

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea Magistrale in Fisica

**QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM:
COMBINATORIAL PROBLEMS AND
CLASSICAL STATISTICAL MODELS**

Supervisor:

**Dr.
Davide Vodola**

Co-Supervisor:

**Prof.
Elisa Ercolessi**

Presented by:

Andrea Basilio Rava

Anno Accademico 2019/2020

ABSTRACT

The Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum-classical algorithm for solving combinatorial optimization problems. Since most of combinatorial optimization problems may be thought as particular instances of Ising Hamiltonians, the study of the QAOA is very relevant from the physical point of view for its potential applications in describing physical systems. In the QAOA a quantum state is prepared and, through $2p$ parameterized quantum evolutions, a final state which represents an extreme of cost function and encodes the approximate solution of the problem is obtained. The $2p$ parameters are determined through a classical parameter optimization process. In this work we apply QAOA to two different problems, the Max Cut and the random bond Ising Model (RBIM). For both problems we perform an analysis of the optimization efficiency, verifying that the quality of the approximation increases with p . For the Max Cut we perform a further analysis of the $p = 1$ case for which we obtain an analytical expression for the cost function and make observations regarding the choice of the initial parameters in the optimization procedure. For the RBIM, for different disordered configurations we obtain the ground states energies and magnetizations for different lattice sizes and different level p of the optimisation. We observe that, even if the magnetisation is obtained for small lattice sizes, its behaviour suggests the presence of a transition separating a ferromagnetic from a paramagnetic phase.

CONTENTS

Introduction	1
1 Quantum computation	7
1.1 Qubits and Hilbert space	7
1.1.1 Open systems and the density operator	10
1.2 Quantum gates and quantum circuits	13
1.2.1 Quantum measurement	16
1.3 Quantum computers and quantum simulators	18
1.3.1 Quantum simulation and classical computing techniques	20
2 Quantum algorithms	23
2.1 Algorithms and complexity classes	23
2.2 Quantum algorithms	25
2.2.1 Quantum search algorithm	26
2.2.2 Variational quantum algorithms	30
2.3 Quantum Approximate Optimization Algorithm	32
2.3.1 Optimization method	36
2.3.2 QAOA applications: Max Cut and RBIM	39
3 QAOA results for Max Cut and RBIM	43
3.1 Max Cut: QAOA results	43
3.1.1 Optimization method and results	46
3.2 RBIM: QAOA results	56
3.2.1 Optimization method and results	58
3.3 Outlook and conclusions	63

CONTENTS

Appendices	67
A Proof of analytical expression of F_1 for Max Cut	69
Bibliography	75

INTRODUCTION

Quantum computing is based on computational models that integrate the laws of quantum mechanics to perform certain computational tasks. The fundamental unit of quantum computation is the qubit, quantum mechanical counterpart of the bit for classical computation. Every quantum computational task consists in the manipulation of qubits with unitary gates and the extraction of the output of the quantum computation through quantum measurement operations. One of the achievements that are hoped to be reached with quantum computation, is the so called "quantum advantage", that will help to implement quantum algorithms for solving problems that are inaccessible for classical computation, due to their extremely high computational cost. Examples of these problems are simulation of large quantum systems, search an unstructured database, solving large-scale linear algebra problems [1], [2]. However, the implementation of such quantum algorithms requires quantum error correction (QEC) protocols, [3], to reduce the effect of quantum noise, an unavoidable effect due to the interaction of the system with the environment, which induces decoherence, i.e. corruption of the quantum information in the qubits manipulation or outcome reading trough quantum measurement. At present days the proposed QEC protocols require millions of physical qubits to allow the implementation of nearly noise-free quantum algorithms [2], however currently realizable quantum devices range from 50 to 100 qubits [1].

Even though the availability of an error-corrected quantum computer with millions of qubits is still decades away [1], currently realized quantum computers are the noisy intermediate-scale quantum (NISQ) devices [4]. NISQs are devices that may work with a limited size number of qubits, about 100 ("intermediate-scale") and have an imperfect control over those qubits ("noisy") [5]. Among the main goals in the near-term, is one to achieve quantum advantage with these devices [5] and to

develop new techniques that will be useful also for long-term devices [2]. Many algorithms have been proposed to be implemented in near-term devices, among these the hybrid quantum-classical algorithms have emerged to be well suited to account for the constraints imposed by the NISQs devices and be promising candidate for reaching quantum advantage [1]. The idea behind these algorithms is that they delegate the classically difficult part of some computation to the quantum computer and perform the classically tractable part on some classical device.

One of the first proposals of hybrid quantum-classical algorithms is the Quantum Approximate Optimization Algorithm (QAOA) [6] proposed to solve combinatorial optimization problems. Besides the fact that QAOA is the most studied gate model approach for optimization using near-term devices [7], it is well suited for physical applications because most of the combinatorial optimization problems may be thought as particular Ising Hamiltonians [7].

In the QAOA a quantum state is prepared as an input of a circuit with p levels, specified by $2p$ variational parameters. Using measurement outputs, the parameters are then optimized by a classical computer and fed back to the quantum computer that evolves the state in a closed loop. At the end the quantum state prepared using these optimal parameters encodes the approximate solution of the problem. Many features and properties of QAOA are yet to be completely understood, like its performance beyond its lowest depth variant [8], the prospects for achieving quantum advantage [7] or the difficulty of the parameter optimization [9], as the difficulty to efficiently optimize in the nonconvex, high-dimensional parameter landscape [8].

In this work we implement the QAOA in a noise-free environment, through classical quantum simulations using *QuTiP*, an open-source framework written in the Python programming language [10] designed for simulating the dynamics of quantum systems. We analyse the QAOA and, through these simulations, we study the performances of the algorithm for different QAOA p -levels, by verifying that they improve as p increases. We provide also an analysis about the initial parameters choice in the optimization procedure for the lowest QAOA level case, by leaving an open question if this analysis may be extended to higher p levels. Lastly we show an application of QAOA to a physical problem and how it behaves efficiently in finding the expected system ground states.

For the parameter optimization we exploit a gradient based optimizer, the gradi-

ent descent optimizer, with *finite differences* [11] as derivative estimator and *Adam* [12] as the algorithm to implement the parameters update.

We apply the QAOA to two different problems: in the first for finding the maximum cut of a given graph and in the second for finding the ground state of a classical spins Hamiltonian. The first problem is called Max Cut, it is a problem applied to graphs, where one looks for a bi-partition of vertices with higher number of edges that go from one partition to the other (the maximum cut). For different graphs we perform an analysis of the optimization behaviour and of the quality of the QAOA approximation for different QAOA p levels. We perform a further analysis of the QAOA level $p = 1$ case, where we study the behaviour of the optimization steps in the parameters landscape and discuss the choice of the initial parameters for efficient optimizations. Then we show that the QAOA returns the expected solutions for the three graphs.

The second problem is the random bond Ising model (RBIM) [13], which describes the interaction of classical spins in a two-dimensional (2D) lattice, where the couplings are ferromagnetic or antiferromagnetic and are distributed randomly with probability λ , where $\lambda = 1$ corresponds to all ferromagnetic couplings. Here, for three different lattices, we apply QAOA many times to find the spin configurations corresponding to the ground states of the system for different probabilities λ . We analyze the optimization procedure for different cases and show that also for this problem the QAOA finds the optimal parameters. In this problem we also analyze the behaviour of the classical magnetization and energies of the ground states obtained from the QAOA for different disorder configurations probabilities and observe that the returned ground states for different probabilities λ represent states that go from ordered regions to the disordered ones as λ decrease, as expected. Further, even if we consider small finite size lattices, we are able to observe that the evaluated magnetization has higher values in the smallest lattice for λ smaller than a certain probability and inverts its behaviour for λ higher than that probability value, which is the expected behaviour of the order parameter near a critical point.

The structure of this work is the following:

- In Chapter 1 we give an overview of the basic concepts of quantum computation. In Section 1.1 we introduce qubits and the fundamental quantum mechanical framework on which quantum computation is based. In Section

1.2 we describe the quantum circuit model, one of the principal gate models in quantum computation. Here we show the most important quantum gates and we present the concept of universality. In Section 1.3 we introduce the basic concepts and ideas behind the realization of a quantum computer. Then we briefly describe few examples of possible experimental realizations. We conclude the chapter with the idea behind the near-term devices (NISQ), because they are the ones that at the present day are experimentally available and we introduce QuTip, the open-source framework written in the Python programming language that we've used for the QAOA implementation.

- In Chapter 2 we give an overview of the principal ideas behind algorithms and quantum algorithms, then we give a more detailed description of the quantum approximate optimization algorithm. In Section 2.1 we introduce algorithms and we give a brief description of the complexity classes, that are used in the study of algorithms' performances and resource requirements. In Section 2.2 we describe briefly few fundamental features of quantum algorithms. Then we describe the Grover's algorithm, a quantum search algorithm, one of the principal quantum algorithm families. We conclude this section by giving an overview of the variational quantum algorithms (VQAs), a quantum algorithms family to which QAOA belongs to, focusing on their main components. In Section 2.3 we introduce and define the QAOA, and we describe its procedure for a generic computational optimization problem. We then describe the optimization method used in our QAOA implementation, a gradient-based method, implemented through the Adam algorithm. In the subsection 2.3.2 we describe the Max Cut problem, and we show how it is formulated its cost functions in terms of the QAOA. After the Max Cut we describe the random bond Ising model, we show its formulations in terms of the QAOA and also we give a brief overview of the predicted phase transition that occurs in the system.
- In Chapter 3 we show our results for the QAOA applied to the Max Cut and the RBIM. In Section 3.1 we describe our application of QAOA to Max Cut on three different graphs and for different QAOA levels. We show the steps of the optimization method for the different cases and we check that it behaves

as expected. We then analyze the special case of the QAOA level $p = 1$, that allows us to discuss about the choice of the initial parameters in the optimization method procedure. For the Max Cut application we show lastly that the QAOA results coincide with the expected solutions. In Section 3.2 we describe the application of QAOA to RBIM on three different lattices, for different QAOA levels, and for different coupling configurations for a picked probability. Here, we show the steps of the optimization for some probabilities and check that it behaves as expected. For the RBIM, we show also the plots of the evaluated classical magnetizations and energies vs. different disorder configurations probabilities from which we see that the QAOA returns for each lattice the expected ground states. We observe that, even in small finite size lattices, the magnetization vs. probability plot presents features of the expected phase transition. In Section 3.3 we present our outlook and draw our conclusions.

QUANTUM COMPUTATION

In this chapter we introduce the fundamental concepts of quantum computation, starting with the qubit and all the implications derived from its quantum mechanical nature followed by the basic building blocks for quantum circuits, a universal language for describing quantum computations.

In the last part of the chapter we present an overview of the actual quantum computers existent nowadays, by ending with a brief description of NISQ devices and quantum simulations, among which there is the classical quantum simulation through *QuTiP* that have been used for the considerations and results about the QAOA exposed in Chapter 3. The main reference used for this part is [14].

1.1 QUBITS AND HILBERT SPACE

All computing systems rely on a fundamental ability to store and manipulate information. Current computers manipulate individual *bits*, which store information as binary 0 and 1 states. Quantum computers leverage quantum mechanical phenomena to manipulate information. To do this, they rely on *quantum bits*, or *qubits*. A qubit is described as a quantum mechanical object which, in analogy to a classical bit, may be in two states $|0\rangle$ and $|1\rangle$. The difference between bits and qubits is that a qubit may also be in a *linear combination*, or *superposition*, of the states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \quad (1.1)$$

The coefficients α and β are complex numbers, so the state of a qubit is a vector in a two-dimensional complex vector space, a *Hilbert space* \mathcal{H} . The special states $|0\rangle$ and $|1\rangle$, known as *computational basis states*, form an orthonormal basis for this vector space.

In a classical system, a bit would have to be in one state or the other, it can be examined to determine whether it is in the state 0 or 1. However it is not possible to measure a qubit directly to determine its quantum state, that is, the values of α and β . To understand that difference, we need to introduce the concept of *evolution* of a quantum state and in particular of *quantum measurement*.

The evolution of a *closed* quantum system is described by a *unitary transformation*. Given the state $|\psi\rangle$ at time t_1 , it evolves to the state $|\psi'\rangle$ at time t_2 by an unitary operator U ,

$$|\psi'\rangle = U|\psi\rangle. \quad (1.2)$$

Applying an unitary operator on a quantum state changes it, but a hypothetically observer cannot get any information about the state, to achieve that, a quantum measurement, which is an interaction that makes the system no longer closed, must be performed.

Quantum measurements are described by a collection $\{M_m\}$ of *measurements operators*, where the index m refers to the measurement outcomes and the measurement operators satisfy the *completeness equation* $\sum_m M_m^\dagger M_m = I$. Given a state $|\psi\rangle$, the probability that the result m occurs is

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle, \quad (1.3)$$

and the state after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{p(m)}}. \quad (1.4)$$

For a single qubit, there's an important measurement, the *measurement of a qubit in the computational basis* or *single qubit projective measurement*, defined by the two measurements operators $M_0 = P_0 = |0\rangle\langle 0|$, $M_1 = P_1 = |1\rangle\langle 1|$, the *projectors* on the computational basis states $|0\rangle$ and $|1\rangle$.

Thus, when a qubit is measured in the computational basis, that measurement leads

to outcome an 0, corresponding to the state $|0\rangle$, with probability $|\alpha|^2$ and an outcome 1, corresponding to the state $|1\rangle$, with probability $|\beta|^2$. It is clear that:

$$|\alpha|^2 + |\beta|^2 = 1, \quad (1.5)$$

thus, in general, a qubit's state is a unit vector in a two-dimensional complex vector space.

From Equation (1.5) follows the geometric interpretation of a qubit, Equation (1.1) may be rewritten, up to a global phase, as

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle, \quad (1.6)$$

where θ and φ are real numbers. Thus, by considering the spherical coordinates

$$\begin{cases} r_x = \sin \theta \cos \varphi \\ r_y = \sin \theta \sin \varphi \\ r_z = \cos \theta \end{cases}, \quad (1.7)$$

the possible quantum states for a single qubit can be visualized on the unit three-dimensional sphere, often call the *Bloch sphere*, as shown in Figure 1.1.

In general, for any computational task, many qubits are required, so we have to consider multiple qubits systems. The state space of a quantum system made up of more than one distinct physical systems, a *composite system*, is the tensor product of the state spaces of the distinct physical systems. Thus, if the component systems are numbered through 1 to n and system number i is prepared in the state $|\psi\rangle_i$, then the joint state of the total system is

$$|\psi\rangle = |\psi\rangle_1 \otimes |\psi\rangle_2 \otimes \cdots \otimes |\psi\rangle_n. \quad (1.8)$$

From its definition, it naturally follows that the vector space of that joint vector is the Hilbert space $\mathcal{H}^{\otimes n} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_n$.

In the context of a composite system is generalized the definition of *projective measurement* seen for the single qubit case. A projective measurement is described

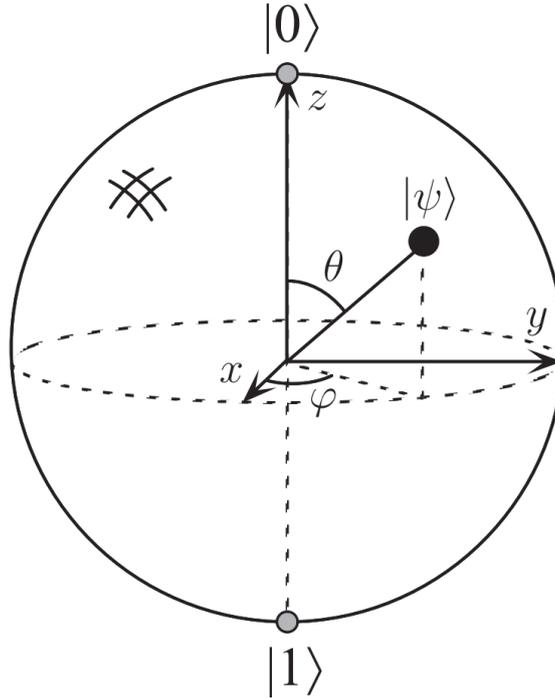


Figure 1.1: Bloch sphere representation of a qubit. Taken from [14]

by an *observable*, M , with spectral decomposition,

$$M = \sum_m m P_m, \quad (1.9)$$

where P_m is the projector onto the eigenspace of M with eigenvalue m . Upon measuring the state $|\psi\rangle$, the probability of getting result m is given by

$$p(m) = \langle \psi | P_m | \psi \rangle. \quad (1.10)$$

Given that outcome m occurred, the state of the quantum system immediately after the measurement is

$$\frac{P_m |\psi\rangle}{\sqrt{p(m)}} \quad (1.11)$$

1.1.1 OPEN SYSTEMS AND THE DENSITY OPERATOR

Real systems are *open systems* that interact with the outside world. Among these interactions, random interactions exist which cannot be completely controlled and

they are referred as *quantum noise*. It is crucial to understand and control such noise processes in order to build useful quantum information processing systems [15].

Until now the description of a quantum system has been formulated in terms of state vectors, whose state is completely known, a *pure state*. An alternate formulation is possible in terms of the *density operator* or *density matrix*. This alternate formulation is mathematically equivalent to the state vector approach, but its functionality relies in the description of quantum systems as *mixed states* whose state is not completely known due to presence of noise. A quantum system is described by a mixed state when it is in one of a number of states $|\psi_i\rangle$ with respective probabilities p_i , with $\{p_i, |\psi_i\rangle\}$ an *ensemble of pure states*. The density operator for the system is defined by

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i|. \quad (1.12)$$

Unitary evolution and measurement can be rephrased in the language of density operators, for proof see [14].

Thus, given an unitary operator U , the evolution of the density operator is described by

$$\rho' = U\rho U^\dagger, \quad (1.13)$$

and, for a measurement described by measurement operators M_m , the probability of obtaining result m is

$$p(m) = \text{tr}(M_m^\dagger M_m \rho), \quad (1.14)$$

with the state after the measurement which yields the result m described by the density operator

$$\rho_m = \frac{M_m \rho M_m^\dagger}{p(m)}. \quad (1.15)$$

The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n , and system number i is prepared in the state ρ_i , then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_n$. We introduce here an important operator in the study of density matrices, the *trace* of a matrix. Given a matrix A , the trace

of A is defined to be the sum of its diagonal elements,

$$\text{tr}(A) \equiv \sum_i A_{ii}. \quad (1.16)$$

Density operators can be characterized by the following theorem [14]:

Theorem 1 (Characterization of density operators). *An operator ρ is the density operator associated to some ensemble $\{p_i, |\psi_i\rangle\}$ if and only if it satisfies the conditions:*

1. (**Trace condition**) $\text{tr}(\rho) = 1$.
2. (**Positivity condition**) ρ is a positive operator.

