Alma Mater Studiorum · University of Bologna

School of Science
Department of Physics and Astronomy
Master Degree in Physics

Optimization schedules for the Quantum Approximate Optimization Algorithm

Supervisor:
Prof. Elisa Ercolessi

Submitted by:
Matteo Grotti

Academic Year 2022/2023

Contents

List of Figures

Introduction

The last two decades have been characterized by the massive spread of new cutting-edge technologies related to the world of quantum computation and quantum information. Their common feature is the exploitation of the new paradigm provided by quantum mechanics and its, nowadays well-known, principles both from a software and a hardware points of view [27]. Even without exploring the enormous complexity of the hardware facilities of a quantum processor, known as QPU (Quantum Processing Unit), in detail, what can be said is that plenty of processors have been developed according to the most disparate technological benchmarks, from Josephson junctions [37] to atomic platforms [17].

From a theoretical point of view, since the second half of the twentieth century the scientific literature counts a limitless number of papers which helpfully paved the way to the development of quantum algorithms or the improvement of the ones already existing. Among the endless number of algorithms, a special class of them has become rather important due to the parallel development of machine learning [28] [29]: the optimization algorithms [12]. They are not, in principle, quantum algorithms, because they are not based on quantum mechanical principles. However, they acquired a fundamental role in modern hybrid quantum-classical algorithms since they are the crux which allows a sort of fusion and connection between current powerful classical computers and the noisy quantum computers, known as NISQ (Noisy Intermediate-Scale Quantum) devices. One of the hybrid algorithm which gained a lot of attention and attracted the curiosity and the efforts of several researchers during the last decade, is the QAOA (Quantum Approximate Optimization Algorithm) [10]. It is basically an optimization algorithm aimed at finding the minimum of a given cost function by means of a gate-based parameterized quantum circuit, whose role is to prepare the quantum state of a complex system, and a classical optimization schedule whose role is, instead, to locate the optimal parameters on the cost function, also called energy, landscape, exploiting its power to solve variational problems on that system. The main application, for which QAOA has been exploited the most, is the ground state preparation of quantum many body systems, such as the very famous Ising model [13]. The QAOA has shown a lot of potentialities in this sense, and we expect that QAOA and other possible hybrid algorithms would play a crucial role in the transition phase from a fully classical computation to a powerful fault tolerant quantum computation. Nobody knows how long this transition will be, but for sure it won't be short and the development of ideal quantum processors is far to be completed. As a consequence, before we get endowed with a massive computational power in our hands, we must settle with the employment of hybrid algorithms, which, however, are deemed very performative for some specific tasks.

The idea behind this thesis project is to test and evaluate several optimization schedules which have been developed and tailored for the QAOA with the purpose of improving and speeding-up the algorithm. As announced earlier, quantum many body theory and its models represents a good arena within which to test different routines for the QAOA since the latter has been devised with the intention of facilitating the preparation of quantum many body states. We then focused our attention to the Ising model with transverse field which we deem a very good candidate model for this scope. More specifically, we wish to clarify which optimization routines are better than others from a time management and solution quality points of view and how can they be possibly combined together in further analysis. It is worth to underline that we did not make use of a quantum processor, but we simulated the QAOA algorithm on a classical device. It is too early to have the possibility to physically apply the QAOA on a real quantum device, because of the issues given by the noisy behavior of the latter and that's why an overwhelming majority of the studies present in the literature have been performed with classical simulations. Nevertheless, simulations are still fundamental to understand how the algorithm works and how will it be possible to eventually apply it, one day, on a quantum computer.

The thesis is organized and structured as follows:

Chapter 1 provides an introduction to digital quantum computation and presents the basic elements to fully understand quantum algorithms;

Chapter 2 reports a state-of-the-art-based explanation of the QAOA and the relevant optimization routines already devised and tailored with the scope of constructing the ground state of many body models. Many of them have been tested on a classical problem, namely the MaxCut [41], which is also described in detail, slightly different from the Ising one. That's another reason which led us to apply these techniques on a more relevant model from a physical point of view;

Chapter 3 provides a mathematical and physical explanation of the various peculiarities of the XY chain [13], of which the Ising with transverse field is a special case, highlighting all the elements needed to understand how the QAOA can be actually applied on it. A deepening on the Ising with transverse field itself is provided in addition at the end of the chapter;

Chapter 4 is finally dedicated to show the outcomes of our simulations and discuss them exhaustively and coherently.

Chapter 1

Digital Quantum Computation

Quantum computation and information can be arguably defined as the study of the information processing tasks which can be achieved exploiting quantum mechanical systems [27]. The origins of this outstanding computational framework date back to the early twentieth century with the invention of quantum mechanics. This fascinating theory drastically changed the way we describe the world around us and this led to a succession of scientific discoveries. One of the most important of them, for the purposes of building up a quantum processor, was the development of an atom trapping technique capable to isolate a single atom and allow us to probe its quantum behavior.
Nowadays, several quantum processors (called QPUs) are available for quantum simulation purposes. Nevertheless, current QPUs are not still completely reliable, they are noisy and cannot be used to perform complicated computational tasks.
This chapter is structured as follows: in the first section we present a brief review of the main postulates of quantum mechanics; in section 2 we introduce the qubit, i.e. the basic unit of quantum computation, and its features; in section 3 we start talking about quantum circuit, listing all the quantum logic gates and how they work; in section 4 we introduce the concept of measurement applied to quantum bits; in section 5 we provide a complete proof of the universality of single qubit gates and CNOTs and the possibility to approximate every unitary transformation with a small set of gates; finally, in section 6, we delve into the quantum simulation algorithm from a theoretical point of view together with an illustrative example.

1.1 Postulates of quantum mechanics

In order to fully understand the basic concepts of quantum computing and for the sake of clarity, it is reasonable to briefly recap the postulates of quantum mechanics on which the whole mathematical and physical framework of quantum computation relies.

State space

The first postulate fixes the mathematical space we use to describe quantum mechanical systems, known as the *Hilbert space.* It is a complex vector space associated to any physical system whose dimension, which depends on the system we're dealing with, may vary, in principle, within two and infinity.

Postulate. *The state of the system is a described by a unit vector* $|\psi\rangle$ *belonging to the Hilbert space.*

The state $|\psi\rangle$ has to be intended as an equivalence class of vectors since the underlying physical state is the same regardless possible overall phases like $e^{i\alpha}|\psi\rangle$, which can be always neglected.
Since the Hilbert space H is a vector space, one can always identify a *basis* of vectors which allows to write any possible other state as a linear combination of them. For a finite dimensional Hilbert space, a generic state can be written as:

$$|\psi\rangle = \sum_{i=1}^{dim(H)} \alpha_i |i\rangle, \tag{1.1}$$

where the coefficients α_i are complex numbers. If at least two coefficients α_i in the previous equation are different from zero, the system is said to be in a *superposition* of the basis states.

Observables

In quantum mechanics, an *observable* is defined as a quantity about which we can get information from the system. Examples of typical observables are the position, the momentum or the energy.
In this mathematical framework, observables are associated with self-adjoint (or hermitian) operators, which can be represented by matrices of dimension $dim(H) \times dim(H)$ and act linearly on vector states.
Each operator A can be decomposed as a sum of the form:

$$A = \sum_{i=1}^{dim(H)} a_i |a_i\rangle\langle a_i| \tag{1.2}$$

where the vectors $|a_i\rangle$ are called *eigenvectors* and the scalars a_i are said, instead, *eigenvalues.* Applying an operator to one of its eigenvectors results in the same eigenvector multiplied by a numerical factor which is, indeed, the corresponding eigenvalue:

$$A|a_i\rangle = a_i|a_i\rangle. \tag{1.3}$$

Moreover, the eigenvalues are real numbers since the operator is hermitian and the set of eigenvectors is, in turn, a basis of the Hilbert space.

Dynamics

Now we may ask ourselves how the state of a system $|\psi\rangle$ changes with time. Quantum mechanics postulates that:

Postulate. *the evolution of an isolated physical system is described by a unitary transformation such that the state of the system at the final time, say,* t_2 *is related to the state at the initial time* t_1 *by a unitary operator which depends on* t_1 *and* t_2*:*

$$|\psi(t_2)\rangle = U(t_1, t_2)|\psi(t_1)\rangle. \tag{1.4}$$

Furthermore, the inverse evolution is described by the hermitian conjugate of the unitary:

$$|\psi(t_1)\rangle = U^{\dagger}(t_1, t_2)|\psi(t_2)\rangle. \tag{1.5}$$

We'll refer to $U(t_1, t_2)$ as the *evolution operator.* The mathematical form of the unitary U can be recovered leveraging the equation of motion of quantum mechanical systems, that is to say, the *Schrödinger's equation*

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \tag{1.6}$$

where H is the *hamiltonian* of the system and, since it is an hermitian operator, can be written as

$$H = \sum_E E|E\rangle\langle E|. \tag{1.7}$$

We'll henceforth set $\hbar = 1$. The states labeled with $|E\rangle$ are the energy eigenvectors and the scalars E are their corresponding eigenvalues.
Solving the Schrödinger's equation one gets

$$U(t_1, t_2) = e^{-iH(t_2 - t_1)} \tag{1.8}$$

such that

$$|\psi(t_2)\rangle = e^{-iH(t_2 - t_1)}|\psi(t_1)\rangle. \tag{1.9}$$

A remarkable point to stress is that any unitary operator one may think of can be always written an as imaginary exponential of a given hamiltonian.

Measurements

A fundamental milestone of quantum mechanics is the concept of *measurement*, which is based on two main cornerstones: the probabilistic nature of the measurement action on physical systems, and its destructive effect on the quantum state of the system. The former says that, when we measure a given observable on our system, we can only predict the probabilities of each possible outcome since a clear information about which outcome

is going to occur is not directly accessible. The latter is pointing out that the action of measurement itself disturbs the system and causes a variation of its quantum state (unless the very particular case where the state of the system just before the measurement was already an eigenvector of the operator associated to the observable).
What has been said so far can be mathematically summarized in the following statement:

Postulate. *Quantum measurements are described by a collection of operators M_m called measurement operators. The subscript m indexes the possible outcomes that may emerge in the experiment.*
If the initial state of the system is $|\psi\rangle$, the probability of recovering the outcome m is given by:

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle \tag{1.10}$$

and the post-measurement state will be then

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{1.11}$$

The change of the quantum state of the system after the measurement procedure is also known as *collapse*. The measurement operators must obey the *completeness relation* which states that

$$\sum_m M_m^\dagger M_m = \mathbb{I} \tag{1.12}$$

and can be eventually rephrased by saying that the probabilities must sum to one. In fact, by plugging eq. (1.10) into eq. (1.12) one gets

$$\sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle = \sum_m p(m) = 1. \tag{1.13}$$

The evaluation of the outcomes probabilities is somewhat easy to compute. For the sake of clarity, consider an observable A we wish to measure on a state $|\psi\rangle$. Since A is hermitian, the set of its eigenvectors form a complete basis of the Hilbert space, as remarked in the previous paragraph, and the state $|\psi\rangle$ can be therefore expressed as a linear combination of them:

$$|\psi\rangle = \sum_m c_m|\psi_m\rangle. \tag{1.14}$$

From eq. (1.2), called the *spectral decomposition formula*, the operator A can be rewritten as

$$A = \sum_m \lambda_m|\psi_m\rangle\langle\psi_m| \tag{1.15}$$

where λ_m is the set of A's eigenvalues and $|\psi_m\rangle$ is its set of eigenstates. The probability of the outcome λ_m is then

$$p(\lambda_m) = \langle\psi|\mathbb{P}_m^\dagger\mathbb{P}_m|\psi\rangle, \tag{1.16}$$

where we defined $\mathbb{P}_m$ as the projector onto the state $|\psi_m\rangle$,

$$\mathbb{P}_m = |\psi_m\rangle\langle\psi_m|. \tag{1.17}$$

It is straightforward to compute the expression in eq. (1.16) by replacing $|\psi\rangle$ with its decomposition in eq. (1.14) and conclude that

$$p(\lambda_m) = c_m^2 \tag{1.18}$$

which is simply the coefficient of the decomposition of $|\psi\rangle$ associated to the eigenvector $|\psi_m\rangle$. The state of the system is said to be *collapsed* into the eigenvector $|\psi_m\rangle$.

Composite systems

Once providing a complete description of the postulates of quantum mechanics regarding single systems, it is natural to move a step forward and ask ourselves what is the proper way to describe quantum systems which are made out of two or more subsystems. In particular, we would like to focus on how the state space of the composite system is built up starting from the state spaces of its components. The very last postulate presented in this overview states the following:

Postulate. *the state space of a composite system is the tensor product of the state spaces of the component physical systems. Furthermore, if the components are labeled by* $i = 1...n$ *and the i-th subsystem is placed in the state* $|\psi_i\rangle$*, the overall state can be written as*

$$|\psi_{tot}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ... \otimes |\psi_n\rangle. \tag{1.19}$$

It is worth to highlight two crucial inferences emerging from this description:

- The dimension of the Hilbert space of a composite system is given by the product of the dimensions of the Hilbert spaces of the components. As a consequence, $dimH$ scales exponentially in the number of subsystems. For instance, having N physical systems whose state space dimensionality is M each, results in the total dimension of the whole Hilbert space to be M^N;
- The expression in eq. (1.19) represent a general way to write the state of a composite system. This equation holds if and only if for each subsystem one can identify a "sub-state", previously called $|\psi_i\rangle$. In this case the state of the whole system is said to be a *product state*. Whether the aforementioned condition cannot be satisfied, it is impossible to express the overall state as a product state. The composite system is, hence, in an *entangled state*. Entanglement plays a crucial role for what the design of quantum algorithms is concerned as we'll see later on in this chapter.

1.2 The Qubit

The basic unit of classical computation and information is the *bit*, which can assume two possible values, namely 0 and 1. Analogously, quantum computation theory is built upon the quantum counterpart of the classical bit, called the *qubit (quantum bit)*. There is an enormous difference between the bit and the qubit since the latter is a mathematical object which obeys the laws of quantum mechanics.

1.2.1 Single qubit states

We can identify a state space of the qubit, consisting in a two-dimensional Hilbert space, and define an orthonormal basis on it, formed by the states denoted as $|0\rangle$ and $|1\rangle$. Hence, the state of a qubit can actually be any linear superposition of the basis states and be written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.20}$$

where the coefficients α and β are, in principle, complex numbers. The set $\{|0\rangle, |1\rangle\}$ is often referred to as the *computational basis*. In addition, since the vector state should be normalized to length 1, the coefficients must satisfy the following constraint:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.21}$$

Another useful way to denote the computational basis states is their vector expression

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1.22}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.23}$$

Quantum processors are capable to measure a single observable on a qubit, specifically the Pauli operator σ^z whose matrix form is

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.24}$$

and its eigenvectors are exactly the states $|0\rangle$ and $|1\rangle$. As a consequence, the qubit state collapses onto one of the computational basis state after a measurement and the probability of collapse on either $|0\rangle$ or $|1\rangle$ is given by the squared modulus of the coefficients α and β, respectively.

Because $|\alpha|^2 + |\beta|^2 = 1$, we may rewrite eq. (1.20) in the following form:

$$|\psi\rangle = e^{i\gamma} \left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi} \sin\frac{\theta}{2}|1\rangle \right) \tag{1.25}$$

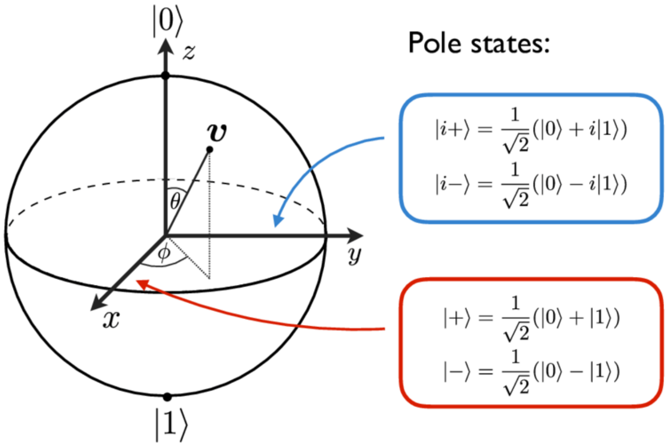

$|0\rangle$
z
$\boldsymbol{v}$
θ
ϕ
y
x
$|1\rangle$
Pole states:
$|i+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$
$|i-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$
$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Figure 1.1: Bloch sphere representing the state of a qubit. The pole states are the ones belonging to the computational basis, namely $|0\rangle$ and $|1\rangle$. The state $|\psi\rangle$ (called v here) is a vector of unit length on this sphere. The angles θ and φ in eq. (1.26) are also displayed. Moreover, the states $|+\rangle, |-\rangle, |i+\rangle$ and $|i-\rangle$ are highlighted as they are of massive interest being the eigenvectors of the Pauli matrices σ^x and σ^y

and, since $e^{i\gamma}$ is an overall phase, for what we have said in the previous section, we can drop it and the effective state of the qubit results in

$$|\psi\rangle = \left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right). \tag{1.26}$$

The last expression suggest a very neat way to show the quantum state of a qubit by displaying it on the so called *Bloch sphere* (fig. 1.1). The angles θ and φ define a point on the unit radius sphere which represents the state of the qubit. The poles of the sphere are the computational basis states. Furthermore, fig. 1.1 emphasizes the four pole states on the x and y dimensions, namely $|+\rangle, |-\rangle, |i+\rangle$ and $|i-\rangle$. As we'll show in the next section, they turn out to be the eigenvectors of the Pauli operators σ^x and σ^y, whose matrix expression is, for the sake of completeness, given by:

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.27}$$

$$\sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{1.28}$$

Unfortunately, despite the Bloch sphere representation is very simple and clean for a single qubit state, it does not exist a generalization to multiple qubit states which preserves this level of clarity.

1.2.2 Multiple qubit states

As mentioned in the previous section, while discussing about the exponential scaling of the dimension of the Hilbert space of a composite system, the basis states of the whole Hilbert space are built up as the tensor product between any possible n-tuple of the basis elements of the subsystems' Hilbert spaces. For an instance system of two qubits, we denote the basis states as $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, and the dimension of the Hilbert space is indeed $2^2 = 4$. Generalizing to an N qubits system, the Hilbert space dimensionality is 2^N and the computational basis is then $\{|00...0\rangle = |0\rangle^{\otimes N}, ..., |11...1\rangle = |1\rangle^{\otimes N}\}$. It is straightforward to notice that if the qubits were approximately a few hundreds, the number of basis states would be surprisingly larger than the estimated number of atoms in the observable universe. This should be taken as a sort of intimation of the computational power we would be endowed if we potentially could have access to an ideal quantum processor.

Now let's come back to our two qubit system. The general form of its state is the following:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{1.29}$$

where the usual normalization condition on the coefficients is applied.

For what measurements is concerned, because this is a multi-qubit system, we may decide to measure either all the qubits or just a subset of them. In the former case, the system is going to collapse on a computational basis state, in analogy with the single qubit case discussed previously. On the contrary, if we decide to measure only one of the qubits, say the first, we may get 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$ and the post-measurement state would be

$$|\psi\rangle_{PM} = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|}}, \tag{1.30}$$

or we may get 1 with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$ leaving the post-measurement state

$$|\psi\rangle_{PM} = \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|}}. \tag{1.31}$$

Separable states vs entangled states

As discussed at the very end of section 1.1, quantum states can be divided into *separable (or product)* states and *entangled* state. Let's apply this distinction to the case of qubits. Separable states can be always written as a tensor product between single qubit states. For instance the balanced superposition of the basis states, i.e. the state

$$\frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) \tag{1.32}$$

is separable since it can be rewritten as

$$\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)_1 \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)_2 = |+\rangle_1 \otimes |+\rangle_2. \tag{1.33}$$

The subscript indicates the qubit the state refers to. In contrast, the following state is entangled since there is no way to write it as a tensor product:

$$|\psi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{1.34}$$

This is one of the four Bell states. The other three are shown below:

$$\begin{aligned} |\psi^-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ |\phi^+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |\phi^-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \end{aligned} \tag{1.35}$$

Other examples of entangled states are the $|GHZ\rangle$, whose expression is:

$$|GHZ\rangle = \frac{1}{\sqrt{2}} \left(|000\rangle + |111\rangle\right), \tag{1.36}$$

and the $|W\rangle$ state which is:

$$|W\rangle = \frac{1}{\sqrt{3}} \left(|100\rangle + |010\rangle + |001\rangle\right). \tag{1.37}$$

The two states above apparently look similar since it is not hard to recognize that they're both entangled state. Nonetheless, they carry a different amount of entanglement. In fact, by measuring only one of the qubit of the $|GHZ\rangle$ state, the entanglement content of the state is immediately lost since the system collapses to either $|000\rangle$, with probability of 1/2, or $|111\rangle$, again with probability 1/2. In the latter state, instead, the measurement of one of the qubits, say the first for instance, does not eliminate entanglement with probability of 2/3. The possible post-measurement states are, in fact, $|100\rangle$ with probability of 1/3 and $|0\rangle \otimes \left(\frac{|01\rangle+|10\rangle}{\sqrt{2}}\right)$ with probability of 2/3. In the first case, entanglement is lost completely while, in the second case, the state of the second and third qubits is still entangled.

The main feature for which entangled states can be distinguished from separable ones is that measurement outcomes are correlated. Roughly speaking, if we measure the first qubit in computational basis and we get a certain outcome, we'll also know the outcome we would get if we measured the second one even without performing such a measurement. As an example, consider the Bell state $|\psi^+\rangle$. If we measure qubit number 1 and we get 0, which by the way occurs with a probability of a half, the post-measurement state will be $|00\rangle$ (which is, instead, separable), and the second qubit will also be in the state $|0\rangle$. On the contrary, if we got 1 the system would collapse to the state $|11\rangle$ and the second qubit will also be in $|1\rangle$. This fascinating characteristic of entangled states gave rise to the famous *EPR paradox* and it is the foundation of many quantum information protocols such as the *superdense coding* and the *quantum teleportation* [27].

1.3 Quantum gates

Once defined the basic unit of quantum computation, i.e. the qubit, it's time to delve into the construction of real quantum circuits and analyze the main elements which characterize them. As a classical circuit is based on bits, wires and logic gates acting on the bits themselves, a quantum circuit is based on qubits, wires and quantum logic gates.

1.3.1 Single qubit gates

Let's start our journey by listing the simplest gates a quantum circuit may be made out of, i.e. those gates which act on a single qubit.

X gate

The X gate is the quantum equivalent of a classical NOT and it is, in fact, also known as *quantum NOT* or *bit-flip*. It acts on a qubit by swapping the state $|0\rangle$ with the state $|1\rangle$ and viceversa. Its matrix form is:

$$\begin{aligned} X &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned} \tag{1.38}$$

The eigenstates of the X gate are the states $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ with eigenvalues $+1$ and -1, respectively.

Z gate

The Z gate, also known as *phase-flip*, adds a phase of $\theta = \pi$ only in front of the state $|1\rangle$ and its matrix form is shown below:

$$\begin{aligned} Z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ Z|0\rangle &= |0\rangle \\ Z|1\rangle &= -|1\rangle \end{aligned} \tag{1.39}$$

Its eigenvectors are the computational basis states $|0\rangle$ and $|1\rangle$ with eigenvalues $+1$ and -1, respectively, as one can straightforwardly notice from eq. (1.39).

Y gate

The Y gate is known as *bit-phase-flip* and can be seen as a sort of combination of the X and Z gates. This gate not only swaps the computational basis states but it also adds an imaginary phase. The matrix form of the Y gate is, in fact:

$$\begin{aligned} Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ Y|0\rangle &= i|1\rangle \\ Y|1\rangle &= -i|0\rangle \end{aligned} \tag{1.40}$$

The eigenstates of the Y gate are the states called $|i\pm\rangle$ in fig. 1.1 hence $|i+\rangle = \frac{|0\rangle+i|1\rangle}{\sqrt{2}}$ and $|i-\rangle = \frac{|0\rangle-i|1\rangle}{\sqrt{2}}$.

It is notable to remark that the X,Y and Z gates correspond to the Pauli matrices σ^x, σ^y and σ^z. As a consequence, they satisfy the well-known relations involving Pauli matrices, i.e.

$$\begin{aligned} \sigma_x^2 = \sigma_y^2 = \sigma_z^2 = -i\sigma_x\sigma_y\sigma_z &= \mathbb{I} \\ [\sigma_i, \sigma_j] &= 2i\epsilon_{ijk}\sigma_k \\ \{\sigma_i, \sigma_j\} &= 2\delta_{ij}\mathbb{I} \end{aligned} \tag{1.41}$$

Hadamard gate

The Hadamard gate, called H, plays a fundamental role in quantum algorithms since it allows to put a set of qubits in the balanced superposition state displayed in eq. (1.32) for two qubits. Before check how this is possible, let's look at the matrix form of this gate:

$$\begin{aligned} H &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ H|0\rangle &= |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ H|1\rangle &= |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{aligned} \tag{1.42}$$

The Hadamard transforms the state $|0\rangle$ into a superposition of $|0\rangle$ and $|1\rangle$ and the state $|1\rangle$ into an equivalent phase-flipped superposition. Viceversa,

$$\begin{aligned} H|+\rangle &= |0\rangle = \frac{|+\rangle + |-\rangle}{\sqrt{2}} \\ H|-\rangle &= |1\rangle = \frac{|+\rangle - |-\rangle}{\sqrt{2}}. \end{aligned} \tag{1.43}$$

This gate, which is therefore a change-of-basis operator, can also be written as $H = \frac{X+Z}{\sqrt{2}}$, squares to identity and its eigenvectors are

$$\begin{aligned} |H_+\rangle &= \cos\left(\frac{\pi}{8}\right)|0\rangle + \sin\left(\frac{\pi}{8}\right)|1\rangle \\ |H_-\rangle &= -\sin\left(\frac{\pi}{8}\right)|0\rangle + \cos\left(\frac{\pi}{8}\right)|1\rangle \end{aligned} \tag{1.44}$$

with eigenvalues ±1 for $|H_+\rangle$ and $|H_-\rangle$, respectively.
As mentioned above, only the Hadamard gate is needed to create a balanced superposition state. In fact, starting with N qubits in the product state $|0\rangle^{\otimes N} = |0...0\rangle$, and applying the Hadamard gate on each of them, we get

$$H^{\otimes N}|0\rangle^{\otimes N} = \frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} |x\rangle \tag{1.45}$$

where, in the summation, the state $|0\rangle = |00...00\rangle$ and the state $|2^N - 1\rangle = |11...11\rangle$, which is another useful way to label the computational basis states in which the bit string is replaced by the corresponding value in basis 10. As one can notice, the right-hand side of eq. (1.45) is the aforementioned superposition state.

S gate

The S gate adds an imaginary phase on the state $|1\rangle$. Its matrix form is:

$$
\begin{aligned}
S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \\
S|0\rangle = |0\rangle \\
S|1\rangle = i|1\rangle.
\end{aligned}
\tag{1.46}
$$

T gate

The T gate, instead, adds a phase of $\frac{\pi}{4}$ on the state $|1\rangle$. Its matrix form is:

$$
\begin{aligned}
T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \\
T|0\rangle = |0\rangle \\
T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle.
\end{aligned}
\tag{1.47}
$$

The T gate is also known as the $\pi/8$ gate since, besides an overall phase, can be rewritten as $\begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$. In addition, notice that $S = T^2$.

The phase angles $\pi/4$ for the T gate, $\pi/2$ for the S gate and π for Z, can be possibly replaced by a generic value φ.

Rotation gates

Starting from Pauli matrices, one can build up generalized single qubit rotations by exponentiating them therefore getting the following matrix expressions:

$$
\begin{aligned}
R_x(\theta) = e^{-iX\theta/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \\
R_y(\theta) = e^{-iY\theta/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \\
R_z(\theta) = e^{-iZ\theta/2} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}.
\end{aligned}
\tag{1.48}
$$

This operators correspond to rotations about the $\hat{x}, \hat{y}$ and $\hat{z}$ axis of the Bloch sphere, respectively.

A generalization of rotation gates, which consists in a rotation about any given axis $\hat{n}$, has the following form:

$$R_{\hat{n}}(\theta) = e^{-i\hat{n}\cdot\vec{\sigma}/2} = \cos\left(\frac{\theta}{2}\right)\mathbb{I} - i\sin\left(\frac{\theta}{2}\right)(n_x X + n_y Y + n_z Z) \tag{1.49}$$

where $\hat{n} = (n_x, n_y, n_z)$ denotes the vector of unit norm along the rotation axis and $\vec{\sigma} = (\sigma^x, \sigma^y, \sigma^z)$ is the vector whose components are the Pauli matrices.

Z-Y decomposition for a single qubit

In the very last section before the summary, we show how a general unitary 2×2 matrix, i.e. an operator acting on a single qubit, can be decomposed as a series of rotations. Given a unitary U, there exist four angles, say, $(\alpha, \beta, \gamma, \delta)$ such that

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{1.50}$$

Because of its unitarity, the row and column vectors of U are orthonormal. Hence, there exist real numbers so that

$$U = \begin{pmatrix} e^{i(\alpha-\beta/2-\delta/2)}\cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)}\sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)}\sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)}\cos\frac{\gamma}{2} \end{pmatrix}. \tag{1.51}$$

From the usual matrix multiplication rules and the definition of rotation matrices, eq. (1.50) follows from the equation above. An analogous decomposition can be recovered involving $X - Y$ or $Z - X$ rotations as well as using two general non-parallel rotation axis.

Summary of single qubit gates

In fig. 1.2 a summary of the main single qubit gates mentioned in the previous paragraphs is presented together with their corresponding symbol and matrix expressions.

1.3.2 Two qubits gates

Let's walk a step forward and pay our attention to more complex gates acting on two input qubits. The typical two qubits gate is a *controlled operation*, which is defined as the application of a single qubit unitary on the so called *target qubit*, say the second one, if and only if the first qubit, known as *control qubit*, is in the state $|1\rangle$. Otherwise, if the control qubit is in the state $|0\rangle$, the target is left invariant. Controlled operations are fundamental in quantum computation since they can be exploited to create entangled states. The general operatorial form of two qubits gates consists in a 4×4 matrix since the two qubits Hilbert space has dimension 4.

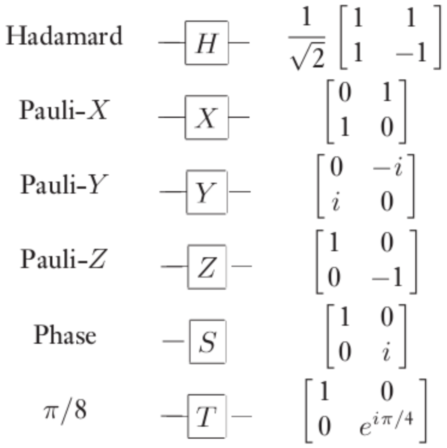

Hadamard
H
$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Pauli-X
X
$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Pauli-Y
Y
$\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$
Pauli-Z
Z
$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$
Phase
S
$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
$\pi/8$
T
$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$

Figure 1.2: Symbols of the main single qubit gates and their associated matrix expression [27]. From top to bottom: Hadamard, X, Y, Z, S and T gates

Controlled-NOT

The controlled-NOT, or CNOT for brevity, is arguably the most important controlled operation in quantum computation. Given the control qubit in the state $|c\rangle$ and the target one in the state $|t\rangle$, it performs the following operation: $|c\rangle|t\rangle \rightarrow |c\rangle|t \oplus c\rangle$. In other words, the CNOT flips the target, i.e. applies an X gate on it, provided that the control is in the state $|1\rangle$.
The matrix form of the CNOT in computational basis is:

$$CNOT = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

$$\begin{aligned} CNOT|00\rangle &= |00\rangle\\ CNOT|01\rangle &= |01\rangle\\ CNOT|10\rangle &= |11\rangle\\ CNOT|11\rangle &= |10\rangle. \end{aligned} \tag{1.52}$$

The symbol of the CNOT gate is displayed in fig. 1.3.

