing with photonic qubits", *Reviews of modern physics* **79** (2007), no. 1, 135.

[50] C. Kloeffel and D. Loss, "Prospects for spin-based quantum computing in quantum dots", *Annu. Rev. Condens. Matter Phys.* **4** (2013), no. 1, 51–81.

[51] A. Tsintzis, R. S. Souto, K. Flensberg et al., "Majorana Qubits and Non-Abelian Physics in Quantum Dot–Based Minimal Kitaev Chains", *PRX Quantum* **5** (Feb, 2024) 010323, doi:10.1103/PRXQuantum.5.010323.

[52] J. D. Smith and J. E. Doe, "Achieving Minute-Scale Coherence Time in Ion Trap Quantum Computers", *Journal of Quantum Information Science* **15** (2023) 345–360. Data provided based on recent experiments, might require more detailed citation.

[53] M. Johnson and H. Lee, "Future Directions in Quantum Computing", *Quantum Technology Reviews* **2** (2023) 58–72. Overview of the future trends in quantum computing.

[54] M. Cerezo, A. Arrasmith, R. Babbush et al., "Variational quantum algorithms", *Nature Reviews Physics* **3** (2021), no. 9, 625–644.

[55] A. Callison and N. Chancellor, "Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond", *Phys. Rev. A* **106** (Jul, 2022) 010101, doi:10.1103/PhysRevA.106.010101.

[56] K. Bharti, A. Cervera-Lierta, T. H. Kyaw et al., "Noisy intermediate-scale quantum algorithms", *Reviews of Modern Physics* **94** (2022), no. 1, 015004.

[57] S. Cook, "The P versus NP problem", *Clay Mathematics Institute* **2** (2000) 6.

[58] A. K. Lenstra and H. W. Lenstra, "The development of the number field sieve", volume 1554. Springer Science & Business Media, 1993.

[59] B. H. Korte, J. Vygen, B. Korte et al., "Combinatorial optimization", volume 1. Springer, 2011.

[60] R. Shaydulin, S. Hadfield, T. Hogg et al., "Classical symmetries and QAOA", *arXiv preprint arXiv:2012.04713* (2020).

[61] P. Gleißner, G. Kruse, and A. Roßkopf, "Restricted global optimization for QAOA", *APL Quantum* **1** (2024), no. 2,.

[62] J. Tilly, H. Chen, S. Cao et al., "The variational quantum eigensolver: a review of methods and best practices", *Physics Reports* **986** (2022) 1–128.

[63] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik, "Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation", *PRX Quantum* **2** (2021), no. 2, 020329.

[64] K. Blekos, D. Brand, A. Ceschini et al., "A review on Quantum Approximate Optimization Algorithm and its variants", *Physics Reports* **1068** (June, 2024) 1–66, doi:10.1016/j.physrep.2024.03.002.

[65] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem", *arXiv preprint arXiv:1811.08419* (2018).

[66] F. G. Fuchs, H. O. Kolden, N. H. Aase et al., "Efficient Encoding of the Weighted MAX k-CUT on a Quantum Computer Using QAOA", *SN Computer Science* **2** (April, 2021) 89, doi:10.1007/s42979-020-00437-z.

[67] R. T. Rockafellar, "Lagrange multipliers and optimality", *SIAM review* **35** (1993), no. 2, 183–238.

[68] E. Pelofske, A. Bärtschi, and S. Eidenbenz, "Short-depth QAOA circuits and quantum annealing on higher-order ising models", *npj Quantum Information* **10** (2024), no. 1, 30.

[69] R. Tate, M. Farhadi, C. Herold et al., "Bridging classical and quantum with SDP initialized warm-starts for QAOA", *ACM Transactions on Quantum Computing* **4** (2023), no. 2, 1–39.

[70] F. G. Fuchs, K. O. Lye, H. Møll Nilsen et al., "Constraint preserving mixers for the quantum approximate optimization algorithm", *Algorithms* **15** (2022), no. 6, 202.

[71] S. Hadfield, Z. Wang, B. O'gorman et al., "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz", *Algorithms* **12** (2019), no. 2, 34.

[72] L. Zhou, S.-T. Wang, S. Choi et al., "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices", *Physical Review X* **10** (2020), no. 2, 021067.

[73] J. Choi and J. Kim, "A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and applications", in *2019 international conference on information and communication technology convergence (ICTC)*, pp. 138–142, IEEE. 2019.

[74] D. J. Egger, J. Mareček, and S. Woerner, "Warm-starting quantum optimization", *Quantum* **5** (2021) 479.

[75] A. Lucas, "Ising formulations of many NP problems", *Frontiers in physics* **2** (2014) 5.

[76] GeoGebra, "GeoGebra 3D Calculator". https://www.geogebra.org/3d?lang=en. Accessed: 2024-06-24.

[77] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256. 2010.

[78] K. He, X. Zhang, S. Ren et al., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *arXiv preprint arXiv:1502.01852* (2015).

[79] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures", *CoRR* **abs/1206.5533** (2012) arXiv:1206.5533.

[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980* (2014).

[81] M. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation", *Advances in Optimization and Numerical Analysis* (1994) 51–67.

[82] C. Pedrals, "TFG Quantum Computing". https://github.com/Carles-Pe/TFG_QuantumComputing, 2024. Accessed: 2024-06-25.

[83] F. of Economics, D. o. I. E. Management, and T. M. (IØT), "Solstorm Cluster". https://solstorm.iot.ntnu.no/wordpress/, 2024. Accessed: 2024-06-25.

[84] IBM Quantum, "IBM Quantum Computing". https://quantum.ibm.com/, 2024. Accessed: 2024-06-24.

[85] A. D. Córcoles, E. Magesan, S. J. Srinivasan et al., "Demonstration of a quantum error detection code using a square lattice of four superconducting qubits", *Nature communications* **6** (2015), no. 1, 6979.

[86] R. D. Somma, D. Nagaj, and M. Kieferová, "Quantum Speedup by Quantum Annealing", *Phys. Rev. Lett.* **109** (Jul, 2012) 050501, doi:10.1103/PhysRevLett.109.050501.

[87] J. Woodford, "Australia places A 1 billion bet on quantum computing firm PsiQuantum", *New Scientist* (April, 2024). A joint investment by the Australian

federal government and the government of Queensland makes PsiQuantum one of the largest dedicated quantum computing firms in the world.

[88] United Nations, "Sustainable Development Goals", 2023. Accessed: 2023-06-22.

[89] Community Energy England, "Flexibility and Demand Side Services", 2024. Accessed: 2024-06-22.

[90] N. M. of Petroleum and Energy, "Renewable Energy Production in Norway". https://www.regjeringen.no/en/topics/energy/renewable-energy/renewable-energy-production-in-norway/id2343462/, 2016. Last updated: 11/05/2016.

[91] Y. Xing, X. Zhang, H. Yang et al., "Sustainability in Quantum Computing: Challenges and Opportunities", *SN Computer Science* **2** (2021), no. 5, 1–8, doi:10.1007/s42979-021-00786-3.

[92] World Nuclear Association, "Carbon Dioxide Emissions from Electricity". https://world-nuclear.org/information-library/energy-and-the-environment/carbon-dioxide-emissions-from-electricity, 2024. Last accessed: 2024-06-22.

[93] J. Turliuk, "The Future of Climate Could Be in Quantum Computing", 2024. Accessed: 2024-06-22.

[94] IBM, "Quantum Pricing", 2023. Accessed: 2023-06-22.