One of the deepest application of the density operator is as a descriptive tool for *subsystems* of a composite quantum system, via the *reduced density operator*. In a system, composed by the subsystems A and B , described by $\rho^{AB} = \rho^A \otimes \rho^B$, the reduced density operator for system A is defined by

$$\rho^A \equiv \text{tr}_B(\rho^{AB}), \quad (1.17)$$

where tr_B is the *partial trace* over system B, defined by

$$\text{tr}_B(|a_1\rangle_A \langle a_2|_A \otimes |b_1\rangle_B \langle b_2|_B) \equiv |a_1\rangle_A \langle a_2|_A \text{tr}(|b_1\rangle_B \langle b_2|_B), \quad (1.18)$$

where $|a_1\rangle_A$ and $|a_2\rangle_A$ are any two vectors in the state space A, and $|b_1\rangle_B$ and $|b_2\rangle_B$ are any two vectors in the state space B.

The last argument exposed in this introductory section regards the concept of *fidelity*. One of the most important requirements for being able to perform a useful computation is to be able to prepare the desired input. In preparing the initial state, as it has been stated previously, it not possible to exactly know the produced state, so it is needed a quantity to express the *distance* between two states. A measure of distance between quantum states is the fidelity, defined, given two states ρ and σ as,

$$F(\rho, \sigma) \equiv \text{tr} \sqrt{\rho^{1/2} \sigma \rho^{1/2}}. \quad (1.19)$$

where, for any ρ and σ :

- $F(\rho, \sigma) = F(\sigma, \rho)$

- $0 \leq F(\rho, \sigma) \leq 1$, and $F(\rho, \rho) = 1$.

1.2 QUANTUM GATES AND QUANTUM CIRCUITS

Changes occurring to a quantum state can be described using the language of *quantum computation*. Analogous to the way a classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a *quantum circuit* containing wires and elementary *quantum gates* to carry around and manipulate the quantum information. Classical computer circuits consist of *wires* and *logic gates*. The wires are used to carry information around the circuit, while the logic gates perform manipulations of the information, converting it from one form to another.

Operations on a qubit must preserve its unitary norm, and thus are described by unitary matrices. Of these, some of the most important are the Pauli matrices:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1.20)$$

Three other fundamental single qubit gates are the Hadamard gate (H), phase gate (S) and $\pi/8$ gate (T):

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; T \equiv \begin{bmatrix} 1 & 0 \\ 0 & \exp(\frac{i\pi}{4}) \end{bmatrix}. \quad (1.21)$$

Symbols for the common single qubit gates are shown in Figure 1.2.

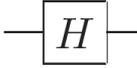
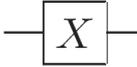
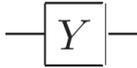
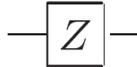
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli- X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli- Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli- Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$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

Figure 1.2: Names, symbols, and unitary matrices for the common single qubit gates. Taken from [14]

The Pauli matrices give rise to three classes of unitary matrices, the *rotation operators* about the \hat{x} , \hat{y} and \hat{z} axes of the Bloch sphere, defined by:

$$R_x(\theta) \equiv e^{-i\theta X/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad (1.22)$$

$$R_y(\theta) \equiv e^{-i\theta Y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad (1.23)$$

$$R_z(\theta) \equiv e^{-i\theta Z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (1.24)$$

In computation, one of the most useful operation is the *controlled operation*. In the generalization from one to multiple qubits, the prototypical multi-qubit quantum logic gate is the *controlled-NOT* or CNOT gate. This gate has two input qubits, known as the *control* qubit ($|c\rangle$) and the *target* qubit ($|t\rangle$), respectively. The circuit representation is shown in Figure 1.3. In terms of the computational basis, the CNOT performs the action $|c\rangle|t\rangle \rightarrow |c\rangle|t \oplus c\rangle$, where $|c\rangle, |t\rangle \in [|0\rangle, |1\rangle]$ and \oplus is the XOR gate. Thus, in the computational basis $|c, t\rangle$ the matrix representation on

CNOT is

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1.25)$$

As for the single qubit case, the requirement that probability be conserved is ex-

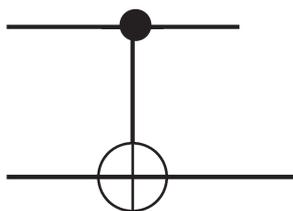


Figure 1.3: Circuit representation for the controlled-NOT gate. The top line represents the control qubit, the bottom line the target qubit.

pressed in the fact that U_{CN} is a *unitary matrix*, that is, $U_{CN}^\dagger U_{CN} = I$. More generally, with U is an arbitrary single qubit unitary operation. A *controlled- U* operation is a two qubit operation, again with a control and a target qubit. If the control qubit is set then U is applied to the target qubit, otherwise the target qubit is left alone; that is, $|c\rangle_1 |t\rangle_2 \rightarrow (\frac{1+Z_1}{2} + \frac{1-Z_1}{2} U_2) |c\rangle_1 |t\rangle_2$. The controlled- U operation is represented by the circuit shown in Figure 1.4.

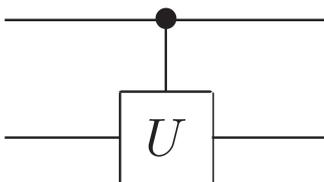


Figure 1.4: Controlled- U operation. The top line is the control qubit, and the bottom line is the target qubit.

We just show here other two very important unitaries in quantum computing. One is the the SWAP operator. The SWAP operator exchanges the state of two qubits. That is, in the computational basis SWAP maps

$$|00\rangle \rightarrow |00\rangle, |01\rangle \rightarrow |10\rangle, |10\rangle \rightarrow |01\rangle \text{ and } |11\rangle \rightarrow |11\rangle \quad (1.26)$$

The SWAP gate is equivalent to the matrix

$$\text{SWAP} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.27)$$

with respect to the computational basis.

In quantum information, perhaps the most common three-qubit unitary discussed is the Toffoli gate [16]. This gate is similar to the CNOT except that the Toffoli gate has two controls instead of one; thus, the Toffoli gate is also known as the CCNOT gate. The Toffoli gate can therefore be represented as

$$\text{Toffoli} \equiv (II - |11\rangle\langle 11|) \otimes I + |11\rangle\langle 11| \otimes X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad (1.28)$$

in the computational basis.

1.2.1 QUANTUM MEASUREMENT

A special kind operation left as last in this section is measurement. As a circuit element, a projective measurement in the computational basis is denoted using a 'meter' symbol, illustrated in Figure 1.5. In the theory of quantum circuits it is conventional to not use any special symbols to denote more general measurements, because they can always be represented by unitary transforms with ancilla qubits followed by projective measurements [14].

Two important principles in quantum circuits regarding measurements are, as stated in [14]:

- **Principle of deferred measurement:** Measurements can always be moved

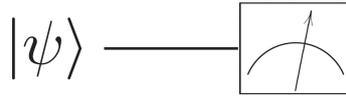


Figure 1.5: Symbol for projective measurement on a single qubit.

from an intermediate stage of a quantum circuit to the end of the circuit; if the measurement results are used at any stage of the circuit then the classically controlled operations can be replaced by conditional quantum operations.

- **Principle of implicit measurement:** Without loss of generality, any un-terminated quantum wires (qubits which are not measured) at the end of a quantum circuit may be assumed to be measured.

This section ends with the concept of *universality* and of it is achieved in quantum computation. A set of gates is said to be *universal for quantum computation* if any unitary operation may be approximated to arbitrary accuracy by a quantum circuit involving only those gates.

Here follows three universality constructions for quantum computation, for proofs see [14],

- **Two-level unitary gates are universal:** an arbitrary unitary operator may be expressed *exactly* as a product of unitary operators that each acts non-trivially only on a subspace spanned by two computational basis states.
- **Single qubit and CNOT gates are universal:** an arbitrary unitary operator may be expressed exactly using single qubit and CNOT gates.
- Any single qubit operation may be approximated to arbitrary accuracy using the Hadamard, phase, and $\pi/8$ gates.

Thus, these three imply that any unitary operation can be approximated to arbitrary accuracy using Hadamard, phase, CNOT, and $\pi/8$ gates.

It may be shown [14] that this universality approximation is achieved also with Hadamard, phase, CNOT and Toffoli gates.

1.3 QUANTUM COMPUTERS AND QUANTUM SIMULATORS

A quantum computer is a device that can arbitrarily manipulate the quantum state or part of itself. The experimental requirements for building a quantum computer concern the elementary units of the theory, the quantum bits. To realize a quantum computer, as stated in [14], we need to:

- give qubits some robust physical representation (to assure that they retain their quantum properties)
- select a system in which they can be made to evolve as desired, in particular where its possible to perform a universal family of unitary transformations
- be able to prepare qubits in some specified set of initial states
- be able to measure the final output state of the system

The challenge of experimental realization is that these basic requirements can often only be partially met [14]. A quantum computer has to be well isolated in order to retain its quantum properties, but at the same time its qubits have to be accessible so that they can be manipulated to perform a computation and to read out the results. Thus, a realistic implementation must strike a delicate balance between these constraints.

Here follows some complete physical models for a realizable quantum computer:

- **Optical photon quantum computer:** single photons can serve as good qubits. Considered two cavities, we use the states $|01\rangle$ and $|10\rangle$, states that represent whether the photon is in one cavity or the other, to be as logical 0 and 1. However conventional nonlinear optical materials which are sufficiently strong to allow single photons to interact inevitably absorb or scatter the photons.
- **Optical cavity quantum electrodynamics:** cavity-QED is a technique by which single atoms can be made to interact strongly with single photons. It provides a mechanism for using an atom to mediate interactions between single photons.

- **Ion traps:** trapped ions can be cooled to the extent that their electronic and nuclear spin states can be controlled by applying laser pulses. By coupling spin states through center-of-mass phonons, logic gates between different ions can be performed.
- **Nuclear magnetic resonance:** nuclear spins are nearly ideal qubits, and single molecules would be nearly ideal quantum computers if their spin states could only be controlled and measured. Nuclear magnetic resonance makes this possible using large ensembles of molecules at room temperature, but at the expense of signal loss due to an inefficient preparation procedure.

Another kind of quantum computer realization is achieved with superconducting qubits, for more exhaustive description see [17]:

- **Superconducting qubits:** in a superconductor, the basic charge carriers are pairs of electrons (known as Cooper pairs), rather than the single electrons in a normal conductor. At every point of a superconducting electronic circuit, the condensate wave function describing the charge flow is well-defined by a specific complex probability amplitude [18]. Research in superconducting quantum computing is conducted by companies such as Google, IBM, IMEC, BBN Technologies, Rigetti, and Intel.

Most of the originally proposed quantum algorithms were thought to be implemented on the theories surrounding fault-tolerant quantum computation [2]. Fault-tolerant quantum computing refers to the framework of ideas that allow qubits to be protected from quantum errors introduced by poor control or environmental interactions (Quantum Error Correction, QEC) and the appropriate design of quantum circuits to implement both QEC and encoded logic operations in a way to avoid these errors cascading through quantum circuits [3]. For tasks theoretically proposed to be performed by this kind of devices, would be required millions of physical qubits to incorporate these QEC techniques successfully [19]. Unfortunately, building an error-corrected quantum computer with millions of physical qubits may take decades [2].

Currently realized quantum computers are the so called “*Noisy Intermediate-Scale Quantum* (NISQ)” devices, i.e. those devices whose qubits and quantum operations are substantially imperfect. One of the goals in the NISQ era is to extract

the maximum quantum computational power from current devices while developing techniques that may also be suited for the long-term goal of the fault tolerant quantum computation [5]. Quantum algorithms specially developed to be run on current quantum computing hardware or those which could be developed in the next few years, the term *near-term* quantum computation has been coined [2]. The NISQ era corresponds to the period when only a few hundred noisy qubits are available, in contrast, the nearterm era involves any quantum computation performed in the next few years.

As exposed previously, in many fields such as quantum optics, trapped ions, superconducting circuit devices, and most recently nanomechanical systems, it is possible to design systems using a small number of effective oscillator and spin components, excited by a limited number of quanta [10]. In 2019, the Google AI Quantum team implemented an experiment with the 53-qubit Sycamore chip, in which single-qubit gate fidelities of 99.85% and two-qubit gate fidelities of 99.64% were attained on average [18].

1.3.1 QUANTUM SIMULATION AND CLASSICAL COMPUTING TECHNIQUES

As stated in subsection 1.1.1, every quantum system encountered in the real world is an open quantum system. For although much care is taken experimentally to eliminate the unwanted influence of external interactions, there remains, if ever so slight, a coupling between the system of interest and the external world. In addition, any measurement performed on the system necessarily involves coupling to the measuring device, therefore introducing an additional source of external influence. Consequently, developing the necessary tools, both theoretical and numerical, to account for the interactions between a system and its environment is an essential step in understanding the dynamics of practical quantum systems [10].

One of the most important practical applications of computation is the simulation of physical systems. Classical computers cannot simulate a quantum system efficiently, the number of complex numbers needed to describe a quantum system generally grows exponentially with the size of the system, rather than linearly, as occurs in classical systems. In general, storing the quantum state of a system with n

distinct components takes something like c^n bits of memory on a classical computer, where c is a constant which depends upon details of the system being simulated, and the desired accuracy of the simulation [14].

By contrast, a quantum computer can perform the simulation using kn qubits, where k is again a constant which depends upon the details of the system being simulated. This allows quantum computers to efficiently perform simulations of quantum mechanical systems that are believed not to be efficiently simulable on a classical computer [14].

In general, for all but the most basic of Hamiltonians, an analytical description of the system dynamics is not possible, and one must resort to numerical simulations [10]. In absence of a quantum computer, these simulations must be carried out using classical computing techniques, where the exponentially increasing dimensionality of the underlying Hilbert space severely limits the size of system that can be efficiently simulated. However, nowadays the available quantum computers are the NISQ devices, that are amenable to classical simulation in a truncated Hilbert space.

Thus, in this this context, can be studied and implemented algorithms suited to be executed with this kind of devices, with a limited number o maximum qubits. Among NISQ algorithms there is the QAOA, which applications and performances we've studied through classical quantum simulations using *QuTiP*. The *Quantum Toolbox in Python*, or QuTiP, is an open-source framework written in the Python programming language, designed for simulating the dynamics of open quantum systems.

QUANTUM ALGORITHMS

Quantum computers are designed to outperform standard computers by running quantum algorithms [19]. In this chapter we introduce algorithms focusing then on how to analyze an algorithm by giving an explanation about computational resources requirement and complexity classes. Then, after the fundamental concepts about algorithms are stated, we describe quantum algorithms, giving descriptions of the Grover algorithm and the Variational Quantum Algorithms (VQAs), family of algorithms which Quantum Approximate Optimization Algorithm (QAOA) belongs to. The last section of the chapter is dedicated to QAOA, the idea behind it, its properties and applications.

2.1 ALGORITHMS AND COMPLEXITY CLASSES

In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

As an effective method, an algorithm can be expressed within a finite amount of space and time, and in a well-defined formal language for calculating a function. Starting from an initial state and initial input, the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state. The

transition from one state to the next is not necessarily deterministic; for example some algorithms, known as randomized algorithms, incorporate random input or quantum algorithms, by the intrinsic probabilistic quantum mechanical nature of quantum states.

The study of algorithms is referred to the model of computation they're implemented to be run, there are many model of computation, both for classical and quantum computation, some examples are the *Turing machine*, *probabilistic Turing machine* or *circuit model* for classical computation and the *quantum circuit model* exposed in Section 1.2.

In the study of algorithms, the two fundamental properties to be analyzed are:

- computational task which can be performed
- required resources to perform it

Different models of computation lead to different resource requirements for computation [14]. However, many general considerations may be done about a quantitative analysis of resource requirements.

One of the tools to perform this analysis is the *asymptotic notation*, which can be used to summarize the *essential* behaviour of a function [19]. The asymptotic notation is used to express information about the growth rate of an algorithm. Given a resource of consumption of f , suppose $f(n)$ and $g(n)$ are two functions on the non-negative integers, there are three notations [14]:

- **O -notation:** $f(n)$ is $O(g(n))$ if there exist M, c such that $|f(n)| \leq c|g(n)|$ for all $n > M$
- **Ω -notation:** $f(n)$ is $\Omega(g(n))$ if there exist M, c such that $|f(n)| \geq c|g(n)|$ for all $n > M$
- **Θ -notation:** $f(n)$ is $\Theta(g(n))$ if there exist M, c_1, c_2 such that $c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|$ for all $n > M$

The asymptotic notation is important also in the introduction to *complexity classes*, another fundamental tool in the algorithms' analysis. Computational complexity is the study of the time and space resources required to solve computational problems. The task of computational complexity is to prove lower bounds on the

resources required by the best possible algorithm for solving a problem, even if that algorithm is not explicitly known [14]. The chief distinction made in computational complexity is between problems which can be solved using resources which are bounded by a *polynomial* in n , or which require resources which grow faster than any polynomial in n . In the latter case it is said that the resources required are *exponential* in the problem size. A problem is regarded as *easy*, tractable or feasible if an algorithm for solving the problem using polynomial resources exists, and as *hard*, intractable or infeasible if the best possible algorithm requires exponential resources. Complexity classes are groupings of problems by hardness, namely the scaling of the cost of solving the problem with respect to some resource, as a function of the “size” of an instance of the problem [2]. Here follows some of the fundamental complexity classes:

- P: decision problems, problems with a yes or no answer, that can be solved in time polynomial with respect to input size by a deterministic classical computer.
- NP: a problem is said to be in NP, if the problem of verifying the correctness of a proposed solution lies in P, irrespective of the difficulty of obtaining a correct solution.
- NP-Complete: NP-complete problems are a set of problems to each of which any other NP-problem can be reduced in polynomial time and whose solution may still be verified in polynomial time.
- NP-Hard: NP-hard problems are those at least as hard as NP problems

In Figure 2.1 there’s a representative scheme.

2.2 QUANTUM ALGORITHMS

Quantum algorithms are algorithms that run on any realistic model of quantum computation. The most commonly used model of quantum computation is the circuit model, and the *quantum Strong Church-Turing thesis* states that the quantum circuit model can efficiently simulate any realistic model of computation [21].