Controlled-U

In principle, the X gate in the CNOT can be replaced by any unitary at will. A generalized controlled-unitary works as following: $|c\rangle|t\rangle \rightarrow |c\rangle U^c|t\rangle$, where $U^0 = \mathbb{I}$ and $U^1 = U$. The symbol of the controlled-unitary, or CU for brevity, is shown in fig. 1.4.

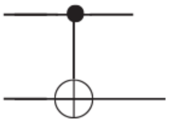


Figure 1.3: Symbol of the CNOT gate [27]. The upper line corresponds to the control qubit while the lower line represents the target

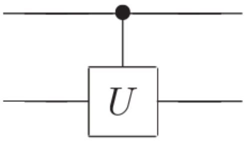


Figure 1.4: Symbol of the general controlled-unitary gate [27]. Similarly for the CNOT, the upper line corresponds to the control qubit whereas the lower one represents the target

Swap gate

Unlike the gates discussed above, the swap gate is not a controlled unitary. It is, instead, a two qubits gate which swaps the states of the qubits. It can be built up by applying three consecutive CNOTs as shown in fig. 1.5, where the control and the target qubits are alternated.
Given two qubits in the product state $|\psi_1\rangle|\psi_2\rangle$, the swap gate acts like that: $SWAP|\psi_1\rangle|\psi_2\rangle = |\psi_2\rangle|\psi_1\rangle$.

Two qubits entangling circuit

In this final paragraph related to two qubits gates we would like to show how is it possible to create an entangled state by means of the CNOT gate, as announced previously. For the sake of simplicity, we'll consider a two qubits register. The procedure to entangle two qubits is the following:

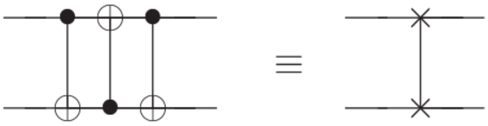


Figure 1.5: Swap gate symbol (right-hand side) [27]. On the left, three consecutive CNOTs, in which the target and control are alternated, are used to construct a swap gate

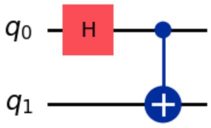


Figure 1.6: Entangling circuit for two qubits. The qubits, called q_0 and q_1, should be initialized in a product state such as $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. After applying an Hadamard on the first qubit and a CNOT, the final state will be one of the four Bell states depending on which initialization has been chosen

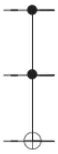


Figure 1.7: Symbol of the Toffoli gate [27]. The first two lines represent the controls while the latter is the target

- Initialize the two qubits in the product state $|00\rangle = |0\rangle \otimes |0\rangle$;
- Apply the Hadamard gate on the first one;
- Apply the CNOT gate where the first qubit is the control and the second is the target;
- The state of the system at the end of this trivial algorithm is the first Bell state $|\psi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$;
- If the qubits are initialized in $|01\rangle, |10\rangle$ or $|11\rangle$, the final state will be $|\psi^-\rangle, |\phi^+\rangle$ or $|\phi^-\rangle$, respectively.

A picture of the entangling circuit is shown in fig. 1.6.

1.3.3 Notable multiple qubits gates

Toffoli gate

The Toffoli gate is a doubly controlled NOT, acting on three qubits where two of them are the controls and one is the target. The X gate (NOT) is performed on the target if and only if both the controls are in the state $|1\rangle$, otherwise the target is left alone. Its symbol is displayed in fig. 1.7.

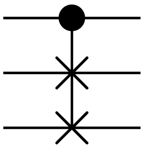


Figure 1.8: Symbol of the Fredkin (controlled swap) gate. The first line is the control qubits while the two lower lines represent the targets on which the swap operates

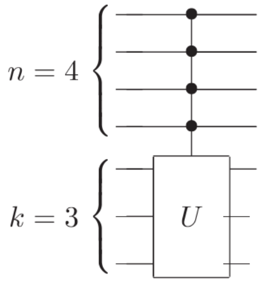


Figure 1.9: Example of a unitary gate acting on $k = 3$ qubits, controlled by $n = 4$ additional qubits. The unitary transformation is performed only if all the controls are in the state $|1\rangle$ [27]

Fredkin gate

The Fredkin gate is a controlled swap. It acts on three qubits where the first is the control while the second and the third are the targets. Fig. 1.8 shows its symbol.

Multi-qubit unitary with multiple controls

The major possible generalization of a quantum gate is a unitary acting on a subset of the quantum register formed by k qubits controlled by n additional qubits. The transformation encoded in the unitary U is performed on the targets if and only if all the controls are in the state $|1\rangle$. An illustrative exmaple of such a general controlled gate is shown in fig. 1.9.

1.4 Measurements

On typical quantum circuit instances, a measurement of at least one of the qubits in performed in order to get information about some quantum states. The only basis in which a real quantum processor can actually perform a measurement is the computational basis i.e. the set $\{|0\rangle, |1\rangle\}$, also known as the Z-basis, since those states are the eigenvectors of the Z operator. Nevertheless, in principle, a measurement on any basis

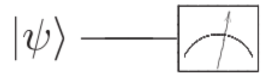


Figure 1.10: Symbol of the measurement process on a qubit [27]

of the Hilbert space whatsoever can be performed. The symbol which the measurement procedure is depicted with, is shown in fig. 1.10. Moreover, the outcome of the measurement is usually depicted, instead, with a double line since after the state collapse, the qubit resembles a classical bit.
Two very important principles regarding measurements in quantum circuits are worth to be emphasized. They are the *principle of deferred measurements* and the *principle of implicit measurements.*

Principle of deferred measurements

Measurements at an intermediate stage of a quantum circuit can always be moved to the end of the circuit. The classical controlled operations which exploit the outcomes of intermediate quantum measurements can be, in this case, replaced by conditional quantum operations.
Roughly speaking, taking CNOTs as an example since they are the most widely used controlled gates, we can swap them with measurements without losing any information about quantum states or affecting the final result of the circuit. An illustrative example of how the principle of deferred measurement works is depicted in fig. 1.11. Here, the equivalence between two circuits is proven. The circuit on the left-hand side shows the creation of a Bell pair followed by a measurement. From what has been said hitherto concerning the evaluation of the outcomes probabilities, we may get the post-measurement state $|00\rangle$ or $|11\rangle$ with probability 1/2 each provided that the qubits have been initialized in $|00\rangle$. In the circuit on the right-hand side, instead, the qubits are measured before applying the CNOT gate. Let's compute how the state changes at each step of the circuit.

- Suppose qubits are initialized in $|00\rangle$. After applying the Hadamard gate on the first one we get the state $|+\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$;
- The two qubits are measured: the second one will be obviously found in the state $|0\rangle$ while the first one may collapse into either $|0\rangle$ or $|1\rangle$, each with probability 1/2. Therefore, the post-measurement state will be $|00\rangle$ with a half probability or $|10\rangle$ with the same probability;
- The CNOT is then performed. If the input state is $|00\rangle$ nothing changes. On the contrary, if the input is $|10\rangle$ we get the state $|11\rangle$. Both these scenarios may occur with probability of 1/2.

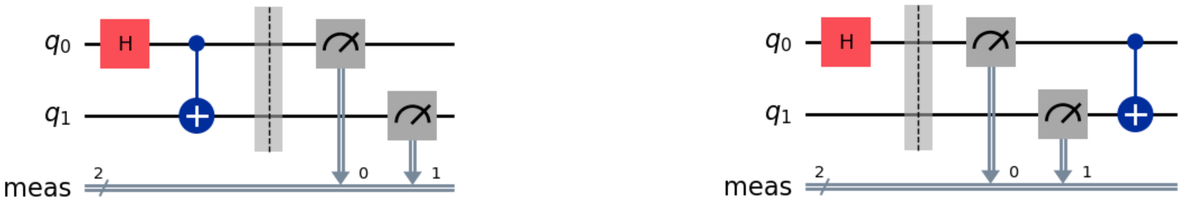


Figure 1.11: Simple exemplifying case of the meaning of the principle of deferred measurements. The two circuits are exactly equivalent. The one on the left builds up a Bell state, say the first, and measure both the qubits at the very end. We know that the possible outcome are $|00\rangle$ and $|11\rangle$ both with probability of a half. The circuit on the right, in contrast, performs the measurement before applying the CNOT gate but that does not influence what we are going to find at the end of the circuit. Again the states $|00\rangle$ and $|11\rangle$ are possible, with 1/2 probability each

The possible quantum states at the end of the circuits and their associated occurrence probabilities are, hence, equal.

Principle of implicit measurements

Without loss of generality, any qubit which is not measured in a quantum circuit may be assumed to be measured.
This principle is stating that the statistics of the measurement outcomes of a qubit is not affected by the action of measurement on the other qubits. In other words, the distribution of the outcomes is completely determined by what is called the density matrix of that qubit itself.

1.4.1 Measurements in a basis different from the computational basis

As mentioned at the beginning of this section, it may be remarkable, for pure theoretical reasons, to illustrate how to provide the possible outcomes of a measurement which is performed in a basis different from the computational basis:

- First of all we must select an hermitian operator whose eigenvectors represent the basis in which we wish to perform our measurement. Call these two states $\{|\psi_1\rangle, |\psi_2\rangle\}$;
- Call $|\psi\rangle$ the qubit state. Rewrite $|\psi\rangle$ as a linear combination of the basis elements, i.e. $|\psi\rangle = \alpha|\psi_1\rangle + \beta|\psi_2\rangle$;
- The post-measurement state will be one of the two basis elements, with associated probability of $|\alpha|^2$ for the state $|\psi_1\rangle$ and $|\beta|^2$ for the state $|\psi_2\rangle$.

It is fundamental to point out that any measurement in a given orthonormal basis can be reduced to a measurement in the computational basis by properly applying a unitary change-of-basis operator U on the qubits just before the measurement. Let's prove it in a generalized form:

- Given two orthonormal basis, say $B_1 = \{|\psi_1\rangle, |\psi_2\rangle\}$ and $B_2 = \{|\phi_1\rangle, |\phi_2\rangle\}$, there always exists a unitary transformation which acts on the basis elements in the following way:

$$\begin{aligned} U|\psi_1\rangle &= |\phi_1\rangle \\ U|\psi_2\rangle &= |\phi_2\rangle \\ U^\dagger|\phi_1\rangle &= |\psi_1\rangle \\ U^\dagger|\phi_2\rangle &= |\psi_2\rangle; \end{aligned} \tag{1.53}$$

- Call $|\psi\rangle$ the state to be measured in the basis B_1 but, unfortunately, our device can only perform measurements in the basis B_2, and suppose we know the expression of $|\psi\rangle$ in terms of the basis elements of B_1, say $|\psi\rangle = \alpha|\psi_1\rangle + \beta|\psi_2\rangle$;

- The outcomes of a measurement in the basis B_1 may give $(|\psi_1\rangle, |\psi_2\rangle)$ with probability $(|\alpha|^2, |\beta|^2)$. Apply the unitary U to the state $|\psi\rangle$:

$$\begin{aligned} U|\psi\rangle = \alpha U|\psi_1\rangle + \beta U|\psi_2\rangle &= \\ = \alpha|\phi_1\rangle + \beta|\phi_2\rangle&; \end{aligned} \tag{1.54}$$

- Now perform the measurement in the basis B_2. The state will collapse to $(|\phi_1\rangle, |\phi_2\rangle)$ with probabilities $(|\alpha|^2, |\beta|^2)$;

- Apply the inverse unitary $U^\dagger$ to bring the post-measurement state back to the B_1 basis.

This is the exact procedure we must perform on current quantum devices when we're interested in measuring qubits in different basis.

1.4.2 Reconstruction of quantum states

We have hitherto highlighted dozens of times the intrinsic probabilistic nature of quantum measurements. In a typical situation, at the end of a quantum circuit, we need to get information about the quantum state of the system. The only way we can have access to it is via measurements. Since normally the final state of the qubits is a superposition of the basis elements including entangled states, we need to reproduce the quantum circuit many times, in each performing a measurement until we get enough data to do statistics and reconstruct the whole quantum state. The number of measurement needed to recover the final state depends on two factors: the tolerance on the amplitudes, i.e. the maximum difference between the real value of the coefficients in front of the basis elements in the superposition and the same value recovered through the measurements, and the complexity of the quantum state, i.e. the number of basis elements with non-zero amplitude. A standard instance of problem where the final reconstruction of the quantum state may be quite difficult are quantum simulations. A brief definition of quantum simulations is that of algorithms devised to reproduce the behavior of a quantum system and possibly find the ground state of that system which evolve according to a given hamiltonian. We'll look at quantum simulations in detail later on. The point which is worth to remark is that quantum systems might have very complex ground states and the amount of measurements required to recover them may easily increase to the order of 10^4 or even more. As a consequence, the total computational time is considerably affected by this, let's say, "inconvenient". The repetition of the quantum circuit has a significant cost in terms of computational resources and this represent the price to pay, the drawback, of quantum computing.

1.5 Universality of quantum gates

A set of gate is said to be *universal* if any unitary can be approximated with arbitrary accuracy by a quantum circuit involving only those gates. What we'll prove in this section is the universality of the set containing all single qubit gates and the CNOT. The final statement is therefore that *any unitary operation can be exactly decomposed as a series of single qubit rotations and CNOTs* [27]. Furthermore, an approximation of a general unitary gate can be obtained with only the Hadamard, the CNOT and the T gate.

1.5.1 Two-level gates are universal

The first step toward the proof of universality of single qubit gates and CNOTs consists in proving that two-level gates are universal. Two-level matrices are defined as operators which act non-trivially only on two or fewer vector components.

Consider a unitary U which acts on a d-dimensional Hilbert space. This matrix U can be decomposed into a product of two-level matrices. To prove this, let's start from the simple case where U is 3×3:

$$U = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & j \end{bmatrix} \tag{1.55}$$

We'll find two-level operators U_1, U_2, U_3 such that

$$U_3 U_2 U_1 U = \mathbb{I} \tag{1.56}$$

from which

$$U = U_1^\dagger U_2^\dagger U_3^\dagger. \tag{1.57}$$

Now, let's see how to construct the three two-level unitaries such that the product of their hermitian conjugated, which are still two-level matrices, gives U:

1. If $b = 0$ set U_1 as:

$$U_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \tag{1.58}$$

   Otherwise, if $b \neq 0$, set U_1 to:

$$U_1 = \begin{bmatrix} \frac{a^*}{\sqrt{a^2+b^2}} & \frac{b^*}{\sqrt{a^2+b^2}} & 0 \\ \frac{b}{\sqrt{a^2+b^2}} & \frac{-a}{\sqrt{a^2+b^2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.59}$$

   Note that this two matrices satisfy the definition of two-level matrices given at the beginning of the section since they act non-trivially at most on two vector components. When we multiply $U_1 U$ we get:

$$U_1 U = \begin{bmatrix} a' & d' & g' \\ 0 & e' & h' \\ c' & f' & j' \end{bmatrix}. \tag{1.60}$$

2. If $c' = 0$ set U_2 as:

$$U_2 = \begin{bmatrix} a'^* & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \tag{1.61}$$

   If, conversely, $c' \neq 0$ set U_2 as:

$$U_2 = \begin{bmatrix} \frac{a'^*}{\sqrt{a'^2+c'^2}} & 0 & \frac{c'^*}{\sqrt{a'^2+c'^2}} \\ 0 & 1 & 0 \\ \frac{c'}{\sqrt{a'^2+c'^2}} & 0 & \frac{-a'}{\sqrt{a'^2+c'^2}} \end{bmatrix}. \tag{1.62}$$

   From matrix multiplication we get:

$$U_2 U_1 U = \begin{bmatrix} 1 & d'' & g'' \\ 0 & e'' & h'' \\ 0 & f'' & j'' \end{bmatrix}. \tag{1.63}$$

   Since U_2, U_1 and U are unitary, it follows that $d'' = g'' = 0$ cause the first row vector must have unit norm.

3. Set then U_3 as:
$$U_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & e''^{*} & h''^{*} \\ 0 & f''^{*} & j''^{*} \end{bmatrix}. \tag{1.64}$$
It is straightforward to check the validity of eq. (1.56).

Generalizing to a unitary matrix acting on a d-dimensional Hilbert space, this procedure allows one to find $d-1$ unitary matrices $U_1, ..., U_{d-1}$ such that the matrix $U_{d-1}...U_1U$ has a one in the top left hand corner and zeros elsewhere in the first row and column. The procedure is repeated for the $d-1 \times d-1$ unitary submatrix in the bottom right corner of $U_{d-1}...U_1U$ until 2×2 matrices are reached. This leads to a $d \times d$ unitary which can be written as:
$$U = V_1...V_k \tag{1.65}$$
where the V_i matrices are two-level unitaries and $k = d(d-1)/2$.

1.5.2 Single qubit and CNOT gates are universal

The next step is to show that single qubit gates together with CNOTs can be used to implement any arbitrary two-level operation and therefore, for what we have seen in the previous section, they constitute a set of universal gates for quantum computation.
Suppose U is a $n \times n$ two-level matrix acting non-trivially on the subspace spanned by the vectors $|s\rangle = |s_1...s_n\rangle$ and $|t\rangle = |t_1...t_n\rangle$ where $s_1...s_n$ and $t_1...t_n$ are the binary expansions of s and t, resepctively. Call $\tilde{U}$ the 2×2 non-trivial submatrix of U which can be considered as a single qubit gate. We wish to build up a circuit which implements U made out of only CNOTs and $\tilde{U}$ operations. To achieve our goal we need to invoke the *Gray code*. The Gray code connecting s and t is defined as a sequence of binary numbers starting from s and ending with t such that two adjacent members of the list differ in exactly one bit. For the sake of clarity, suppose $s = 101001$ and $t = 110011$. A Gray code from s to t is given by:
$$\begin{matrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{matrix}. \tag{1.66}$$

Let $g_1, ..., g_m$ be the elements of a Gray code connecting $g_1 = s$ and $g_m = t$. Notice that $m \leq n+1$ since s and t, being bit strings of length n, can differ at most in n locations. In order to implement U we should devise a quantum circuit which basically changes the states $|g_1\rangle \rightarrow |g_2\rangle \rightarrow ... \rightarrow |g_m\rangle$ and performs a controlled-$\tilde{U}$ operation with the target qubit located at the single bit where $|g_{m-1}\rangle$ and $|g_m\rangle$ differ. A detailed description of the algorithm is provided below:

1. First of all, we need to swap $|g_1\rangle$ and $|g_2\rangle$. Since the corresponding bit strings differ in one single digit, we can apply a X gate on the qubit where the discrepancy is located. Then a controlled operation is used to swap $|g_2\rangle$ with $|g_3\rangle$. Keep on going this way until the state $|g_{m-1}\rangle$ is reached;
2. Suppose $|g_{m-1}\rangle$ and $|g_m\rangle$ differ in the j-th qubit. We hence apply a controlled-$\tilde{U}$ with the j-th qubit as target conditional on the other qubits having the same value as appear in both $|g_{m-1}\rangle$ and $|g_m\rangle$;
3. Undo the swap operations, bringing back the state $|g_{m-1}\rangle$ to $|g_1\rangle$.

To see how the circuit is built in practice, consider a simple example with three qubits. Suppose we wish to decompose the following two-level unitary:

$$U = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & b \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ c & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix} \tag{1.67}$$

where $\tilde{U}$ is then the 2×2 matrix $\tilde{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. The unitary U acts non-trivially on the basis states $|000\rangle$ and $|111\rangle$. The Gray code conecting 000 with 111 is the following:

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}. \tag{1.68}$$

The required circuit to perform U is depicted in fig. 1.12. As first step, it brings the string 000 to the string 011, i.e. the second to last string of the Gray code above. Then a controlled-$\tilde{U}$ is performed on the first bit, which is the one where the second to last and the last bit strings of the Gray code differ, and finally the states are swapped back from 011 to 000.

1.5.3 Approximating unitaries with a discrete set of operators

The result we've got hitherto may look completely satisfactory, since we have proved that any unitary can be reduced to a sequence of CNOTs and single qubit rotations. Nevertheless, the set of single qubit gates is continuous and, for practical purposes, we

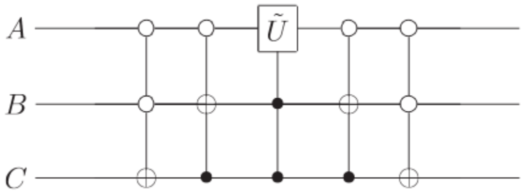


Figure 1.12: Quantum circuit performing U in eq. (1.67) [27]. Controlled NOTs are perfomed where the control qubits are the one which do not undergo any change in each step. The white circles indicate that the control qubit should be in the state $|0\rangle$ (not $|1\rangle$) to enable the X gate on the target. When the states have been swapped, a controlled-$\tilde{U}$ is performed on the qubit where the string 011 and 111 differ, i.e. the first one, and the states are then swapped back

need to find a way to approximate any single qubit rotations with a discrete set of gates in such a way that we can effectively build exploitable quantum circuits.
The very last step of this discussion regards proving that any single qubit rotation can be actually approximated within an arbitrarily small tolerance by a set of gates including only the Hadamard, the CNOT and the T gates.

Approximating unitary operators

Consider two unitaries U and V. The former is the target transformation we wish to implement while the latter is the unitary supposed to approximate the unitary U we can implement in practice. We define the error when V is implemented instead of U as:

$$E(U,V) = \max_{|\psi\rangle} ||(U-V)|\psi\rangle|| \tag{1.69}$$

where $|\psi\rangle$ is a generic state of the Hilbert space. If the error $E(U,V)$ is small, the measurement statistics got by implementing V rather than U is approximately the same. In fact, call M an operator belonging to a measurement set and call P_U and P_V the probability of obtaining the associated measurement outcomes if the operator U or V is applied, respectively. Therefore,

$$|P_U - P_V| = |\langle\psi|U^\dagger M U|\psi\rangle - \langle\psi|V^\dagger M V|\psi\rangle|. \tag{1.70}$$

Using the Cauchy-Schwarz inequality it is straightforward to conclude that

$$|P_U - P_V| \leq 2E(U,V). \tag{1.71}$$

In the general case we may apply a set $\{V_1, V_2, ..., V_m\}$ of gates intended to approximate the ideal set $\{U_1, U_2, ..., U_m\}$. In this case, the error does not increase faster than linearly in the number of gates m, i.e.

$$E(U_m...U_1, V_m...V_1) \leq \sum_{j=1}^{m} E(U_j, V_j). \tag{1.72}$$

To prove this let's start from the simple case where $m = 2$.

$$\begin{aligned} E(U_2U_1, V_2V_1) = ||(U_2U_1 - V_2V_1)|\psi\rangle|| = \\ = ||(U_2U_1 - V_2U_1)|\psi\rangle + (V_2U_1 - V_2V_1)|\psi\rangle||. \end{aligned} \tag{1.73}$$

From the triangle relation,

$$E(U_2U_1, V_2V_1) \leq E(U_2, V_2) + E(U_1, V_1). \tag{1.74}$$

The general proof is obtained by induction.
This results are extremely useful. In fact, suppose we wish to get the same measurement outcomes probabilities within a tolerance Δ by approximating a sequence of operators $U_1, ..., U_m$ with $V_1, ..., V_m$. It suffices that $E(U_j, V_j) \leq \frac{\Delta}{2m}$ to get the desired result.

Universality of Hadamard, CNOT and T gates

The Hadamard and the T gates can be used to approximate any single qubit operation. The T gate is, up to an overall phase, a rotation about $\pi/4$ over the $\hat{z}$ axis. The composite gate HTH is, instead, a rotation of the same angle around the $\hat{x}$ axis on the Bloch sphere. Composing the T alone with the HTH operator gives the general transformation

$$\begin{aligned} e^{-i\frac{\pi}{8}Z}e^{-i\frac{\pi}{8}X} = \\ = \cos^2\frac{\pi}{8}\mathbb{I} - i\left(\cos\frac{\pi}{8}(X+Z) + \sin\frac{\pi}{8}Y\right)\sin\frac{\pi}{8}, \end{aligned} \tag{1.75}$$

which represent a rotation about the axis with unit vector $\hat{n} = \frac{1}{N}\left(\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8}\right)$, where N is just the vector normalization, of an angle θ such that $\cos\frac{\theta}{2} = \cos^2\frac{\pi}{8}$.
Repeated iterations of $R_{\hat{n}}(\theta)$ rotations can be used to approximate a generic rotation $R_{\hat{n}}(\alpha)$. Let $\delta > 0$ be the accuracy we desire to reach and let N be an integer larger than $2\pi/\delta$. Now define $\theta_k = (k\theta) \mod 2\pi$ where each $\theta_k \in [0, 2\pi)$. There are two distinct $j, k \in [1, N]$ such that $|\theta_k - \theta_j| < \delta$. Assume for the sake of simplicity, without loss of generality, that $k > j$. Therefore the sequence $\theta_{l(k-j)}$ fills up the interval $[0, 2\pi)$ as l is varied in such a way that each term within the sequence is no more than δ apart from its neighbor terms. It follows that for any ϵ there exists an n so that

$$E(R_{\hat{n}}(\alpha), R_{\hat{n}}(\theta)^n) < \frac{\epsilon}{3}. \tag{1.76}$$

As already discussed so far in this chapter, every single qubit unitary transformation can be always written as a sequence of rotations in the following way:

$$U = R_{\hat{n}}(\beta)R_{\hat{m}}(\gamma)R_{\hat{n}}(\delta). \tag{1.77}$$

As a consequence of that and eq. (1.72), there exist three positive integers such that

$$E(U, R_{\hat{n}}(\theta)^{n_1}R_{\hat{n}}(\theta)^{n_2}R_{\hat{n}}(\theta)^{n_3}) < \epsilon. \tag{1.78}$$

This implies that, as we wished to prove, given a single qubit unitary U, it is always possible to approximate U with an arbitrary small accuracy called ϵ with a quantum circuit made out only of Hadamard and T gates.

Solovay-Kitaev theorem

One may wonder how efficiently the approximation of a single qubit unitary can be done in terms of the number of Hadamard and T gates. The *Solovay-Kitaev theorem* [6] provides us an answer to this non-trivial question. It states that:
Any single qubit unitary can be approximated within an accuracy ϵ using $O(\log^c(1/\epsilon))$ gates from our discrete set, where $c \simeq 2$.
This theorem therefore implies that in order to approximate a circuit containing m CNOTs and single qubit unitaries we need a number of gates of the order of $O(m \log^c(m/\epsilon))$, which is a polylogarithmic slope and it can be considered acceptable for any typical practical application.

This simple proof brings a very important result in quantum computation. The possibility to reduce any matrix, including evolution operators, to sequences of single qubit rotations and CNOTs drastically simplifies the construction of every quantum circuit, including quantum simulation circuits.

1.6 Quantum simulation algorithm

In this section we analyze how is it possible to construct a quantum circuit to simulate the evolution of a given system under a certain hamiltonian [27].
For what classical simulations is concerned, the typical problem asks to solve a differential equation e.g.

$$\frac{dy}{dt} = f(y) \tag{1.79}$$

to first order, where it's known that

$$y(t + \Delta t) \approx y(t) + f(y)\Delta t. \tag{1.80}$$

The quantum counterpart of this scenario consists in solving the Schrödinger's equation (1.6). The solution, for a time-independent hamiltonian H, is:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle. \tag{1.81}$$

Nevertheless, hamiltonians regarding typical many body systems are difficult to exponentiate entirely because of the exponential growth of its size in the number of qubits. As a consequence, one may settle for the first order approximation of the solution which is:

$$|\psi(t+\Delta t)\rangle = (\mathbb{I} - iH\Delta t)|\psi(t)\rangle. \tag{1.82}$$

The computation of first order solution is typically straightforward since it's not complicated to construct a quantum circuit to simulate $\mathbb{I} - iH\Delta t$. Nonetheless, the first order solution appears, in many cases, very unsatisfactory from a physical point of view. However, for many physical systems, the hamiltonian can be written as a sum over local interaction, i.e. two-body terms, or one-body terms. This represents a very typical scenario in quantum many body theory e.g. the Ising model [13]. Specifically, for a system of n particles,

$$H = \sum_{k=1}^{L} H_k \tag{1.83}$$

where each term H_k involves at most a constant number of qubits, say c, and L is a polynomial function of n. Now, since H_k gathers a smaller number of qubits it is much easier to approximate it with a quantum circuit. Nevertheless, we shall take into consideration that the terms H_k of the hamiltonian do often not commute, i.e. $[H_j, H_k] \neq 0$. The obvious consequence is that $e^{-iHt} \neq \prod_{k=1}^{L} e^{-iH_k t}$. However, though it might seem an unpleasant inconvenient, there is still a way to accomplish a good approximation of e^{-iHt}. We must exploit the *Trotter's formula* [34].

1.6.1 The Trotter's formula

Let A and B be hermitian operators. Then for any real time t,

$$\lim_{n\to\infty} \left(e^{iAt/n} e^{iBt/n}\right)^n = e^{i(A+B)t}. \tag{1.84}$$

Roughly speaking, the Trotter's formula is basically saying that the evolution under an hamiltonian H formed by two noncommuting terms A and B can be obtained by alternately evolving the system with A and B for a small time $\epsilon = t/N$ N times, until the total evolution time is reached. If N is finite but large enough, the discretization of the evolution will produce an approximation of the state we would get under the exact evolution of the original state under H.

Proof

The Taylor's approximation at first order of $e^{iAt/n}$ is

$$e^{iAt/n} = \mathbb{I} + \frac{1}{n} iAt + O\left(\frac{1}{n^2}\right). \tag{1.85}$$

Hence,

$$e^{iAt/n}e^{iBt/n} = \mathbb{I} + \frac{1}{n}(A+B)t + O\left(\frac{1}{n^2}\right). \tag{1.86}$$

Exponentiating the above equation we get

$$\left(e^{iAt/n}e^{iBt/n}\right)^n = \mathbb{I} + \sum_{k=1}^{n} \binom{n}{k} \frac{1}{n^k} \left[i(A+B)t\right]^k + O\left(\frac{1}{n}\right). \tag{1.87}$$

Now, since

$$\binom{n}{k} \frac{1}{n^k} = \left(1 + O\left(\frac{1}{n}\right)\right) / k!, \tag{1.88}$$

the original limit becomes

$$\lim_{n\to\infty} \left(e^{iAt/n}e^{iBt/n}\right)^n = \lim_{n\to\infty} \sum_{k=0}^{n} \frac{(i(A+B)t)^k}{k!} \left(1 + O\left(\frac{1}{n}\right)\right) + O\left(\frac{1}{n}\right) = e^{i(A+B)t}. \tag{1.89}$$

Further modifications of the Trotter's formula provide a way to compute the approximation at higher order by means of quantum simulations. Some instances are displayed below:

$$e^{i(A+B)\Delta t} = e^{iA\Delta t}e^{iB\Delta t} + O(\Delta t^2); \tag{1.90}$$

$$e^{i(A+B)\Delta t} = e^{i\frac{A}{2}\Delta t}e^{iB\Delta t}e^{i\frac{A}{2}\Delta t} + O(\Delta t^3); \tag{1.91}$$

$$e^{i(A+B)\Delta t} = e^{iA\Delta t}e^{iB\Delta t}e^{-\frac{1}{2}i[A,B]\Delta t^2} + O(\Delta t^3). \tag{1.92}$$

The last one is called the *Baker-Campbell-Hausdorf formula* [16].

1.6.2 The Algorithm

Inputs:

- An hamiltonian $H = \sum_k H_k$ acting on N qubits where each term H_k acts on a finite subsystem whose size is independent on N;
- An initial state $|\psi_0\rangle = |\psi(t_0)\rangle$ in which to place the system at the beginning of the simulation;

- A non-zero accuracy δ;
- A final time t_f at which the evolved state is desired.

Outputs:

- A state $|\tilde{\psi_f}\rangle = |\psi(t_f)\rangle$ satisfying $|\langle\tilde{\psi_f}|e^{iH(t_f-t_0)}|\psi_0\rangle|^2 \geq 1-\delta$

Procedure:

1. Initialize the system in the state $|\psi_0\rangle$, $j=0$;
2. Evolve the state $|\psi_j\rangle$ by applying $U_{\Delta t}$, $|\psi_{j+1}\rangle = U_{\Delta t}|\psi_j\rangle$. $U_{\Delta t}$ is the operator which approximates $e^{i\sum_k H_k \Delta t}$. The complexity of $U_{\Delta t}$ depends on both the accuracy δ and the approximation method, e.g. the Baker-Hausdorf formula;
3. Increase $j \leftarrow j+1$ and repeat step 2 until the final value j is reached, i.e. when $j\Delta t = t_f$;
4. The final result is the desired state $|\tilde{\psi_f}\rangle$. In order to recover it one should repeat the simulation performing a measurement at the end and do statistics on the outcomes as explained in sec. 1.4.2.

1.6.3 Illustrative example

Here we provide a simple illustrative example of the quantum simulation algorithm applied to the hamiltonian $H = Z_1 \otimes Z_2 \otimes Z_3$ acting on a three qubits system [27]. The simulation circuit is depicted in fig. 1.13. Let's see how the circuit works and how can it effectively reproduce the evolution of a system under the hamiltonian H:

1. Let's start with the system in the initial state $|q_1q_2q_3\rangle$ where each q_i can be either 0 or 1;
2. Add an ancilla qubit to the system and place it in the state $|0\rangle$ (in fig. 1.13 the ancilla is the last qubit). The four qubits state is now $|q_1q_2q_30\rangle$
3. Apply three CNOTs, each of them acting on a different control and the ancilla as target. The state now becomes $|q_1q_2q_3p\rangle$ where $p=0$ if the parity of the first three qubits in computational basis is even and $p=1$ otherwise;
4. Apply a phase shift $e^{-iZ\Delta t}$ on the ancilla where Δt is the evolution time. The state becomes $e^{\pm i\Delta t}|q_1q_2q_3p\rangle$ where if $p=0$ the exponent has negative sign and if $p=1$ it has, instead, positive sign;
5. Bring back the ancilla to $|0\rangle$ by applying again the same CNOTs as in 2;
6. The final state is $e^{\pm i\Delta t}|q_1q_2q_3\rangle \otimes |0\rangle$. The ancilla can therefore be excluded and the state of the three initial qubits has been then evolved under the hamiltonian H.

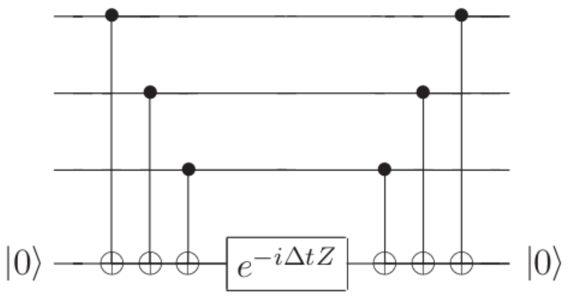


Figure 1.13: Simulation of the evolution of a three qubits system under the hamiltonian $H = Z_1 \otimes Z_2 \otimes Z_3$ [27]. The idea is to utilize an ancilla qubit, initialized in the state $|0\rangle$ on which a phase shift is applied. The phase shift depends on the parity of the qubits state in computational basis and it can be either $e^{-i\Delta t}$ (even parity) or $e^{i\Delta t}$ (odd parity). The ancilla is then brought back to $|0\rangle$ and the final state of the three qubits is then $e^{-iH\Delta t}|q_1 q_2 q_3\rangle$ (see the main text for more details)

Chapter 2

The Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) is a cutting edge hybrid quantum-classical algorithm in the field of quantum computing and represent a powerful tool for the solution of optimization problems. It allows to solve typical optimization problems, such as MaxCut, MIS and Ising model-based problems [41], by finding the ground state of an associated variational hamiltonian exploiting the principles of quantum mechanics, specifically the entanglement and the superposition of quantum states [27].

This chapter, which consists in a collection of the main outcomes of previous researches, is structured as follows: in the first section we provide the definition of the QAOA and how it works; in section 2 we summarize concrete state-of-the-art results regarding the application of the QAOA to the MaxCut problem on graphs, with a specific focus on the peculiar features one may leverage to successfully expand the application of QAOA to other types of problems; in section 3 we reveal the close connection between the QAOA and the Quantum Adiabatic Algorithm (QAA), while in section 4 we list the main characteristics of classical optimizers and initialization strategies, where the latter have been devised specifically for the QAOA to reduce the probability of failure of optimization procedures.

2.1 The algorithm

The QAOA has been introduced by Fahri et al. [10], at the dawn of quantum computing. Combinatorial optimization problems are specified by n bits and m clauses. The goal of the algorithm is to maximize an objective function of the form:

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z) \tag{2.1}$$

which represent the number of satisfied clauses. $C_\alpha(z) = 1$ if the clause α is satisfied by the string $z = z_1 z_2 ... z_n$ or $C_\alpha(z) = 0$ if it is not. Approximate optimization asks for a bit-string which makes $C(z)$ close to its maximum (or minimum, depending on the specific problem). To solve such a problem by means of a quantum computer, which works in a 2^n-dimensional Hilbert space of basis $|z\rangle$ where $z = 00..00, 00..01, ..., 11..11$ and n is the number of qubits, we think the objective function (2.1) as a diagonal operator in the computational basis.
Define:

$$U(C,\gamma) = e^{-i\gamma \hat{C}} = \prod_{\alpha=1}^{m} e^{-i\gamma C_\alpha} \tag{2.2}$$

which represent the γ-dependent evolution operator under the objective hamiltonian $\hat{C}$. The range of γ can be restricted to $[0, 2\pi]$ as C_α is an integer value. Now define:

$$\hat{B} = \sum_{j=1}^{n} \sigma_j^x \tag{2.3}$$

and the β-dependent product of single qubit operators:

$$U(B,\beta) = e^{-i\beta \hat{B}} = \prod_{j=1}^{n} e^{-i\beta \sigma_j^x} \tag{2.4}$$

where β lays in the interval $[0, \pi]$. In addition, notice that $[\hat{C}, \hat{B}] \neq 0$.
The initial state of the n qubits is the uniform superposition of the basis states:

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle \tag{2.5}$$

Given $2p$ parameters $\gamma_1 ... \gamma_p$ and $\beta_1 ... \beta_p$ where the integer $p \geq 1$ is called *depth* of the algorithm, define the parameterized state:

$$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle = U(B, \beta_p) U(C, \gamma_p) ... U(B, \beta_1) U(C, \gamma_1) |s\rangle. \tag{2.6}$$

The operators $U(C, \gamma_i)$ form the so called *cost layers* of the algorithm while the operators $U(B, \beta_i)$ form the *mixer layers*.
Call $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ the expectation value of the objective function C on the above angle-dependent state (2.6):

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle \tag{2.7}$$

and let M_p be the maximum of (2.7) over the angles $(\boldsymbol{\gamma}, \boldsymbol{\beta})$:

$$M_p = \max_{(\boldsymbol{\gamma}, \boldsymbol{\beta})} F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}). \tag{2.8}$$

The maximization at depth $p-1$ can be viewed as a constrained maximization at depth p:

$$M_p \geq M_{p-1}. \tag{2.9}$$

In the large p limit, it is expected, though Farhi et al. [10] do not provide a mathematical theorem to prove it, that the maximum M_p approaches the real maximum of the cost function C:

$$\lim_{p \to \infty} M_p = \max_z C(z). \tag{2.10}$$

At a finite depth one gets an approximate solution for the bitstring which maximizes the cost function.

In order to get such a bitstring one must choose an initial set of angles, create the state in eq. (2.6) using a quantum computer and evaluate the expectation of the cost function (2.1) on that state. To compute $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ a measurement on the final state is required to get the bitstring corresponding to that choice of angles. Our goal is to find a set of angles that makes $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ as close as possible to M_p which, in principle, is the best expectation value of the cost function one can get at depth p. It is straightforward to notice that it is impossible to know in advance how to pick good angles. This is why the algorithm needs support from a classical optimizer whose job is to provide new angles such that the expectation value of the objective function on the corresponding state is greater than the previous one and repeat these steps until convergence to the optimal parameters is reached. There are plenty of classical algorithms which can perform the optimization of the angles [12]. Each algorithm has its own features, which are presented in section 2.4, and the choice of the algorithm has a fundamental influence on the success of the optimization procedure.

In the following sections an analysis of the QAOA algorithm applied to the MaxCut problem on graphs is provided.

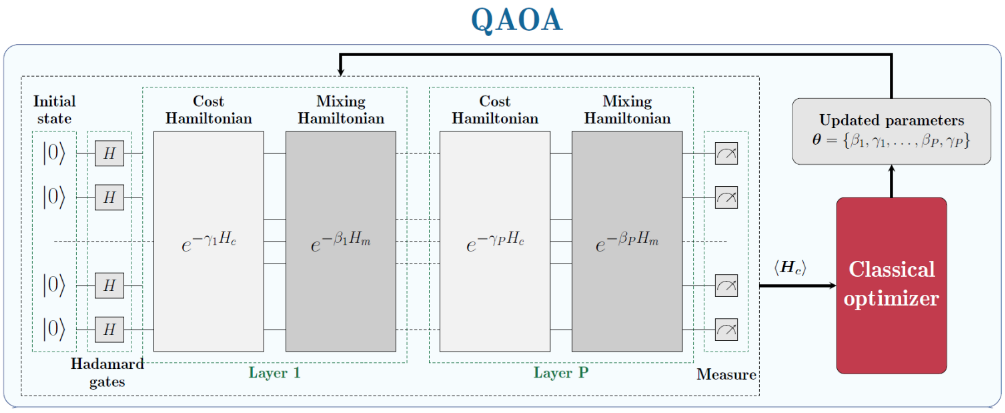


Figure 2.1: Schematic representation of the Quantum Approximate Optimization Algorithm. The qubits are initially placed in the state $|+\rangle^{\otimes N}$ and alternately evolved under the cost and the mixer parameterized hamiltonians p times, where p is the depth of the circuit. Finally, the quantum state is reconstructed by taking several measurements and the expectation value of the cost function is computed. This value is sent to the local optimizer which updates the values of the parameters until the best one are found and the optimal state is recovered

2.2 QAOA applied to the MaxCut problem on graphs

In this section we define the MaxCut problem [25] and present a state-of-the-art review which analyzes some interesting features of this problem on regular graphs and how the QAOA can be successfully applied to solve it.

2.2.1 Description of the MaxCut problem

Given a graph G=(V,E) [2] where V is the set of vertices/nodes and E is the set of edges, the MaxCut asks to split the graph G into two subsets such that the number of edges connecting two nodes belonging to different subsets is the largest possible. From now on, we'll consider graphs with bounded degree, where the *degree* of a node is defined as the number of nearest neighbors of that node. If all the nodes of a graph have the same degree, the graph is said *regular* [40].

Mapping the problem to the QAOA framework, solving the MaxCut on a graph G is equivalent to find the the bitstring $|z\rangle = |z_1 z_2 .. z_n\rangle$ such that the corresponding eigenvalue of the cost hamiltonian

$$C = \sum_{\langle jk \rangle} C_{\langle jk \rangle} \tag{2.11}$$

$$C_{\langle jk \rangle} = \frac{1}{2}(1 - \sigma_j^z \sigma_k^z) \tag{2.12}$$

is as large as possible, where $j, k \in V$ and $\langle jk \rangle \in E$. At the level of quantum computation, each node of the graph is mapped into one qubit.
Recall the expectation value of the cost hamiltonian defined in equation (2.7) and consider the operator associated with the edge $\langle jk \rangle$:

$$U^\dagger(C, \gamma_1)...U^\dagger(B, \beta_p)C_{\langle jk \rangle}U(B, \beta_p)...U(C, \gamma_1) \tag{2.13}$$

This operator involves only qubits j, k and all the qubits whose distance in the graph from qubits j and k is at most equal to p. This can be empirically seen at $p = 1$. The operator in eq. (2.13) now becomes:

$$U^\dagger(C, \gamma_1)U^\dagger(B, \beta_1)C_{\langle jk \rangle}U(B, \beta_1)U(C, \gamma_1). \tag{2.14}$$

The terms which do not involve qubits j or k in the factor $U^\dagger(B, \beta_1)$ commute with $C_{\langle jk \rangle}$ and cancel out with the corresponding terms in $U(B, \beta_1)$. As a consequence the single-edge operator reduces to:

$$U^\dagger(C, \gamma_1)e^{i\beta_1(\sigma_j^x + \sigma_k^x)}C_{\langle jk \rangle}e^{-i\beta_1(\sigma_j^x + \sigma_k^x)}U(C, \gamma_1). \tag{2.15}$$

As the mixer layer now depend only on the vertices j and k, the terms which do not involve qubits j and k in the cost layer operator $U(C, \gamma_1)$ commute through and cancel out. Therefore, the single-edge operator depends only on qubits j,k and their nearest neighbors. This proof can be generalized to any p. In other words, each term in the cost hamiltonian (2.12) depends on the qubits forming the central edge $\langle jk \rangle$ and on the edges which are at most p steps away from the central one and the qubits on them. The group formed by the central edge and the nearest neighbor is called a *subgraph* of G. As an example, fig. 2.2 shows all the possible subgraphs at $p = 1$ of a 3-regular graph, namely the tree, the single triangle and the double triangle. The number of qubits in each subgraph is independent of n and this allows to compute the expectation value $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ as a sum of contributions coming from subsystems whose size does not depend on n which is the size of the whole graph.
Given a subgraph g, define:

$$C_g = \sum_{\langle jk \rangle \in g} C_{\langle jk \rangle} \tag{2.16}$$

$$B_g = \sum_{j \in g} \sigma_j^x \tag{2.17}$$

$$|s, g\rangle = \prod_{j \in g} |+\rangle_j \tag{2.18}$$

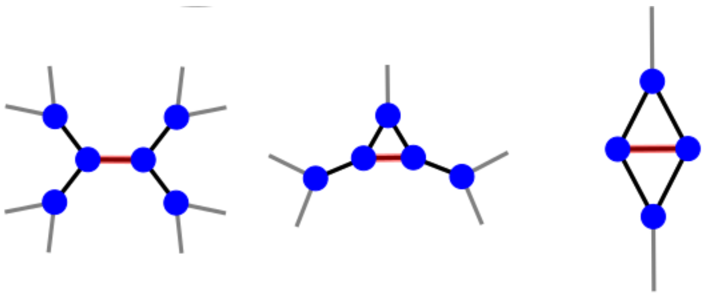


Figure 2.2: The three different subgraphs of a 3-regular graph at $p = 1$ [41]. From left to right: the tree subgraph, the single triangle and the double triangle. The edge in red in each subgraph is the central one

which are the usual QAOA operators and initial state restricted to g. Each edge in the graph G, $\langle jk \rangle$, is associated with a subgraph $g(j, k)$ whose expectation value is a contribution to F_p. If two edges $\langle jk \rangle$ and $\langle j'k' \rangle$ give rise to isomorphic subgraphs then the corresponding function of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is the same. F_p can be then written as [10]:

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \sum_g w_g f_g(\boldsymbol{\gamma}, \boldsymbol{\beta}) \tag{2.19}$$

where

$$f_g(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle s, g| U^\dagger(C_g, \gamma_1) ... U^\dagger(B_g, \beta_p) C_{\langle jk \rangle} U(B_g, \beta_p) ... U(C_g, \gamma_1) |s, g\rangle \tag{2.20}$$

and w_g are the weights of each subgraph type g which count the number of occurrences of the g in the global edge sum. Notice that the terms f_g do not depend on the number of bits n and the number of clauses m. The maximum number of qubits which can appear in each term f_g is the number of qubits of the tree subgraph at depth p which is:

$$q_{tree} = 2 \left[\frac{(d-1)^{p+1} - 1}{d - 2} \right] \tag{2.21}$$

or $2p+2$ if $d = 2$ where d is the maximum degree of the graph. The resource requirement does not grow with n. However, if p is very large, they may be beyond our current classical computational power and that's why the development of quantum computers is expected to bring a great advantage in this sense.
If p and d are fixed, the distribution of $C(z)$ is concentrated near its mean. To see this, consider

$$\langle \boldsymbol{\gamma}, \boldsymbol{\beta}| C^2 |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle - \langle \boldsymbol{\gamma}, \boldsymbol{\beta}| C |\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle^2 = \tag{2.22}$$

$$= \sum_{\langle jk\rangle,\langle j'k'\rangle} \langle s|U^\dagger(C,\gamma_1)...U^\dagger(B,\beta_p)C_{\langle jk\rangle}C_{\langle j'k'\rangle}U(B,\beta_p)...U(C,\gamma_1)|s\rangle$$
$$- \langle s|U^\dagger(C,\gamma_1)...U^\dagger(B,\beta_p)C_{\langle jk\rangle}U(B,\beta_p)...U(C,\gamma_1)|s\rangle$$
$$* \langle s|U^\dagger(C,\gamma_1)...U^\dagger(B,\beta_p)C_{\langle j'k'\rangle}U(B,\beta_p)...U(C,\gamma_1)|s\rangle. \quad (2.23)$$

If the subgraphs $g(j,k)$ and $g(j',k')$ have no common qubits, the term in eq. (2.23) is 0. Going deeper, it means that there is no path which connects the two central edges $\langle jk\rangle$ and $\langle j'k'\rangle$ shorter than $2p+2$. The maximum number of edges which may contribute to the sum in eq. (2.23) is

$$2\left[\frac{(d-1)^{2p+2}-1}{d-2}\right] \quad (2.24)$$

where we have used eq. (2.21) substituting $2p+1$ instead of p. As a consequence,

$$\langle\boldsymbol{\gamma},\boldsymbol{\beta}|C^2|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle - \langle\boldsymbol{\gamma},\boldsymbol{\beta}|C|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle^2 \leq 2\left[\frac{(d-1)^{2p+2}-1}{d-2}\right]m \quad (2.25)$$

since each summand has at most unit norm. The standard deviation of $C(z)$ is therefore proportional to $\sqrt{m}$ so that the mean of a sample of order m^2 of $C(z)$ would be at most 1 far from F_p with probability $1-\frac{1}{m}$. If the number of clauses of the problem is sufficiently large, the probability that the algorithm produces a bitstring with $C(z)$ much larger than F_p is negligible.

2.2.2 MaxCut on 2-regular graphs

Before addressing the MaxCut problem applied to 2-regular graphs, let's define two important quantities:

- The **cut value** [10] is the number of edges connecting nodes belonging to the two different subsets divided by the total number of edges of the instance graph G. The goal of the MaxCut, in fact, is to find the maximum cut value for that graph, clarifying the meaning of the name;

- The **approximation ratio** [10] is the ratio between the cut value found by the algorithm and the maximum cut value of the instance graph. Roughly speaking, the approximation ratio is an indicator of the "quality" of the solution found by the algorithm. If it finds the best solution, the approximation ratio is identically 1.

As a simple example of application of the QAOA algorithm to the MaxCut problem let's consider connected 2-regular graphs with n vertices which are rings. The real MaxCut value depends on the parity of n: if n is even, the maximum cut is 1 and the solution bitstring is $|z\rangle = |0101...0101\rangle$; if n is odd, there will be one edge connecting two nodes of the same subsets so that the MaxCut value is $\frac{n-1}{n} = 1 - \frac{1}{n}$ and the solution string is $|z\rangle = |0101...010\rangle$. The only possible subgraph type is a line of length $2p+2$ with the given edge in the middle. Having n vertices, the weight for that subgraph type is n. Maximizing eq. (2.20) for small values of p and recovering the general solution by induction, Fahri et al. [10] found that

$$M_p = n\frac{2p+1}{2p+2} \tag{2.26}$$

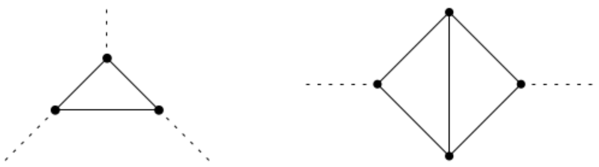
with a corresponding cut value of $\frac{2p+1}{2p+2}$. It is straightforward to see that by taking p large enough the cut value can be made arbitrarily close to 1, proving the large-p limit in eq. (2.10).

2.2.3 MaxCut on 3-regular graphs and performance guarantees

3-regular graphs have been useful to study the performance guarantees of QAOA, that are the minimum approximation ratios the algorithm will achieve, which depend on p, regardless the instance graph it has been fed with. Furthermore, some interesting properties concerning the solution of the MaxCut which massively simplify the practical usage of the algorithm on regular graphs have been conjectured [41].

p = 1 performance guarantees

Consider a generic 3-regular graph with n vertices. Suppose there are T isolated triangles and S crossed squares like the following ones [10]:



To say that a triangle is isolated, the dotted outgoing lines must end to distinct vertices. Because no isolated triangle and crossed square can share a vertex, in general $3T+4S \leq n$. In each crossed square there is one edge whose associated subgraph is the double triangle showed in fig. 2.2, call it g_4, and 4 edges whose subgraph type is the single triangle, call it g_5. In each isolated triangle, 3 edges have subgraph g_5. All other edges have an associated tree subgraph, call it g_6. Therefore, the expectation value of the cost function can be written as:

$$F_1(\gamma,\beta) = Sf_{g_4} + (3T+4S)f_{g_5} + \left(\frac{3n}{2} - 5S - 3T\right)f_{g_6} \tag{2.27}$$

where the function f_g has been defined in eq. (2.20). The maximum of F_1 is [10]:

$$M_1(n, S, T) = \max_{\gamma,\beta} F_1(\gamma, \beta). \tag{2.28}$$

Now, in each crossed square or isolated triangle there is at least one edge out of the cut. Hence, that the maximum cut value is upper bounded by $1 - \frac{T}{3n/2} - \frac{S}{3n/2}$. Scaling out n and numerically evaluating the approximation ratio, it has been noticed that it reaches its minimum when neither isolated triangle nor crossed square is present in the graph, that is to say, $T = S = 0$ and all the edges have a g_6 associated subgraph. Let's call these special graphs 1-trees. The minimum approximation ratio is 0.6924.

p $\geq$ 1 guarantees conjectures

Wurtz et al. [41] provided an alternative proof of the $p = 1$ performance guarantee which leverages a so called *graph hierarchy.* Their proof can be summarized by saying that given a graph G, one can find a new graph G' whose approximation ratio is worse. To construct the new graph G', one needs to locate cycles whose length is smaller or equal to $2p + 1$ in the graph G and replace all the edges which are part of those cycles with particular subgraphs shown in fig. 2.3. Replacing an edge with this subgraph eliminates the small cycles, where with the term "small" we mean that its length is $\leq 2p + 1$, so that the algorithm has a harder time solving since it cannot identify any closed easily-solvable loop in any subgraph. The subgraph type associated to each edge in the replacing subgraph is the tree one.

At the very end, it has been found for $p = 1$ and $p = 2$ that 1-trees and 2-trees graphs are the worst in terms of approximation ratio. In particular, for $p = 2$ the performance guarantee has a value of 0.7559. Wurtz et al. [41] supposed that what they observed for $p = 1, 2$ should hold also for $p \geq 2$. This led to the following two conjectures:

- **Graph hierarchy conjecture**: Given an instance graph G and circuit depth p, there exist optimal angles for the tree subgraph and there exists an edge replacement with a subgraph of the same kind of the ones shown in fig. 2.3 such that the new resulting graph has a smaller approximation ratio. The replacing subgraph should not have any cycle of length smaller than $2p + 2$;

- **Large loop conjecture**: p-trees are worst case graphs. These graphs are constructed only out of the tree subgraph and do not have cycles smaller than $2p + 2$. If the algorithm cannot distinguish between even and odd length cycles it is not able to make good prediction about the cut estimate which require this distinction. Moreover, the algorithm "sees" less of the full graph and this leads to worse solutions.

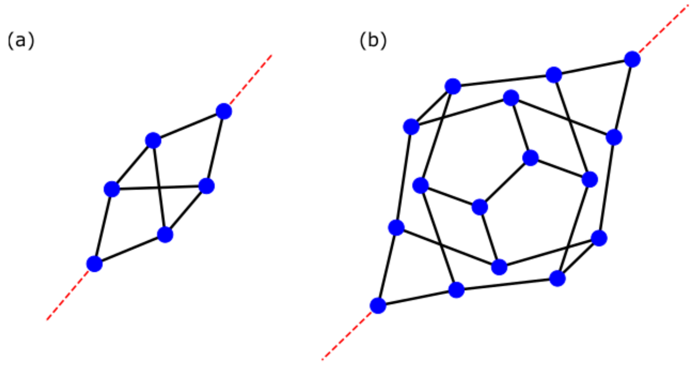


Figure 2.3: Edge replacement for $p = 1$ (a) and $p = 2$ (b) [41]

The fixed angle conjecture

In [41], Wurtz et al. discovered a very interesting and useful property regarding the optimal parameters of the QAOA which can be synthesized by the so called *fixed angle conjecture* which states the following:
Any graph evaluated at fixed angles optimal for the tree subgraph will have an approximation ratio larger than the guarantee.
These angles can be used as starting points for a further optimization procedure which returns optimal angles for the specific graph of interest.
The fixed angles have been computed by Wurtz et al. [42] for 3-regular graphs and they are shown in fig. 2.4. Furthermore, the fixed angle conjecture has been proved for the $n \leq 16$ 3-regular graphs up to $p = 11$ and the results are shown in fig. 2.5. The worst approximation ratio obtained among all the graphs of the ensemble, represented by the lower bound of the shaded region, is always above the guarantee, represented by the yellow dotted line, proving the fixed angle conjecture. As mentioned above, the fixed angles can be used as a starting point for a local optimizer to find the global optimum for that specific graph. Fig. 2.6 shows a comparison between average approximation ratio of the $n \leq 16$ 3-regular graphs computed at fixed angles and at optimal angles, found by the optimizer. For all the graphs at $p = 1$ and an overwhelming majority of the graphs at $p = 2$, the optimizer found the real global optimum. As a consequence, fixed angles are very good guesses for local optimizers and can be successfully exploited in this kind of analysis significantly reducing the computational time needed by the optimizer.

			1	2	3	4	5	6	7	8	9	10	11
$p = 1$	$C_1 \geq 0.6925$	γ	0.616										
		β	0.393										
$p = 2$	$C_2 \geq 0.7559$	γ	0.488	0.898									
		β	0.555	0.293									
$p = 3$	$C_3 \geq 0.7924$	γ	0.422	0.798	0.937								
		β	0.609	0.459	0.235								
$p = 4$	$C_4 \geq 0.8169$	γ	0.409	0.781	0.988	1.156							
		β	0.600	0.434	0.297	0.159							
$p = 5$	$C_5 \geq 0.8364$	γ	0.360	0.707	0.823	1.005	1.154						
		β	0.632	0.523	0.390	0.275	0.149						
$p = 6$	$C_6 \geq 0.8499$	γ	0.331	0.645	0.731	0.837	1.009	1.126					
		β	0.636	0.534	0.463	0.360	0.259	0.139					
$p = 7$	$C_7 \geq 0.8598$	γ	0.310	0.618	0.690	0.751	0.859	1.020	1.122				
		β	0.648	0.554	0.490	0.445	0.341	0.244	0.131				
$p = 8$	$C_8 \geq 0.8674$	γ	0.295	0.587	0.654	0.708	0.765	0.864	1.026	1.116			
		β	0.649	0.555	0.500	0.469	0.420	0.319	0.231	0.123			
$p = 9$	$C_9 \geq 0.8735$	γ	0.279	0.566	0.631	0.679	0.726	0.768	0.875	1.037	1.118		
		β	0.654	0.562	0.509	0.487	0.451	0.403	0.305	0.220	0.117		
$p = 10$	$C_{10} \geq 0.8785$	γ	0.267	0.545	0.610	0.656	0.696	0.729	0.774	0.882	1.044	1.115	
		β	0.656	0.563	0.514	0.496	0.469	0.436	0.388	0.291	0.211	0.112	
$p = 11$	$C_{11} \geq 0.8828$	γ	0.257	0.528	0.592	0.640	0.677	0.702	0.737	0.775	0.884	1.047	1.115
		β	0.656	0.563	0.516	0.504	0.482	0.456	0.421	0.371	0.276	0.201	0.107

Figure 2.4: Performance guarantees up to $p = 11$ (on the left) and optimal angles for the tree subgraph of 3-regular graphs up to $p = 11$ (on the right). These are the so called fixed angles. If a graph is evaluated at fixed angles, the fixed angle conjecture states that the approximation ratio will be larger than the guarantee in column 2 [42]. The conjecture has been proved for the ensemble of the $n \leq 16$ 3-regular graphs

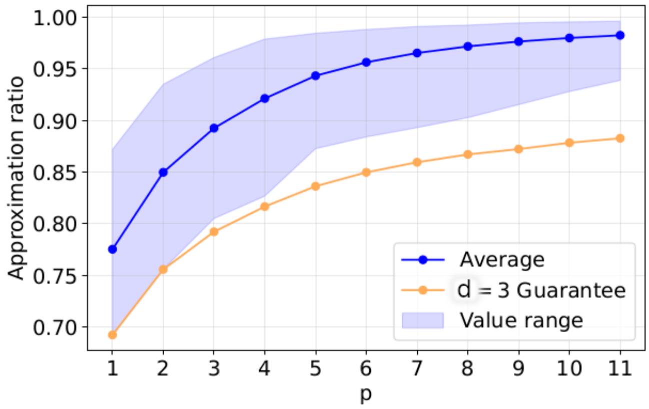
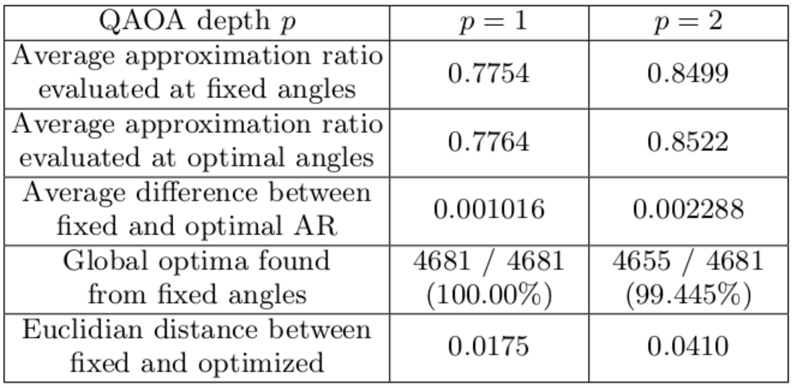


Figure 2.5: Approximation ratio for the $n \leq 16$ complete ensemble of 3-regular graphs [42]. The yellow line represents the performance guarantees shown in fig. 2.4. The blue shaded region shows the range between the best and worst approximation ratio obtained among all the graphs of the ensemble. As one can notice, the lower bound of the shaded region is always above the guarantee.