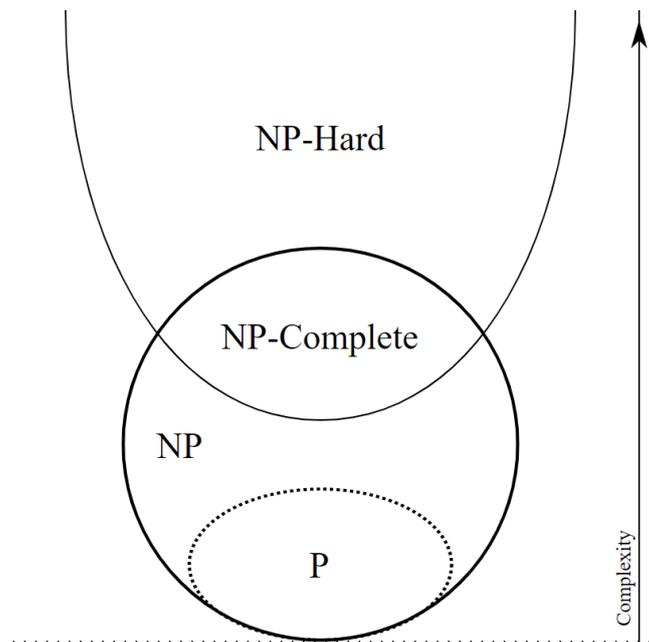


Figure 2.1: Euler diagram for P, NP, NP-complete, and NP-hard set of problems. Taken from [20]

The study of quantum algorithms is very important for several reasons. Computationally secure cryptography is widely used in society today, and relies on the believed difficulty of a small number of computational problems. Quantum computation appears to redefine what is a tractable or intractable problem, and one of the first breakthroughs in the development of quantum algorithms was Shor's discovery of efficient algorithms for factoring and finding discrete logarithms [22].

In the next section we describe one of the principal quantum algorithms, the *quantum search algorithm* or *Grover's algorithm*. Then, in the following section, we expose the variational quantum algorithms, a family of quantum algorithms that has been proposed for essentially all applications that researchers have envisioned for quantum computers, and they appear to be the best hope for obtaining quantum advantage [1].

2.2.1 QUANTUM SEARCH ALGORITHM

The quantum search algorithm was invented by Grover, and it offers a quadratic speed-up over classical algorithms for searching an unsorted database [23]. Suppose you are given a telephone directory sorted according to the names in alphabetic order. Suppose we know the telephone number or address of a person whose name

we want to find. For that purpose, the telephone directory is a usual unsorted database. To find out the person's name, we would need to identify the line ω that contains also his telephone number and address.

Given an unstructured database search, with $N \gg 1$ items the task is to locate one particular term ω . Denoted the size of the database $N = 2^n$, it is possible to encode the entire database in the quantum state $|s\rangle$ given by

$$|s\rangle \equiv |+\rangle^{\otimes n} = \sum_{z \in \{0,1\}^n} \frac{1}{\sqrt{N}} |z\rangle, \quad (2.1)$$

The idea of the algorithm is that the wanted state in all the database is $|\omega\rangle$, which is one of the states encoded in $|s\rangle$

$$\begin{aligned} |s\rangle &= \frac{1}{\sqrt{N}} |\omega\rangle + \frac{1}{\sqrt{N}} \sum_{\substack{z \in \{0,1\}^n \\ z \neq \omega}} |z\rangle \\ &= \frac{1}{\sqrt{N}} |\omega\rangle + \sqrt{\frac{N-1}{N}} |\omega^\perp\rangle, \end{aligned} \quad (2.2)$$

where

$$|\omega^\perp\rangle \equiv \frac{1}{\sqrt{1-N}} \sum_{z \neq \omega} |z\rangle. \quad (2.3)$$

The state $|s\rangle$ is stated as a state vector in the two-dimensional Hilbert space $\mathcal{H}_2 = \text{Span}\{|\omega\rangle, |\omega^\perp\rangle\}$, as shown in Figure 2.2. Therefore, with

$$\theta \equiv \arcsin \frac{1}{\sqrt{N}}, \quad (2.4)$$

we have

$$|s\rangle = \sin \theta |\omega\rangle + \cos \theta |\omega^\perp\rangle. \quad (2.5)$$

The idea of the algorithm is that, by applying certain rotation operators on $|s\rangle$, eventually $|s\rangle$ will end up in the wanted state $|\omega\rangle$. Here we will show how to find $|\omega\rangle$ from $|s\rangle$ in the following steps.

1° Apply

$$U_\omega \equiv I_2 - 2|\omega\rangle\langle\omega|, \quad (2.6)$$

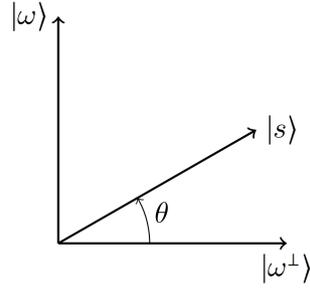


Figure 2.2: State $|s\rangle$ as a state vector in the two-dimensional Hilbert space $\mathcal{H}_2 = \text{Span}\{|\omega\rangle, |\omega^\perp\rangle\}$.

to $|s\rangle$ and obtain

$$\begin{aligned} |s'\rangle &= U_\omega |s\rangle \\ &= -\sin \theta |\omega\rangle + \cos \theta |\omega^\perp\rangle. \end{aligned} \tag{2.7}$$

The geometric implication of operator U_ω is reflecting the state $|s\rangle$ around the $|\omega^\perp\rangle$ axis, as can be seen in Figure 2.3.

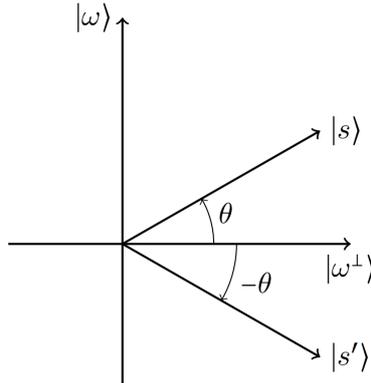


Figure 2.3: The first reflection operator U_ω in the geometrical picture. The state $|s\rangle$ is reflected around the axis $|\omega^\perp\rangle$.

2° Apply

$$U_s \equiv 2|s\rangle\langle s| - I_2, \tag{2.8}$$

to $|s'\rangle$ and obtain, for proof see [24],

$$\begin{aligned} |s_1'\rangle &= U_s |s'\rangle \\ &= -\sin 3\theta |\omega\rangle + \cos 3\theta |\omega^\perp\rangle. \end{aligned} \tag{2.9}$$

The geometric implication of operator U_s is reflecting the state around the $|s\rangle$ axis, as can be seen in Figure 2.4

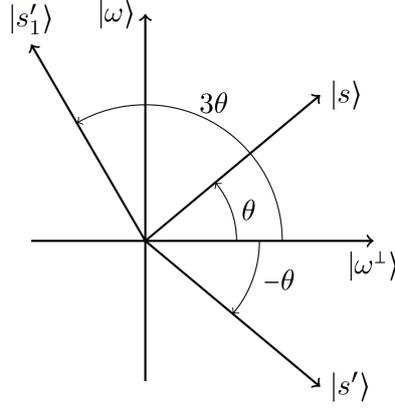


Figure 2.4: The second reflection from the state $|s'\rangle$ to the $|s'_1\rangle$ with the second reflection operator U_s .

$$R_{grov} \equiv U_s \circ U_\omega, \quad (2.10)$$

therefore

$$|s'_1\rangle = R_{grov}|s\rangle. \quad (2.11)$$

As we can see that the R_{grov}^n rotates the original state $|s\rangle$ by the angle 2θ . Thus, we can define further that

$$R_{grov}^n = \underbrace{R_{grov} \circ R_{grov} \circ \dots \circ R_{grov}}_n, \quad (2.12)$$

and we obtain

$$R_{grov}^n|s\rangle = \sin(2n+1)\theta|\omega\rangle + \cos(2n+1)\theta|\omega^\perp\rangle, \quad (2.13)$$

which is called *Grover iteration*.

So, supposing that with T -iteration we can get the marked $|\omega\rangle$, therefore

$$\begin{cases} (2T+1)\theta = \frac{\pi}{2} \\ \sin\theta = \frac{1}{\sqrt{N}} \end{cases} \Rightarrow \begin{cases} T = \frac{\pi}{4\theta} - \frac{1}{2} \\ \theta \simeq \frac{1}{\sqrt{N}}, \end{cases} \quad (2.14)$$

namely

$$T = \frac{\sqrt{N}\pi}{4} [1 + O(N^{-1/2})] \propto \sqrt{N}. \quad (2.15)$$

Thus, we have shown that the steps of the Grover algorithm find the solution with

$O(\sqrt{N})$ operations, surpassing the classical counterpart which would require $O(N)$ operations. Moreover, the quantum search algorithm is general in the sense that it can be applied far beyond the database search example just described to speed up many classical algorithms that use search heuristics [14].

2.2.2 VARIATIONAL QUANTUM ALGORITHMS

Most of the current NISQ algorithms relies on harnessing the power of quantum computers in a hybrid quantum-classical arrangement [2]. Such algorithms delegate the classically difficult part of some computation to the quantum computer and perform the classically tractable part on some sufficiently powerful classical device [1]. These algorithms variationally update the parameters of a parametrized quantum circuit and hence are referred to as *Variational Quantum Algorithms* (VQA), sometimes also called *Hybrid Quantum-Classical Algorithms*. One of the first proposed VQA was the Quantum Approximate Optimization Algorithm [6], proposed to solve combinatorial optimization problems.

A variational algorithm comprises several modular components that can be readily combined, extended and improved with developments in quantum hardware and algorithms [1]. Chief among these are:

- the cost function, the equation to be variationally minimized/maximized;
- one or several Parameterized Quantum Circuit (PQC), with the quantum circuit unitaries whose parameters are manipulated in the minimization/maximization of the cost function;
- the measurement scheme, which extracts the expectation values needed to evaluate the cost function;
- and the classical optimization, the method used to obtain the optimal circuit parameters that minimize/maximize the cost function.

Here follows a more detailed description of this components, to show features and potentiality of VQAs:

Objective function: the Hamiltonian H encodes information about a given system, and it naturally arises in the description of many physical systems, such

as in quantum chemistry problems. Its expectation value yields the energy of a quantum state, which is often used as the minimization target of a VQA. Other problems not related to real physical systems can also be encoded into a Hamiltonian form, thereby opening a path to solve them on a quantum computer. Hamiltonian operators are not all that can be measured on quantum devices; in general, any expectation value of a function written in an operational form can be extracted from a quantum computer. After the Hamiltonian of a problem has been determined it must be decomposed into a set of particular operators that can be measured with a quantum processor. Such a decomposition is an important step of many quantum algorithms in general and of VQA in particular [2].

Parameterized quantum circuits (PQC): following the objective function, the next essential constituent of a VQA is the quantum circuit that prepares the state that best meets the objective. It is generated by means of a unitary operation $U_{\vec{\theta}}$ that depends on a series of parameters $\vec{\theta}$, the PQC. Finding the parameter values of the PQC that deliver the optimal unitary, on the other hand, is the task of the classical optimization subroutine. The PQC is applied from an initial state $|\psi_0\rangle$ that can be the state at which the quantum device is initialized or a particular choice motivated by the problem at hand. Similarly, the initial parameters and the structure of the circuit are unknown a priori. However, knowledge about the particular problem can be leveraged to predict and postulate its structure [2].

Measurement: The main task in most quantum algorithms is to gain information about the quantum state that has been prepared on the quantum hardware. In VQA, this translates to estimate the expectation value of the objective function $\langle H \rangle_{U_{\vec{\theta}}}$. The most direct approach to estimate expectation values is to apply a unitary transformation on the quantum state to the diagonal basis of the observable H . Then, one can read out the expectation value directly from the probability of measuring specific computational states corresponding to an eigenvalue of H . Here, one needs to determine whether a measured qubit is in the $|0\rangle$ or $|1\rangle$ state. However, on NISQ devices, the transformation to the diagonal basis mentioned before can be an overly costly operation [2]. As a NISQ friendly alternative, most observables of interest can be efficiently parameterized in terms of Pauli strings.

Parameters optimization: In many VQAs, a core task that is allocated to the classical computer is the optimization of quantum circuit parameters. In principle,

this problem is not different from any multivariate optimization procedure and standard classical methods can be applied [25]. However, in the NISQ era, the coherence time is short, which means that complicated analytical gradient circuits can not be implemented [2]. At the same time, objective measurements take a long time, which means that algorithms with few function evaluations should be favored. As a last criterion, the optimizer should be resilient to noisy data coming from current devices and precision on expectation values that is limited by the number of shots in the measurement. These three requirements make that certain existing algorithms are better suited for PQC optimization and are more commonly used, and that new algorithms are being developed specifically for PQC optimization [2].

In the next section we introduce the Quantum Approximate Optimization Algorithm, where all the elements and considerations exposed for VQAs may be found specifically for this algorithm.

2.3 QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

The quantum approximate optimization algorithm (QAOA) is a quantum algorithm that produces approximate solutions for *combinatorial optimization problems* or *constraint satisfaction problems* (CSP), which can in general be reduced to that of finding the ground state of particular Ising models [9].

Combinatorial optimization problems are specified by n bits and a set $\{C_\alpha(z)\}$ of m clauses, or constraints. Each clause is a constraint on a subset of the bits which is satisfied for certain assignments of those bits and unsatisfied for the other assignments, of the form

$$C_\alpha(z) = \begin{cases} 1 & \text{if } z \text{ satisfies the clause } \alpha \\ 0 & \text{otherwise,} \end{cases} \quad (2.16)$$

where $z = z_1 z_2 \dots z_n$ is the bit string, $z \in \{0, 1\}^n$. Typically, clauses only evaluate a few bits; therefore, we will restrict all $C_\alpha(z)$ to evaluate the satisfiability of at most k bits, where k is some fixed integer. In other words, each clause is k -local [16].

The objective function, defined on n bit strings, is the number of satisfied clauses,

$$C(z) = \sum_{\alpha=1}^m C_{\alpha}(z), \quad (2.17)$$

and, depending on the problem request, solutions are the bitstring for which the cost function is maximized or minimized.

The aim of the algorithm is to explore the space of bitstring states for a superposition which is likely to yield a close value for the H^C operator to the classical C_{max} or C_{min} value upon performing a measurement in the computational basis. Thus, the idea is to find, through an optimization parameters procedure, the optimal parameters for which the quantum estimation is close as possible to the ideal classical value of the cost function.

The quantum computer works in a 2^n dimensional Hilbert space with computational basis vectors $|z\rangle$, to encode the problem, the classical cost function (2.17) can be converted to a quantum problem Hamiltonian by promoting each binary variable z_i to a quantum spin Z_i [8]:

$$H^C = C(Z_1, Z_2, \dots, Z_n), \quad (2.18)$$

which is diagonal in the computational basis. This Hamiltonian, for combinatorial optimization problems, has the general expression of the Ising model one

$$H^C = \sum_{j < k} l_{jk} Z_j Z_k, \quad (2.19)$$

where l_{jk} correspond to scalar weights with real values [7].

We now describe the QAOA steps, a schematic of a p -level Quantum Approximation Optimization Algorithm is shown in Figure 2.5, where p is a positive integer which defines the number of alternate parameterized operations applied to the initial state.

QAOA starts with a uniform superposition over the n bitstring basis states,

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{2^n} \sum_{z \in \{0,1\}^n} |z\rangle. \quad (2.20)$$

2. QUANTUM ALGORITHMS

Then, chosen $2p$ variational parameters $(\vec{\gamma}, \vec{\beta}) = (\gamma_1, \dots, \gamma_p, \beta_1, \dots, \beta_p)$, a variational state is obtained through $2p$ evolutions applied to the initial state

$$|\vec{\gamma}, \vec{\beta}\rangle = U(H^B, \beta_p)U(H^C, \gamma_p) \dots U(H^B, \beta_1)U(H^C, \gamma_1)|s\rangle, \quad (2.21)$$

where the evolution operators are defined as

$$U(H^C, \gamma) = e^{-i\gamma H^C} = \prod_{\alpha=1}^m e^{-i\gamma H_\alpha^C}, \quad (2.22)$$

and

$$U(H^B, \beta) = e^{-i\beta H^B} = \prod_{j=1}^m e^{-i\beta H_j^B}, \quad (2.23)$$

with

$$H^B = \sum_{j=1}^n X_j, \quad (2.24)$$

indicated as *mixing hamiltonian*.

Consequently, the expectation value of H^C is determined in this variational state

$$F_p(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | H^C | \vec{\gamma}, \vec{\beta} \rangle, \quad (2.25)$$

which is done by repeated measurements of the quantum system in the computational basis. After a measurement a bitstring $z_i \in \{0, 1\}^n$ is obtained as outcome and, after N measurements each z_i is obtained with occurrency w_i , thus F_p is evaluated as

$$F_p = \frac{\sum_i^N w_i C(z_i)}{\sum_i^N w_i}. \quad (2.26)$$

A classical computer is used to search for the optimal parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ so as to maximize the averaged measurement output $F_p(\vec{\gamma}^*, \vec{\beta}^*)$,

$$(\vec{\gamma}^*, \vec{\beta}^*) = \arg \max_{\vec{\gamma}, \vec{\beta}} F_p(\vec{\gamma}, \vec{\beta}). \quad (2.27)$$

This is typically done by starting with some initial guess of the parameters and performing simplex or gradient based optimization. The idea of the QAOA is to perform an optimization method over the circuit parameters $(\vec{\gamma}, \vec{\beta})$ to maximize F_p .

This will specify a state $|\vec{\gamma}, \vec{\beta}\rangle$ which is likely to yield an approximately optimal partition $|z\rangle$ upon performing a measurement in the computational basis. A figure of merit for benchmarking the performance of QAOA is the *approximation ratio*

$$r = \frac{F_p(\vec{\gamma}^*, \vec{\beta}^*)}{C_{max}}. \quad (2.28)$$

In this work we use the approximation ratio to check performances of QAOA for different instances and QAOA levels. The approximation ratio, from how it is defined, lies in the range $0 < r \leq 1$, where the quality of the approximation is proportional to r . We expect the quality of the approximation improves as p is increased, as stated in [6].