QAOA depth p	$p = 1$	$p = 2$
Average approximation ratio evaluated at fixed angles	0.7754	0.8499
Average approximation ratio evaluated at optimal angles	0.7764	0.8522
Average difference between fixed and optimal AR	0.001016	0.002288
Global optima found from fixed angles	4681 / 4681 (100.00%)	4655 / 4681 (99.445%)
Euclidian distance between fixed and optimized	0.0175	0.0410

Figure 2.6: Comparison between the average approximation ratio of $n \leq 16$ ensemble of 3-regular graphs evaluated at global optima vs. fixed angles at $p = 1, 2$ [42]. Rows 1 and 2 represent the average approximation ratio evaluated at fixed and optimal angles. Row 3 represents the average difference between the approximation ratio evaluated at fixed and optimal angles. Row 4 indexes the number and percentage of graphs for which the gradient ascent optimizer initialized at fixed angles finds the global optima. Row 5 represents the euclidean distance between the fixed angles and optimal ones in parameters space. For both $p = 1, 2$ the fixed angles are very good initial guess for the optimizers because the global optima has been found for almost every graph.

2.2.4 Transferability of optimal solutions

The discovery of fixed angles on 3-regular graphs [42], which can be exploited to find the optimal parameters quicker, leads to an additional question: is there the possibility that optimal parameters for a given graph G_1 are also optimal for another graph G_2 and, in this case, under which conditions this happens?
A detailed analysis which answers to this question is present in [14]. Before looking at the result Galda et al. [14] got, let's define the so called *Transferability* of parameters:
given the optimal parameters for a small "donor" graph, the transferability of parameters is defined as the usage of the optimal parameters of the donor graph to prepare the QAOA state which maximizes the expectation value of the same hamiltonian on a larger "acceptor" graph
The exploitation of quickly computable parameters, which are optimal for the donor graph and are also optimal or near-optimal for the acceptor graph, may lead to a dramatic reduction of the time required to compute the optimal parameters for the acceptor graph which, in principle, is the graph of interest. This can be seen as a sort of trick to speed up the computation.
In order to find out, if there is, a rule capable to predict when the transferability is good or bad between two graphs, it is important to understand if there is a pattern in the

location of maxima and minima in the cost function, or, equivalently, energy, landscape of the subgraphs of regular graphs. Fig. 2.7 shows the energy landscape of the subgraphs of 3-, 4- and 5-regular graphs as a function of angles γ and β. As one can notice, maxima and minima are located approximately in the same position among subgraphs of 3- and 5-regular graphs while only half of them match the corresponding maxima and minima of subgraphs of 4-regular graphs. However, for what subgraphs of 4-regular graphs only is concerned, maxima and minima are close between themselves. These observations led to the following three conjectures [14]:

- Optimized parameters can be successfully transferred between any two random d-regular graphs;
- Optimized parameters can be successfully transferred between two random d_1- and d_2-regular graphs provided that d_1 and d_2 have the same parity, i.e. both even or odd;
- Optimized parameters cannot be successfully transferred between two random d_1- and d_2-regular graphs when d_1 is even and d_2 is odd or viceversa.

A simple test of these three conjectures is shown in fig. 2.8. This heatmap represent the transferability coefficient, which is an indicator of good or bad transferability, between all the $p = 1$ subgraphs of d-regular graphs with d ranging from 1 to 8. The transferability coefficient has been computed by optimizing the parameters (γ, β) on the donor subgraph, from which the parameters are being transferred, starting from 20 different random instances, and evaluating the ratio between the average expectation value of the cost function using each of the 20 couples of optimized parameters on the acceptor subgraph and the maximum energy contribution of the acceptor subgraph in the same optimization procedure. The main result is that the transferability is very good between subgraphs of even or odd regular graphs while the transferability is bad across subgraphs of even and odd regular graphs. What has been shown by Galda et al. [14] is a partial analysis. Further studies of the transferability at larger p are required. However, the observations are compatible with the ones of Brandao et al. [4] shown in fig. 2.9. Here, the expectation value of 25 different 3-regular graphs with 20 nodes each and a verified MaxCut value of 26 has been evaluated using 3 sets of parameters for each p: the first one is chosen such that the cost function value is low; the second one is randomly chosen and the third one is chosen to give a high cost function expectation. The analysis has been repeated for each p ranging from 2 to 8.
It is straightforward to notice that the standard deviation of the expectation is very low compared to the average cost function value. This means that each set of angles (γ_i, β_i) gives approximately the same expectation value of the cost function for each graph instance. Therefore, at least for 3-regular graphs, the transferability of parameters between random graphs is good.

An example of how transferability can be exploited to evaluate large graphs starting from optimal parameters for small graphs is provided in [4]. Brandao et al. generated a random 10-nodes 3-regular graph and optimized the 16 parameters at $p = 8$ starting from 200 random instances and the best set of angles was kept. The approximation ratio they found was 0.984. These angles has been used to evaluate 25 random 3-regular graphs with 24 vertices. The average approximation ratio was 0.934 with a standard deviation of 0.014. The optimal parameters for the 10-node graph were quasi-optimal for all the 25 24-nodes graphs. Because no optimization was performed on the larger graphs, the computational time was severely reduced and the quality of the solution was still considerable.

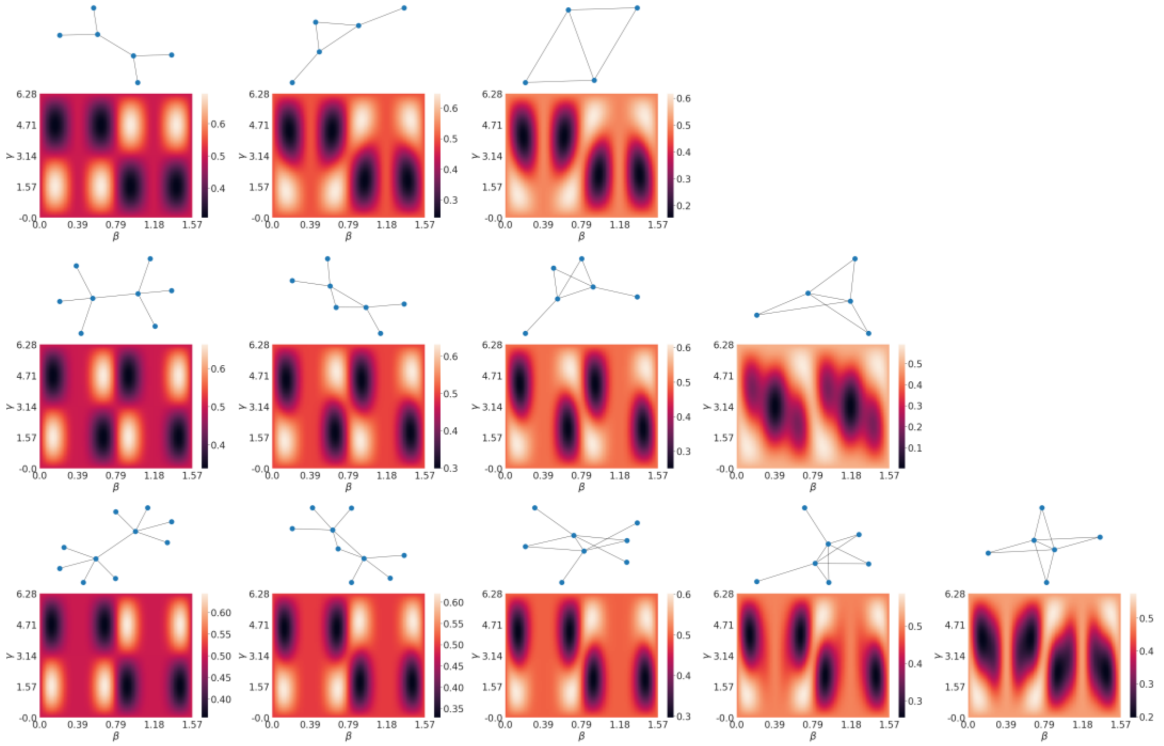


Figure 2.7: Energy landscape of the $p = 1$ subgraphs of 3- (top row), 4- (middle row) and 5- (bottom row) regular graphs as a function of QAOA parameters [14]. Energy maxima and minima are located approximately in the same position among the subgraphs of 3- and 5- regular graphs. For subgraphs of 4-regular graphs, maxima and minima are still located in the relative vicinity between themselves though only half of them match the ones of subgraphs of 3- and 5-regular graphs.

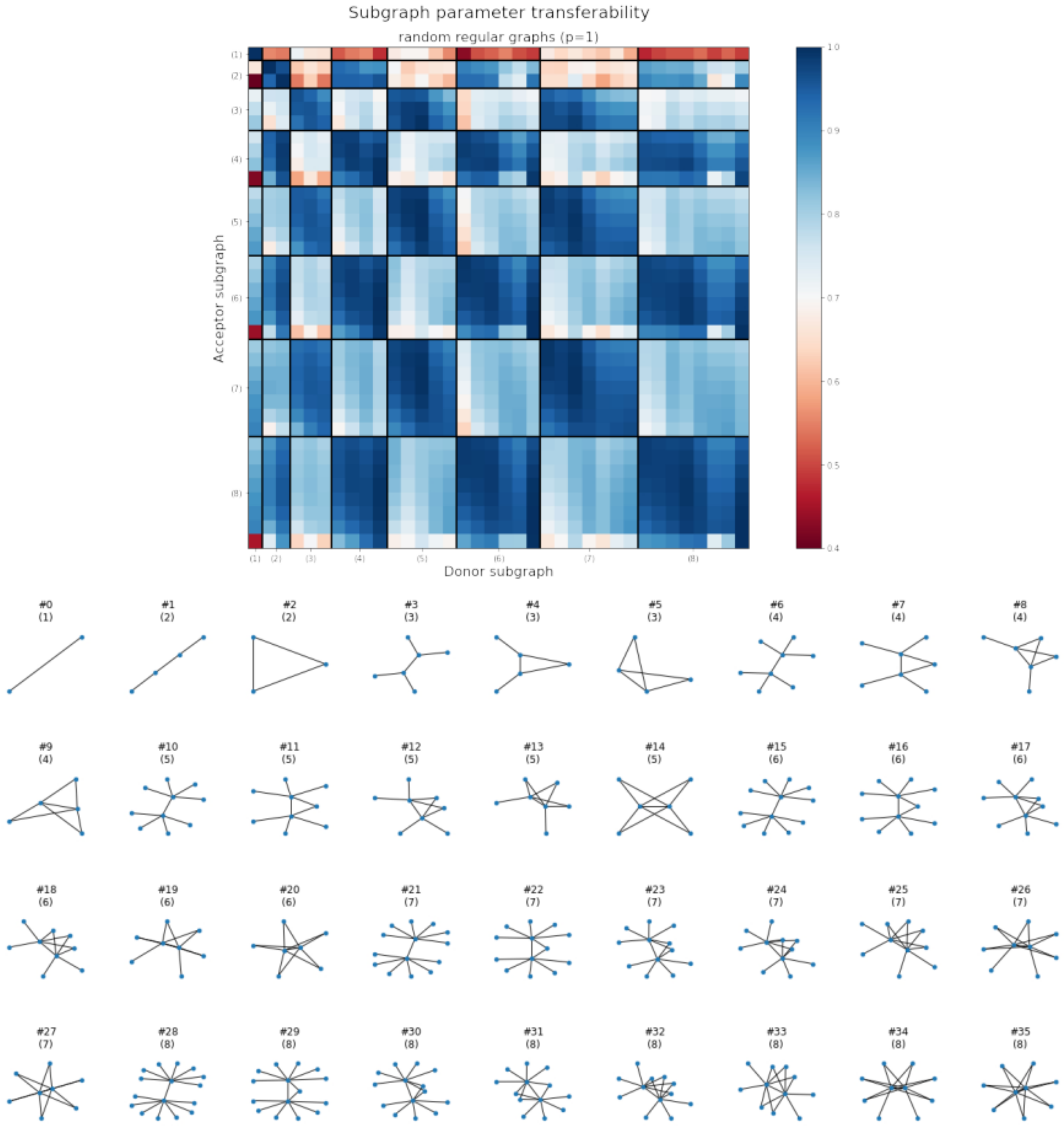


Figure 2.8: Transferability heatmap between all subgraphs of d-regular graphs with d ranging from 1 to 8 and $p = 1$ [14]. Blue and red values represent good and bad transferability, respectively. Good transferability is observed between subgraphs of odd or even regular graphs while a bad transferability is observed across subgraphs of odd and even regular graphs

	Low		Random		High	
p	Mean	Std.	Mean	Std.	Mean	Std.
2	6.636	0.319	14.691	0.036	22.409	0.228
3	5.218	0.294	15.125	0.042	23.109	0.175
4	3.933	0.259	14.627	0.157	23.822	0.272
5	3.132	0.159	15.725	0.113	24.349	0.179
6	2.550	0.100	16.404	0.140	24.918	0.266
7	1.954	0.088	15.975	0.096	25.110	0.221

Figure 2.9: Mean and standard deviation of objective function values across 25 random 20-nodes 3-regular graph instances each of them having a MaxCut value of 26 [4]. For each p there are 3 sets of fixed parameters. The first is chosen to give a low cost function value, the second is randomly chosen and the third is chosen to make the cost function high.

2.3 Connection with the Quantum Adiabatic Algorithm

The Quantum Approximate Optimization Algorithm originates from the Quantum Adiabatic Algorithm (QAA) [11]. Before analyzing the connection between these two variational algorithms, it is important to recall two milestones of quantum mechanics and quantum information: the quantum adiabatic theorem [5] and the Trotter's decomposition [34].

The Quantum Adiabatic Theorem

Consider a time-dependent hamiltonian $H(s)$ where $s \in [0,1]$ and it is defined as $s = \frac{t}{T}$ with $t \in [0,T]$ the time parameter and T the total evolution time.
The Quantum Adiabatic Theorem, according to [5], states that:
Suppose $H(s)$ is an hamiltonian which has a non-degenerate ground state for any $s \in [0,1]$, and suppose that the total evolution time T satisfies

$$T \geq \frac{2}{\varepsilon}\left[c_1\frac{||\dot{H}(0)||}{\Delta(0)^2} + c_1\frac{||\dot{H}(1)||}{\Delta(1)^2} + \int_0^1 ds\left((3c_1^2 + c_1 + c_3)\frac{||\dot{H}(s)||^2}{\Delta(s)^3} + c_2\frac{||\ddot{H}(s)||}{\Delta(s)^2}\right)\right]. \tag{2.29}$$

Then the evolution of the initial state $|\psi(0)\rangle = |\phi(0)\rangle$ under the Schrödinger equation (1.6) produces a final state $|\psi(1)\rangle$ satisfying

$$|||\psi(1)\rangle - |\phi(1)\rangle|| \leq \varepsilon \tag{2.30}$$

for some constant c_1, c_2, c_3. $|\psi(s)\rangle$ is defined as the actual state of the system and $|\phi(s)\rangle$ is the instantaneous ground state of $H(s)$. $\Delta(s)$ is the gap between the smallest eigenvalue $E(s)$ of $H(s)$ and the nearest distinct eigenvalue of $H(s)$ [5].

The Trotter's decomposition

The proof of the Trotter's theorem [27] [34] has been shown in detail in sec. 1.6.1. For the sake of completeness of the current discussion, here we report only the statement: *Given an hamiltonian* $H = A + B$ *where* $[A, B] \neq 0$*, the evolution of a system under* H *for a time* t *can be written as*

$$e^{it(A+B)} = \lim_{N\to\infty} \left(e^{i\frac{t}{N}A} e^{i\frac{t}{N}B} \right). \tag{2.31}$$

The quantum adiabatic algorithm [11] is designed to find the optimal solution of a given problem exploiting the quantum adiabatic theorem. The QAA starts by putting the system in the ground state of an hamiltonian B and slowly changes the hamiltonian until it reaches the problem hamiltonian C, which does not commute with B. The instantaneous hamiltonian can be written as

$$H(t) = (1 - t/T)C + (t/T)B \tag{2.32}$$

where T is the total evolution time. The quantum adiabatic theorem ensures that, if T is sufficiently large, the system will be found in the ground state of C. Now, let's set B as in eq. (2.3), C as in eq. (2.1) and the initial state as in eq. (2.5). Leveraging Trotter's decomposition formula, we can discretize the adiabatic evolution of eq. (2.32) by alternately applying evolution operators $U(C, \gamma)$, $U(B, \beta)$ to obtain the final state $|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$, which is the final QAOA state. Looking from this perspective, the sum of the angles is the total time T. We want angles to be small, so that the QAOA resembles the adiabatic evolution, and the evolution time to be large, which forces p to be large. In other words, we can always find a set of angles $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ and a depth p such that we make the expectation value of the cost function C, previously defined as $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ in eq. (2.7), close to its maximum M_p as desired.
A remarkable difference between QAA and QAOA is that QAA works by producing a state with a large overlap with the optimal bitstring. On the contrary, QAOA produces a state whose energy is close to the maximum, but the overlap between this state and the best one in terms of energy may be very small. This different behavior of the two algorithms is due to the fact that QAA works by leveraging the quantum adiabatic theorem [5] described above which directly controls the evolution of an initial state, which is already the ground state of a given hamiltonian, towards the ground state of the final cost hamiltonian. The QAOA, instead, exploits the Trotter's formula but the

quantity which undergoes optimization is the energy, i.e. the expectation value of the cost function. As a consequence, if the QAOA algorithm does not perfectly succeed, it will find a state whose energy is close to the maximum, but nothing guarantees that the state itself is geometrically close to the ground state, i.e. that it has a large overlap with the vacuum.

The overlap between two states $|\psi\rangle$ and $|\phi\rangle$ is measured by computing the squared modulus of the scalar product between the states and it's called *fidelity*:

$$F(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|^2 \tag{2.33}$$

This quantity will play a crucial role in the determination of the success or failure of an optimization procedure of the QAOA.

2.4 Classical optimizers

Hitherto we have stressed that, in order to optimize the QAOA parameters, a classical optimizer coupled to the quantum circuit is required. There are plenty of classical optimization algorithms, each of them having its own peculiar features. A first classification of optimization algorithms can be made by splitting them into *local optimization algorithms* and *global optimization algorithms.* A general description of the main algorithms is provided in the following sections [12]. A technical analysis of all of them is, instead, beyond the scope of this work.

2.4.1 Local optimization algorithms

Local search algorithms are designed to find the maximum or minimum (suppose for the sake of simplicity to be a minimum) of an objective function $f(x)$, where x is a parameter vector, whose energy landscape is, roughly speaking, "simple", meaning that only a single minimum is present in the search space i.e. local minima are absent. From another perspective, we may think a local optimizer as an algorithm which is able to locate a local minimum within a small region of the whole search space, where, in principle, other local minima and a global minimum are present.

Some local search algorithms require the evaluation of the gradient of the cost function, while others do not. The former are called *gradient-based algorithms* while the latter are called *gradient-free algorithms.*

Gradient-based algorithms

Gradient-based algorithms compute the gradient of the objective function in order to find a search direction at a given point on which an improvement in terms of energy is expected. On this direction a new point to evaluate the cost function is selected. At the new point, the energy is smaller (remember that our goal is to minimize the objective). This is the idea of the simplest gradient-based algorithm which is the *Gradient descent.* In some cases, the calculation of the derivatives of $f(x)$ may be computationally demanding. On the other hand, gradient-based algorithms are preferred when the search space is large. The main gradient-based algorithms, which refine the idea of the gradient descent are listed below [12]:

- **Adam** [18] is a stochastic gradient descent algorithm with adaptive learning rate which helps dampening oscillations and accelerating the whole procedure. It is suitable for non-stationary objective functions and requires little memory

- **AMSGRAD** [36] is a variant of the Adam. It uses long-term memory of past gradients to improve convergence properties

- **SPSA(Simultaneous Perturbation Stochastic Approximation)** [32] is suitable for objective functions with multiple unknown parameters. Its main feature is the gradient approximation which requires only two measurements of the cost function regardless the dimension of the objective.

- **CG(Conjugate Gradient)** [26] uses only first derivatives to compute an improving approximate solution x for an objective function which can be written as

  $$f(x) = \frac{1}{2}x^T Ax - x^T b \tag{2.34}$$

  where A is a matrix and b is a vector. Minimizing $f(x)$ is equivalent to solve the equation

  $$Ax = b. \tag{2.35}$$

- **TNC(Truncated Newton)** [8] is a more sophisticated version of CG. It computes the search direction d_k by applying CG to solve

  $$\nabla^2_{xx} f(x_k) d_k = -\nabla f(x_k) \tag{2.36}$$

- **L-BFGS-B(Limited-memory BFGS Bound)** [22] belongs to the class of quasi-Netwon methods. It uses the first derivative of f to compute the steepest descent direction and an approximated inverse hessian matrix which is stored in little memory. It is suitable for problems involving several variables

- **SLSQP(Sequential Least SQuares Programming)** [23] involves BFGS method to minimize a function with any combination of equality and inequality constraints.

The search direction is computed solving the quadratic subproblem

$$\min_d f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \sigma_k) d \tag{2.37}$$

$$such\, that\; b(x_k) + \nabla b(x_k)^T d \leq 0 \tag{2.38}$$

$$and\; b(x_k) + \nabla b(x_k)^T d = 0 \tag{2.39}$$

where $\mathcal{L}(x, \lambda, \sigma) = f(x) - \lambda b(x) - \sigma c(x)$ for a single equality and single inequality constraints problem

Gradient-free algorithms

In some optimization problems, the information about the derivative of $f(x)$ may be unavailable or expensive to obtain in terms of computational resources because the objective itself is noisy or non-smooth. In these cases, gradient-based algorithms cannot provide any relevant utility as they rely on derivatives or finite differences. Therefore, gradient-free algorithms, which do not compute any derivative, are favorable. The main gradient-free algorithms are [12]:

- **COBYLA(Constrained Optimization BY Linear Approximation)** [30] computes the expectation of the objective function at the vertices of a trust region, which is a simplex, in its mathematical meaning, in N dimension, where N is the number of variables of the problem. A linear polynomial is then used to create an interpolation of the objective function. The same is done for the constraints

- **Nelder-Mead** [15] is a direct search method which initially evaluates the objective in $N + 1$ test points arranged at the vertices of a simplex. It then computes a new test point where the function is going to be evaluated and, if the expectation value is better than at least one of the old test point, it replaces that point with the new test point. This is repeated until convergence. This method is suitable for problems whose objective varies smoothly and is unimodal

- **Powell** [30] method is a conjugated search method which finds local minima of a real-valued function of N real-valued inputs. The algorithm starts from a point x_0 on which the objective is evaluated and defines N search vectors, typically the normals $\{s_1, ..., s_N\}$ aligned with each axis. It then performs a bi-directional search along each vector s_i. The minima found during each bi-directional search are

$$x_0 + s_1\alpha_1, x_0 + \sum_{i=1}^{2} s_i\alpha_i, ..., x_0 + \sum_{i=1}^{N} s_i\alpha_i \tag{2.40}$$

  where the scalars α_i are obtained during the search procedure. The point $x_0 + \sum_{i=1}^{N} s_i\alpha_i$ is the new point to start a new bi-directional search. This is repeated until convergence

2.4.2 Global optimization algorithms

Global optimization algorithms are devised to search for the global optimum of an objective function $f(x)$ whose landscape is populated by other local optima. When this happens, it is very tricky for a local optimizer to correctly identify the global minimum as it'll probably get stuck in some local minima. A global search is then required. Unlike local optimizers, for which the techniques to solve the problem rely on similar basic ideas, the strategies may be completely different and, in general, there are a few if any grantees that the algorithm will locate the global optimum. However, when the energy landscape is too complicated to handle for a local optimizer, global search algorithms are very useful and powerful tools. A global search often imply the evaluation of some candidate solutions from which new candidates are generated and kept if there is an improvement. Some global optimizers are listed below.

- **Differential Evolution(DE)** [9] [31] is based on population breeding and recombination. Given an N-dimensional search space, DE starts by choosing a population of M points in which the cost function is evaluated picking the best one, x_{best}. After that, for each population's member, a "mutant" point is created in this way:

$$x_{mutant} = x_{best} + F(x_1 - x_2) \tag{2.41}$$

  where F is called mutation factor and x_1, x_2 are chosen randomly among the members. In the recombination phase some parameters of the member are replaced by the mutant's ones according to a several possible strategies. Finally, if the cost function is lower in the modified target the point is kept, otherwise it's discarded. This is repeated until convergence

- **Bayesian Optimization Algorithm** [35] is made out of three phases. The warmup phase is the same of DE, a set of points $X = \{\boldsymbol{\theta}_j\}_{j=1}^{M}$ is chosen and the best is picked. In the kernel optimization phase the hyperparameters $(\tilde{\sigma}, \tilde{l})$ are fixed in order to minimize the likelihood function

$$\log p(\mathbf{y}|\boldsymbol{\Theta}) = -\frac{1}{2}\mathbf{y}^T \cdot \mathbf{K}^{-1} \cdot \mathbf{y} - \frac{1}{2}\log\det\mathbf{K} - \frac{N}{2}\log 2\pi \tag{2.42}$$

  where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N)$ is the design matrix, $\mathbf{y}$ is the observation vector and $\mathbf{K}$ is the covariance matrix whose elements are

$$k(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \sigma^2 \left(1 + \frac{\sqrt{3}||\boldsymbol{\theta} - \boldsymbol{\theta'}||_2}{l}\right) e^{-\frac{\sqrt{3}||\boldsymbol{\theta}-\boldsymbol{\theta'}||_2}{l}} \tag{2.43}$$

  where $|| \cdot ||_2$ indicates the 2-norm. The final step consists in the proposal of a new point $\boldsymbol{\theta'}$ to evaluate the cost function such that the expected improvement is maximized. The point $\boldsymbol{\theta'}$ is added to the dataset and the procedure is repeated until convergence

- **Basin-hopping** [39] is a stochastic global search algorithm. It combines local optimization with a global-like perturbation of the candidate solution. In particular, given the starting point x, the algorithm purpose a new candidate solution x' which is then perturbed. The perturbation allows to hop from the current basin of the landscape to another one, where a global optimum may be present. The perturbed candidate solution is accepted with a probability which depends on a "temperature" parameter which is lowered at each step so that, at the very beginning of the optimization routine, jumps across different basins are more probable while, at the end, the algorithm becomes more and more selective

- **Simulated Annealing** [43] has some similarities with the Basin-hopping algorithm. In fact, the main idea is to purpose a new candidate solution located in the neighborhood of the starting point and accept that solution with a probability related to a decreasing "temperature" parameter. Nonetheless, unlike basin-hopping, the acceptance probability depends also on the energy difference between the starting and the candidate points and no local optimization is performed

2.4.3 Comparison between global optimizers

A comparison of the performances of the four global optimizers described in the previous section is presented in [35]. Tibaldi et al. asked the optimizers to solve the MaxCut problem for the 10 nodes graph in fig. 2.10. The outcome of their analysis is depicted in fig. 2.11. In particular, panel (a) shows the number of calls N_c to the quantum circuit as a function of the circuit depth p to get the same approximation ratio of the Bayesian optimizer. Panels (b) and (c) show, instead, the complement to 1 of the approximation ratio $1-R$ and the fidelity F as a function of the number of optimization steps, averaged over 30 runs, where $p=7$. It is evident that the Bayesian optimizers outperforms the other algorithms from every point of view. In fact, it is capable to achieve a higher approximation ratio (or a smaller $1-R$, equivalently) and a higher fidelity than other algorithms, being equal the number of steps. The other three optimizers need a much larger number of steps to reach the same approximation ratio. The Differential Evolution is, on the contrary, the worst in the large steps limit, reaching the smallest approximation ratio and fidelity among the set of tested algorithms.
Another interesting feature of bayesian optimizer is worth to be highlighted. Fig. 2.12 shows the approximation ratio and the fidelity got for the same MaxCut problem as before as a function of the number of measurements (or shots) N_s performed at the end of QAOA to reconstruct the final state. $1/N_s = 0$ corresponds to the approximation ratio and fidelity got by means of an exact simulation, i.e. without performing any measurement but taking the vector form of the final QAOA state. The remarkable point consists in the fact that in passing from $N_s = 1024$ to $N_s = 128$ the approximation ratio reduces of only the 5% and the same occurs passing from $N_s = 128$ to $N_s = 16$. The

relatively small number of shots needed to reconstruct the quantum state could be seen as a massive asset from the perspective of an application of the QAOA on NISQ devices, since it allows to reduce the computational effort needed to reconstruct the final solution.

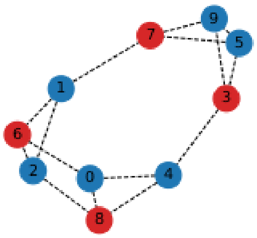


Figure 2.10: 10 nodes graph used to compare the performances of global optimizers in [35]

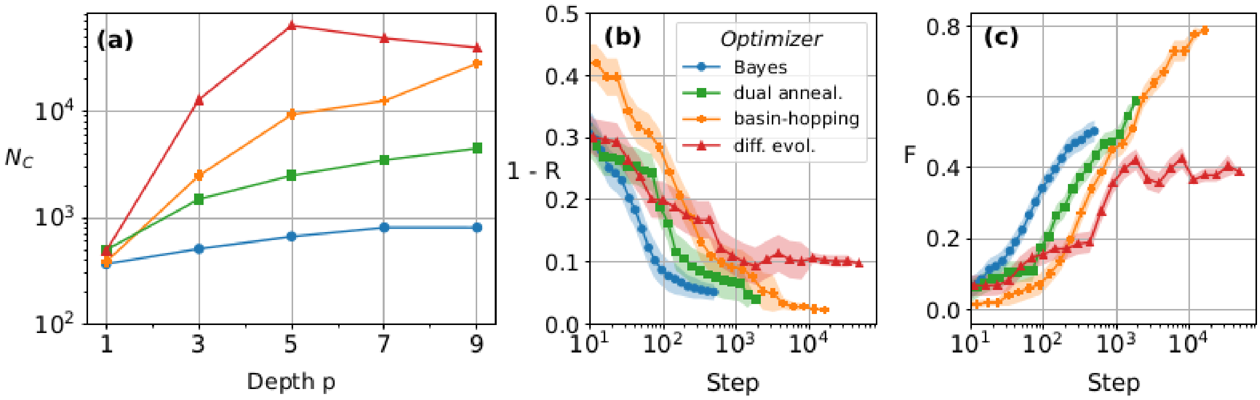


Figure 2.11: (a) panel shows the number of calls N_c to the quantum circuit for the Bayesian optimizer, the Basin-Hopping, the Simulated Annealing and the Differential evolution in order to obtain the same approximation ratio of the Bayesian algorithm as a function of the circuit depth [35]. (b) and (c) panels display the complement to 1 of the approximation ratio, i.e. $1 - R$, and the fidelity F got by the aforementioned four optimizers averaged over 30 runs for $p = 7$ as a function of the number of the optimization steps, respectively. Those data refers to the solution of the graph in fig. 2.10

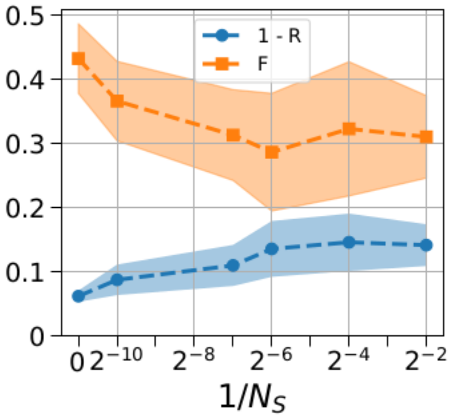


Figure 2.12: Approximation ratio (plotted as $1 - R$) and fidelity obtained by perform N_s measurements of the final QAOA state for the same MaxCut problem on the graph of fig. 2.10 [35]. $1/N_s = 0$ correspond to the approximation ratio and fidelity got with exact simulations. Taking $N_s = 128$ shots reduces the approximation ratio only of a 5% with respect to $N_s = 1024$ and the same occurs passing from $N_s = 128$ to $N_s = 16$

2.4.4 Initialization strategies

As mentioned in the previous section, local optimizers may get trapped in local minima when the energy landscape is populated by many local peaks. Furthermore, when the problem involves a lot of qubits or the circuit depth becomes considerable, an additional annoying issue comes out: the presence of barren plateaus in the landscape. Barren plateaus are regions where the gradient of the objective function becomes negligible, hindering local optimizers in performing their task [1]. In order to avoid these inconvenients, several initialization strategies have been developed [20][21][24]. The term ”initialization strategy” means an elaborate way to initialize the QAOA parameters with the idea of massively reducing the probability of failure of the optimization procedure. Most of the initialization strategies are depth-progressive, meaning that the initial parameter at depth p are computed starting from the optimal parameters at depth $p - 1$. The most recent initialization strategies are listed below:

- **Parameters fixing strategy(PFS)** [20] works as follows:

  1. Start from $p = 1$ and choose two random parameters (γ_1, β_1);
  2. Optimize the two parameters at $p = 1$;
  3. Repeat until the desired depth:
     (a) Generate $M = O(10)$ random instances of two additional parameters at depth $p + 1$, $(\gamma_{p+1}, \beta_{p+1})$;

  - (b) For each instance, evaluate the expectation of the objective using the parameters $(\gamma_1, \beta_1, ..., \gamma_{p+1}, \beta_{p+1})$ and choose the best;
  - (c) Optimize the full list of parameters.

  The parameters fixing strategy, as will be discussed later on, provides a very high quality solution. The drawback is the considerable amount of computational resources required to perform such a routine

- **Layerwise approach** [21] is similar to PFS but there is one fundamental difference which negatively affects the solution quality: only the last two parameters are optimized at each depth step. The main advantage of this procedure is the time saving in running the algorithm

- **Bilinear strategy** [21] works as follows:

  1. Choose random angles and find the optimum for $p = 1$ and $p = 2$. In order to do this one may follow the same steps of PFS up to $p = 2$;
  2. Start from $p = 3$ and repeat until the desired depth:
     - (a) Define $\Delta^p_{j,j-1} = \phi^p_j - \phi^p_{j-1}$;
     - (b) Define $\Delta^{p,p-1}_j = \phi^p_j - \phi^{p-1}_j$;
     - (c) $\forall j \leq p-2, \phi^p_j = \phi^{p-1}_j + \Delta^{p-1,p-2}_j$;
     - (d) $\phi^p_{p-1} = \phi^{p-1}_{p-1} + \Delta^{p-1,p-2}_{p-1}$;
     - (e) $\phi^p_p = 2\phi^p_{p-1} - \phi^p_{p-2}$;
     - (f) Optimize the full set of parameters.

  The symbol ϕ^p_j stands for the j-th γ or β parameter at depth p. Since the initial parameters are computed directly from optimal parameters at depth $p-1$ without querying the quantum circuit, the computational time is expected to be smaller than that needed by PFS

- **INTERP** algorithm [24] takes its name by the fact that the optimal parameters at depth p are interpolated in order to generate the initial parameters at depth $p+1$. The procedure works as follows:

  1. Start by choosing a guess for $p = 1$ parameters (γ_1, β_1);
  2. Optimize $p = 1$ parameters;
  3. Repeat up to the desired depth:
     - (a) $\forall j \leq p+1, \phi^{p+1}_j = \frac{j-1}{p}\phi^p_{j-1} + \frac{p-j+1}{p}\phi^p_j$
     - (b) Optimize the above parameters

2.4.5 Performances of initialization strategies

In this section we'll analyze the performance of the aforementioned initialization strategies in various scenarios.

Approximation ratio

The parameters fixing strategy has been compared to a random start routine on the solution of the MaxCut problem on the whole set of 3-regular graphs with $n = 6, 8, 10, 12, 16$, though the total number of instances is not clearly specified [20]. The results are shown in fig. 2.13 where the average approximation ratio, taken as the mean of the approximation ratio got for every graph instance, is plotted as a function of the circuit length. The local optimizer chosen was Nelder-Mead. The approximation ratio increases with p when PFS is applied. On the contrary, a random start approach cannot guarantee such a monotonic behavior of the approximation ratio. Furthermore, at large p, the approximation ratio is clearly better when using PFS and it approaches 1. The standard deviation related to the average appoximation ratio is, instead, showed in fig. 2.14. At large p the standard deviation decreases, meaning that the algorithm is capable to correctly identify a near-optimal solution in most of the graph instances.

The bilinear strategy, which has been developed later than PFS, has been compared to both PFS and the layerwise approach in terms of both approximation ratio and required computational resources on 20 instances of 3-regular graphs and 11 instances of 4-regular graphs [21]. The optimizer used in this analysis was L-BFGS-B. The results are shown in fig. 2.15. The approximation ratio got using the bilinear strategy matches the one obtained with PFS. A remarkable difference between the two strategies is clearly visible in panel (c) and (d) of fig. 2.15. As one can notice, the number of optimization steps n_{fev} required by the bilinear strategy is significantly smaller compared to PFS. For what layerwise approach is concerned, the average approximation ratio is smaller than the one obtained with the other two methods. As mentioned in the previous section, because in the layerwise approach only the last two parameters are optimized in each step, it is expected that this method does not achieve the desired solution quality though the computational time needed for the optimization is, of course, smaller than that needed by the other two approaches for sufficiently large p.

Smooth optimal solutions

Typically, when applying such depth-progressive initialization strategies, the optimal solutions come out to be "smooth" in the layer index. The analysis performed by Mele et al. [24] shows exactly this peculiar feature of optimal solutions obtained via INTERP for the ground state preparation of Heisenberg XYZ and longitudinal transversal field

Ising model (LTFIM) whose hamiltonians are:

$$H_{XYZ} = \sum_{j=1}^{N} (\sigma_j^x \sigma_{j+1}^x + \Delta_y \sigma_j^y \sigma_{j+1}^y + \Delta_z \sigma_j^z \sigma_{j+1}^z) \tag{2.44}$$

$$H_{LTFIM} = \sum_{j=1}^{N} \sigma_j^z \sigma_{j+1}^z + g_x \sum_{j=1}^{N} \sigma_j^x + g_z \sum_{j=1}^{N} \sigma_j^z \tag{2.45}$$

with $\Delta_y, \Delta_z, g_x, g_z > 0$. The initial $p = 1$ parameters have been chosen to be $(\gamma_1, \beta_1) = (0.1, 0.1)$ in every analysis, including the one presented in the following paragraph, since the authors verified that this choice allows a convergence to a well-defined minimum in the $p = 1$ search space. Smooth optimal solution for both models with $p = 16$ and different system sizes are shown in fig. 2.16.
The optimal solutions are not only evidently smooth in the layer index, but the smoothness can be also seen when comparing two lines referring to two neighbor system sizes, e.g. $N = 16$ and $N = 14$, or $N = 12$ and $N = 10$. The two lines are, in fact, almost overlapping each other. The increasing or decreasing trend of the smooth parameters is, instead, dictated by the model.

Transferability to larger graphs

Still in the context of INTERP strategy [24], the optimal solutions remain almost the same varying N, suggesting a good transferability to larger systems. This type of transferability presents a main difference with the one discussed in sec. 2.2.4. In fact, transferring optimal parameters to a larger graph, with the idea of speeding-up the computation, can be actually seen as a process where what changes is the number of vertices of the graph but not its substantial nature, since we are still handling rings. However, because the donor small graph and the larger acceptor are still 2-regular graphs, the inferences related to a good transferability of parameters discussed in sec. 2.2.4 keeps holding. Furthermore, transferring optimal smooth solutions to systems with different size allows to avoid barren plateaus in the energy landscape as proved in [24]. Fig. 2.17 shows the average gradient evaluated around the optimal smooth solution found for an $N = 8$ system, precisely in the ϵ-neighborhood of the solution, with $\epsilon = 0.05$ and globally, as the average of the gradient in 1000 random points of the search space. The global gradient exponentially decreases as N gets larger and this represents the evidence of the presence of barren plateaus in the landscape. On the contrary, around transferred smooth solution the gradient is almost constant at a non-zero value as p gets larger and decreases very slowly with N. As a consequence, a local optimization procedure on the larger system would probably succeed. Smooth solutions can be exploited as an educated guess for a further refinement optimization.

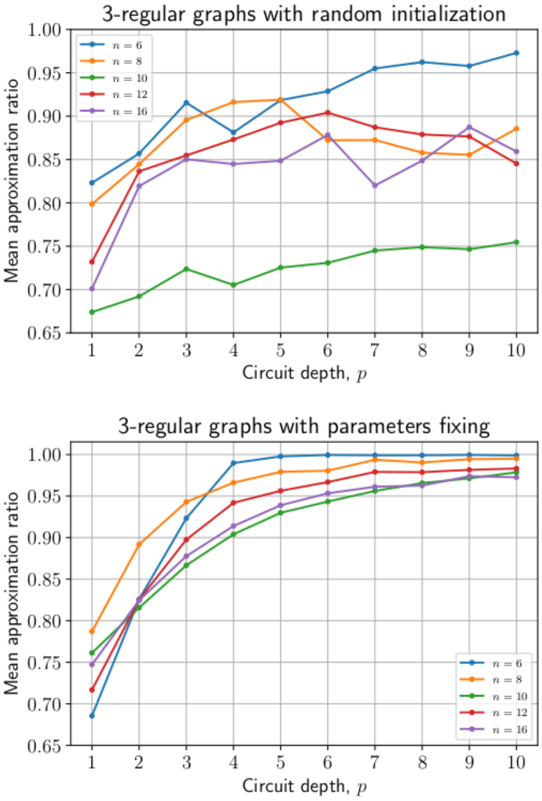


Figure 2.13: Mean approximation ratio as a function of the circuit depth for both the random start procedure and the parameters fixing strategy for the solution of MaxCut on the whole set of 3-regular graphs with $n = 6, 8, 10, 12, 16$ [20]. The mean value is computed by averaging the approximation ratio got for every graph instance. Approximation ratio is clearly better when PFS is applied. Moreover, the random start routine does not guarantee that increasing the depth leads to a larger approximation ratio. When PFS is applied, instead, the behavior of the approximation ratio is monotonic

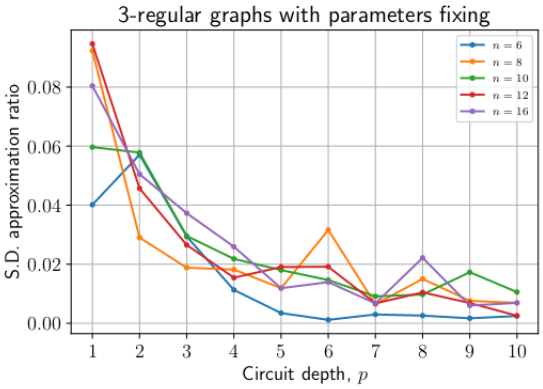


Figure 2.14: Standard deviation of the approximation ratio in with parameters fixing strategy [20]. The standard deviation decreases when p is large, that is to say, when the average approximation ratio approaches 1. Therefore, at large p, the algorithm becomes more capable to identify the best solution in most graph instances

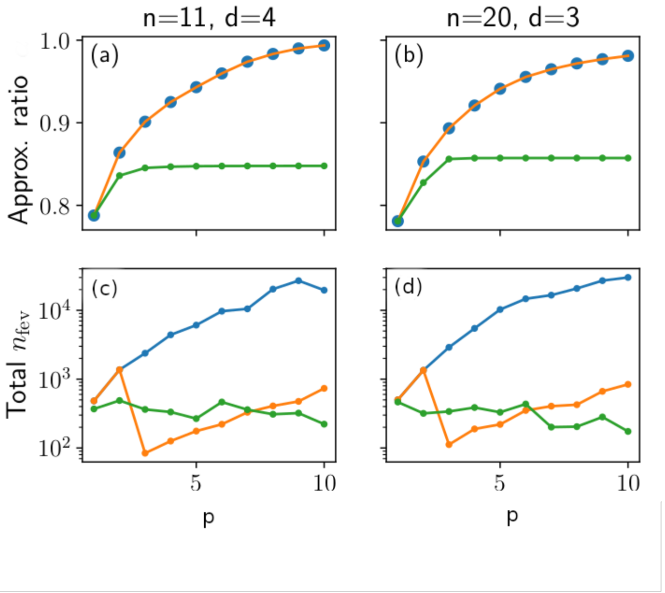


Figure 2.15: Comparison of the results of PFS, layerwise approach and the bilinear strategy on 20 random instances of 3-regular graphs and 11 instances of 4-regular graphs [21]. Blue dots refer to PFS, the orange line represents the bilinear strategy and the green line stands for layerwise approach. n_{fev} is the number of optimization steps required by L-BFGS-B to locate the optimal solution. The approximation ratio obtained using the bilinear strategy matches the one obtained with PFS, though the computational time needed by the bilinear is much smaller compared to the time needed by PFS. Layerwise approach required a low computational time as low is the quality of the solution

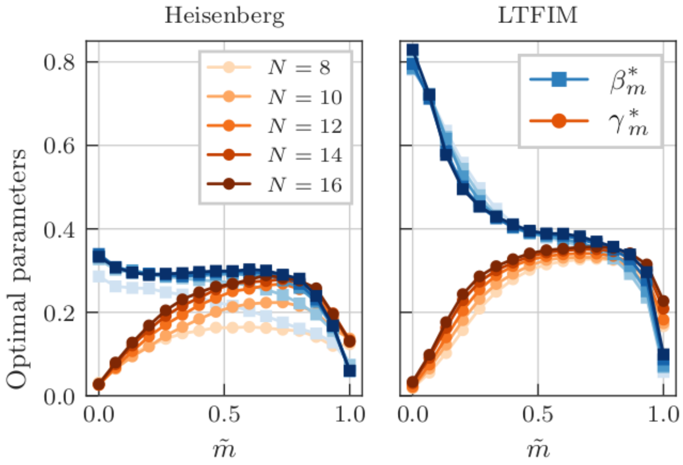


Figure 2.16: Optimal smooth solution $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ for both the Heisenberg and the LTFIM models with $p = 16$ and different system sizes obtained via INTERP [24]. On the x-axis the index $\tilde{m} = (m-1)/(p-1)$ is plotted where m is the layer index which ranges from 1 to p

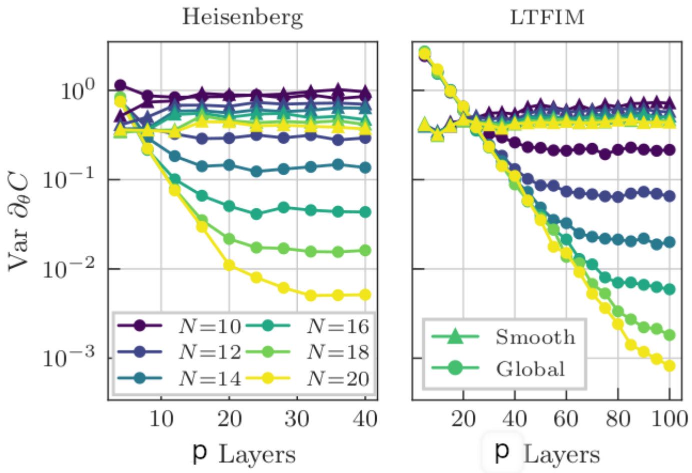


Figure 2.17: Single partial derivative of the cost hamiltonian in eq. (2.44,2.45) with respect to the parameter α_1 [24]. The derivative has been evaluated both in the ϵ-neighborhood (with $\epsilon = 0.05$) of the transferred smooth solution (denoted as "smooth") and globally, averaging the gradient in 1000 points all around the search space (denoted as "global"). The parameters used here are the optimal ones for a $N = 8$ spin chain

Chapter 3

The Ising model with transverse field

The XY model, which is a generalization of the simpler one dimensional quantum Ising model, is a milestone of quantum many body theory because of its rich phase diagram and the possibility of potentially calculating every single quantity of interest. This model became fundamental to test new conjectures, ideas and hypotheses including the ones coming from the world of quantum computation. In our simulations, discussed in the next chapter, we considered, in fact, the one dimensional Ising model with transverse field, which is a special subclass of the more general XY model [13].
In this chapter we present a general overview of the XY model listing its main properties, with a particular focus on the ones useful for our simulation purposes. We then show how to recover the exact solution of the XY chain by diagonalizing the hamiltonian and exploiting the Jordan-Wigner and the Bogoliubov transformations. We will then proceed to analyze the behavior of the spin chain in the ordered and disordered phases, with a deepening on the possible degeneracy of the ground state. In the second to last section of the chapter we briefly show how to compute the correlation functions. The very last section is designated to illustrate the specific model we worked in our simulations, i.e. the Ising model with transverse field, providing a brief overview of the most interesting properties for what quantum simulations is concerned.

3.1 General overview

The one dimensional XY model is the simplest non-trivial integrable model and its excitations are non-local free fermions. The two parameters which characterize the model are called γ and h. The former encodes the anisotropy of the interaction while the latter represents the external magnetic field. The phase diagram presents two quantum phase transitions: one belonging to the universality class of the Heisenberg chain, where $\gamma = 0$,

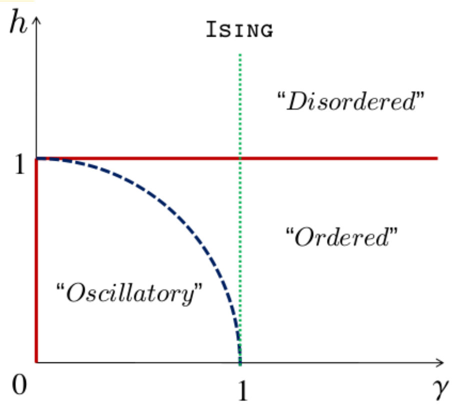


Figure 3.1: Phase diagram of the XY model [13]. The quantum phase transitions are located at the $\gamma = 0$ line and at the critical magnetic field $h = 1$. For $\gamma = 1$ we recover the 1D Ising model with transverse field

$|h| \leq 1$ and the model reduces to the XX model, and the second one belonging to the universality class of the Ising model, i.e. a simple Ising transition, where $|h| = 1$. We'll focus on the sector of the phase diagram where $\gamma, h \geq 0$, therefore dropping the sidebars of the absolute value for the magnetic field.
The hamiltonian of the XY model is displayed below:

$$H = \frac{J}{2} \sum_{j=1}^{N} \left[\left(\frac{1+\gamma}{2} \right) \sigma_j^x \sigma_{j+1}^x + \left(\frac{1-\gamma}{2} \right) \sigma_j^y \sigma_{j+1}^y + h\sigma_j^z \right]. \tag{3.1}$$

From now on, we will set $J = -1$, thus considering a ferromagnetic interaction.
The spins interact anisotropically with their nearest neighbors and with an external magnetic field on the $\hat{z}$ direction.
The model has two fundamental symmetries:

- A rotation of an angle $\pi/2$ about the $\hat{z}$ axis interchanges the roles of x and y and is equivalent to reverse the sign of γ, i.e. $\gamma \to -\gamma$;

- A reflection of the spin across the $x - y$ plane is compensated by reversing the sign of h, i.e. $h \to -h$.

It is straightforward to notice that the 1D Ising model can be recovered when $\gamma = 1$. Fig. 3.1 shows the phase diagram of the XY model. The two quantum phase transitions are located at the isotropic line $\gamma = 0$, $h \leq 1$ and on the critical magnetic field $h = 1$. The critical line, in particular, splits the phase diagram into two different regions: the *disordered phase* and the *ordered phase*. In the disordered phase the ground state is non-degenerate in the thermodynamic limit, while in the ordered region it is doubly-degenerate. The order parameter of the transition is the magnetization along the $\hat{x}$ axis,

$\langle \sigma^x \rangle$. It is identically zero in the disordered phase and non-vanishing m_x in the ordered phase. Consider the point where $(\gamma, h) = (1, 0)$, i.e. the Ising model with no field. Assuming period boundary conditions, the two ground states in the ordered phase are:

$$
\begin{aligned}
|GS\rangle_1 &= |\rightarrow\rightarrow\rightarrow ...\rangle = \prod_{j=1}^{N} \frac{1}{\sqrt{2}} (|\uparrow_j\rangle + |\downarrow_j\rangle) \\
|GS\rangle_2 &= |\leftarrow\leftarrow\leftarrow ...\rangle = \prod_{j=1}^{N} \frac{1}{\sqrt{2}} (|\uparrow_j\rangle - |\downarrow_j\rangle) \qquad (3.2)
\end{aligned}
$$

where the states $|\uparrow_j\rangle$ and $|\downarrow_j\rangle$ indicate the single spin state with positive or negative projection onto the $\hat{z}$ axis, respectively. In these states $\langle \sigma^x \rangle = \pm 1$. The exact degeneracy of the ground state is present only in the thermodynamic limit while for chains of finite length there is still an energy gap between them. However, the factorized form of the degenerate ground state is preserved on the line $\gamma^2 + h^2 = 1$ represented in blue in fig. 3.1. In fact, they can be always written as:

$$
\begin{aligned}
|GS\rangle_1 &= \prod_{j=1}^{N} (\cos\theta |\uparrow_j\rangle + \sin\theta |\downarrow_j\rangle) \\
|GS\rangle_2 &= \prod_{j=1}^{N} (\cos\theta |\uparrow_j\rangle - \sin\theta |\downarrow_j\rangle) \qquad (3.3)
\end{aligned}
$$

where $\cos^2(2\theta) = \frac{1-\gamma}{1+\gamma}$. The peculiarity of this line is that the degeneracy is exact for any length of the chain. On the bicritical point, i.e the intersection point between the isotropic and the critical line $(\gamma, h) = (0, 1)$, the energy spectrum becomes perfectly quadratic.

Now let's look at the step-by-step solution to algebraically recover these interesting features [13].

3.2 Diagonalization of the hamiltonian

Since we are treating a one-dimensional spin chain, we can exploit the so called *Jordan-Wigner transformations* which map a spin chain into a chain of spinless fermions. The reason behind the employment of these transformation is that it is not convenient to work directly with spin variables because they are fermionic but commute bosonically between different sites. The price to pay is that the mapping is highly non-local.

3.2.1 Jordan-Wigner transformations

First of all, define:

$$\sigma^{\pm} = (\sigma^x \pm i\sigma^y)/2 \qquad (3.4)$$

Then proceed to define new fermionic variables ψ_j such that:

$$\begin{aligned} \sigma_j^+ &= e^{i\pi\sum_{l<j}\psi_l^\dagger\psi_l}\psi_j = \prod_{l=1}^{j-1}\left(1-2\psi_l^\dagger\psi_l\right)\psi_j \\ \sigma_j^- &= \psi_j^\dagger e^{-i\pi\sum_{l<j}\psi_l^\dagger\psi_l} = \prod_{l=1}^{j-1}\left(1-2\psi_l^\dagger\psi_l\right)\psi_j^\dagger \\ \sigma_j^z &= 1-2\psi_j^\dagger\psi_j \end{aligned} \tag{3.5}$$

Jordan-Wigner transformations map a spin up into an empty site and a spin down into an occupied site. The non-local part of the transfromation is called *Jordan-Wigner string* and fixes the anticommutation relation between sites of the spinless fermions chain, breaking also the translational invariance of the model by selecting site 1 as the starting point of the chain. The Jordan-Wigner string, i.e. the sum in the exponent in eq. (3.5), counts the number of fermions at the left with respect to site j. Rewriting the whole hamiltonian in terms of these new fermionic variables we get:

$$\begin{aligned} H = -\frac{1}{2}\sum_{j=1}^{N-1}\left(\psi_j^\dagger\psi_{j+1}+\psi_{j+1}^\dagger\psi_j+\gamma\psi_j^\dagger\psi_{j+1}^\dagger+\gamma\psi_{j+1}\psi_j\right) \\ +h\sum_{j=1}^{N}\psi_j^\dagger\psi_j-\frac{hN}{2}+\frac{\mu_N^x}{2}\left(\psi_N^\dagger\psi_1+\psi_1^\dagger\psi_N+\gamma\psi_N^\dagger\psi_1^\dagger+\gamma\psi_1\psi_N\right) \end{aligned} \tag{3.6}$$

where μ_N^x is defined as

$$\mu_N^x = \prod_{j=1}^{N}\left(1-2\psi_j^\dagger\psi_j\right) = \prod_{j=1}^{N}\sigma_j^z \tag{3.7}$$

and it's called *parity operator*. For the sake of clarity, the *parity* is defined as the evenness/oddness of the number of fermions in the chain. If the chain contains an even number of fermions, $\mu_N^x = 1$, otherwise it is -1.

Let's focus our attention for a while on the hamiltonian in eq. (3.6). The first two terms in the sum describe fermions hopping on a lattice and are hence called *hopping terms*. The latter two terms are the *interaction terms*. They're superconducting-like terms which create or destroy fermions in pairs. The terms multiplied by μ_N^x is the *boundary term* and can be disregarded in the thermodynamic limit. Nevertheless, it is important to establish the degeneracy of the model in the ordered phase. It is straightforward to notice that, if $\gamma \neq 0$, and, by the way, this is true also in the particular case of the Ising model with transverse field, the hamiltonian does not conserve the number of fermions. However, since fermions are created/destroyed in pairs, the total parity of the chain is invariant. As a consequence,

$$[\mu_N^x, H] = 0. \tag{3.8}$$

This peculiarity allows us to give separate descriptions of the system when the parity is even/odd. In particular, the hamiltonian can be written as:

$$H = \frac{1+\mu_N^x}{2} H^+ + \frac{1-\mu_N^x}{2} H^- \tag{3.9}$$

where $H^\pm$ have the same form as in (3.6) with $\mu_N^x = \pm 1$, respectively.
In order to satisfy the boundary terms, we must impose antiperiodic boundary conditions in the even particle sector and periodic boundary conditions in the odd particle sector, i.e.:

$$\begin{aligned} \psi_{j+N}^{(+)} &= -\psi_j^{(+)} \\ \psi_{j+N}^{(-)} &= \psi_j^{(-)}. \end{aligned} \tag{3.10}$$

Each sector is governed by the same hamiltonian but the Fock space is different due to the different boundary conditions.

3.2.2 Fourier transform

The next step is to Fourier transform the hamiltonian. In order to take into account the different boundary conditions each sector is governed by, we will sum over integers in the odd particle sector and over half-integers in the even particle sector. The Fourier index is therefore moving in the following ranges:

$$\begin{aligned} q \in \frac{1}{2}, \frac{3}{2}, ..., N - \frac{1}{2} \quad \mu_N^x &= +1 \\ q \in 0, 1, ..., N-1 \quad \mu_N^x &= -1. \end{aligned} \tag{3.11}$$

Now we are able to rewrite the Jordan-Wigner variables as Fourier sums in this way:

$$\begin{aligned} \psi_j^{(\pm)} &= \frac{1}{N} e^{i\frac{\pi}{4}} \sum_{q\in\Gamma_\pm} e^{i\frac{2\pi}{N}qj} \psi_q \\ \psi_q &= e^{-i\frac{\pi}{4}} \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}qj} \psi_j^{(\pm)} \end{aligned} \tag{3.12}$$

where $\Gamma_\pm$ are the ranges within which q belongs for $\mu_N^x = \pm 1$, respectively, as displayed in eq. (3.11). The hamiltonian in Fourier space becomes:

$$H^\pm = \frac{1}{N} \sum_{q\in\Gamma_\pm} \left[h - \cos\left(\frac{2\pi}{N} q\right) \right] \psi_q^\dagger \psi_q + \frac{\gamma}{2N} \sum_{q\in\Gamma_\pm} \sin\left(\frac{2\pi}{N} q\right) \left[\psi_q \psi_{-q} + \psi_{-q}^\dagger \psi_q^\dagger \right] - \frac{hN}{2}. \tag{3.13}$$

3.2.3 Bogoliubov transformations

The next step toward the diagonalization of the hamiltonian is to make use of the *Bogoliubov transformations* which are defined below:

$$\begin{aligned} \chi_q &= \cos\theta_q\,\psi_q - \sin\theta_q\,\psi^\dagger_{-q} \\ \chi_{-q} &= \cos\theta_q\,\psi_{-q} + \sin\theta_q\,\psi^\dagger_q \end{aligned} \tag{3.14}$$

where the angle θ_q, called the *Bogoliubov angle*, is defined such that it obeys the following relation:

$$\tan(2\theta_q) = \frac{\gamma\sin(\frac{2\pi}{N}q)}{h-\cos(\frac{2\pi}{N}q)}. \tag{3.15}$$

Basically, the Bogoliubov transformations define new fermionic variables so that the hamiltonian, in terms of them, becomes diagonal. In fact,

$$H^\pm = \frac{1}{N}\sum_{q\in\Gamma_\pm}\varepsilon\left(\frac{2\pi}{N}q\right)\left[\chi^\dagger_q\chi_q - \frac{N}{2}\right]. \tag{3.16}$$

The Bogoliubov variables describe so called *quasi-particles*, behaving as free fermions. The energy spectrum of quasi-particles is:

$$\varepsilon(\alpha) = \sqrt{(h-\cos\alpha)^2 + \gamma^2\sin^2\alpha}. \tag{3.17}$$

3.3 Even/odd particle sectors

After diagonalizing the hamiltonian, let's deepen our knowledge regarding the behavior of the system in the even and odd particle sectors by highlighting the main features characterizing each sector and emphasizing the differences between them. In particular, we'll focus on the ground states and their associated energies in order to understand how they look like. These in-depth analysis come out to be very useful for the contextualization of the results we got from the application of the QAOA to the Ising model.