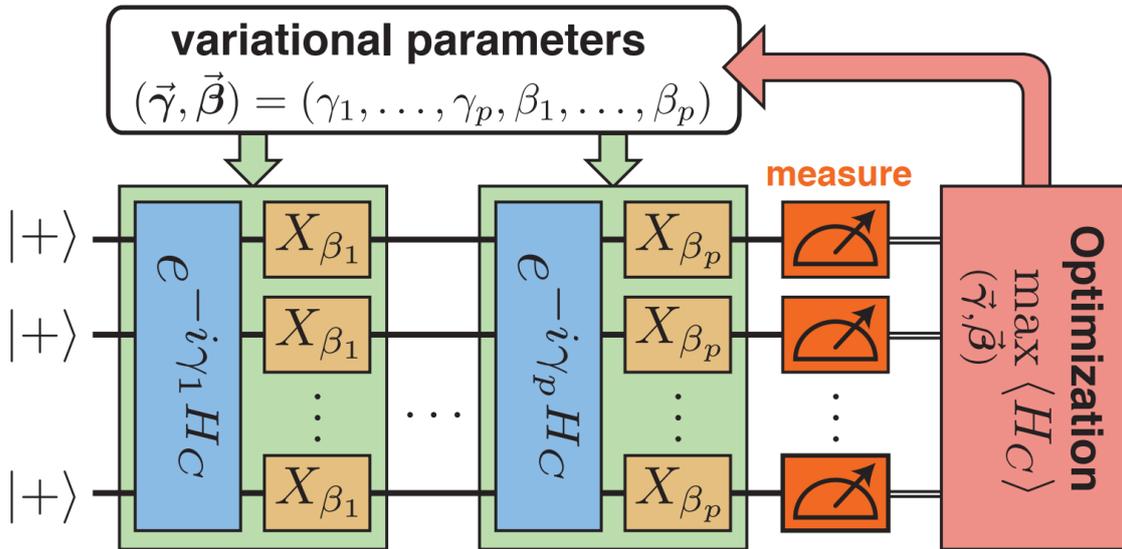


Figure 2.5: Schematic of a p -level Quantum Approximation Optimization Algorithm. In this scheme the rotation operator $R_j^X(\beta_i)$ that the operator H^B applies on the j th qubit is indicated as X_{β_i} . Taken from [8]

The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints [6].

Just to give a grasp of a physical realization of QAOA in a real quantum computer, we give here realization in quantum circuit model the initial state preparation

and of the QAOA unitaries . The initial state is prepared Hadamard gates as

$$\begin{aligned}
 |0\rangle & \text{---} \boxed{H} \text{---} \\
 |s\rangle = H^{\otimes n}|0\rangle^{\otimes n} & = |0\rangle \text{---} \boxed{H} \text{---} \\
 & \vdots \\
 & \vdots \\
 |0\rangle & \text{---} \boxed{H} \text{---}
 \end{aligned} \tag{2.29}$$

Each of the Pauli-X interactions in the mixing Hamiltonian (2.24) can be implemented with a one-qubit gate [9],

$$e^{-\frac{i}{2}\beta X_j} \equiv \text{---} \boxed{R_x(\beta)} \text{---} \tag{2.30}$$

and each of the two-qubit $Z_u Z_v$ interactions in the cost Hamiltonian (2.19) can be implemented with two CNOT gates, plus a local one-qubit gate [9],

$$e^{-\frac{i}{2}\gamma(I-Z_u Z_v)} \equiv \begin{array}{c} \text{---} \bullet \text{---} \\ | \\ \oplus \text{---} \boxed{R_z(-\gamma)} \text{---} \oplus \\ | \\ \text{---} \bullet \text{---} \end{array} . \tag{2.31}$$

2.3.1 OPTIMIZATION METHOD

Hybrid quantum-classical optimization with parameterized quantum circuits provides a promising approach for understanding and exploiting the potential of noisy intermediate-scale quantum (NISQ) devices [26]. While a variety of gradient-free optimization methods have been proposed and studied [27], in this work we exploit a gradient based optimizer, the gradient descent optimizer, with *finite differences* as derivative estimator and *Adam* as the algorithm to implement the parameters update.

Given a model parameterized by $\vec{\theta} \in \mathbb{R}^d$ and a cost function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ which depends on $\vec{\theta}$, the exact gradient descent update rule is

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \eta \nabla F(\vec{\theta}^{(t)}), \tag{2.32}$$

where, for QAOA, $\vec{\theta} = (\vec{\gamma}, \vec{\beta})$, $d = 2p$ and $F = F_p$, the expectation value of the cost

Hamiltonian H^C for the QAOA level p . In the context of hybrid quantum-classical optimization, it is not possible to evaluate the exact gradient of the cost function, but it is computed by repeating the state preparation and measurement steps and accumulating enough outcome statistics to estimate the expectation value of H^C . Thus, the gradient descent update rule is replaced with a stochastic update rule of the form

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - \eta \nabla g^{(t)}(\vec{\theta}^{(t)}), \quad (2.33)$$

where $g^{(t)}(\vec{\theta})$ is the estimator of the gradient.

A general way of approximating the gradient is to use a finite-difference estimator, which requires to experimentally measure expectation values for slightly different values of the parameters. We evaluate the estimators of the gradient with the *central finite difference method*. Given a fixed step size $h > 0$, the symmetric finite difference estimator for the i_{th} element of the gradient can be defined as:

$$g_i^{(t)}(\vec{\theta}^{(t)}) = \frac{F(\vec{\theta} + h\vec{e}_i) - F(\vec{\theta} - h\vec{e}_i)}{2h}, \quad (2.34)$$

where \vec{e}_i is the i_{th} element of the canonical base vector in the d -dimensional parameters space.

There are many algorithms for gradient descent optimization [28], in this work we use the Adaptive Moment Estimation (Adam) method proposed in [12]. *Adam* is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [12]. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters [12]. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods [12].

Here we show the pseudo-code of the algorithm,

Algorithm 1 *Adam*, with hyperparameters set as $\eta = 0.05$, $\hat{\beta}_1 = 0.9$, $\hat{\beta}_2 = 0.999$, $\epsilon = 10^{-8}$, $F(\vec{\theta})$ evaluated as (2.26) with 100 samplings and $h = 0.1$, $\vec{\theta}_0$ initialized with each θ_i chosen randomly near 0.

Require: η : Stepsize

Require: $\hat{\beta}_1, \hat{\beta}_2 \in [0, 1)$: Exponential decay rates for the moment estimates

Require: ϵ : smoothing term that avoids division by zero

Require: $F(\vec{\theta})$: Stochastic objective function with parameters $\vec{\theta}$

Require: $\vec{\theta}_0$: Initial parameter vector

$m_0 \leftarrow 0$ (Initialize 1st moment vector)

$v_0 \leftarrow 0$ (Initialize 2nd moment vector)

$t \leftarrow 0$ (Initialize timestep)

for N steps **do**

$t \leftarrow t + 1$

$g_t \leftarrow \vec{g}^{(t)}(\vec{\theta}^{(t-1)})$ (Get gradients w.r.t. stochastic objective at timestep t as in (2.34))

$m_t \leftarrow \hat{\beta}_1 \cdot m_{t-1} + (1 - \hat{\beta}_1) \cdot g_t$ (Update biased first moment estimate)

$v_t \leftarrow \hat{\beta}_2 \cdot v_{t-1} + (1 - \hat{\beta}_2) \cdot g_t^2$ (Update biased second raw moment estimate)

$\hat{m}_t \leftarrow m_t / (1 - \hat{\beta}_1^t)$ (Compute bias-corrected first moment estimate)

$\hat{v}_t \leftarrow v_t / (1 - \hat{\beta}_2^t)$ (Compute bias-corrected second raw moment estimate)

if cost function has to be minimized

$\vec{\theta}_t \leftarrow \vec{\theta}_{t-1} - \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)

if cost function has to be maximized

$\vec{\theta}_t \leftarrow \vec{\theta}_{t-1} + \eta \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)

end for

return $\vec{\theta}_t$ (Resulting parameters)

The algorithm updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t) where the hyper-parameters $\hat{\beta}_1, \hat{\beta}_2 \in [0, 1)$ control the exponential decay rates of these moving averages. The moving averages themselves are estimates of the 1st moment (the mean) and the 2nd raw moment (the uncentered variance) of the gradient. However, these moving averages are initialized as (vectors of) 0's, leading to moment estimates that are biased towards zero, especially during the initial timesteps, and especially when the decay rates are small (i.e. the $\hat{\beta}$ s are close to 1). This initialization bias is counteracted, resulting in bias-corrected estimates \hat{m}_t and \hat{v}_t .

2.3.2 QAOA APPLICATIONS: MAX CUT AND RBIM

In this work, we perform the classical simulation of QAOA for the Max Cut problem and in the study of ground state configurations of spins in the Random Bond Ising Model (RBIM). Here we briefly introduce the two problems.

MAX CUT

The Max Cut problem is an example of a CSP and can be defined as follows: given a set of vertices V and a set of edges E between the vertices in V , Max Cut on a simple graph $G = (V; E)$ is the problem of finding a bi-partition of V , a *maximum cut*, that maximizes the number of edges that run between the two partitions. Note that the edge between vertices u and v will be indicated by $\langle u, v \rangle$. Since we will only consider undirected graphs, $\langle u, v \rangle$ is equivalent to the set $\{\langle u, v \rangle\}$. The choice of objective function for Max Cut is

$$C(z) = \sum_{\langle u, v \rangle \in E} C_{uv}(z), \quad (2.35)$$

where

$$C_{uv}(z) = \frac{1}{2}(z_u - z_v)^2. \quad (2.36)$$

On inspection, we see that for an edge $\langle u, v \rangle$ the clause $C_{\langle u, v \rangle}(z)$ assigns the value 1 if $z_u \neq z_v$ and 0 if $z_u = z_v$. Thus, $C(z)$ is maximized when the number of edges whose vertices are not in the same partition is maximized.

Thus, in the QAOA implementation, the Max Cut cost function translates into the Hamiltonian

$$H^C = \sum_{\langle u,v \rangle \in E} \frac{1}{2}(I - Z_u Z_v), \quad (2.37)$$

which shows that this problem can be reduced to that of finding the ground state of an antiferromagnetic Ising model [9], defined on the graph G .

For $p > 1$ the QAOA procedure is the one exposed previously, however, for MAX Cut, there's a theorem, [29], which gives an analytical expression of F_p for the $p = 1$ case

$$\begin{aligned} F_1(\gamma, \beta) = \langle C \rangle &= \sum_{\langle u,v \rangle \in E} \langle C_{uv} \rangle = \\ & \sum_{\langle u,v \rangle} \frac{1}{2} + \frac{1}{4}(\sin 4\beta \sin \gamma)(\cos^{d_u} \gamma + \cos^{d_v} \gamma) - \frac{1}{4}(\sin^2 2\beta \cos^{d_u+d_v-2\lambda_{uv}} \gamma)(1 - \cos^{\lambda_{uv}} 2\gamma), \end{aligned} \quad (2.38)$$

with d_u : degree of vertex $u - 1$, d_v : degree of vertex $v - 1$, λ_{uv} : number common neighbours of vertices u and v , for proof see appendix A. So, for the $p = 1$, QAOA for Max Cut has a complete classical implementation, and its performances are confronted with the hybrid implementation for higher QAOA depths.

Max Cut is the most used problem for many QAOA analysis, [9],[8], [7] just to cite few of the many references about QAOA performance studies applied for Max Cut. In this work we analyze my QAOA implementation, correct behaviour and performances firstly for Max Cut. Consequently we apply the QAOA in another, more physical relevant, situation, the random bond Ising model, where we check the classical behaviour of the spin configurations of 2D lattices, for different probability distributions of the two different natures (ferromagnetic or antiferromagnetic) of the links between nearest neighbours sites.

RANDOM BOND ISING MODEL (RBIM)

Ising-type models play an important role as the simplest nontrivial models for describing magnetically ordered phases and corresponding phase transitions [13]. Here the focus is on the random bond Ising model (RBIM), which serves as a simple model for phase transitions in a system with quenched disorder [13].

The random bond Ising model consists of the standard Ising Hamiltonian with

nearest neighbour couplings on the two-dimensional (2D) square lattice

$$\mathcal{H}_I = -J \sum_{\langle i,j \rangle} K_{ij} S_i S_j, \quad (2.39)$$

where S_i, S_j are Ising variables that take on values ± 1 , J is the coupling constant and K_{ij} is a random coupling given by the probability distribution $P(K_{ij})$. In particular, by considering the RBIM on a square lattice with a binary probability distribution

$$P(K_{ij}) = \lambda \delta(K_{ij} - K) + (1 - \lambda) \delta(K_{ij} + K), \quad (2.40)$$

with $K > 0$. In words, the coupling K_{ij} are ferromagnetic with probability λ and antiferromagnetic with probability $1 - \lambda$. For this system, a phase transition is expected from the ferromagnetic to paramagnetic phases as shown in Figure 2.6. For $\lambda = 1$,

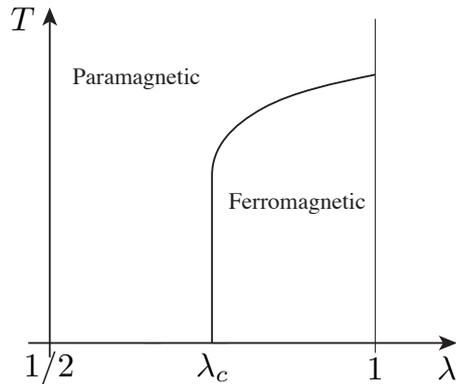


Figure 2.6: Phase diagram (λ vs. $T = 1/K$) for the 2D RBIM on a square lattice with fixed points, phase boundaries (solid line), ferromagnetic and paramagnetic phases. Taken from [13]

the system is a pure ferromagnetic Ising model with $T_c^{-1} = \ln(\sqrt{2} + 1)/2 \approx 0.44$ [13]. The pure ferromagnetic case can be mapped onto the antiferromagnetic case by sending $S_i \rightarrow -S_i$ for i on one sublattice [13]. More generally, this transformation is equivalent to sending $p \rightarrow 1 - \lambda$. Hence only the region $0 \leq \lambda \leq 1/2$ is considered. The solid line is a phase boundary which separates the ferromagnetically-ordered from the paramagnetic phase. For $\lambda < 1$, this critical temperature is reduced by frustration induced by the bond configuration until it vanishes at $\lambda = \lambda_c \approx 0.88$ [13].

In this work, we consider only the ground state of the system for the $T = 0$ limit.

For different probabilities, the Ising Hamiltonian

$$H_I = -J \sum_{\langle j,k \rangle} K_{jk} Z_j Z_k, \quad (2.41)$$

is minimized through the QAOA and the classical energy E and magnetization M of the system are evaluated. What we expect is that, around p_c , the behaviours of E and M show an observable change, with respect to the probability. It is important to notice that, the considerations and results exposed in [13] about the phase transition expected for $T = 0$ at $\lambda \approx 0.88$ has been obtained in the *thermodynamic limit*. Thus, here, my purpose is principally to show another application of my QAOA implementation and to see if, even for small number of lattice sites, a more qualitative than quantitative result may be obtained.

QAOA RESULTS FOR MAX CUT AND RBIM

In this chapter we show and describe the results of our implementation of the quantum approximate optimization algorithm (QAOA) applied to Max Cut and random bond Ising model (RBIM) optimization problems. The Max Cut is a combinatorial optimization problem, well suited for optimization and performances analysis. The other problem, the RBIM, which describe the interaction of classical spins in a two-dimensional (2D) lattice, offers an application example of our algorithm to a physical problem. In Section 3.1 we show the results for the Max Cut problem, in Section 3.2 we show the results for the RBIM.

3.1 MAX CUT: QAOA RESULTS

The Max Cut problem introduced in chapter 2 is a combinatorial optimization problem and can be defined as follows: given a set of *vertices* (or *nodes*) V and a set of *edges* E between the vertices in V , Max Cut on a simple graph $G = (V; E)$ is the problem of finding a bi-partition of V , a *maximum cut*, that maximizes the number of edges that run between the two partitions.

The Max Cut problem may be stated in terms of finding the maxima of the cost function

$$C(z) = \sum_{\langle u,v \rangle \in E} \frac{1}{2} (z_u - z_v)^2, \quad (3.1)$$

3. QAOA RESULTS FOR MAX CUT AND RBIM

where $\langle u, v \rangle$ indicates the edge between vertices u and v , see for details subsection 2.3.2 in chapter 2. expressed in (2.35), i.e. finding the configurations of nodes after a cut, expressed as $z = z_1 z_2 \dots z_n$, where n is the number of nodes of the graph and each z_u takes value 1 the node u belongs to one partition or 0 otherwise. If we denote with z^* a possible maximum cut partition we have that $C(z^*) = C_{max}$, where C_{max} is maximum number of edges that go from one partition to the other.

We apply the QAOA on three different graphs, with 5, 7 and 9 nodes, labelled as G_1 , G_2 and G_3 respectively. The three graphs are shown in Figure 3.1, where we highlight one of the possible maximum cuts for each graph, i.e. one of the solutions to the Max Cut problem. For example, in the graph G_1 a maximum cut is expressed by the bitstring $z^* = 00011$ and $C_{max} = 6$, in the graph G_2 a maximum cut is expressed by the bitstring $z^* = 0111010$ and $C_{max} = 11$ and in the graph G_3 a maximum cut is expressed by the bitstring $z^* = 001101010$ and $C_{max} = 14$.

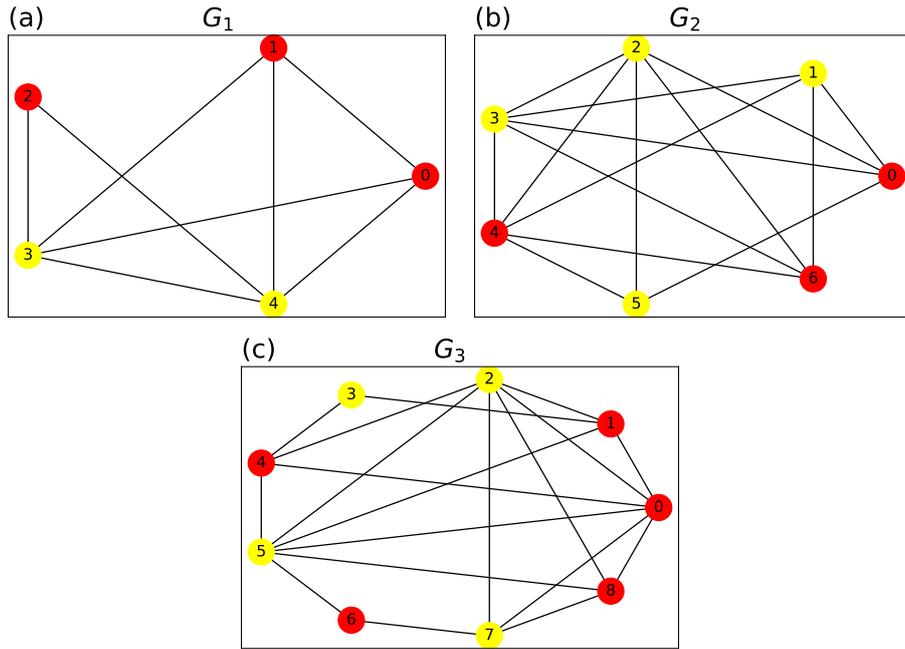


Figure 3.1: (a): 5 nodes graph G_1 , (b): 7 nodes graph G_2 , (c): 9 nodes graph G_3 . For each graph we highlight one of the maximum cuts, for (a): $z^* = 00011$ and $C_{max} = 6$, (b): $z^* = 0111010$ and $C_{max} = 11$, (c): $z^* = 001101010$ and $C_{max} = 14$.

We describe here the QAOA steps to find the approximate solution (for details see Section 2.3 of chapter 2):

1. we consider a graph G of which we want to compute the Max Cut. This defines

the number of qubits, the Hilbert space the cost function $C(z)$ (3.1) and the cost Hamiltonian

$$H^C = \sum_{\langle u,v \rangle \in E} \frac{1}{2}(I - Z_u Z_v), \quad (3.2)$$

where the configuration nodes z_u and z_v in $C(z)$ translate into the Pauli operators Z_u and Z_v , for details see subsection 2.3.2 in chapter 2. For example here we want to find solution to Max Cut for a given graph and the number of nodes corresponds to the number of qubits and the edges defines the expression of the cost function.

2. all the possible configurations are encoded in the initial state

$$|s\rangle = \sum_{z \in \{0,1\}^n} |z\rangle, \quad (3.3)$$

which is the uniform superposition of all the computational base states $|z\rangle$ with $z \in \{0, 1\}^n$. That is, each $|z\rangle$ is the quantum state which encodes the classical counterpart bitstring z , one of the possible configurations of the problem, for details see Section 2.3 in chapter 2.

3. we choose a level p of the QAOA. This implies that we use $2p$ parameters $(\vec{\gamma}, \vec{\beta}) = (\gamma_1, \dots, \gamma_p, \beta_1, \dots, \beta_p)$ and then apply $2p$ evolution operations to $|s\rangle$ to obtain the final state

$$|\vec{\gamma}, \vec{\beta}\rangle = U(H^B, \beta_p)U(H^C, \gamma_p) \dots U(H^B, \beta_1)U(H^C, \gamma_1)|s\rangle, \quad (3.4)$$

where $U(H^C, \gamma) = e^{-i\gamma H^C}$ and $U(H^B, \beta) = e^{-i\beta H^B}$, with $H^B = \sum_j^n X_j$ (for more details see Section 2.3 in chapter 2). The idea of the QAOA is that for certain optimal parameters, indicated as $(\vec{\gamma}^*, \vec{\beta}^*)$, the obtained final state encodes the problem solution. That is, the optimal final state is expected to be a superposition of all the possible configurations, where the optimal solutions are encoded as the states with higher probability in the superposition.

4. find the optimal parameters by applying an optimization method. Here we use a gradient based method, implemented through the Adaptive Moment Estimation (Adam) algorithm, described in Section 2.3.1. At each step of the

3. QAOA RESULTS FOR MAX CUT AND RBIM

optimization, we perform the steps 2. and 3., evaluate the gradient of the expectation value of the cost function $F_p(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | H^C | \vec{\gamma}, \vec{\beta} \rangle$ over the final state and use the gradient to update new parameters. F_p is obtained through N quantum measurements on the computational states, where each returns as output a classical bitstring z , corresponding to one of the possible problem configurations and F_p is evaluated through the classical mean over all the N outcomes, as stated in (2.26).

5. thus, after the classical optimization procedure, the obtained parameters are the ones which maximize the expectation value of the cost function F_p and for which the optimal final state $|\vec{\gamma}^*, \vec{\beta}^*\rangle$ encodes the problem solution.

We use the *approximation ratio*

$$r = \frac{F_p(\vec{\gamma}^*, \vec{\beta}^*)}{C_{max}}, \quad (3.5)$$

with $0 < r < 1$, that gives a benchmark in the analysis of the quality of the QAOA approximation, i.e. its performance. The idea is that, the closer r is to 1, the better the approximation is. We expect the QAOA performances to increase with the QAOA level p , in particular theoretically is expected that

$$r \xrightarrow[p \rightarrow \infty]{} 1, \quad (3.6)$$

as stated in [6].

In the next subsection we show the results about the optimization method and the performances of the QAOA for different p -levels in terms of r , applied to the three graphs G_1 , G_2 and G_3 . We also present results and analysis for the special case of $p = 1$ and the solutions for the three graphs obtained with QAOA for the $p = 1$ and $p = 6$ levels.

3.1.1 OPTIMIZATION METHOD AND RESULTS

We show in Figure 3.2 the approximation ratio as a function of the optimization steps for the three graphs G_1 , G_2 and G_3 . In all the optimization procedures, we initialize the parameters randomly near zero, and perform 200 Adam steps. For all

3. QAOA RESULTS FOR MAX CUT AND RBIM

the graphs, the approximation ratio increases and reach a maximum value in the Adam steps and the performance of the optimization increase with the QAOA level p as expected.

3. QAOA RESULTS FOR MAX CUT AND RBIM

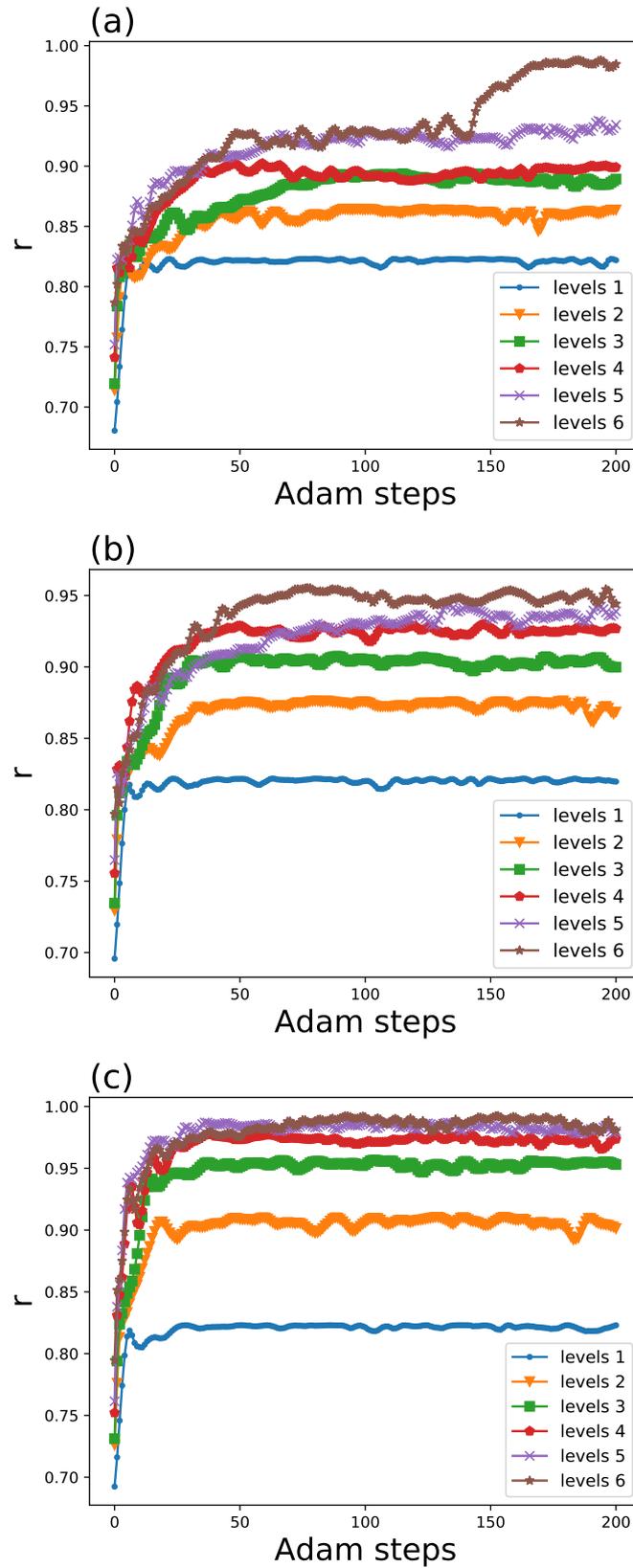


Figure 3.2: The approximation ratio $r = F_p/C_{max}$ of QAOA on the Max Cut optimization problem as a function of gradient descent steps, for different QAOA levels, for (a): 5 nodes graph, (b): 7 nodes graph, (c): 9 nodes graph. The approximation ratio increases with p , as expected.

We may perform a further analysis about the optimization procedure for the QAOA level $p = 1$ case. For the $p = 1$ case, we have an analytical expression of the expectation value of the cost function

$$F_1(\gamma, \beta) = \sum_{\langle u,v \rangle} \frac{1}{2} + \frac{1}{4} (\sin 4\beta \sin \gamma) (\cos^{d_u} \gamma + \cos^{d_v} \gamma) - \frac{1}{4} (\sin^2 2\beta \cos^{d_u+d_v-2\lambda_{uv}} \gamma) (1 - \cos^{\lambda_{uv}} 2\gamma), \quad (3.7)$$

with d_u : degree of vertex $u - 1$, d_v : degree of vertex $v - 1$, λ_{uv} : number common neighbours of vertices u and v , for proof see appendix 3 and (for more details see subsection 2.3.2 in chapter 3). Thus, for the $p = 1$ case, the parameters optimization may be performed classically by maximizing F_1 and by finding the optimal (γ^*, β^*) . Additionally we are able to study the parameters landscape with respect to F_1 . We plot the values of the approximation ratio $r = F_1/C_{max}$ in a grid of parameters (γ, β) , with $\beta \in [-\frac{\pi}{4}; \frac{\pi}{4}]$ and $\gamma \in [-\frac{\pi}{2}; \frac{\pi}{2}]$. We show, for the graph with 5 nodes, G_1 , in Figure 3.3 the Adam steps towards the maximum in the parameters landscape. The optimization starts with parameters generated randomly near zero in the point $A \approx (0, 0)$ and moves towards a maximum in the point $B \approx (-0.48, -0.27)$ with 200 Adam steps.

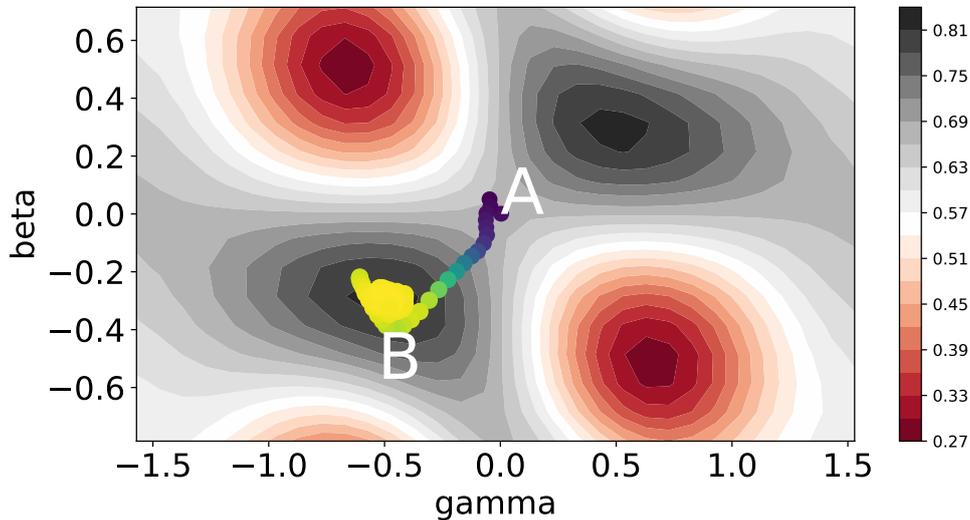


Figure 3.3: Adam steps towards maximum of the approximation ratio $r = F_p/C_{max}$ for graph with 5 nodes G_1 . The optimization starts at the point $(\gamma, \beta) \approx (0, 0)$ labeled as "A" in the figure and moves towards the maximum labeled as "B" in the figure. The maxima are indicated as an increasing opacity of black and the minima are indicated as an increasing opacity of red. The steps of the optimization are indicated as green dots that change color from darker to lighter green.

Another consideration that we can do with the $p = 1$ case is about the initial

3. QAOA RESULTS FOR MAX CUT AND RBIM

parameter of the optimization. We always start with parameters near zero in the optimization procedure. We show in Figure 3.4 the F_1 values in grid of different type of graphs. We may see that for not-regular graphs, for fixed γ , there are regions where, for any β , the cost function shows plateaux. Thus, for not-regular graphs, the parameters optimization may starts in one of these points and will not be able to reach one of the global maxima. For this reason we initialize the parameters near zero, to avoid starting points regions of function plateaux.

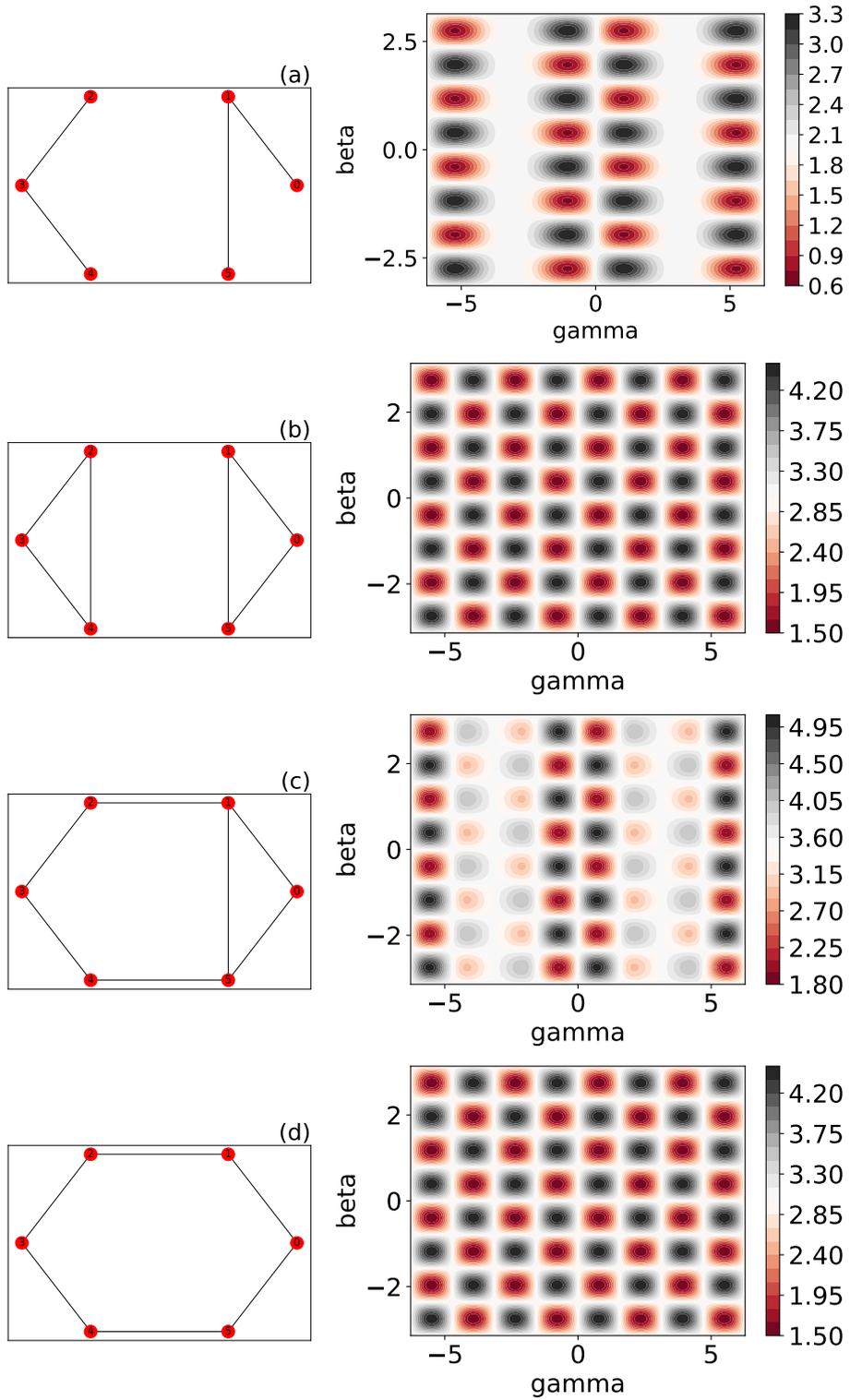


Figure 3.4: Plot of analytical F_1 in the parameter landscape for (a): not-regular, not-connected graph, (b): regular, not-connected graph, (c): not-regular, connected graph, (d): regular, connected graph. The periodicity and shape of F_1 depend on the regularity of the graph and for not-regular graphs, the function presents plateaux, which we want to avoid as starting points of the optimization

We have shown that the optimization step in our implementation of the QAOA behaves as expected. We show now that the QAOA returns the expected approximation solution. In the QAOA steps, after the classical optimization is performed, we have the optimal parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ from which we obtain the final state $|\vec{\gamma}^*, \vec{\beta}^*\rangle$. In Figures 3.5, 3.6, 3.7 we show the probability distributions of the optimal final states for the graph G_1, G_2, G_3 respectively, for QAOA levels $p = 1$ and $p = 6$, where the optimal parameters are the ones obtained after the optimizations showed in Figure 3.2. For G_1 the only maximum cut is the configuration $z^* = 00011$ (or $z^* = 11100$ equivalently). For G_2 the only maximum cut is the configuration $z^* = 0111010$ (or $z^* = 1000101$ equivalently). For G_3 the maximum cuts are the configurations $z^* = 001101010, 010010011, 010010101, 010010111, 010011010, 010011011, 011010101$ (or the equivalently configurations with the z_u flipped). We observe that, for all the three graphs, both for $p = 1$ and $p = 6$ in the superposition of all the possible states, the ones with higher probability in the final states are the ones corresponding to the solutions. In particular, we also show that the quality of the approximation for $p = 6$ is higher than the one for $p = 1$.