3.3.1 Even particle sector

Since the spectrum is always positive, the ground state is simply defined as the state which can be annihilated by the destruction operator χ_q:

$$\chi_q|GS\rangle_+ = 0 \quad q = \frac{1}{2}, ..., N - \frac{1}{2}. \tag{3.18}$$

In the state $|GS\rangle_+$, where the subscript $+$ refers to the even particle sector, there are no quasi-particles. The vacuum of quasi-particles, expressed in terms of physical fermions, becomes:

$$|GS\rangle_+ = \prod_{q=0}^{(N-1)/2} \left(\cos\theta_{q+1/2} + \frac{1}{N} \sin\theta_{q+1/2} \psi^\dagger_{q+1/2} \psi^\dagger_{-q-1/2} \right) |0\rangle \tag{3.19}$$

where the state $|0\rangle$ is the vacuum state of physical fermions, annihilated by the operator ψ_q. The ground state energy is, instead:

$$E_0^+ = -\frac{1}{2} \sum_{q=0}^{N-1} \varepsilon \left[\frac{2\pi}{N} \left(q + \frac{1}{2} \right) \right] \xrightarrow{N\to\infty} -\frac{N}{2} \int_0^{2\pi} \frac{dq}{2\pi} \varepsilon(q) \tag{3.20}$$

where, in the last expression, the thermodynamic limit is taken.
The Hilbert space is generated by applying creation operators on the ground state. In order to maintain the parity and remain in the even particle sector, where the number of quasi-particles should be even, creation operators must be applied in pairs. Each excitation adds an energy equal to $\varepsilon(2\pi q/N)$. As a remark, the quasi-particles can be also called *elementary excitations.*

3.3.2 Odd particle sector

The odd particle sector is more interesting than the even particle sector because of the presence of zero mode elementary excitations. This causes the splitting of the phase diagram into the disordered and the ordered phase, which present a different degeneracy of the ground state. The two phases are separated by the critical line $h = 1$, across which an Ising transition occurs.
Intuitively, we may be tempted to define the ground state in the odd particle sector as the state annihilated by the operator χ_q where now q runs from 0 to $N-1$ in this way:

$$\chi_q |GS'\rangle_- = 0 \quad q \in 0, ..., N-1 \tag{3.21}$$

but one can consistently reply that this state is not allowed by the constraint of odd excitations. Therefore, the actual ground state in this sector is the one which involves the minimum odd number of excitations, i.e. 1, with the minimum energy, i.e. $q = 0$. This state has hence the following form:

$$|GS\rangle_- = \frac{1}{\sqrt{N}} \chi_0^\dagger |GS'\rangle_- = \frac{1}{\sqrt{N}} \psi_0^\dagger \sum_{q=1}^{N/2} \left(\cos\theta_q + \frac{1}{N} \sin\theta_q \psi_q^\dagger \psi_{-q} \dagger \right) |0\rangle. \tag{3.22}$$

The zero mode has to be treated separately. In fact, for $q = 0$ the superconducting-like interaction term in the hamiltonian vanishes and the physical fermionic operator

ψ_0 coincides with its corresponding Bogoliubov transform, i.e. χ_0. Moreover, the zero mode excitation carries an energy contribution of exactly $h-1$. As a consequence, the hamiltonian H^- has to be written as:

$$H^- = (h-1)\left[\frac{1}{N}\chi_0^\dagger\chi_0 - \frac{1}{2}\right] + \sum_{q=1}^{N-1}\varepsilon\left(\frac{2\pi}{N}q\right)\left[\frac{1}{N}\chi_q^\dagger\chi_q - \frac{1}{2}\right]. \tag{3.23}$$

The prefactor $h-1$ in the hamitonian can be either positive or negative and its sign has important consequences on the degeneracy of the ground state. Let's analyze in detail what happens both in the ordered and in the disordered phases.

Disordered phase

For $h > 1$:

$$h - 1 = \varepsilon(0) > 0 \tag{3.24}$$

and therefore the hamiltonian in eq. (3.23) is the same as the one displayed in eq. (3.16). The Hilbert space is, as well as the one of the even particle sector, generated by applications of pairs of creation operators such as $\chi_q^\dagger\chi_{q'}^\dagger$ or, alternatively, by application of a creation/destruction pair such as $\chi_q^\dagger\chi_0$, where we're replacing the zero momentum quasi-particle with another one with a higher energy.
The states generated in the disordered phase of the odd particle sector and the ones generated in the even particle sector intertwine in the thermodynamic limit and one can effectively disregard the separation into the two sectors. The ground state is therefore non-degenerate in the disordered phase.

Ordered phase

For $h < 1$:

$$h - 1 = -\varepsilon(0) < 0 \tag{3.25}$$

and the presence of the zero mode lowers the energy of the system which becomes:

$$E_0^- = \frac{1}{2}(h-1) - \frac{1}{2}\sum_{q=1}^{N-1}\varepsilon\left(\frac{2\pi}{N}q\right) \xrightarrow{N\to\infty} -\frac{N}{2}\int_0^{2\pi}\frac{dq}{2\pi}\varepsilon(q). \tag{3.26}$$

In the thermodynamic limit, the ground state with no excitations $|GS\rangle_+$ and the lowest energy state in this sector, i.e. $|GS\rangle_-$, become degenerate since $E_0^+ = E_0^-$. Furthermore, the gap between $|GS\rangle_+$ and $|GS\rangle_-$ closes exponentially in the system size N. In the ordered phase, the ground state is then doubly degenerate, meaning that there are two states with the same minimum energy.

3.4 Finite temperature behavior and correlation functions

For the sake of completeness of the description of the XY model, a brief discussion about the finite temperature behavior of the chain and the correlation functions of spins is presented in this section even though this concepts are not strictly of concern for what the thesis project is regarded.

3.4.1 Partition function at finite temperature

The partition function at finite temperature for $h < 1$, i.e. in the ordered phase, is given by:

$$\begin{aligned} \mathcal{Z} = \sum e^{-\beta E_i} = \\ = 2^{N-1} \left\{ \prod_{q=0}^{N-1} \cosh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q + \frac{\pi}{N} \right) \right] + \prod_{q=0}^{N-1} \sinh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q + \frac{\pi}{N} \right) \right] \right\} \\ + 2^{N-1} \left\{ \prod_{q=0}^{N-1} \cosh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q \right) \right] + \prod_{q=0}^{N-1} \sinh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q \right) \right] \right\} \end{aligned} \tag{3.27}$$

Taking the thermodynamic limit, the free energy per site comes out to be:

$$\begin{aligned} \mathcal{F} = -\frac{1}{\beta} \lim_{N \to \infty} \left(-\frac{1}{N} \ln \mathcal{Z} \right) = \\ = -\frac{1}{\beta} \ln 2 - \frac{1}{\pi \beta} \int_0^{\pi} \ln \cosh \left(\frac{\beta}{2} \varepsilon(\omega) \right) d\omega - \frac{1}{\beta} \lim_{N \to \infty} \frac{1}{N} \ln \left[1 + \prod_{q=0}^{N-1} \tanh \frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q \right) \right], \end{aligned} \tag{3.28}$$

where the last term, which encodes the degeneracy of the model, is negligible in the thermodynamic limit.
In the disordered phase, i.e. when $h > 1$, similar expressions for both the partition function and the free energy can be recovered and they are even simpler since the degeneracy term is not present. They read:

$$\begin{aligned} \mathcal{Z} = 2^{N-1} \left\{ \prod_{q=0}^{N-1} \cosh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q + \frac{\pi}{N} \right) \right] + \prod_{q=0}^{N-1} \sinh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q + \frac{\pi}{N} \right) \right] \right\} \\ + 2^{N-1} \left\{ \prod_{q=0}^{N-1} \cosh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q \right) \right] - \prod_{q=0}^{N-1} \sinh \left[\frac{\beta}{2} \varepsilon \left(\frac{2\pi}{N} q \right) \right] \right\} \end{aligned} \tag{3.29}$$

$$\mathcal{F} = -\frac{1}{\beta}\ln 2 - \frac{1}{\pi\beta}\int_0^\pi \ln\cosh\left(\frac{\beta}{2}\varepsilon(\omega)\right)d\omega. \tag{3.30}$$

From the partition function and the free energy, all the other thermodynamic quantities can be computed obtaining that the whole thermodynamics of the model is essentially that of free fermions as expected.

3.4.2 Correlation functions at zero temperature

Since every n-point correlation function can be expressed as a product of 2-point correlation functions by means of the Wick's theorem, we'll thus focus our attention to the latter. Because the destruction operator χ_q annihilates the vacuum of quasi-particles $|GS\rangle = (|GS\rangle_+ + |GS\rangle_-)/\sqrt{2}$, the only non-zero two-point function, in terms of the Bogoliubov operators, is:

$$\langle GS|\chi_q\chi_k^\dagger|GS\rangle = N\delta_{q,k}. \tag{3.31}$$

Now, rewriting the χ_q operators in terms of the physical fermions in Fourier space, exploiting reverse Bogoliubov transformations, the two-point functions become:

$$\begin{aligned}
\langle GS|\psi_q^\dagger\psi_k|GS\rangle &= \frac{1-\cos\theta_q}{2}N\delta_{q,k}\\
\langle GS|\psi_q\psi_k^\dagger|GS\rangle &= \frac{1+\cos\theta_q}{2}N\delta_{q,k}\\
\langle GS|\psi_q\psi_k|GS\rangle &= -\frac{\sin 2\theta_q}{2}N\delta_{q,-k}\\
\langle GS|\psi_q^\dagger\psi_k^\dagger|GS\rangle &= \frac{\sin 2\theta_q}{2}N\delta_{q,-k}
\end{aligned} \tag{3.32}$$

where θ_q is the Bogoliubov angle defined in eq. (3.15).
By Fourier transform we can get the correlators in coordinate space. In the thermodynamic limit they read:

$$\begin{aligned}
F_{jl} = i\langle GS|\psi_j\psi_l|GS\rangle = -i\langle GS|\psi_j^\dagger\psi_l^\dagger|GS\rangle &= \int_0^{2\pi}\frac{dq}{2\pi}\frac{\sin 2\theta(q)}{2}e^{iq(j-l)}\\
G_{jl} = \langle GS|\psi_j\psi_l^\dagger|GS\rangle &= \int_0^{2\pi}\frac{dq}{2\pi}\frac{1+\cos 2\theta(q)}{2}e^{iq(j-l)},
\end{aligned} \tag{3.33}$$

where the function $\theta(q)$ is defined as $\theta(q) = \frac{1}{2}\arctan\frac{\gamma\sin q}{h-\cos q}$ and it represents the continuum limit of the Bogoliubov angle.
At this point we need to take into account for the Jordan-Wigner transformation in order to get the spin-spin correlators. First of all, let's define the generic expectation value:

$$\rho_{lm}^\nu = \langle GS|\sigma_l^\nu\sigma_m^\nu|GS\rangle \tag{3.34}$$

where $\nu \in \{x, y, z\}$. In terms of spin creation and annihilation operators, each possible correlator becomes:

$$\begin{aligned} \rho^x_{lm} &= \langle GS| \left(\sigma^+_l + \sigma^-_l\right) \left(\sigma^+_m + \sigma^-_m\right) |GS\rangle \\ \rho^y_{lm} &= -\langle GS| \left(\sigma^+_l - \sigma^-_l\right) \left(\sigma^+_m - \sigma^-_m\right) |GS\rangle \\ \rho^z_{lm} &= \langle GS| \left(1 - 2\sigma^+_l \sigma^-_l\right) \left(1 - 2\sigma^+_m \sigma^-_m\right) |GS\rangle. \end{aligned} \tag{3.35}$$

In terms of the Jordan-Wigner operators ψ_j, the spin-spin correlators can be rewritten as expectation values of products a string of operators:

$$\begin{aligned} \rho^x_{lm} &= \langle GS|B_l A_{l+1} B_{l+1} ... A_{m-1} B_{m-1} A_m|GS\rangle \\ \rho^y_{lm} &= (-1)^{m-1} \langle GS|A_l B_{l+1} A_{l+1} ... B_{m-1} A_{m-1} B_m|GS\rangle \\ \rho^z_{lm} &= \langle GS|A_l B_l A_m B_m|GS\rangle, \end{aligned} \tag{3.36}$$

where $A_l = \psi_l + \psi^\dagger_l$ and $B_l = \psi_l - \psi^\dagger_l$.
It is straightforward to check that $\langle GS|A_l A_m|GS\rangle = \langle GS|B_l B_m|GS\rangle = 0$. Therefore, using the Wick's theorem as announced at the beginning of this section, one can express the zz correlator in the following way:

$$\rho^z_{lm} = H^2(0) - H(m-l)H(l-m) \tag{3.37}$$

where the function $H(m-l)$, which has nothing to do with the hamiltonian though the same letter H has been used, is defined as:

$$H(m-l) = \langle GS|B_l A_m|GS\rangle = \frac{1}{2} \int_0^{2\pi} \frac{dq}{2\pi} e^{2i\theta(q)} e^{iq(m-l)}. \tag{3.38}$$

The other two correlators, namely the xx and yy correlators, involving a string of $m-l$ operators, can be expressed as the determinant of $(m-l) \times (m-l)$ matrices called *Toeplitz matrices.* They read:

$$\rho^x_{lm} = \begin{vmatrix} H(-1) & H(-2) & \cdots & H(-n) \\ H(0) & H(-1) & \cdots & H(1-n) \\ \vdots & \vdots & \ddots & \vdots \\ H(n-2) & H(n-3) & \cdots & H(-1) \end{vmatrix} \tag{3.39}$$

$$\rho^y_{lm} = \begin{vmatrix} H(1) & H(0) & \cdots & H(2-n) \\ H(2) & H(1) & \cdots & H(3-n) \\ \vdots & \vdots & \ddots & \vdots \\ H(n) & H(n-1) & \cdots & H(1) \end{vmatrix}. \tag{3.40}$$

The behavior of all the correlation functions are displayed in fig. 3.3 and fig. 3.4 in terms of the length-scale parameters $\lambda_\pm$ which are defined as:

$$\lambda_\pm = \frac{h \pm \sqrt{\gamma^2 + h^2 - 1}}{1 + \gamma} \tag{3.41}$$

and they are the zeros of the extension to the complex plane of the energy spectrum in eq. (3.17). The logarithms of such parameters give the correlation lengths of the chain, in unit of the lattice spacing a:

$$\xi_\pm = \frac{a}{|\ln \lambda_\pm|}. \tag{3.42}$$

The behavior of the $\lambda_\pm$ parameters in the different phases of the system is, instead, displayed in fig. 3.2.

In the disordered phase, both $\lambda_\pm$ are real, thus providing a measurable finite correlation length. The magnetization along the $\hat{x}$ axis is zero in this phase. At criticality, i.e. when $h = 1$, λ_+ becomes exactly equal to one and the correlation length approaches infinity, therefore entailing a scale-invariant behavior of the system. In the ordered phase, when $h^2 + \gamma^2 > 1$, the correlation length is finite and measurable again cause the $\lambda's$ are smaller than unity. In this phase, moreover, the system acquires a net magnetization along the $\hat{x}$ axis, i.e. $\langle \sigma^x \rangle = m_x \neq 0$. When, on the contrary, $h^2 + \gamma^2 < 1$ we are in the so called *oscillatory phase*. The name of this phase comes from the behavior of the xx correlator, which assumes a periodic modulation since $\lambda_\pm$ become complex conjugated. For what the other regions of the phase diagram is concerned, going to negative magnetic fields requires the inversion of the roles of λ_+ and λ_- while for negative γ we must replace $\lambda_\pm$ with their reciprocal $\lambda_\pm^{-1}$. Additional details on the XY model, which are beyond the scope of this thesis work, can be found in [13].

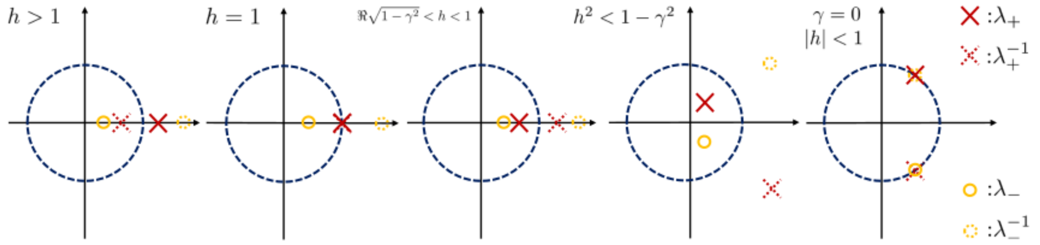
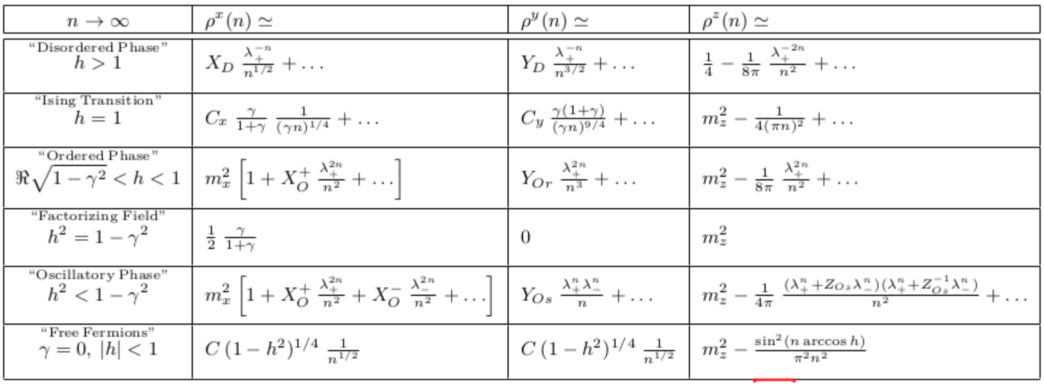
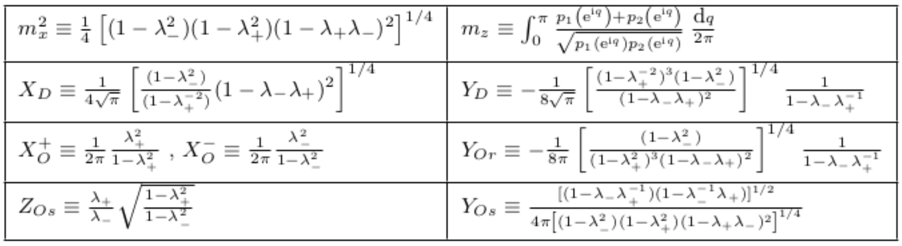


Figure 3.2: Positions of the length-scale parameters $\lambda_\pm$ with respect to the unit circle in the complex plane in the different phases of the system [13]



$n \to \infty$	$\rho^x(n) \simeq$	$\rho^y(n) \simeq$	$\rho^z(n) \simeq$
"Disordered Phase" $h > 1$	$X_D \, \frac{\lambda_+^{-n}}{n^{1/2}} + \ldots$	$Y_D \, \frac{\lambda_+^{-n}}{n^{3/2}} + \ldots$	$\frac{1}{4} - \frac{1}{8\pi} \, \frac{\lambda_+^{-2n}}{n^2} + \ldots$
"Ising Transition" $h = 1$	$C_x \, \frac{\gamma}{1+\gamma} \, \frac{1}{(\gamma n)^{1/4}} + \ldots$	$C_y \, \frac{\gamma(1+\gamma)}{(\gamma n)^{9/4}} + \ldots$	$m_z^2 - \frac{1}{4(\pi n)^2} + \ldots$
"Ordered Phase" $\Re\sqrt{1-\gamma^2} < h < 1$	$m_x^2 \left[1 + X_O^+ \, \frac{\lambda_+^{2n}}{n^2} + \ldots\right]$	$Y_{Or} \, \frac{\lambda_+^{2n}}{n^3} + \ldots$	$m_z^2 - \frac{1}{8\pi} \, \frac{\lambda_+^{2n}}{n^2} + \ldots$
"Factorizing Field" $h^2 = 1 - \gamma^2$	$\frac{1}{2} \, \frac{\gamma}{1+\gamma}$	0	m_z^2
"Oscillatory Phase" $h^2 < 1 - \gamma^2$	$m_x^2 \left[1 + X_O^+ \, \frac{\lambda_+^{2n}}{n^2} + X_O^- \, \frac{\lambda_-^{2n}}{n^2} + \ldots\right]$	$Y_{Os} \, \frac{\lambda_+^n \lambda_-^n}{n} + \ldots$	$m_z^2 - \frac{1}{4\pi} \, \frac{(\lambda_+^n + Z_{Os}\lambda_-^n)(\lambda_+^n + Z_{Os}^{-1}\lambda_-^n)}{n^2} + \ldots$
"Free Fermions" $\gamma = 0, \ \lvert h \rvert < 1$	$C\,(1-h^2)^{1/4} \, \frac{1}{n^{1/2}}$	$C\,(1-h^2)^{1/4} \, \frac{1}{n^{1/2}}$	$m_z^2 - \frac{\sin^2(n \arccos h)}{\pi^2 n^2}$

Figure 3.3: Asymptotic behavior of the two-point correlators [13]. $C_x = \frac{1}{2A^3}e^{1/4}2^{1/12}$, $C_y = -\frac{1}{32A^3}e^{1/4}2^{1/12}$, $C = \frac{1}{A^6}e^{1/2}2^{2/3}$ and $A \simeq 1.282$. The other prefactors are listed in the figure below



$m_x^2 \equiv \frac{1}{4}\left[(1-\lambda_-^2)(1-\lambda_+^2)(1-\lambda_+\lambda_-)^2\right]^{1/4}$	$m_z \equiv \int_0^\pi \frac{p_1(e^{iq})+p_2(e^{iq})}{\sqrt{p_1(e^{iq})p_2(e^{iq})}} \, \frac{dq}{2\pi}$
$X_D \equiv \frac{1}{4\sqrt{\pi}}\left[\frac{(1-\lambda_-^2)}{(1-\lambda_+^{-2})}(1-\lambda_-\lambda_+)^2\right]^{1/4}$	$Y_D \equiv -\frac{1}{8\sqrt{\pi}}\left[\frac{(1-\lambda_+^{-2})^3(1-\lambda_-^2)}{(1-\lambda_-\lambda_+)^2}\right]^{1/4} \frac{1}{1-\lambda_-\lambda_+^{-1}}$
$X_O^+ \equiv \frac{1}{2\pi}\frac{\lambda_+^2}{1-\lambda_+^2}$, $X_O^- \equiv \frac{1}{2\pi}\frac{\lambda_-^2}{1-\lambda_-^2}$	$Y_{Or} \equiv -\frac{1}{8\pi}\left[\frac{(1-\lambda_-^2)}{(1-\lambda_+^2)^3(1-\lambda_-\lambda_+)^2}\right]^{1/4} \frac{1}{1-\lambda_-\lambda_+^{-1}}$
$Z_{Os} \equiv \frac{\lambda_+}{\lambda_-}\sqrt{\frac{1-\lambda_+^2}{1-\lambda_-^2}}$	$Y_{Os} \equiv \frac{[(1-\lambda_-\lambda_+^{-1})(1-\lambda_-^{-1}\lambda_+)]^{1/2}}{4\pi\left[(1-\lambda_-^2)(1-\lambda_+^2)(1-\lambda_+\lambda_-)^2\right]^{1/4}}$

Figure 3.4: Dependence on the parameters $\lambda_\pm$ of the prefactors in the table above [13]

3.5 Ising model with transverse field

In the last section of this chapter dedicated to the review of the XY model and its peculiar features, we would like to stress the main characteristics of this model which are interesting from the point of view of the simulations and analysis performed during this thesis project whose results and outcomes are presented completely in the next chapter.

3.5.1 Hamiltonian

The model we have focused on in our simulations is the Ising model with transverse field. It is a particular case of the XY model where $\gamma = 1$. Furthermore, we considered the case where the interaction is set along the $\hat{z}$ direction and the magnetic field lies along the $\hat{x}$ axis, which is the reversal of the hamiltonian of the XY model we treated so far. At the end of the story, the expression of the hamiltonian we worked with is the following:

$$H = -\sum_{j=1}^{N} \sigma_j^z \sigma_{j+1}^z - h \sum_{j=1}^{N} \sigma_j^x \tag{3.43}$$

where we assumed periodic boundary conditions, i.e. $\sigma_{N+1}^z = \sigma_1^z$.
Replacing the $\sigma' s$ with their corresponding single qubit gates, just to rewrite the hamiltonian in a way which recalls the world of quantum computation, it becomes:

$$H = -\sum_{j=1}^{N} Z_j Z_{j+1} - h \sum_{j=1}^{N} X_j. \tag{3.44}$$

3.5.2 Phases and ground states

The Ising model with transverse field correspond to the green line on the phase diagram of fig. 3.1. The phase transition, as explained previously in this chapter, occurs for $h = 1$. For $h < 1$ the system is in the ordered phase and the ground state is doubly degenerate. On the contrary, for $h > 1$, the system is in the disordered phase and the ground state is non-degenerate. Those ground states can be possibly found analytically, by diagonalizing the hamiltonian. In two cases they can be easily written down without resorting to the diagonalization of the hamiltonian:

- When the external field $h = 0$, the model reduces to a ferromagnetic Ising without magnetic field. The non-commuting term vanishes and the ground states (they are two, since we are in the ordered phase), in the thermodynamic limit, are:

$$\begin{aligned} |GS\rangle_1 &= |\uparrow\uparrow\uparrow ... \uparrow\rangle = |000...0\rangle \\ |GS\rangle_2 &= |\downarrow\downarrow\downarrow ... \downarrow\rangle = |111...1\rangle \end{aligned} \tag{3.45}$$

  It is worth to highlight that these two ground state are equivalent to the states of eq. (3.2);

- When $h \to \infty$, the only relevant term in the hamiltonian is the one depending on the field and the model reduces to a chain of non-interacting spins exposed to an external magnetic field. The non-degenerate ground state in the thermodynamic limit is:

$$|GS\rangle = |\rightarrow\rightarrow\rightarrow ... \rightarrow\rangle = \prod_{j=1}^{N} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)_j = |+++...+\rangle. \tag{3.46}$$

When we consider short chains, i.e. far from the thermodynamic limit, the degeneracy of the ground state in the ordered phase is not present if the field is different from zero.

It is therefore possible to identify a unique ground state of the system even for $h < 1$. Nevertheless, as remarked in sec. 3.3.2, when the chain becomes longer and longer, the two "ground states" get exponentially closer in energy, though a correct identification of the ground state is still possible. However, to find such a state, analytical methods are required because the mathematical form of the ground state is much more complicated than the ones displayed above for the two simple cases $h \to 0$ and $h \to \infty$.

Chapter 4

QAOA applied to the Ising model

The last chapter of this thesis is dedicated to illustrate the outcomes of the simulations we performed. The main idea of the thesis project is to test the INTERP optimization schedule [24] for the Quantum Approximate Optimization Algorithm on the ground state preparation of the Ising model with transverse field [13]. The former has been described in detail in sec. 2.4.4, while the latter has been described in sec. 3.5. The main properties of the XY model, of which the Ising with transverse field is a particular case, are illustrated in chapter 3. However, a brief recap of the most valuable properties and peculiarities of both the physical model and the algorithm are provided also at the beginning of this chapter for the sake of clarity and completeness of the description. Since the Ising model with transverse field is exactly solvable, it represents a good candidate for the role of test model of new quantum algorithms, including initialization strategies for the QAOA, which include also the aforementioned INTERP algorithm. In particular, what we would like to prove by means of the simulations, is that the combination of initialization strategies, powerful local optimizers, correctly designed quantum circuits and possibly the transferability of optimal solutions mechanism, provide a clear and undeniable advantage from a computational time and solution quality points of view, with respect to standardized classical procedures for the application of the QAOA on the solution of quantum many body problems.

This chapter is organized as follows: sec. 4.1 is dedicated to a brief review of the INTERP algorithm and the main features of the Ising model with transverse field; sec. 4.2 describes the common methodologies of all the simulations we performed; in sec. 4.3 we show how to construct the QAOA quantum circuit; sec. 4.4 provides the smooth optimal solutions for the Ising model with $h = 0.5$ both listing them in a table and plotting them in a figure; sec. 4.5 displays a comparison between the most commonly-used local optimizers, proving the L-BFGS-B to outperform its challengers; sec. 4.6 shows that in the disordered phase of the Ising model, the QAOA at low depth is able to locate the real ground state anyway; sec. 4.7 provides the reconstruction of the quantum state via measurements; sec. 4.8 shows how the computational time and the optimization steps behave with respect to N and p; in sec. 4.9 the transferability of optimal solution is exploited to speed-up the optimization procedure; in sec. 4.10 a modified version of the INTERP strategy is presented as another tailored algorithm to speed-up the optimization; sec. 4.11 discuss the results from a wider perspective, looking also at further developments.

4.1 The INTERP strategy and the Ising model

The INTERP initialization strategy is a depth-progressive schedule. The meaning of *depth-progressive* is that the QAOA is applied, at the beginning of the procedure, at lowest depth, i.e. $p = 1$, optimizing the two parameters γ_1 and β_1 for that circuit. The initial choice of the lowest-depth parameters is, from now on, fixed to $(0.1, 0.1)$ since, as suggested in [24], it reduces the probability of failure of a good local optimizer. The depth is then progressively increased one by one until the desired depth is reached, and at each step an optimization of the parameters is performed. The optimal parameters at depth p are interpolated to compute the starting parameters for the optimization procedure at depth $p+1$. The main advantages provided by the INTERP algorithm are related to the avoidance of barren plateaus by the local optimizer which is then capable to correctly locate the optimal point without annoyance. Because of that, the quality of the solution is generally improved with respect to the one got applying a random optimization procedure. Furthermore, the application of the INTERP algorithm produces so called *smooth solutions*, where the optimal parameters vary slowly in the layer index. This fact, in principle, is not a symptom of some advantage, but it is a very peculiar feature of this kind of strategies and it will be taken into account as a characteristic we expect to reproduce in our simulations, where we apply the INTERP strategy on a slightly different model, the Ising with transverse field, than the ones in [24].
The Ising model with transverse field is a particular case of the XY model where the interaction is present along a single direction (for us it will be the $\hat{z}$ direction) and it is ferromagnetic, i.e. we take the hamiltonian in eq. (3.1) and choose $J = -1$ and $\gamma = 1$. The external magnetic field, instead, is directed along $\hat{x}$. Therefore the hamiltonian of the model is:

$$H = -\sum_{j=1}^{N} Z_j Z_{j+1} - h \sum_{j=1}^{N} X_j \tag{4.1}$$

where Z_j is the Pauli Z matrix acting on the spin on site j, X_j is the Pauli X matrix, N is the length of the spin chain, h is the magnetic field and periodic boundary conditions are assumed, so that the chain "closes" and resembles a ring. Notice that we have already translated the Pauli matrices into a gate notation. The main features of the Ising with transverse field are that:

- It can be exactly solved by diagonalizing the hamiltonian, as illustrated in chapter

3, though the diagonalization requires a lot of computational resources, because the dimension of the matrix scales exponentially in N, and it is then practically feasible only for small systems unless a supercomputer is available;

- It has two phases [13]: in the *disordered phase*, i.e. when $h > 1$, the ground state is non-degenerate and, in the large field limit, it becomes:

$$|GS\rangle = |+++...+\rangle = \prod_j \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle \right)_j \tag{4.2}$$

  In the *ordered phase*, i.e. $h < 1$, the ground state is doubly degenerate and, in the low field limit, it is:

$$\begin{aligned} |GS\rangle_1 &= |000...0\rangle \\ |GS\rangle_2 &= |111...1\rangle; \end{aligned} \tag{4.3}$$

- The degeneracy of the ground state in the ordered phase arises only in the thermodynamic limit, unless $h = 1$ or $h = 0$, where the ground state is degenerate for any length of the chain. It is therefore possible to identify a unique ground state also in the ordered phase but the energy gap between the two "ground states" closes exponentially in N. As a consequence, for very large systems, it becomes non-trivial to distinguish between them. Nevertheless, since we will work with small systems ($N \leq 14$), this inconvenient will not affect much our simulations.

As a remark, the reader is invited to notice that the hamiltonian in eq. (4.1) is made out of two non-commuting terms since the Z and X operators do not commute. The interaction term will generate the *cost layer* of the QAOA while the field term will produce the *mixer layer*. This will be discussed later on in this chapter.

4.2 General methodologies

In the thesis project, several types on analysis and simulations have been performed, each of them having a specific idea behind it. However, some common features characterize all the simulations. First of all, the simulations have been performed with Python programming language adopting the *Qiskit* library [19] for building up the QAOA. This library contains specific commands to construct gate-based (even parameterized) quantum circuits and perform many kind of analysis including optimization of parameters, which is what we are really after. Nonetheless, other possible equivalent choices, for what the programming language and the libraries is concerned, are available, e.g. the *Pennylane* library for Python [3].

The following common feature of all the simulations regards the initialization strategy

used to produce the results, which is always the INTERP algorithm, unless if specified differently. Furthermore, the QAOA circuit preparation is the same and a dedicated section will describe how the quantum circuit has been implemented and how does it look like.
An additional remarkable point is that for simulation purposes, the final state vector the QAOA returns is directly saved and no measurement process is performed, unless if specified differently. Since the optimization schedules are deterministic, if the same optimization is repeated multiple times under the same conditions, the algorithm will return the same optimal parameters employing the same number of optimization steps. The only feature which may be affected by statistical error is the computational time, and, in case the simulation involves the evaluation of computational time, the statistical error is shown.
Each subsequent section, besides the next one, which regards the implementation of the QAOA, contains the outcomes of a particular kind of analysis together with an exhaustive description of the specific methodologies adopted for that simulation.

4.3 QAOA circuit preparation

The construction of the correct QAOA circuit, which allows us to find the ground state of an Ising chain, follows five main steps:

1. **Embedding**;
2. Construction of the **cost layer**, also known as **entangling layer**;
3. Construction of the **mixer layer**;
4. Alternated juxtaposition of p cost layers and p mixer layers;
5. Possible insertion of **measurements**, if the reconstruction of the final state is made by averaging the outcomes of many repetitions of the algorithm as described in sec. 1.4.2. On the contrary, for simulation purposes, the exact final state vector can be directly saved. Nevertheless, this is not the standard procedure occurring on a real quantum device.

Each step will be described in detail in the following sections.

4.3.1 Embedding

With the term *embedding* [7] we define a procedure with which the physical system we would like to study, in our case the Ising spin chain, is somehow mapped onto the qubits in the QPU and the quantum state of the qubits is initialized in a precise way.

In our specific case, suppose the N spins belonging to the chain can be labeled as $s_0, s_1, s_2, ..., s_{N-1}$. The embedding mapping is quite straightforward, since each spin of the chain is mapped onto a single qubit, the latter labeled, for the sake of simplicity, as $q_0, q_1, q_2, ..., q_{N-1}$. Since the initial state of the qubits in the QAOA algorithm is chosen to be $|+++...+\rangle = |+\rangle^{\otimes N}$, an Hadamard gate is applied to each qubit as depicted in fig. 4.1.

4.3.2 Cost layer

Consider the hamiltonian in eq. (4.1). The cost layer of the QAOA is the evolution operator generated from the interaction term of this hamiltonian, i.e.

$$e^{i\gamma_i \sum_{j=1}^{N} Z_j Z_{j+1}} \tag{4.4}$$

where γ_i is the QAOA parameter related to that layer with the index $i = 1...p$ being p the depth of the circuit. Now, since the sum is a linear operator and the terms in the sum commute between each others, it can be brought out of the exponential becoming a product. Hence, we look at each single term of the product, i.e.

$$e^{i\gamma_i Z_j Z_{j+1}} \tag{4.5}$$

acting on qubits j and $j+1$. In order to implement this evolution operator involving two qubits, we exploit the universality of CNOTs and rotations to simplify down the evolution operator in the following way:

1. Apply a CNOT gate, where the control qubit is the j-th and the target is the $(j+1)$-th;
2. Perform a rotation about the $\hat{z}$ axis of an angle $2\gamma_i$ on the $(j+1)$-th qubit, i.e. the target of the CNOT;
3. Repeat step 1;

The above three steps must be repeated for each pair of neighbor qubits in order to implement all the evolution operators in the product where j ranges from 1 to N.

4.3.3 Mixer layer

The mixer layer is, instead, generated by the field term of the hamiltonian in eq. (4.1). Its mathematical expression is then

$$e^{ih\beta_i \sum_{j=1}^{N} X_j} \tag{4.6}$$

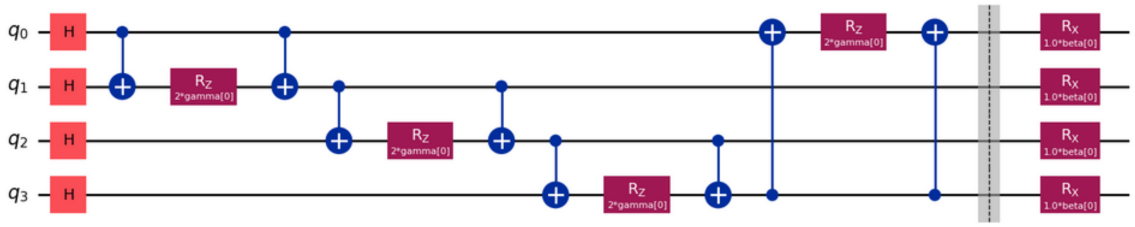


Figure 4.1: Preparation of the quantum circuit which implements the QAOA algorithm for $N = 4$, $p = 1$ and $h = 0.5$. First of all, we apply Hadamard gates to all the qubits in order to initialize the state to $|+\rangle^{\otimes N}$. We then apply a cost and a mixer layer where the former consists in two CNOTs spaced out by a rotation about the $\hat{z}$ axis, repeated for all the possible pairs of neighbor qubits, whereas the latter is made out of a simple rotation about the $\hat{x}$ axis of all the qubit. The rotation angles are proportional to the QAOA parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$. Further details are given in the main text

where h is the magnetic field and β_i is the QAOA parameter associated to that mixer layer. Again, transforming the exponential of a sum, because the terms in the sum commute between each others, into a product of one-qubit exponentials and singling out one of those terms we get the following evolution operator:

$$e^{ih\beta_i X_j} \tag{4.7}$$

acting on qubit j. The implementation of this operator is rather easier than the one of the cost layer since it involves only one qubit. It can be therefore simplified into a rotation about the $\hat{x}$ axis of an angle $2h\beta_i$ and it has to applied to all qubits to construct the whole layer.

4.3.4 Combination of many layers

Once the cost and the mixer layers have been built up, the only remaining step is to join together p cost layers and p mixer layers in an alternated fashion after the embedding, starting from a cost layer. Fig. 4.1 displays an example of QAOA circuit where $N = 4$, $p = 1$ and $h = 0.5$, hence we have four terms per each layer and two total layers, one cost and one mixer, respectively. Don't forget that the last term in the cost layer exists due to the periodic boundary conditions we chose.

4.3.5 Remark: desirable circuit depth and fidelity

There is a fundamental remark to point out before talking about the simulation we performed in this project and the results we got, that is to say, the desirable circuit depth to reach in order to correctly identify the ground state of the system and the

quantity we use to check the goodness of our predictions, namely the *fidelity*.
In the second chapter of this thesis, we stressed on an important feature regarding the QAOA algorithm in general which states that, in order to make good predictions, *the depth of the circuit should be properly chosen such that there are no closed loops whose length is larger than* $2p + 1$ [41]. Now, consider our 1D N-spins chain which, because of periodic boundary conditions, is closed and resembles a ring. The aforementioned conjecture provides an essential lower bound for what the circuit depth is concerned. In fact, it suggests that, in order to have a chance to get the actual vacuum state of the Ising chain, we must implement a circuit whose depth is at least $p \geq int(\frac{N}{2})$, where the function $int()$ selects the closest integer equal or greater than its argument.
The physical quantity we compute in order to get an estimate of the goodness of a given outcome of the algorithm is called *fidelity*. In principle, the fidelity between two quantum states, say $|\psi_1\rangle$ and $|\psi_2\rangle$ is defined as the squared modulus of the scalar product of the two states, hence:

$$F = |\langle\psi_1|\psi_2\rangle|^2. \tag{4.8}$$

For our simulation purposes, the fidelity is computed to check if the algorithm produced the exact ground state of the chain or not. The actual ground state can be directly computed via diagonalization of the hamiltonian and call it $|GS\rangle$. The QAOA circuit produces a final state which, in principle, may be the correct ground state or not, call it $|\psi_{QAOA}\rangle$. If the algorithm succeeded, we expect

$$F = |\langle GS|\psi_{QAOA}\rangle|^2 = 1. \tag{4.9}$$

If F is smaller than unity, we conclude that something went wrong and the procedure failed. The possible reasons of a failure include the non-optimal choice of the circuit depth and a fault of the optimization procedure which, for some reasons, fell through. In this sense, it is worth to keep stressing that some optimizers are better than other ones and nothing guarantees that all the optimization schedules, each of them implementing a different strategy, would succeed in locating the optimal point on the landscape.

4.4 Smooth optimal solutions

In this section we show that the application of the INTERP procedure produced optimal parameters which are smooth, in the sense clarified at the beginning of the chapter, also for the Ising model with transverse field. To prove the smoothness of the optimal parameters for the Ising model, we chose $h = 0.5$ and computed the optimal point through the QAOA with INTERP method, selecting the L-BFGS-B [22] as the local optimizer. The latter choice is not random, since, as we'll show later on in another section of this chapter, this particular optimizer comes out to perform the best among a set of 8 candidates local optimizers provided by the Python's package *scipy.optimize*

[38]. As an heads-up of what the reader will find in the dedicated section, the L-BFGS-B is always able to correctly identify the ground state of the Ising chain, at least for $N \leq 12$, provided a suitable circuit depth is chosen. It is worth to stress that, since the optimization schedule is deterministic and no random components are present and no statistical error affects the optimal parameters. Repeating the optimization several times under the same conditions, in fact, one effectively obtains the same exact parameters. This holds for all the local optimizers we will use to perform the simulations, but it is not true in general. In fact, for instance, the bayesian optimizer [35] and the differential evolution algorithm [31] involve a random component which lead to slightly different optimal parameters each run. As a consequence, the fidelity itself does not suffer from statistical error in our simulations. Table 4.1 lists the optimal parameters for different values of N and p. N ranges from 2 to 14 with step 2 while $p = \frac{N}{2}$ has been set equal to half of the chain length for the reason discussed in the previous section. With this circuit depth, the optimization always succeeded, as one can notice from the values of the infidelity, defined as $1 - F$, reported in the last column of table 4.1. The optimal parameters are truncated at the third decimal place for the sake of clarity. However, the infidelity displayed in the last column refers to the optimal parameters in their entirety, though the digits beyond the third decimal do not affect considerably the infidelity. The same parameters have been plotted in fig. 4.2 where the parameters are displayed as a function of the layer index. Hence, each line, which refers to a specific value of N, has a different length. The values of N and p a line refers to, can be deduced by the length of the line, which has to be equal to the circuit depth p, and N will be then its double. The smoothness of the parameters got for the Ising model is comparable to that got by Mele et al. [24] for the Heisenberg and longitudinal transversal field Ising models, displayed in fig. 2.16. It is worth to notice that, in each line of fig. 4.2, i.e. for every system size N, the parameter β_p, where $p = \frac{N}{2}$, is always much smaller than the other β parameters. This behavior is also reported in [24] (fig. 2.16).
Similar smooth solutions can be found for different values of the magnetic field. Since we have mainly worked with a field value of $h = 0.5$, we reported the optimal parameters for this specific case.

N	p	optimal parameters $(\beta_1, ..., \beta_p, \gamma_1, ..., \gamma_p)$	$1 - F$
2	1	(0.785, 0.277)	$\leq 10^{-9}$
4	2	(1.142, 0.815, 0.353, 0.489)	$\leq 10^{-9}$
6	3	(1.214, 1.194, 0.838, 0.376, 0.577, 0.534)	$\leq 10^{-9}$
8	4	(1.247, 1.268, 1.229, 0.852, 0.386, 0.604, 0.622, 0.551)	$\leq 10^{-9}$
10	5	(1.265, 1.303, 1.305, 1.250, 0.861, 0.392, 0.617, 0.649, 0.640, 0.559)	$\leq 10^{-8}$
12	6	(1.276, 1.323, 1.340, 1.327, 1.264, 0.866, 0.396, 0.625, 0.663, 0.667, 0.650, 0.565)	$\leq 10^{-8}$
14	7	(1.284, 1.336, 1.360, 1.363, 1.342, 1.274, 0.870, 0.399, 0.630, 0.671, 0.681, 0.677, 0.656, 0.568)	$\leq 10^{-8}$

Table 4.1: Optimal parameters for different values of N and $p = \frac{N}{2}$ for the Ising chain with $h = 0.5$. The optimal parameters are truncated at the third decimal place for the sake of clarity. However, the infidelity displayed in the last column refers to the optimal parameters in their entirety, though the digits beyond the third decimal do not affect considerably the infidelity

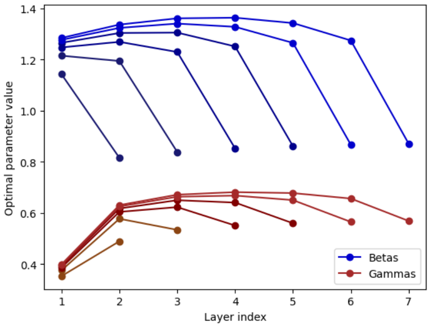


Figure 4.2: Optimal smooth parameters for the Ising chain with $h = 0.5$ for different values of N and $p = \frac{N}{2}$. The number of qubits ranges from 2 to 14 with step 2. The parameters are plotted as a function of the layer index. Hence, each line, which refers to a specific value of N, has a different length. The values of N and p a line refers to, can be deduced by the length of the line, which has to be equal to the circuit depth p, and N will be then its double

4.5 Comparison between local optimizers

Once verified that the INTERP procedure actually produces smooth solutions, we decided to focus our attention on the comparison between local optimizers, with the idea of establishing which optimizer performs the best and is the most reliable among the others. The reliability of the optimizer is a very significant factor to take into account in such kind of analysis which exploit hybrid algorithms. In fact, choosing a bad optimization schedule has massive negative consequences on the outcome of the analysis, which is probably doomed to a total failure.
There are many factors which characterize an optimizer from a practical point of view. We list the ones we deem the most valuable below:

- **Computational time**: we defined it as the time the optimizer needs to perform an optimization procedure. This parameter is primarily influenced by the complications of the optimization due to the complicated nature of the energy landscape, which may present many or few peaks or plateaus. Furthermore, the computational time is straightforwardly conditioned also by the power of the device on which we run the simulation. Nevertheless, if the simulation is run in the same conditions, the computational time becomes relevant to determine if a given algorithm is intrinsically slower or faster than another one;

- **Optimization steps**: It is the number of optimization steps the algorithm requires to perform the optimization. Though this parameter is not be strictly proportional to the time, because some optimization steps may be faster or slower than other ones, depending again on the landscape shape, it is an important parameter to be taken into account anyway. In fact, the number of steps of the optimization procedure is somehow an intimation of the computational resources required to perform such optimization;

- **Reliability**: it's time to provide a better definition of what we mean with "reliability" of an optimizer. It is defined as the capability of the algorithm to correctly identify the ground state of the model we're studying, provided a suitable circuit depth, regardless the number of qubits the circuit involves. In other words, the reliability is related to the maintenance of a unit fidelity for any value of N.

The aforementioned descriptors have been studied for a stocked set of local optimizers. This set contains the following algorithms:

- Nelder-Mead (NM);

- Powell;

- CG;

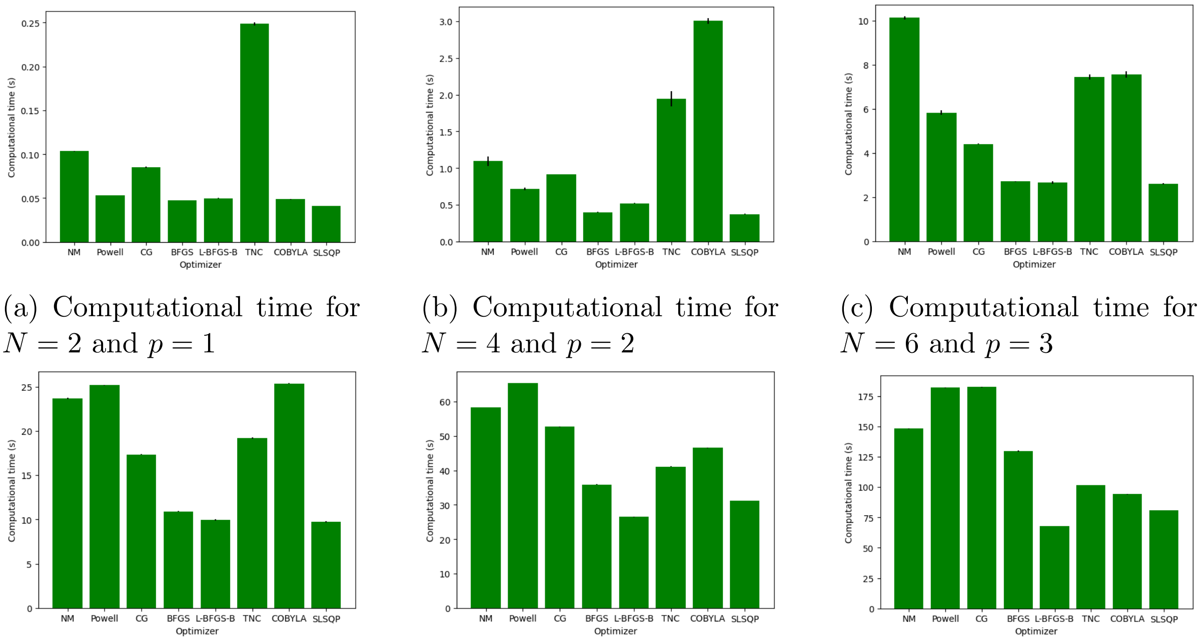
- BFGS;
- L-BFGS-B;
- TNC;
- COBYLA;
- SLSQP.

These algorithms are provided by the *scipy.optimize* package of Python programming language [38] and are the most commonly used. In particular, Nelder-Mead [15], Powell and COBYLA [30] are gradient-free while CG [26], BFGS, L-BFGS-B [22], TNC [8] and SLSQP [23] are gradient-based. These local optimizers have been presented in sec. 2.4.1 together with others not used in this analysis.

For each algorithm, the computational time, the optimization steps and the fidelity have been evaluated for several optimization procedures. For the usual Ising model with $h = 0.5$, we performed a QAOA with INTERP strategy choosing the local optimizer from the list above. Each optimizer has been asked to find the ground state of the Ising chain varying N from 2 to 12 with step 2 and choosing $p = \frac{N}{2}$. The optimization has been repeated 10 times to get the statistical error on the computational time and the average value is plotted.

Fig. 4.3, 4.4 and 4.5 show the histograms, each of them referring to a specific value of N and p, of the computational time, the optimization steps and the fidelity of all the local optimizers in the test set, respectively. In addition, fig. 4.6, 4.7 and 4.8 display the computational time, the optimization steps and the fidelity as a function of N for every optimizer, respectively.

What one immediately notices is that the L-BFGS-B algorithm outperforms the other ones from all possible points of view: it is the fastest, requires the smallest amount of resources, i.e. performs the smallest number of optimization steps, and it's totally reliable, since it identifies the correct ground state getting a unit fidelity regardless the number of qubits. That's why the L-BFGS-B has been chosen as our reference optimizer and, besides this analysis, we exploited it for all the other simulations reported in this chapter. Another interesting result to point out is that gradient-free search algorithms are typically the worst or near-worst from every point of view. They are slow and cannot maintain the maximum fidelity as the spin chain becomes larger. However, for the TNC method, which is gradient-based, the fidelity rapidly drops down also for small N. The SLSQP does the same but it could maintain a unit fidelity until $N = 10$, but it is, unlike TNC, rather fast.



(a) Computational time for $N = 2$ and $p = 1$

(b) Computational time for $N = 4$ and $p = 2$

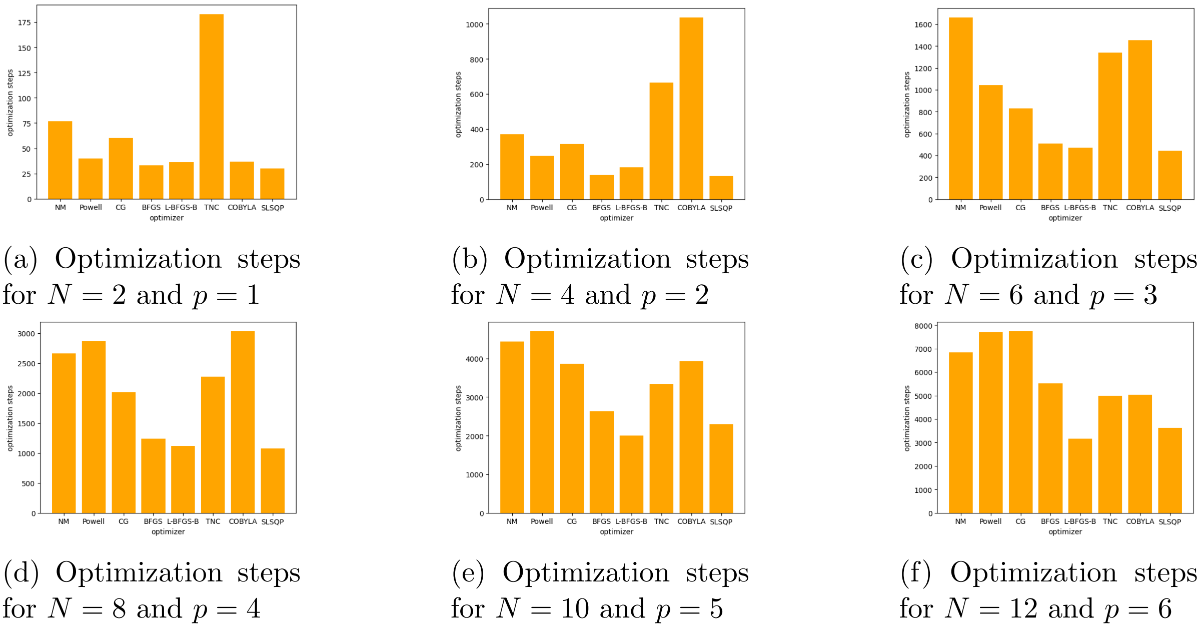
(c) Computational time for $N = 6$ and $p = 3$

(d) Computational time for $N = 8$ and $p = 4$

(e) Computational time for $N = 10$ and $p = 5$

(f) Computational time for $N = 12$ and $p = 6$

Figure 4.3: Histograms of the computational time of the local optimizers. From left to right and from top to bottom: $(N, p) = (2, 1), (4, 2), (6, 3), (8, 4), (10, 5), (12, 6)$. Each value represents the average over 10 runs of the simulation



(a) Optimization steps for $N = 2$ and $p = 1$

(b) Optimization steps for $N = 4$ and $p = 2$

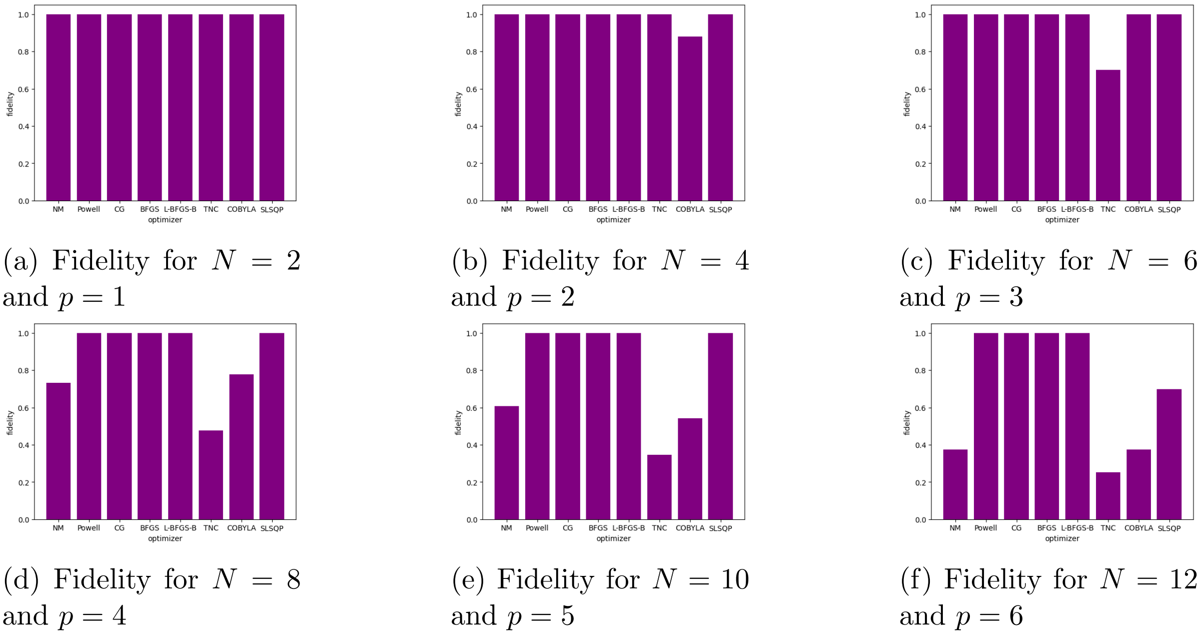
(c) Optimization steps for $N = 6$ and $p = 3$

(d) Optimization steps for $N = 8$ and $p = 4$

(e) Optimization steps for $N = 10$ and $p = 5$

(f) Optimization steps for $N = 12$ and $p = 6$

Figure 4.4: Histograms of the optimization steps of the local optimizers. From left to right and from top to bottom: $(N, p) = (2, 1), (4, 2), (6, 3), (8, 4), (10, 5), (12, 6)$



(a) Fidelity for $N = 2$ and $p = 1$

(b) Fidelity for $N = 4$ and $p = 2$

(c) Fidelity for $N = 6$ and $p = 3$

(d) Fidelity for $N = 8$ and $p = 4$

(e) Fidelity for $N = 10$ and $p = 5$

(f) Fidelity for $N = 12$ and $p = 6$

Figure 4.5: Histograms of the fidelity of the local optimizers. From left to right and from top to bottom: $(N, p) = (2, 1), (4, 2), (6, 3), (8, 4), (10, 5), (12, 6)$

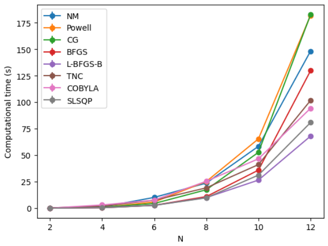


Figure 4.6: Summary of the computational time (in seconds) of all the tested local optimizers as a function of the chain length. The L-BFGS-B, for large N, the fastest optimizer. On the contrary, gradient-free algorithms come out to be the slowest as N increases. Each point of each line is the average over 10 runs

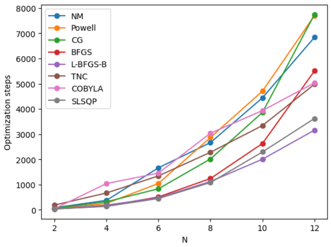


Figure 4.7: Summary of the optimization steps of all the tested local optimizers as a function of the chain length. The algorithm which requires the smallest amount of computational resources is, again, the L-BFGS-B, closely followed by the SLSQP. Moreover, gradient-free algorithms proved another time not to be so performative for large N

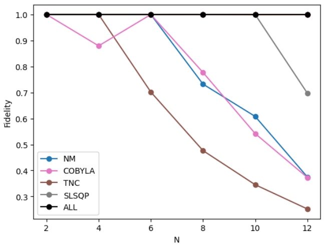


Figure 4.8: Summary of the fidelity reached by all the tested local optimizers as a function of the chain length. The black line represents all the algorithms which gets unit fidelity for that value of N and p. The L-BFGS-B, together with other gradient-based optimizers, was capable to obtain a unit fidelity for any length of the chain, i.e. finding the real ground state. On the contrary, Nelder-Mead and COBYLA (which are gradient-free) together with TNC and SLSQP (which are gradient-based) could not maintain a maximum fidelity. In particular, as N increases, the fidelity they reached severely decreases, proving, at least in this types of analysis, to be unreliable

4.6 Fidelity and phase transition

Meanwhile discussing the features of the Ising model with transverse field, we pointed out that the non-degenerate ground state in the large field limit was exactly $|+\rangle^{\otimes N}$. As one may notice, this state is the initial quantum state of the qubits at the beginning of the QAOA. The obvious deduction is that, if we consider the large field Ising chain, and we perform a QAOA to find its ground state, the embedding puts the qubits in the exact state we wish to find and the evolution is led by an hamiltonian which the qubits state is ground state of. Hence, the algorithm will guess the ground state of the Ising regardless the depth of the circuit. Moreover, one can hypothesize that, after the phase transition occurring at $h = 1$, since the ground state is, roughly speaking, "closer" to the initial QAOA state, also circuits shallower than the required, may get an higher fidelity with respect to the fidelity they would get before the phase transition, i.e. for $h < 1$.
Inspired by this apparently flawless reasoning, we decided to check if our observation had some grounds. We considered the Ising model with different values of the magnetic field: we chose $h = 0.1, 0.3, 0.5, 0.8, 1.0, 2.0, 5.0, 10.0$. For each value of the field, we considered three possible values of the chain length, i.e. $N = 6, 8, 10$. For each possible combination of (h, N) we performed a QAOA with INTERP strategy and L-BFGS-B as local optimizer, to find the ground state of the model where the depth varies in the range $p \in [1, \frac{N}{2}]$. In this way we could potentially verify that also shallow circuits identify the ground state when the field is larger than 1.
Fig. 4.9, 4.10 and 4.11 depict the behavior of the fidelity with respect to the magnetic field for $N = 6, 8, 10$, respectively. Each figure displays different lines, one for every possible value of the circuit depth ranging from 1 to the required value to get unit fidelity.
This analysis led us to prove two important things:

- The former is the one discussed a while ago. The larger is the field beyond $h = 1$, the shallower can be the circuit to get a fidelity greater than a given threshold;

- The latter regards the lower bound of the circuit depth. In order always to get a fidelity $F = 1$ regardless anything, we must set the depth $p = int(\frac{N}{2})$.

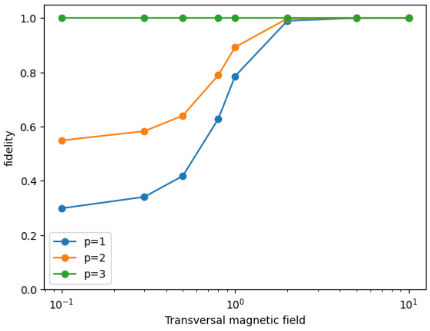


Figure 4.9: Fidelity as a function of the magnetic field for $N = 6$. $p = 3$ is the required depth to get maximum fidelity regardless the model

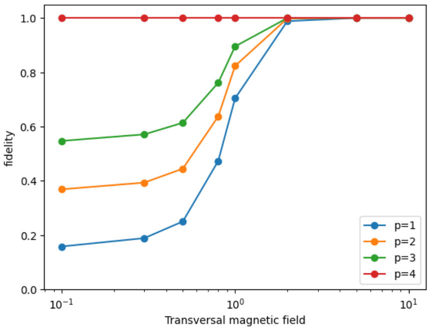


Figure 4.10: Fidelity as a function of the magnetic field for $N = 8$. $p = 4$ is the required depth to get maximum fidelity regardless the model

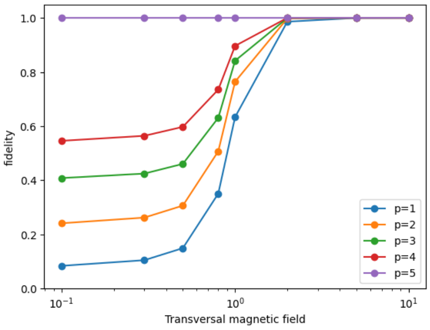


Figure 4.11: Fidelity as a function of the magnetic field for $N = 10$. $p = 5$ is the required depth to get maximum fidelity regardless the model

4.7 Reconstruction of the quantum state via measurements

What if, at the end of the QAOA algorithm, we wish to reconstruct the quantum state by averaging measurement outcomes as it should occur on a real quantum device? What is the minimum amount of measurements to get enough statistics to reconstruct the quantum state with arbitrarily small approximation?