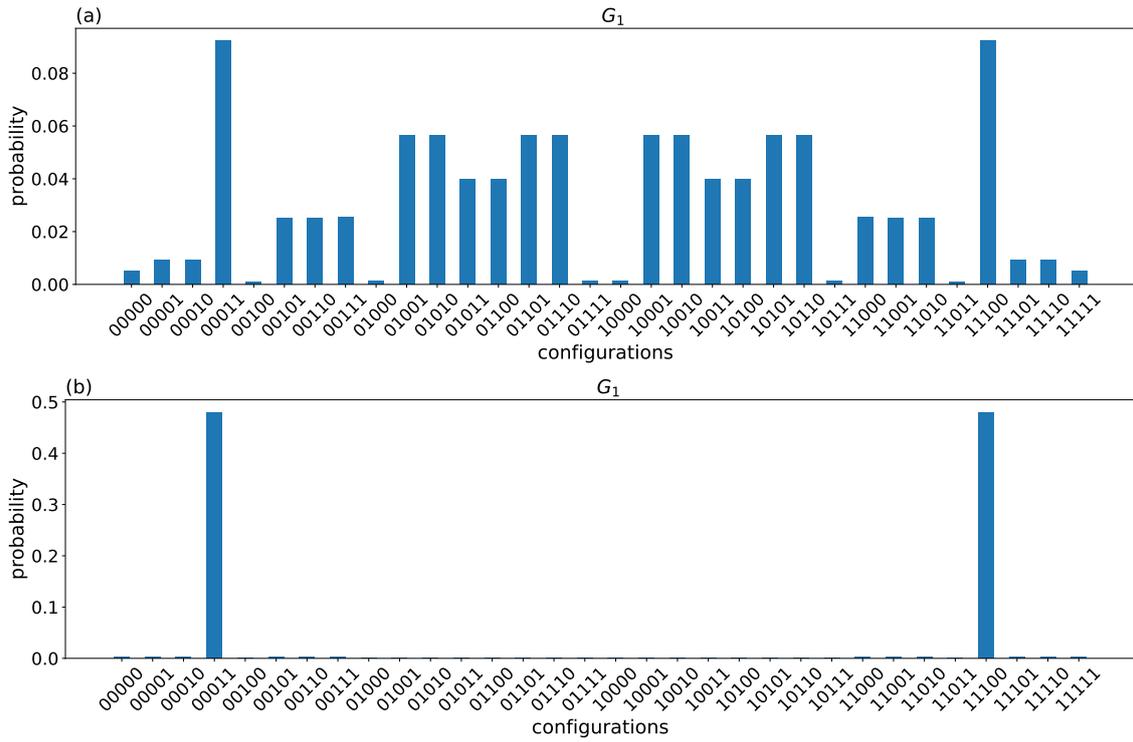


Figure 3.5: Probability distributions of all the possible configurations in the Max Cut problem for the 5 nodes graph G_1 encoded in the final states obtained after the optimizations showed in Figure 3.2 for (a): $p = 1$ QAOA level and (b): $p = 6$ QAOA level. The configurations are encoded as the computational basis states $|z\rangle$, with $z \in \{0, 1\}^5$, and the final state is a superposition of these states. For G_1 the maximum cut is $z^* = 00011$ (or $z^* = 11100$ equivalently) and for both levels the most probable states are the ones corresponding to the maximum cut. In particular, for the level $p = 6$ the approximation is higher than the $p = 1$ case.

3. QAOA RESULTS FOR MAX CUT AND RBIM

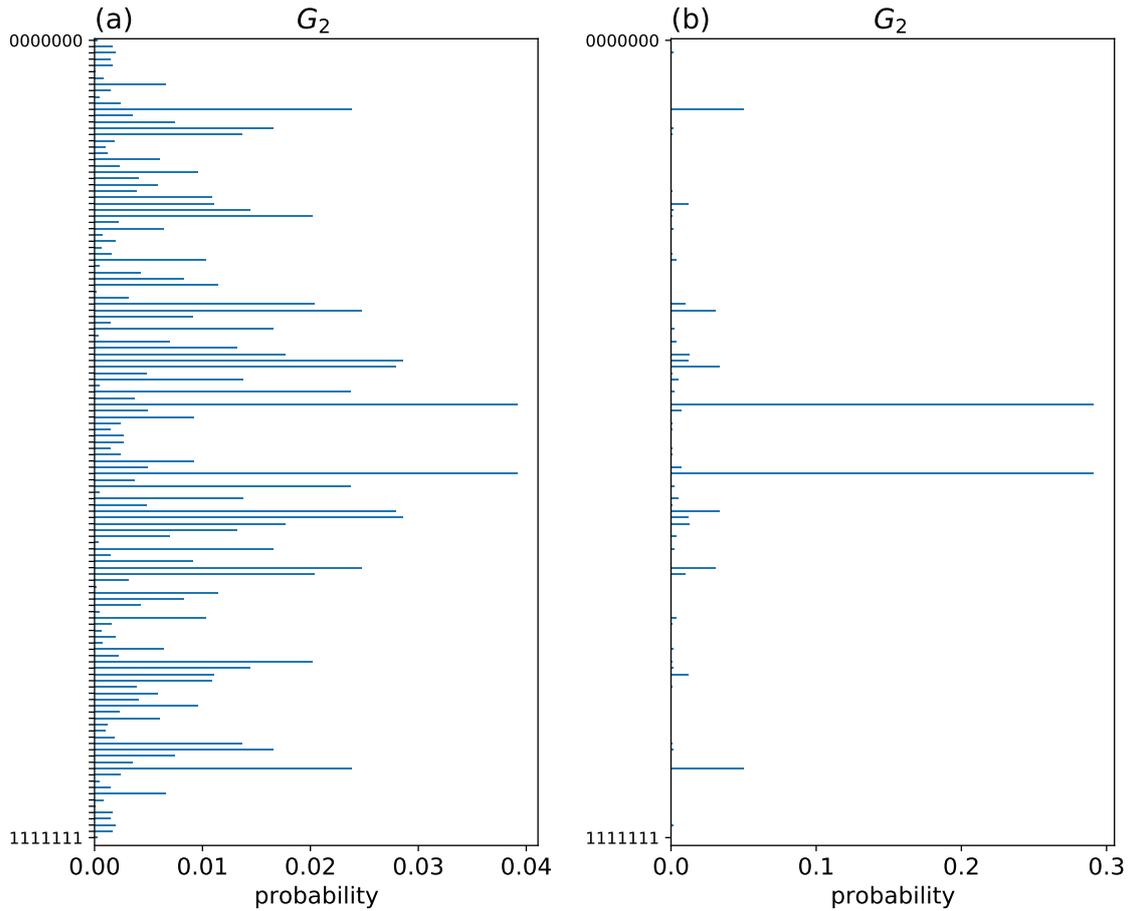


Figure 3.6: Probability distributions of all the possible configurations in the Max Cut problem for the 7 nodes graph G_2 encoded in the final states obtained after the optimizations shown in Figure 3.2 for (a): $p = 1$ QAOA level and (b): $p = 6$ QAOA level. The configurations are encoded as the computational basis states $|z\rangle$, with $z \in \{0,1\}^5$, and the final state is a superposition of these states. For G_2 the maximum cut is $z^* = 0111010$ (or $z^* = 1000101$ equivalently) and for both levels the most probable states are the ones corresponding to the maximum cut. In particular, for the level $p = 6$ the approximation is higher than the $p = 1$ case. Note that here we label only the first tick 0000000 and the last tick 1111111.

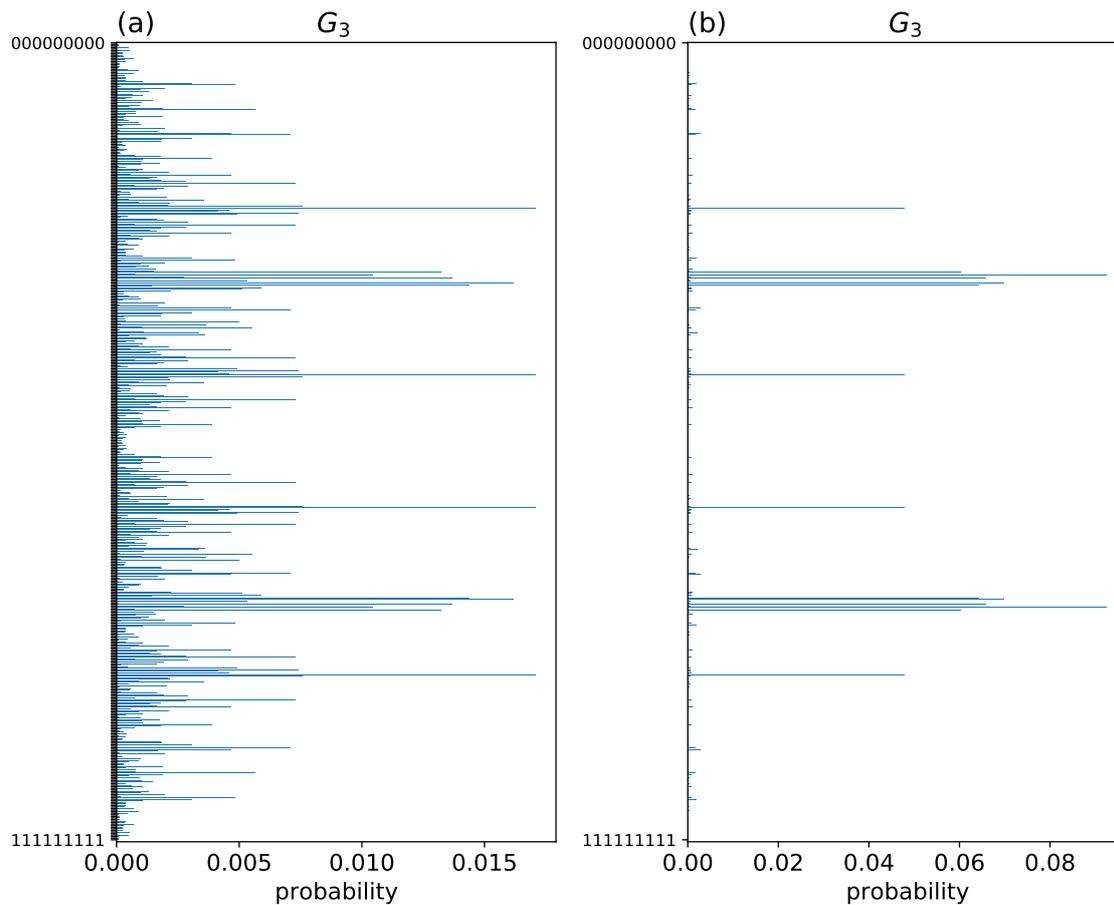


Figure 3.7: Probability distributions of all the possible configurations in the Max Cut problem for the 9 nodes graph G_3 encoded in the final states obtained after the optimizations showed in Figure 3.2 for (a): $p = 1$ QAOA level and (b): $p = 6$ QAOA level. The configurations are encoded as the computational basis states $|z\rangle$, with $z \in \{0,1\}^5$, and the final state is a superposition of these states. For G_3 the maximum cut are $z^* = 001101010, 010010011, 010010101, 010010111, 010011010, 010011011, 011010101$ (or the equivalently configurations with the z_u flipped) and for both levels the most probable states are the ones corresponding to the maximum cut. In particular, for the level $p = 6$ the approximation is higher than the $p = 1$ case. Note that here we label only the first tick 000000000 and the last tick 111111111.

3.2 RBIM: QAOA RESULTS

The random bond Ising model consists of the standard Ising Hamiltonian with nearest neighbour couplings on the two-dimensional (2D) square lattice

$$\mathcal{H}_I = -J \sum_{\langle i,j \rangle} K_{ij} S_i S_j, \quad (3.8)$$

where S_i, S_j are Ising variables, spins, that take on values ± 1 , J is the coupling constant and K_{ij} is a random coupling given by the probability distribution $P(K_{ij})$. In particular, by considering the RBIM on a square lattice with a binary probability distribution

$$P(K_{ij}) = \lambda \delta(K_{ij} - K) + (1 - \lambda) \delta(K_{ij} + K), \quad (3.9)$$

with $K > 0$. Thus, the coupling K_{ij} are ferromagnetic with probability λ and antiferromagnetic with probability $1 - \lambda$. In the following we fix the energy scale by setting $J = 1$. In this model, in the thermodynamic limit at $T = 0$, a phase transition from the ferromagnetic to the paramagnetic phase is expected for $\lambda_c \approx 0.88$ (for more details see Section 2.3.2). We remark that many combinatorial optimization problems can be thought as particular Ising Hamiltonians (3.8). Thus the RBIM may be treated with the QAOA, where the problem consists in finding the minima of the classical Hamiltonian (3.8), i.e. the energy of the system. The minima are the groundstate configurations of the spins lattices, expressed as $S^* = S_1^* S_2^* \dots S_n^*$, where n is the number of sites of the lattice, are obtained after the Adam parameters optimization described in Section 3.1. Thus, an analysis about the phase transition may be performed by finding the groundstate configurations S^* for different probabilities λ and then evaluate the classical magnetization $M = \sum_i^n S_i^*$ and the classical energy $E = - \sum_{\langle i,j \rangle} K_{ij} S_i^* S_j^*$.

We apply the QAOA on three different lattices, with 4, 6 and 9 sites, shown in Figure 3.8, for different probability links configurations $\lambda = [0.70, 0.98]$ with steps of 0.04.

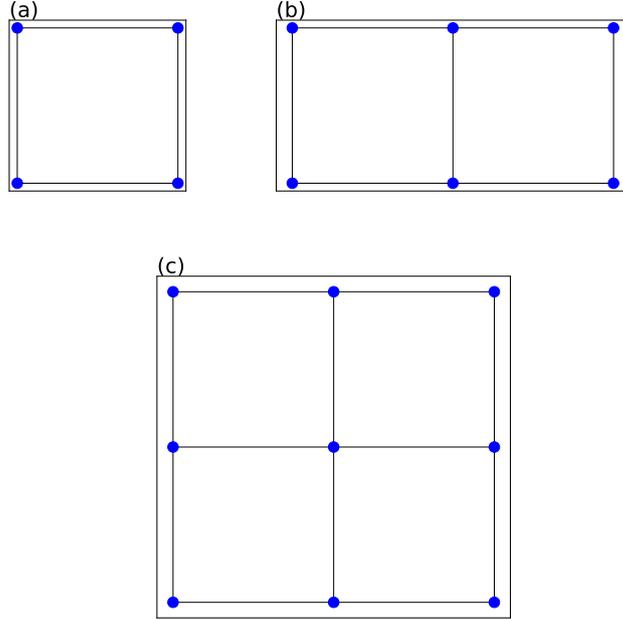


Figure 3.8: Lattices with (a): 4 sites, (b): 6 sites, (c): 9 sites

As described in 3.1, with the QAOA we obtain an approximate state $|\vec{\gamma}^*, \vec{\beta}^*\rangle$ which encodes the solution to the problem, where the classical Hamiltonian (3.8) translates into the quantum Hamiltonian operator

$$H_I = \sum_{\langle i,j \rangle} K_{ij} Z_i Z_j. \quad (3.10)$$

Thus, for the RBIM, we can find the groundstate spin configurations S^* of the system. We describe here the steps that we use to evaluate the classical magnetization and energy for different probabilities λ in the chosen lattice:

1. we choose a probability λ and obtain the couplings K_{jk} of the lattice through the binary probability distribution (see (2.40))
2. we perform the optimization method described in Section 3.1 for 100 steps, with the only difference that here the expectation value of the problem Hamiltonian $F_p = \langle \vec{\gamma}, \vec{\beta} | H_I | \vec{\gamma} \vec{\beta} \rangle$ is minimized
3. with the optimal parameters $(\vec{\gamma}^*, \vec{\beta}^*)$ given by the optimization procedure, we obtain the optimal final state which encodes the ground state spin configura-

3. QAOA RESULTS FOR MAX CUT AND RBIM

tions of the lattice

4. from the final state we obtain the most probable sites configuration $S^* \in \{-1, 1\}^n$, with n is number of sites in the lattice
5. evaluate the classical magnetization M and energy E for this configuration

The previous steps are repeated for $N = 100$ different disorder configurations, so, for each probability, we evaluate the mean values of the magnetization $\bar{M} = \frac{\sum_i^N M_i}{N} \pm \Delta M$ and energy $\bar{E} = \frac{\sum_i^N E_i}{N} \pm \Delta E$, where ΔM and ΔE are the respective standard errors of M and E .

Here we also use the approximation ratio defined in (2.28) for benchmarking the QAOA performances, where, for the RMIM, we evaluate it as

$$r = \frac{F_p(\vec{\gamma}^*, \vec{\beta}^*)}{\mathcal{H}_{min}}, \quad (3.11)$$

where \mathcal{H}_{min} is the minimum value of the classical Ising Hamiltonian in (3.8). Here, for each probability λ , in each of the $N = 100$ trials, the distribution of the couplings in the lattice is different each time. Thus, for each trial, we determine classically which is the \mathcal{H}_{min} to evaluate the approximation ratio. In this problem the approximation ratio as a tool in the analysis of the QAOA performances is quite useful because of the random nature of the cost function that is to be minimized. Here, with small number of sites, we may allow ourselves to evaluate classically each time \mathcal{H}_{min} for the known coupling distribution.

3.2.1 OPTIMIZATION METHOD AND RESULTS

Here we show, for $\lambda = 0.7$ and 0.9 , the Adam steps for three lattices, in figure 3.9, 3.10, 3.11.

We show in Figures 3.9, 3.10, 3.11 the approximation ratio as a function of the parameters optimization steps for the three lattices with 4, 6 and 9 nodes respectively. The optimizations are shown for one of the $N = 100$ trials for $\lambda = 0.7$ and 0.9 . For each optimization, we find the classical \mathcal{H}_{min} , by knowing the randomly generated binomial distribution of the couplings and we evaluate the approximation ratio with (3.11). In all the optimization procedures, we initialize the parameters randomly

near zero, and perform 100 Adam steps. For all the graphs, the approximation ratio increases and reach a maximum value in the Adam steps. The performance of the optimization increases with the QAOA level p for most of the cases shown. For some cases, the $p = 2$ level gives an approximation slightly better than the $p = 3$, but that probably is due to the fact that, for different couplings distribution \mathcal{H}_{min} in general is different each time and so the comparison of the approximation ratio for different levels could not be relevant in this case.

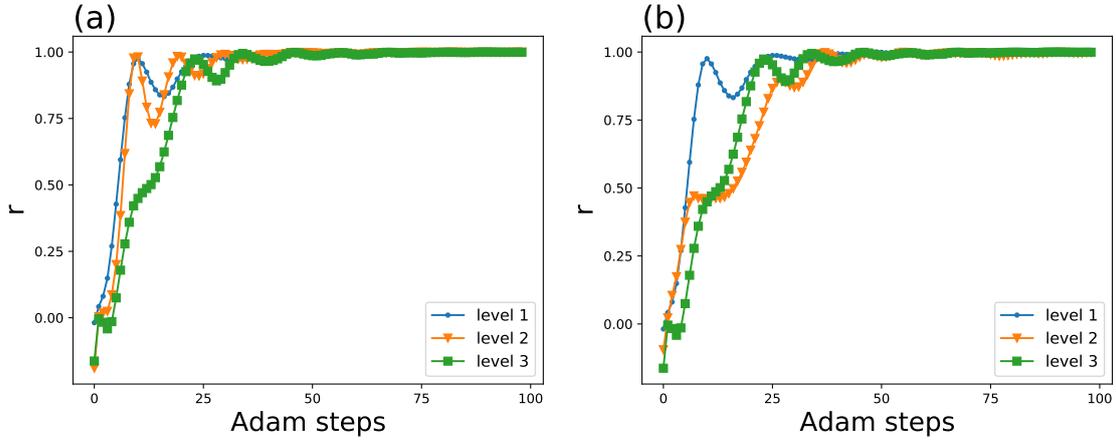


Figure 3.9: The approximation ratio $r = F_p/\mathcal{H}_{min}$ of QAOA on the RBIM optimization problem as a function of gradient descent steps in a lattice with 4 sites, for different QAOA levels, for (a): probability $\lambda = 0.7$, (b): probability $\lambda = 0.9$.

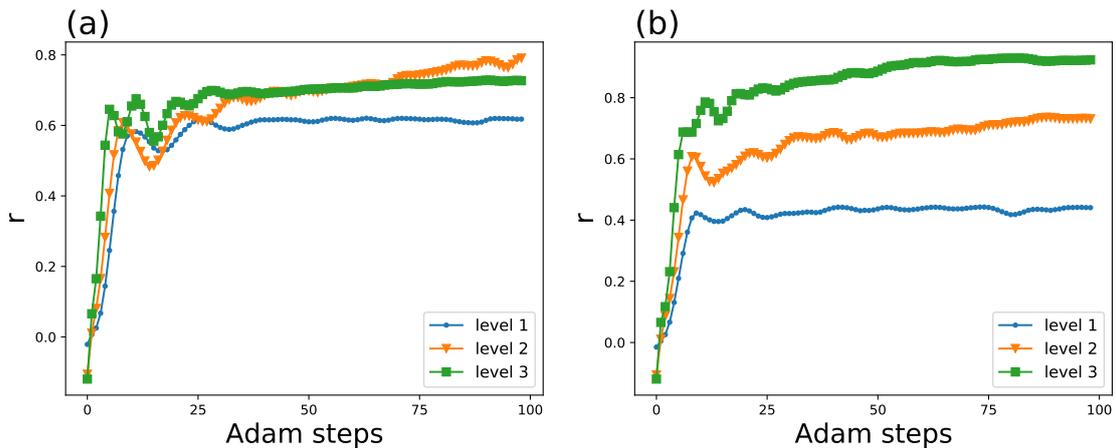


Figure 3.10: The approximation ratio $r = F_p/\mathcal{H}_{min}$ of QAOA on the RBIM optimization problem as a function of gradient descent steps in a lattice with 6 sites, for different QAOA levels, for (a): probability $\lambda = 0.7$, (b): probability $\lambda = 0.9$.

3. QAOA RESULTS FOR MAX CUT AND RBIM

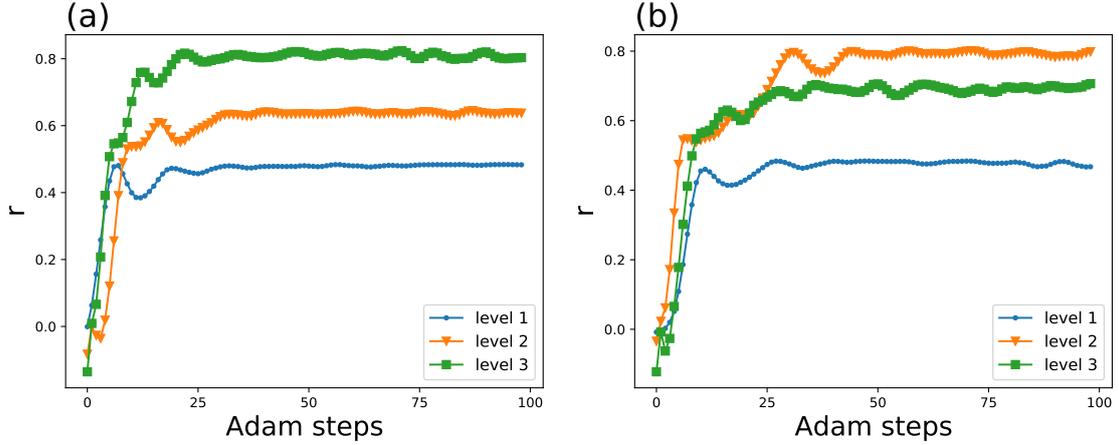


Figure 3.11: The approximation ratio $r = F_p/\mathcal{H}_{min}$ of QAOA on the RBIM optimization problem as a function of gradient descent steps in a lattice with 9 sites, for different QAOA levels, for (a): probability $\lambda = 0.7$, (b): probability $\lambda = 0.9$.

We show now the obtained classical values of the magnetization and energy for different probabilities λ . In Figure 3.12, we show the plot of M per site and E per site for different probabilities λ in the lattices with 4, 6 and 9 sites, where, for each lattice, the QAOA levels $p = 1, 2$ and 3 are compared. In Figure 3.13, we show the plot of M and E per site for the QAOA levels $p = 1, 2$ and 3, where, for each level, the lattices with 4, 6 and 9 sites the are compared. Even though strong observations about the phase transition cannot be stated because we analyze small lattices and so it is impossible for us to make considerations concerning the thermodynamic limit condition, some features may still be highlighted. All the magnetizations, for all lattice and levels, increase with λ and the energy decreases, as expected. This shows that a phase transition occurs for a certain probability, because, by decreasing the probability λ , the magnetization per site goes from the maximum possible value of 1, representing an ordered region, to lower values, indicating more disordered regions. The same is deduced from the energy, which, by decreasing the probability, goes from the minimum energy per site, -1 , $-\frac{7}{6}$ and $-\frac{12}{9}$ for lattices with 4, 6 and 9 sites respectively, representing a completely polarized state, a ferromagnetic state, to higher values, were the ferromagnetic state is lost. In particular, in Figure 3.13 we may observe that, before $\lambda = 0.9$ the magnetization is higher for the smallest lattice size and decreases as the lattice size increases, whereas after that point, we observe the opposite behaviour. That is the expected behavior of an order parameter in proximity of a critical point and that is more evident for the $p = 3$ case, the one

with higher QAOA approximation quality.

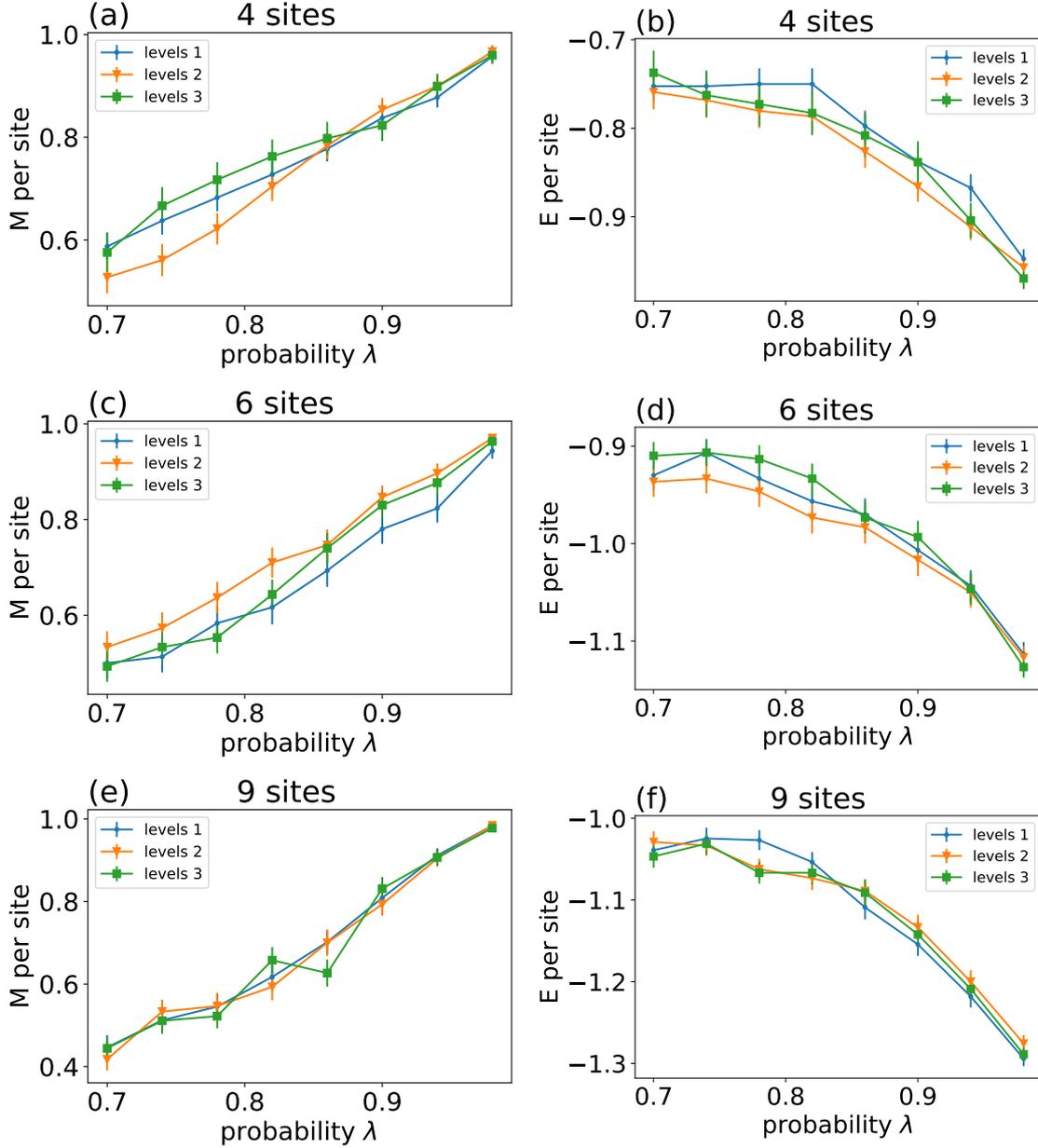


Figure 3.12: Magnetization M and energy E per site for the lattices with 4 ((a) and (b) respectively), 6 ((c) and (d) respectively) and 9 ((e) and (f) respectively) sites, where, for each lattice, the QAOA levels $p = 1, 2$ and 3 are compared. All the magnetizations, for all levels, increase with λ and the energy decreases, as expected. This is an evidence for a phase transition for a certain probability, because, by decreasing the probability λ , the magnetization per site goes from the maximum possible value of 1, representing an ordered region, to lower values, indicating more disordered regions. The same is deduced from the energy, which, by decreasing the probability, goes from the minimum energy per site, -1 , $-\frac{7}{6}$ and $-\frac{12}{9}$ for lattices with 4, 6 and 9 sites respectively, representing a completely polarized state, a ferromagnetic state, to higher values, were the ferromagnetic state is lost.

3. QAOA RESULTS FOR MAX CUT AND RBIM

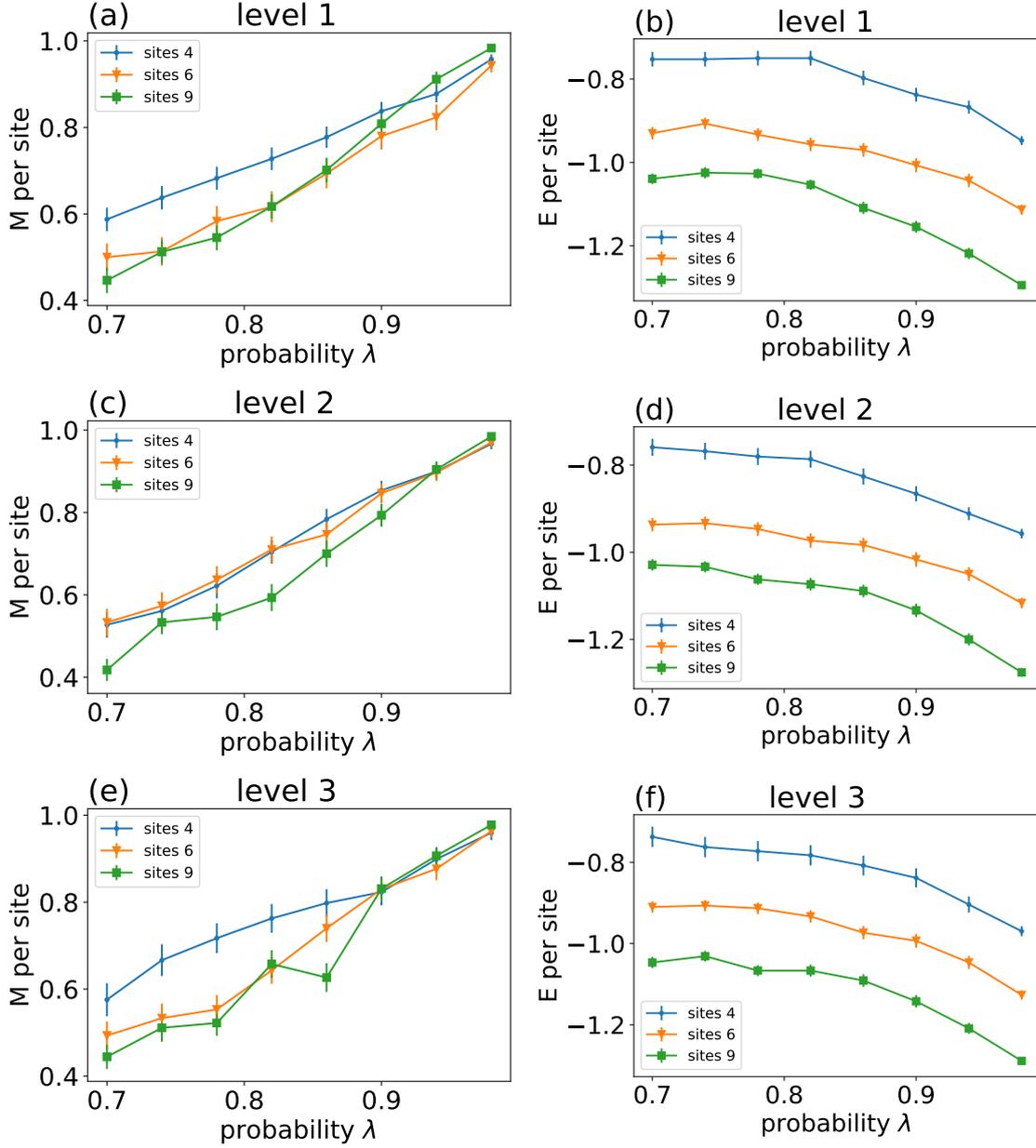


Figure 3.13: M and E per site for the level $p = 1$ ((a) and (b) respectively), $p = 2$ ((c) and (d) respectively) and $p = 3$ ((e) and (f) respectively) sites, where, for each level, the lattices with 4, 6 and 9 sites are compared. All the magnetizations, for all sites, increase with λ and the energy decreases, as expected. This is an evidence for a phase transition for a certain probability, because, by decreasing the probability λ , the magnetization per site goes from the maximum possible value of 1, representing an ordered region, to lower values, indicating more disordered regions. The same is deduced from the energy, which, by decreasing the probability, goes from the minimum energy per site, -1 , $-\frac{7}{6}$ and $-\frac{12}{9}$ for lattices with 4, 6 and 9 sites respectively, representing a completely polarized state, a ferromagnetic state, to higher values, where the ferromagnetic state is lost. In particular, we may observe that, before $\lambda = 0.9$ the magnetization is higher for the smallest lattice size and decreases as the lattice size increases, whereas after that point, we observe the opposite behaviour. That is the expected behavior of an order parameter in proximity of a critical point and that is more evident for the $p = 3$ case, the one with higher QAOA approximation quality.

3.3 OUTLOOK AND CONCLUSIONS

In this work we have implemented the QAOA through classical simulations. We have applied it for finding the Max Cut of a graph and finding the ground state of the random bond Ising model (RBIM). In the QAOA steps we performed the classical optimization procedure using a gradient descent method implemented via the Adam algorithm, with hyper parameters $\eta = 0.05$, $\hat{\beta}_1 = 0.9$, $\hat{\beta}_2 = 0.999$ and $\epsilon = 10^{-8}$. In the gradient descent method the gradients were evaluated with the central finite differences method with increment $h = 0.1$. The quantum expectation value of the cost Hamiltonian F_p (see (2.25)), which was to be optimized, was estimated at each step of the optimization by sampling 100 outcomes of simulated quantum measurements. For Max Cut in the optimization procedure we expected, starting from parameters picked randomly near zero, the optimization to move in the parameter landscape towards parameters for which the expectation value F_p becomes closer to the maximum of the classical cost function $C(z)$ (see (3.1)) which encodes the problem. In particular, we expected the optimization efficiency to increase with the QAOA level p . For Max Cut we have verified both these expected behaviours for different graphs, thus verifying that our QAOA implementation behaves correctly.

Further, in the $p = 1$ case for Max Cut, we have studied the two-dimensional parameters landscape using the analytical expression of the expectation value of the cost function F_1 (see (2.38)), and we discuss that the parameter landscape presents plateaux for not-regular graphs that we want to avoid as starting point in the parameters optimization. These plateaux never include the point $(0, 0)$, that is why, we started all our optimization procedures near $(0, 0)$. A further investigation about this observation is beyond the scope of this work, but it leaves open a more in-deep study for a general demonstration or to generalize it to the $p > 1$ cases.

For the RBIM, for each randomly generated coupling configuration, we expected the optimization to move in the parameter landscape towards parameters for which the expectation value F_p becomes closer to the minimum of the classical Ising Hamiltonian \mathcal{H}_i (see (3.8)) for these couplings. In the RBIM we evaluated, for different coupling configuration, the magnetizations M and energies E of the groundstates obtained after the optimization procedures. Since a probability $\lambda = 1$ corresponds to all ferromagnetic couplings, decreasing λ means moving from an order region to

3. QAOA RESULTS FOR MAX CUT AND RBIM

the disordered one. We have verified that the QAOA for each probability returned the actual groundstates for the specific coupling configuration, because the energies values per site decreases as λ increases whereas the magnetization increases along with λ verifying the expected behaviour.

In the RBIM, in the thermodynamic limit at $T = 0$, a phase transition occurs for a certain critical probability λ_c and we observed that, even if we considered only small finite size lattices, the ordered parameter magnetization behaves as expected near the critical point. Infact the magnetization is higher in the smallest lattice for probabilities smaller than the critical one and it's higher in the larger lattice for probabilities higher than the critical one. These two results strengthened further the fact that the QAOA is well suited for the study of groundstates of physical systems described by Ising Hamiltonians. In particular, even for small lattices, with QAOA we are be able to perform analysis of the phase transition of a physical system.