In this section we provide an exhaustive answer to these non-trivial and non-negligible questions if we wish, one day, to implement quantum algorithms on real quantum devices. Consider once again the Ising model with $h = 0.5$. We performed a QAOA with INTERP strategy and L-BFGS-B as local optimizer to find the ground state of the model for $N \in \{2, 4, 6, 8, 10\}$ and $p = \frac{N}{2}$. The starting assumption states that, because of what we have proved in the previous section, the algorithm outputs the correct ground state of the model. For each chain length, we tried to reconstruct the quantum state performing M measurements, with $M \in \{10, 30, 50, 100, 300, 500, 1000, 3000, 5000, 10000\}$. We refer to each run of the algorithm with measurements as a shot. So M can be also called the "number of shots". Each reconstruction of the state for a N spins chain and M shots has been repeated 20 times to provide a statistically significant result.

Fig. 4.12 shows the fidelity as a function of the number of shots. The algorithm produces the exact ground state with $F = 1$ but, because of the statistical nature of measurements, the actual fidelity approaches unity asymptotically. However, this is what we actually expected from the beginning. The longer the chain is, the larger the number of shots should be to reach the same fidelity. This analysis paves the way for possible future measurements-based simulations. In fact, given a fidelity threshold F_{min} and a chain of length N, one can compute the lower bound of the number of shots required to reach a fidelity $F \geq F_{min}$ with an arbitrarily high probability. It is worth to remark, however, that, in order to reach a considerably high fidelity, also for small chains the minimum amount of measurements to perform is significantly large. As a consequence, the implementation of this kind of algorithms is badly affected by this inconvenient, since the total amount of computational resources required experiences a massive growth.

As a remark, the total amount of measurements needed to reconstruct the final state for $N = 10$ with a fidelity $\frac{F_{exp}}{F_{exact}}$, where F_{exp} is the fidelity got with M measurement and F_{exact} is the fidelity got with exact simulation (for us it is $F_{exact} = 1$), larger than a given threshold is comparable to the number of measurements required by the bayesian [35] algorithm, displayed in fig. 2.12 in chapter 2.

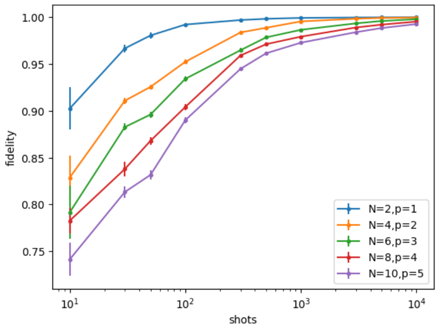


Figure 4.12: Fidelity obtained for each different number of shots. Each line represent the same analysis for different chain length $N \in \{2, 4, 6, 8, 10\}$

4.8 Time and step's trend

In this section we'll provide an in-depth analysis for what the trend of the computational time and the optimization steps as a function of N and p is concerned. What we would like to observe is how two out of three fundamental descriptors of a local optimization algorithm vary when either the length of the spin chain or the circuit depth changes. In particular their slopes may provide some indications and estimates for the computational power needed to perform analysis on possibly larger systems.

In this simulation we considered the QAOA with INTERP strategy applied to the already familiar Ising model with $h = 0.5$ with the well-known L-BFGS-B as local optimizer. We let the chain length range in the interval $N \in \{2, 14\}$ with step 2 and the circuit depth within $p \in \{1, 7\}$ with step 1. For each possible combination of (N, p) we evaluated the computational time and the number of optimization steps required by L-BFGS-B to locate the optimum on the energy landscape and plotted the outcomes in various ways, described in a while. As a remind, for some combinations of (N, p) we may reach a unit fidelity, i.e. when $p \geq \frac{N}{2}$; on the contrary, when $p < \frac{N}{2}$ we cannot reach the maximum fidelity. Furthermore we believed that it could be interesting to evaluate the trend of the computational time and optimization steps for the possible pairs (N, p) where N ranges in the usual interval and $p = \frac{N}{2}$. In this way the algorithm is able to identify the real ground state of the system, and since p is exactly equal to half of the chain length, it does so with the minimum effort, computationally speaking. The simulations are exact, therefore there is statistical error only on the computational time, for which we provide a result averaged over 10 runs. However, the error bars are very small and one can barely glimpse them.

The outcomes of the simulations have been plotted in a complementary way. Fig. 4.13 and 4.15 show the behavior of the time as a function of N, in the former plot, or p, in the latter. Fig. 4.14 and 4.16 display the number of optimization steps, again as a function of either N or p, respectively. Since the scale of the y axis is logarithmic, we can conclude that:

- The computational time varies exponentially in N, since the curve has a linear trend in logarithmic scale, and slower than exponentially in p;
- The optimization steps tend to become constant in N as it increases, while it grows slower than exponentially in p.

Fig. 4.17 and 4.18 show the trend of time and steps for the pairs (N, p) where $N \in \{2, 14\}$ with step 2 and $p = \frac{N}{2}$, respectively. Once again, the y axis scale is logarithmic. The time grows exponentially or almost exponentially while the steps increase slower than exponentially. Nevertheless, it would be desirable, as a possible deepening, to reproduce this simulation on larger systems in order to provide an exhaustive and detailed description of the behavior of time and steps as a function of (N, p). Because of the limited computational capability at our disposal in this project, it was not possible to increment the size of the system beyond $N = 14$ due to exponential behavior of the computational time required to perform such simulations. If, in principle, one could have access to a supercomputer, it would be possible to enlarge the system at least beyond $N = 20$ and run the algorithm in a reasonable amount of time.

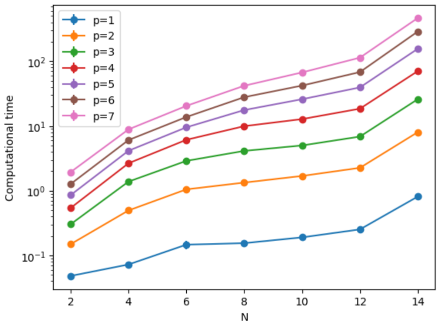


Figure 4.13: Computational time (in seconds) as a function of the chain length N. Each line correspond to a different value of the circuit depth p. Each point is the average over 10 runs, though the error bars are too small to be visible

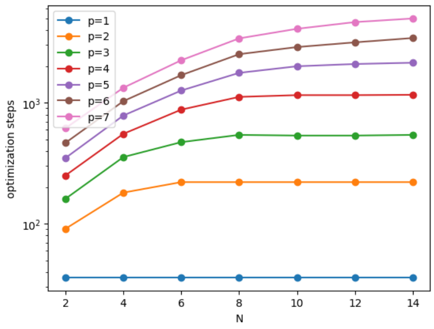


Figure 4.14: Optimization steps as a function of the chain length N. Each line correspond to a different value of the circuit depth p

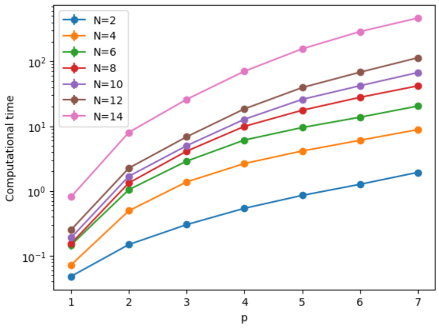


Figure 4.15: Computational time (in seconds) as a function of the circuit depth p. Each line correspond to a different value of the chain length N. Each point is the average over 10 runs, though the error bars are too small to be visible

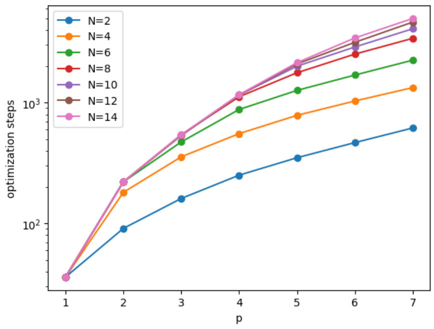


Figure 4.16: Optimization steps as a function of the circuit depth p. Each line correspond to a different value of the chain length N

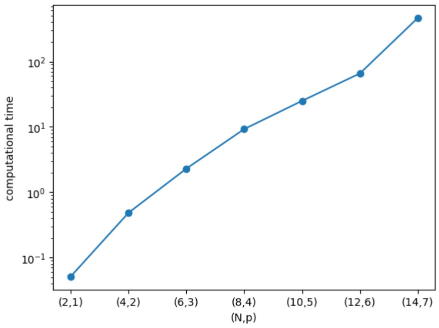


Figure 4.17: Computational time (in seconds) for the pairs (N, p) with N within the range $[2, 14]$ with step 2 and $p = \frac{N}{2}$ in order to obtain with minimum effort a unit fidelity. Each point is the average over 10 runs, though the error bars are too small to be visible

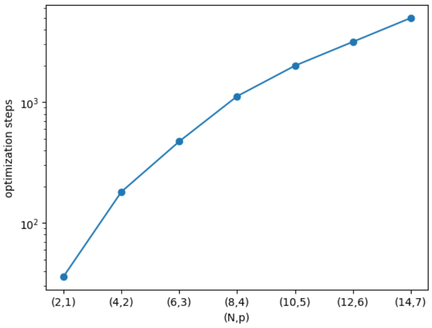


Figure 4.18: Optimization steps for the pairs (N, p) with N within the range $[2, 14]$ with step 2 and $p = \frac{N}{2}$ in order to obtain with minimum effort a unit fidelity

4.9 Transferability of optimal solutions

In section 2.2.4, we introduced the reader to the concept of *Transferability* of optimal solutions from a smaller sized system to a larger one [14] and the conditions under which it could effectively bring an advantage for simulation purposes. In this section we analyze how the exploitation of the transferability mechanism affects the computational time and optimization steps. In particular, what we would like to prove is that, if optimal parameters are transferred from a smaller spin chain to a larger one, and a refinement optimization is performed on the larger system, some time is saved and this procedure is faster than a standard optimization routine. We considered the Ising model with $h = 0.5$ and applied QAOA with INTERP startegy on it. The circuit depth has been fixed to $p = 7$. In this way, we are sure to reach $F = 1$ also for the largest system involved in this analysis, which has $N = 14$. We chose three different sizes of the starting chain, $N_{start} = 4, 6, 8$. For each of them, the parameters have been optimized for $p = 7$ and transferred to larger systems with size equal to $N_{target} \in [N_{start} + 2, 14]$ with step 2. The refinement optimization has been then performed on the larger system. A standard optimization routine has been performed separately.

Fig. 4.19 depicts the ratio between the time required to perform the optimization via transferability, as described above, and the time needed for the standard routine. Basically, the smaller is this ratio, the greater is the advantage brought by the employment of transferability. The optimization has been performed 10 times to make statistics on the computational time. Fig. 4.20 displays the same ratio referred, instead, to the number of optimization steps. Three notable results have come out:

- The smaller is N_{start}, the smaller are the time ratio and the step ratio being equal N_{target}. This was straightforwardly expected since the idea of transferability hinges on the fact that the optimal parameters also for a very small system are, however, near-optimal for the larger one. Because, as proved in the previous section, the computational time grows exponentially in N, is it legitimate to expect that a smaller N_{start} leads to a bigger benefit from this point of view. The same logic can be applied also to the step ratio;

- In fig. 4.19, all the three curves somehow flow into the same value of the time ratio for $N = 14$. This result looks quite non-trivial to explain. A possibility concerns the existence of a sort of critical time ratio, below which it is impossible to go regardless the value of N_{start} and, in the large N_{target} limit, each curve approaches the criticality. If that was true, and by the way nothing guarantees it, we could provide a lower bound on the advantage entailed by the employment of transferability and state that no transferability procedure could employ less than 35% of the time required by a standard optimization routine. Nonetheless, this is just an hypotheses. The right way to possibly get a proof of that is to perform the same simulation increasing N_{target} and look at the behavior of the curves from a more global perspective;

- What has been said for the computational time in the previous point can be said also for the number of steps but with a main difference (fig. 4.20). The ratios of the steps for different starting systems do not converge to the same value. They rather seem to maintain a constant difference or ratio between them in the large N limit. Again, it would be interesting to perform the same simulation increasing N_{target} and check if the difference or the ratio between the lines stabilizes to a constant value and possibly provide an estimate of the latter. Moreover, we could estimate the maximum advantage in terms of optimization steps brought by the employment of the transferability mechanism.

As discussed also in the previous section, the impossibility to perform simulations on larger scales is the main reason behind the incompleteness of our results. However, they can doubtlessly be considered as starting point for further developments in this sense and possibly they could pave the way for additional analysis on real QPUs, when available. We are, indeed, satisfied of the quality and interest of what we got up to now. Moreover, the optimization routine relying on transferability returns the exact optimal parameters of the standard schedule, reaching a unit fidelity. This is another point in favor to its adoption in further simulations.

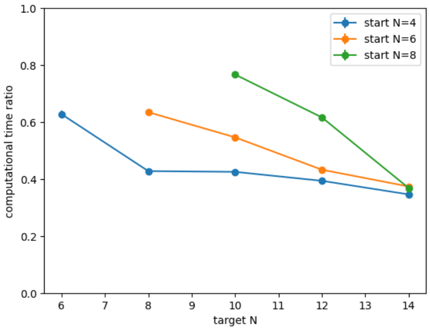


Figure 4.19: Ratio between the computational time required to perform the optimization involving the transferability mechanism, i.e. optimizing the parameters for the smaller system and perform a refinement on the larger one, and the time needed to perform a standard optimization. $p = 7$, in order to get unit fidelity at the end of the refinement also for $N = 14$. Each line refers to a different size of the starting system while, on the $\hat{x}$ axis, the size of the target system is indicated. Each point is the average over 10 runs

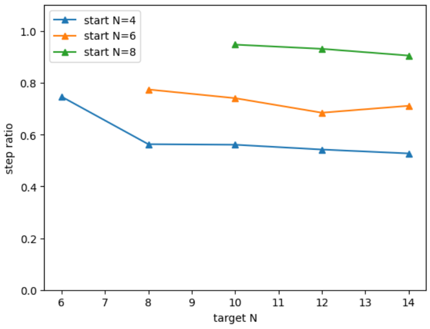


Figure 4.20: Ratio between the number of steps required to perform the optimization involving the transferability mechanism, i.e. optimizing the parameters for the smaller system and perform a refinement on the larger one, and the number of steps needed to perform a standard optimization. $p = 7$, in order to get unit fidelity at the end of the refinement also for $N = 14$. Each line refers to a different size of the starting system while, on the $\hat{x}$ axis, the size of the target system is indicated

4.10 Modified INTERP strategy

The final section of this chapter is dedicated to the presentation of a slightly modified version of the INTERP depth-progressive strategy. As remarked at the beginning of the chapter, the INTERP algorithm works by uploading the circuit depth at every iteration until the desired depth is reached. By the way, we have seen that the desired value of p coincides with $\frac{N}{2}$, to get the actual ground state of the model employing the smallest possible effort. We have also verified that the transferability mechanism is helpful in this sense by lowering the amount of time needed to perform the optimization routine. This method holds because the optimal solution for an Ising chain with N quite small, are, indeed, near-optimal for larger spin chains, whatever is its length, provided that it is a even number. This fundamental observation led us to test an improvement of the INTERP strategy which works as follows:

1. Start from $(N, p) = (2, 1)$ and set $(\gamma_1, \beta_1) = (0.1, 0.1)$
2. Optimize the two parameters;
3. Upload both the system size N and the circuit depth p as follows:
    - $N \to N + 2$;
    - $p \to p + 1$;
4. Compute the starting value for the new parameters by interpolating the optimal point at previous depth, following the exact scheme of the standard INTERP schedule;

5. Optimize the parameters and repeat from point 3 until the desired pair (N, p) is reached.

This improved version of INTERP is definitely tailored to the specific problem of the ground state preparation of an Ising model, or, more in general, of a many body theory 1D model. This could possibly represent, in principle, a sort of limitation.
We decided to compare, again, the computational time and optimization steps required by this modified procedure with respect to the ones needed by the standard INTERP method. We considered the Ising with $h = 0.5$, our reference model, and the pair $(N, p) \in [(2, 1), ..., (14, 7)]$.
Fig. 4.21 displays the ratio, in green for the steps and in red for the time, between those quantities for the modified version and those for the standard version of INTERP. For what time ratio is concerned, each point is the average over 10 runs. As one may notice, the employment of the improved version brings an advantage, greater from a computational time point of view rather than from an optimization steps perspective. Furthermore, the larger is the system size, the smallest are the ratio, especially the one referred to time, and therefore the greater is the advantage provided by the new version of INTERP. An important remark about the fidelity: the new procedure outputs the same exact best parameters thus getting the highest fidelity $F = 1$. A final consideration regards the possible combination of the modified version of the INTERP with the transferability mechanism. We may expect it leads to a further improvement of simulations concerning the solution, i.e. ground state preparation, of many body problems.

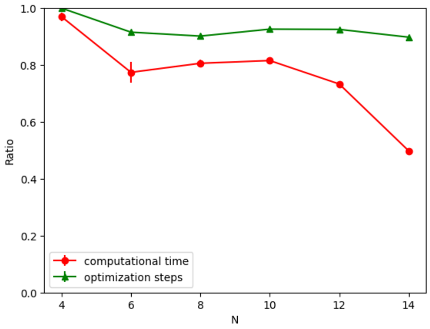


Figure 4.21: The green line represents the ratio between the optimization steps required by the modified INTERP and those required by the standard INTERP. The red line displays the same ratio referred, instead, to the computational time. One can notice that the improved version provides a considerable advantage in terms of computational time and a slight advantage from what the steps is concerned. Furthermore, the larger is the system, the greater is the advantage. Each point of the line regarding computational time is the average over 10 runs

4.11 Discussion

After the gathering of all the outcomes of our simulations, it's time to take stock of the main results we got and summarize them once for all.

First of all, the application of the QAOA with INTERP strategy to the Ising model with transverse field has been extremely successful from every point of view. We recovered the main feature of the INTERP algorithm, i.e. the smoothness of the best solutions, also in our simulations. We verified that the L-BFGS-B is the best local optimizer, and this is the reason of its massive employment also for what machine learning is concerned. We tested several optimization schedules, including a modified version of INTERP and the adoption of the transferability mechanism, already well-known in the literature, to improve the QAOA in general, as it should be considered in its entirety, that is to say, a hybrid quantum-classical algorithm, where the classical part with all its possible flaws and limitations, plays an extremely relevant, not to say fundamental and crucial, role. Further improvements can be surely made and, because of the importance of the optimization schedules both for QAOA itself and from a machine learning perspective, this represents a very interesting and fascinating research field to delve into.

Moreover, we investigated the properties of the Ising model itself, focusing our attention on the phase transition occurring at the critical field $h = 1$, proving that, above the transition, the QAOA is more efficient in the identification of the ground state of the system.

We analyzed also the complications emerging in the exploitation of QAOA as a real quantum algorithm. I'm obviously talking about the implementation of measurement processes, without which nobody can effectively employ any quantum hardware for computing and information purposes. The large number of shots required to recover a quantum state is definitely worsening our future perspective concerning quantum information, together with faulty quantum gates which all alone make the outcomes worse. However, the possible employment of some superconducting architectures, for instance IBM's [33], would severely reduce the time needed to perform simulations and measurement. With IBM quantum computer, one can easily perform a number of measurements of the order of 10^3 within little time.

The results presented in this chapter ultimately helps in paving the way to future investigations on possible further improvements and modifications of already-existing hybrid algorithms, such as QAOA.

Conclusions

This thesis project focused on one of the most celebrated hybrid quantum-classical algorithm, namely the Quantum Approximate Optimization Algorithm [10]. After a brief but, in my opinion, fundamental introduction to the mathematical and physical framework of quantum computation and information [27], we introduced the reader to the main features of the QAOA, highlighting its suitability for what the problem of the preparation of ground states is concerned and its typical applications to quantum many body theory models. It is worth to remark that both the quantum and the classical part of the algorithm play a relevant role and an optimal functioning of both is crucial to a successful application of QAOA. As a consequence of that, we illustrated a review, based on recent scientific literature, of all the techniques, with a deepening on the transferability of optimal solutions and initialization strategies [14] [20] [21] [24], which have been verified to provide a significant improvement of the efficiency of the algorithm from a computational point of view. After that, we introduced the Ising model with transverse field [13], a well-known many body model suitable for testing hybrid algorithms. In the last chapter of the thesis we provided the outcomes of all the simulations we performed. The idea of such simulations was double: first of all, we wanted to verify if the QAOA, coupled to the INTERP initialization strategy, was effectively able to identify the ground state of the Ising model in many possible scenarios, i.e. solving it, and reproduce a peculiar characteristic of INTERP strategy which is the smoothness, in the layer index, of the best parameters; in addition, we wanted to check what is the amount of computational resources required by the algorithm to run and its possible dependence on the optimization schedule chosen, including the adoption of a measurement process at the end of the quantum circuit. The capability of the algorithm to be both accurate in the location of the ground state and efficient in terms of computational time is of fundamental importance in the perspective of applying, hopefully in the near future, this kind of algorithms on NISQ devices. NISQ devices are primarily negatively affected by the noise produced by the quantum gates we apply on the qubits system. This massive flaw entails the usage of a complex error correction system which makes an application of quantum algorithms on many-qubits systems unfeasible from a practical point of view. That's the main reason of the adoption of simulation techniques of the QAOA on classical devices. The results got from these simulations are, indeed, very important for a better understanding of the potentialities of the algorithm, both nowadays and for future. The QAOA, in fact, plays a relevant role also from a machine learning perspective [28] [29], which is nowadays widely spread and employed, however still classically, for many useful tasks in a lot of disparate sectors, from medicine to engineering, from economics to climatology. It is reasonable to think that the future of mankind is going towards an overwhelming transformation of classical computation methods into their quantum counterparts. After all, we have already started the so called "second quantum revolution". Hybrid algorithms and then fully quantum algorithms will therefore be the protagonists of this drastic change. No one knows how much time this transformation will take. What we can do is to accept this challenge and try to give as many contributions as possible in this sense. This is also the scope of this project and surely it is the scope also of previous and future related works.

In our analysis, we proved that the adoption of the transferability mechanism [14] and the INTERP strategy [24] provided a concrete advantage in the speed-up of the entire algorithm, severely reducing the total computational time needed to perform the optimization procedure. Furthermore, we identified the best local optimizer to employ for QAOA. This is the L-BFGS-B search algorithm, which proved itself to be both the fastest in terms of time and optimization steps and the most accurate, since it perfectly locates correct the ground states of the Ising model, at least for a number of qubits $N \leq 14$. Following this branch, we described and tested a modified INTERP procedure where both the number of qubits and the depth of the quantum circuit are optimized at each step. This modification brings an advantage, computationally speaking, still maintaining a perfect accuracy. Because the computational time increases exponentially in the system size N, it is rather important to take into account all the possible methods to improve the algorithm in this sense. For what global optimizers is concerned, the bayesian algorithm [35] looks very promising. The main reason is that, since it is a global optimizer it does not require a depth-progressive strategy because it is able to avoid barren plateaus without them. In addition, it outperformed other global search algorithms and it is competitive, from a resources perspective, with local optimizers.

For what the Ising model itself is concerned, we checked that the employment of the INTERP strategy ensures to find smooth optimal solutions as expected from previous works [24]. Moreover, we proved that, when the magnetic field of the Ising chain is $h > 1$, i.e. we are in the disordered phase, the QAOA identifies the ground state regardless the circuit depth p, in particular in the large h limit. On the contrary, in the ordered phase, i.e. $h < 1$, we need a quantum circuit whose p is at least half of the Ising chain length N to locate the ground state with maximum fidelity F. We decided to focus our attention also on the employment of a measurement process at the end of the circuit to reconstruct the ground state. The number of measurements needed to construct the quantum state rapidly increases with the system size and this may represent a limitation we have to consider from the perspective of an application of the QAOA on NISQ devices.

For what possible further developments is concerned, in order to check the validity of what we have proved and discovered also for larger systems, it would be rather important to repeat the analysis on systems with larger N, which requires a computational power we couldn't have at our disposal. Hopefully, on bigger systems, the employment of strategies to improve the performance of the QAOA would allow to perform such simulations with the minimum effort, strengthening our statements and the validity of our analysis. Furthermore, it would be very interesting to test this algorithm on quantum devices, when available, even with a small number of qubits, and see what we can deduce about the actual feasibility of the implementation of hybrid algorithms on NISQ devices.

Acknowledgements

I would like to thank primarily my family for supporting me from every point of view during this tough period of academic studies.
Moreover, I also would like to thank all my friends, both the ones who accompanied and followed me during the Master's degree course and the ones who supported me from outside, who have been rather crucial for me to be in good spirits.
Last but not least, I thank my supervisor and her PhD students team for helping me during the thesis project.
All the people that I have mentioned gave, from different points of view, a contribution to the realization of this work.

Bibliography

[1] Andrew Arrasmith, Marco Cerezo, Piotr Czarnik, Lukasz Cincio, and Patrick J Coles. Effect of barren plateaus on gradient-free optimization. *Quantum*, 5:558, 2021.

[2] Claude Berge. *The theory of graphs*. Courier Corporation, 2001.

[3] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[4] Fernando GSL Brandao, Michael Broughton, Edward Farhi, Sam Gutmann, and Hartmut Neven. For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. *arXiv preprint arXiv:1812.04170*, 2018.

[5] Andrew Childs. Lecture 18: The quantum adiabatic theorem. *University of Waterloo*, 2008.

[6] Christopher M Dawson and Michael A Nielsen. The solovay-kitaev algorithm. *arXiv preprint quant-ph/0505030*, 2005.

[7] R Di Candia, B Mejia, H Castillo, JS Pedernales, J Casanova, and E Solano. Embedding quantum simulators for quantum computation of entanglement. *Physical Review Letters*, 111(24):240502, 2013.

[8] Francisco Facchinei, Stefano Lucidi, and Laura Palagi. A truncated newton algorithm for large scale box constrained optimization. *SIAM Journal on Optimization*, 12(4):1100–1125, 2002.

[9] Daniel Faílde, José Daniel Viqueira, Mariamo Mussa Juane, and Andrés Gómez. Using differential evolution to avoid local minima in variational quantum algorithms. *arXiv preprint arXiv:2303.12186*, 2023.

[10] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[11] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

[12] Mario Fernández-Pendás, Elías F Combarro, Sofia Vallecorsa, José Ranilla, and Ignacio F Rúa. A study of the performance of classical minimizers in the quantum approximate optimization algorithm. *Journal of Computational and Applied Mathematics*, 404:113388, 2022.

[13] Fabio Franchini et al. *An introduction to integrable techniques for one-dimensional quantum systems*, volume 940. Springer, 2017.

[14] A Galda, X Liu, D Lykov, Y Alexeev, and I Safro. Transferability of optimal qaoa parameters between random graphs. *arXiv preprint arXiv:2106.07531*, 2021.

[15] Fuchang Gao and Lixing Han. Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277, 2012.

[16] R Gilmore. Baker-campbell-hausdorff formulas. *Journal of Mathematical Physics*, 15(12):2090–2092, 1974.

[17] Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.

[18] Imran Khan Mohd Jais, Amelia Ritahani Ismail, and Syed Qamrun Nisa. Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, 2(1):41–46, 2019.

[19] Naoki Kanazawa, Daniel J Egger, Yael Ben-Haim, Helena Zhang, William E Shanks, Gadi Aleksandrowicz, and Christopher J Wood. Qiskit experiments: A python package to characterize and calibrate quantum computers. *Journal of Open Source Software*, 8(84):5329, 2023.

[20] Xinwei Lee, Yoshiyuki Saito, Dongsheng Cai, and Nobuyoshi Asai. Parameters fixing strategy for quantum approximate optimization algorithm. In *2021 IEEE international conference on quantum computing and engineering (QCE)*, pages 10–16. IEEE, 2021.

[21] Xinwei Lee, Ningyi Xie, Dongsheng Cai, Yoshiyuki Saito, and Nobuyoshi Asai. A depth-progressive initialization strategy for quantum approximate optimization algorithm. *Mathematics*, 11(9):2176, 2023.

[22] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[23] Yingjie Ma, Nan Zhang, and Jie Li. Improved sequential least squares programming–driven feasible path algorithm for process optimisation. In *Computer Aided Chemical Engineering*, volume 51, pages 1279–1284. Elsevier, 2022.

[24] Antonio A Mele, Glen B Mbeng, Giuseppe E Santoro, Mario Collura, and Pietro Torta. Avoiding barren plateaus via transferability of smooth solutions in a hamiltonian variational ansatz. *Physical Review A*, 106(6):L060401, 2022.

[25] Bojan Mohar and Svatopluk Poljak. Eigenvalues and the max-cut problem. *Czechoslovak Mathematical Journal*, 40(2):343–352, 1990.

[26] John L Nazareth. Conjugate gradient method. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):348–353, 2009.

[27] Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information. *Phys. Today*, 54(2):60, 2001.

[28] Mateusz Ostaszewski, Lea M Trenkwalder, Wojciech Masarczyk, Eleanor Scerri, and Vedran Dunjko. Reinforcement learning for optimization of variational quantum circuit architectures. *Advances in Neural Information Processing Systems*, 34:18182–18194, 2021.

[29] Johannes S Otterbach, Riccardo Manenti, Nasser Alidoust, A Bestwick, M Block, B Bloom, S Caldwell, N Didier, E Schuyler Fried, S Hong, et al. Unsupervised machine learning on a hybrid quantum computer. *arXiv preprint arXiv:1712.05771*, 2017.

[30] Michael JD Powell. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007.

[31] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media, 2006.

[32] James C Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems*, 34(3):817–823, 1998.

[33] Matthias Steffen, David P DiVincenzo, Jerry M Chow, Thomas N Theis, and Mark B Ketchen. Quantum computing: An ibm perspective. *IBM Journal of Research and Development*, 55(5):13–1, 2011.

[34] Masuo Suzuki. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190, 1976.

[35] Simone Tibaldi, Davide Vodola, Edoardo Tignone, and Elisa Ercolessi. Bayesian optimization for qaoa. *arXiv preprint arXiv:2209.03824*, 2022.

[36] Phuong Thi Tran et al. On the convergence proof of amsgrad and a new version. *IEEE Access*, 7:61706–61716, 2019.

[37] Salvador E Venegas-Andraca, William Cruz-Santos, Catherine McGeoch, and Marco Lanzagorta. A cross-disciplinary introduction to quantum annealing-based algorithms. *Contemporary Physics*, 59(2):174–197, 2018.

[38] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[39] David J Wales and Jonathan PK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[40] Nicholas C Wormald et al. Models of random regular graphs. *London mathematical society lecture note series*, pages 239–298, 1999.

[41] Jonathan Wurtz and Peter Love. Maxcut quantum approximate optimization algorithm performance guarantees for p¿ 1. *Physical Review A*, 103(4):042612, 2021.

[42] Jonathan Wurtz and Danylo Lykov. The fixed angle conjecture for qaoa on regular maxcut graphs. *arXiv preprint arXiv:2107.00677*, 2021.

[43] Yang Xiang, DY Sun, W Fan, and XG Gong. Generalized simulated annealing algorithm and its application to the thomson model. *Physics Letters A*, 233(3):216–220, 1997.