This work and its results may be pursued further on several aspects. Here we have applied the QAOA to only two different problems, one possible aspect to be checked can be the application of this QAOA implementation to other combinatorial optimization problems. For how it was designed, the code of this algorithm was thought to be implemented on an actual real quantum device, with just a needed transposition between the operators and the corresponding quantum gates. Thus, its performances and limitations when implemented on a real device may be subject to further study. Another aspect that can be investigate further is the implementation of different optimization methods. Where balance between performances and resource requirements may be compared. This analysis would be very relevant contextualized in the possible eventuality of implementation in a real quantum computer, where the actual device put strong constraints on the resource requirements. As stated above, a further study may be done about the parameters landscape, by investigating more thoroughly the analytical expression for the $p = 1$ case for Max Cut and in general try to understand better the $p > 1$ cases for different problems.

Appendices

A

PROOF OF ANALYTICAL EXPRESSION OF F_1 FOR MAX CUT

We have to prove that, for edge $\langle u, v \rangle$,

$$\langle C_{uv} \rangle = \frac{1}{2} + \frac{1}{4}(\sin 4\beta \sin \gamma)(\cos^{d_u} \gamma + \cos^{d_v} \gamma) - \frac{1}{4}(\sin^2 2\beta \cos^{d_u+d_v-2\lambda_{uv}} \gamma)(1 - \cos^{\lambda_{uv}} 2\gamma),$$

with d_u : degree of vertex $u - 1$, d_v : degree of vertex $v - 1$, λ_{uv} : number common neighbours of vertices u and v .

Proof :

$$\begin{aligned} \langle C_{uv} \rangle &= \langle s | e^{i\gamma H^C} e^{i\beta H^B} \left[\frac{1}{2}(I - Z_u Z_v) \right] e^{-i\beta H^B} e^{-i\gamma H^C} | s \rangle \\ &= \frac{1}{2} - \frac{1}{2} \langle s | e^{i\gamma H^C} e^{i\beta H^B} Z_u Z_v e^{-i\beta H^B} e^{-i\gamma H^C} | s \rangle \end{aligned}$$

$$\begin{aligned} \langle Z_u Z_v \rangle &= \langle s | e^{i\gamma H^C} \underbrace{e^{i\beta H^B} Z_u Z_v e^{-i\beta H^B}}_{(*)} e^{-i\gamma H^C} | s \rangle \\ &= \underbrace{\langle s | e^{i\gamma H^C} (Z_u Z_v \cos^2 2\beta) e^{-i\gamma H^C} | s \rangle}_{(I)} \\ &\quad + \underbrace{\langle s | e^{i\gamma H^C} (Z_u Y_v + Y_u Z_v) \cos 2\beta \sin 2\beta e^{-i\gamma H^C} | s \rangle}_{(II)} \end{aligned}$$

$$+ \underbrace{\langle s | e^{i\gamma H^C} (Y_u Y_v \sin^2 2\beta) e^{-i\gamma H^C} | s \rangle}_{(III)}$$

$$\begin{aligned} (\star) &= \prod_j e^{i\beta X_j} Z_u Z_v \prod_j e^{-i\beta X_j} = e^{i\beta X_u} Z_u e^{-i\beta X_u} e^{i\beta X_v} Z_v e^{-i\beta X_v} \\ &\stackrel{(1')}{=} [Z_u \cos 2\beta + Y_u \sin 2\beta][Z_v \cos 2\beta + Y_v \sin 2\beta] \\ &= Z_u Z_v \cos^2 2\beta + (Z_u Y_v + Y_u Z_v) \cos 2\beta \sin 2\beta + Y_u Y_v \sin^2 2\beta \end{aligned}$$

(I):

$$\langle s | e^{i\gamma H^C} Z_u Z_v e^{-i\gamma H^C} | s \rangle \stackrel{(2')}{=} \langle s | Z_u Z_v | s \rangle \stackrel{(3')}{=} 0$$

(II):

$$\begin{aligned} \langle s | e^{i\gamma H^C} Y_u Z_v e^{-i\gamma H^C} | s \rangle &= \langle s | \prod_{jk} e^{i\frac{\gamma}{2}(I-Z_j Z_k)} Y_u Z_v \prod_{jk} e^{-i\frac{\gamma}{2}(I-Z_j Z_k)} | s \rangle \\ &= \langle s | \prod_{jk} e^{-i\frac{\gamma}{2} Z_j Z_k} Y_u Z_v \prod_{jk} e^{i\frac{\gamma}{2} Z_j Z_k} | s \rangle \\ &\stackrel{(4')}{=} \langle s | \prod_r^{d+1} [\cos \frac{\gamma}{2} - i Z_u Z_{w_r} \sin \frac{\gamma}{2}] Y_u Z_v \prod_r^{d+1} [\cos \frac{\gamma}{2} + i Z_u Z_{w_r} \sin \frac{\gamma}{2}] \\ &\stackrel{(1')}{=} \langle s | [\cos \gamma - i Z_u Z_v \sin \gamma] \prod_r^d [\cos \gamma - i Z_u Z_{w_r} \sin \gamma] Y_u Z_v | s \rangle \\ &\stackrel{(3')}{=} \langle s | (-i \sin \gamma \cos^d \gamma \underbrace{Z_u Z_v Y_u Z_v}_{(-iX_u)}) | s \rangle + \underbrace{0 + 0 + \dots + 0}_{2(d+1)-1 \text{ terms}} \\ &= \langle s | -i \sin \gamma \cos^d \gamma X_u | s \rangle = -\sin \gamma \cos^d \gamma \end{aligned}$$

$$\langle s | e^{i\gamma H^C} Y_u Z_v e^{-i\gamma H^C} | s \rangle = \dots = -\sin \gamma \cos^e \gamma,$$

(III):

$$\begin{aligned}
 \langle s | e^{i\gamma H^C} Y_u Y_v e^{-i\gamma H^C} | s \rangle & \stackrel{\uparrow}{=} \langle s | e^{i\frac{\gamma}{2}} e^{-i\frac{\gamma}{2} Z_u Z_v} e^{i\gamma H_u^C} e^{i\gamma H_v^C} e^{i\gamma H^C} Y_u Y_v e^{-i\frac{\gamma}{2}} e^{i\frac{\gamma}{2} Z_u Z_v} e^{-i\gamma H_u^C} e^{-i\gamma H_v^C} e^{-i\gamma H^C} | s \rangle \\
 & \stackrel{(5')}{=} \langle s | e^{i2\gamma H_u^C} e^{i2\gamma H_v^C} Y_u Y_v | s \rangle \\
 & \stackrel{\uparrow}{=} \langle s | \prod_r^d [\cos \gamma + i Z_u Z_{w_r} \sin \gamma] \prod_q^e [\cos \gamma + i Z_u Z_{w_q} \sin \gamma] Y_u Y_v | s \rangle \\
 & \stackrel{\uparrow}{=} \frac{1}{2} \cos^{d+e-2f} \gamma (1 - \cos^f 2\gamma) \\
 & \stackrel{(6')}{=}
 \end{aligned}$$

with $d = d_u$, $e = d_v$ and $f = \lambda_{uv}$.

(1'): with $L \subseteq K$, $K \subseteq [n]$, $[n] = \{1, \dots, n\}$, $\sigma_L^i = \bigotimes_{u \in L} \sigma_u^i$, σ_u^i Pauli matrix and $i, j, k \in \{x, y, z\}$

$$\begin{aligned}
 e^{i\alpha \sigma_L^i} \sigma_L^j e^{-i\alpha \sigma_L^i} & = [\cos \alpha + i \sigma_L^i \sin \alpha] \sigma_L^j [\cos \alpha - i \sigma_L^i \sin \alpha] \\
 & = \sigma_L^j \cos^2 \alpha + \sigma_L^i \sigma_L^j \sigma_L^i \sin^2 \alpha + i [\sigma_L^i, \sigma_L^j] \sin \alpha \cos \alpha \\
 & \stackrel{\uparrow}{=} \sigma_L^j \cos^2 \alpha + (i)^{2|L|} \sigma_L^j \sin^2 \alpha + (i)^{|L|+1} (-i \varepsilon_{jik} \sigma_L^j \sigma_L^i) [(\varepsilon_{ijk})^{|L|} - (\varepsilon_{jik})^{|L|}] \sin \alpha \cos \alpha \\
 & \stackrel{(1.1')}{=} \sigma_L^j \{ \cos^2 \alpha + (-1)^{|L|} \sin^2 \alpha + (i)^{|L|+2} [(\varepsilon_{ijk})^{|L|+1} + (\varepsilon_{jik})^{|L|+1}] \sigma_L^i \sin \alpha \cos \alpha \} \\
 & = \begin{cases} \sigma_L^j \{ \cos 2\alpha - i \sigma_L^i \sin 2\alpha \} = \sigma_L^j e^{-i2\alpha \sigma_L^i} = \varepsilon_{jik} \sigma_L^k \sin 2\alpha, & \text{if } L = \{u\} \\ \sigma_L^j \{ \cos^2 \alpha + \sin^2 \alpha \} = \sigma_L^j, & \text{if } L = \{u, v\} \end{cases}
 \end{aligned}$$

(1.1'):

$$\begin{aligned}
 \sigma_L^i \sigma_L^j \sigma_L^i & = \sigma_L^i \bigotimes_{u \in L} \sigma_L^j \otimes \sigma_L^i = \sigma_L^i (i)^{|L|} (\varepsilon_{jik})^{|L|} \sigma_L^k \\
 & = (i)^{2|L|} (\varepsilon_{jik})^{|L|} (\varepsilon_{ikj})^{|L|} \sigma_L^j = (i)^{2|L|} \sigma_L^j
 \end{aligned}$$

$$\begin{aligned}
 i(\sigma_L^i \sigma_L^j - \sigma_L^j \sigma_L^i) & = i[(i)^{|L|} (\varepsilon_{ijk})^{|L|} \sigma_L^k - (i)^{|L|} (\varepsilon_{jik})^{|L|} \sigma_L^k] \\
 & = [(\varepsilon_{ijk})^{|L|} - (\varepsilon_{jik})^{|L|}] (i)^{|L|+1} \sigma_L^k
 \end{aligned}$$

(2'):

$$H^C = \sum_{\langle j,k \rangle} \frac{1}{2}(I - Z_j Z_k) \text{ and } \sigma^i \sigma^j = \delta_{ij} I \Rightarrow H^C \text{ and } Z_u Z_v \text{ commute}$$

(3'):

$$\langle s | \sigma^i | s \rangle = \begin{cases} 1, & \text{if } i = x \\ 0, & \text{otherwise} \end{cases}$$

(4'):

$$Y_u Z_v \prod_{\langle j,k \rangle} e^{i\frac{\gamma}{2} Z_j Z_k} = \prod_{\substack{\langle j,k \rangle \\ k, j \neq u}} e^{-i\frac{\gamma}{2} Z_j Z_k} Y_u Z_v \prod_i^{d+1} e^{i\frac{\gamma}{2} Z_u Z_{w_i}},$$

where, in $d = d_u$ are taken all the edges with vertex u , except $\langle u, v \rangle$

(5'):

$$H^C = \frac{1}{2}(I - Z_u Z_v) + H_u^C + H_v^C + \bar{H}^C,$$

where H_u^C : terms with only u , H_v^C : terms with only v , \bar{H}^C : terms with neither u nor v

(6'):

$$\langle s | \prod_r^d [\cos \gamma + i Z_u Z_{w_r} \sin \gamma] \prod_q^e [\cos \gamma + i Z_u Z_{w_q} \sin \gamma] Y_u Y_v | s \rangle = (*)$$

in (*) the only non-null terms are the ones for which $Z_{w_r} Z_{w_q} = I$, so, by listing the terms:

$$(i \sin \gamma)^2 \underbrace{Z_u Z_{w_r} Z_v Z_{w_q}}_{-X_u X_v} Y_u Y_v \cos^{e+d-2} \gamma \lambda_{uv} +$$

$$\begin{aligned}
 & (i \sin \gamma)^4 \underbrace{Z_u Z_{w_r} Z_v Z_{w_q} Z_u Z_{w'_r} Z_v Z_{w'_q} Y_u Y_v}_{Y_u Y_v} \cos^{e+d-4} \gamma \binom{\lambda_{uv}}{2} + \\
 & (i \sin \gamma)^6 (-X_u X_v) \cos^{e+d-6} \gamma \binom{\lambda_{uv}}{3} + \\
 & \vdots
 \end{aligned}$$

the idea of are written the above terms is that in the product there are λ_{uv} " $i \sin^2 \gamma$ " terms that can be "arranged" in groups of 1, 3, 5, ... terms. So, with $\lambda_{uv} = f$

$$\begin{aligned}
 (*) &= \binom{f}{1} \cos^{d+e-2} \gamma \sin^2 \gamma + \binom{f}{3} \cos^{d+e-6} \gamma \sin^6 \gamma + \binom{f}{5} \cos^{d+e-10} \gamma \sin^{10} \gamma + \dots \\
 &= \sum_{i=1,3,5,\dots}^f \binom{f}{i} \cos^{d+e-2i} \gamma \sin^{2i} \gamma = \cos^{d+e-2f} \gamma \sum_{i=1,3,5,\dots}^f \binom{f}{i} \cos^{2(f-i)} \gamma \sin^{2i} \gamma \\
 &\stackrel{\uparrow}{=} \underset{(6.1')}{\cos^{d+e-2f} \gamma} \frac{1}{2} [(\cos^2 \gamma + \sin^2 \gamma)^f - (\cos^2 \gamma - \sin^2 \gamma)^f] = \frac{1}{2} \cos^{d+e-2f} (1 - \cos^f 2\gamma)
 \end{aligned}$$

(6.1): from binomial theorem, with $a, b \in \mathbb{R}$

$$\begin{aligned}
 (a+b)^f &= \sum_{i=0}^f \binom{f}{i} a^{f-i} b^i = \sum_{i=0,2,4,\dots}^f \binom{f}{i} a^{f-i} b^i + \sum_{i=1,3,5,\dots}^f \binom{f}{i} a^{f-i} b^i \\
 (a-b)^f &= \sum_{i=0}^f \binom{f}{i} (-1)^i a^{f-i} b^i = \sum_{i=0,2,4,\dots}^f \binom{f}{i} a^{f-i} b^i - \sum_{i=1,3,5,\dots}^f \binom{f}{i} a^{f-i} b^i \\
 \Rightarrow \sum_{i=1,3,5,\dots}^f \binom{f}{i} a^{f-i} b^i &= \frac{1}{2} [(a+b)^f - (a-b)^f]
 \end{aligned}$$

Thus, we have that:

$$\begin{aligned}
 \langle C_{uv} \rangle &= \frac{1}{2} [1 - \cos 2\beta \sin 2\beta (-\sin \gamma \cos^d \gamma - \sin \gamma \cos^e \gamma) + \sin^2 2\beta (\frac{1}{2} \cos^{d+e-2f} \gamma (1 - \cos^f 2\gamma))] \\
 &= \frac{1}{2} + \frac{1}{4} (\sin 4\beta \sin \gamma) (\cos^{d_u} \gamma + \cos^{d_v} \gamma) - \frac{1}{4} (\sin^2 2\beta \cos^{d_u+d_v-2\lambda_{uv}} \gamma) (1 - \cos^{\lambda_{uv}} 2\gamma)
 \end{aligned}$$

BIBLIOGRAPHY

- [1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms (2020), [arXiv:2012.09265 \[quant-ph\]](#) .
- [2] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum (nisq) algorithms (2021), [arXiv:2101.08448 \[quant-ph\]](#) .
- [3] A. Paler and S. J. Devitt, An introduction to fault-tolerant quantum computing (2015), [arXiv:1508.03695 \[quant-ph\]](#) .
- [4] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New Journal of Physics* **18**, 023023 (2016).
- [5] J. Preskill, *Quantum* **2**, 79 (2018).
- [6] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), [arXiv:1411.4028 \[quant-ph\]](#) .
- [7] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, and et al., *Nature Physics* [10.1038/s41567-020-01105-y](#) (2021).
- [8] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, *Physical Review X* **10**, [10.1103/physrevx.10.021067](#) (2020).
- [9] G. E. Crooks, Performance of the quantum approximate optimization algorithm on the maximum cut problem (2018), [arXiv:1811.08419 \[quant-ph\]](#) .

BIBLIOGRAPHY

- [10] J. Johansson, P. Nation, and F. Nori, *Computer Physics Communications* **183**, 1760 (2012).
- [11] G. G. Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms (2017), [arXiv:1701.01450 \[quant-ph\]](https://arxiv.org/abs/1701.01450) .
- [12] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), [arXiv:1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980) .
- [13] I. A. Gruzberg, N. Read, and A. W. W. Ludwig, *Physical Review B* **63**, [10.1103/physrevb.63.104422](https://arxiv.org/abs/10.1103/physrevb.63.104422) (2001).
- [14] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
- [15] M. M. Wilde, *Quantum Information Theory*, 2nd ed. (Cambridge University Press, 2017).
- [16] C. Ryan-Anderson, Quantum algorithms, architecture, and error correction (2018), [arXiv:1812.04735 \[quant-ph\]](https://arxiv.org/abs/1812.04735) .
- [17] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, *Applied Physics Reviews* **6**, 021318 (2019).
- [18] B. R. e. a. Arute F., Arya K., *Nature* 574, 505–510 <https://doi.org/10.1038/s41586-019-1666-5> (2019).
- [19] A. Montanaro, *Quantum algorithms: an overview* (2016).
- [20] Brilliant.org., *Complexity classes* (2021).
- [21] M. Mosca, Quantum algorithms (2008), [arXiv:0808.0369 \[quant-ph\]](https://arxiv.org/abs/0808.0369) .
- [22] P. W. Shor, in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994) pp. 124–134.
- [23] P. Lambropoulos and D. Petrosyan, *Fundamentals of Quantum Optics and Quantum Information* (Springer-Verlag, Berlin, Heidelberg, 2006).
- [24] Y. Zhang, *Lecture notes on quantum information and computation* (2015).

- [25] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, Classical optimizers for noisy intermediate-scale quantum devices (2020), [arXiv:2004.03004 \[quant-ph\]](#) .
- [26] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Faehrmann, B. Meynard-Piganeau, and J. Eisert, *Quantum* **4**, 314 (2020).
- [27] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, *Science Advances* **5**, [10.1126/sciadv.aaw9918](#) (2019), <https://advances.sciencemag.org/content/5/10/eaaw9918.full.pdf> .
- [28] S. Ruder, An overview of gradient descent optimization algorithms (2017), [arXiv:1609.04747 \[cs.LG\]](#) .
- [29] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, *Physical Review A* **97**, [10.1103/physreva.97.022304](#) (2018).