eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Degree in Computer Engineering

Computer science

End of degree work

Solving combinatorial optimization problems using quantum computing: A case study for the QAP

Author

Unai Lizarralde Imaz

2020

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Degree in Computer Engineering

Computer science

End of degree work

Solving combinatorial optimization problems using quantum computing: A case study for the QAP

Author

Unai Lizarralde Imaz

Director(s)

Jose A. Pascual Saiz

Alexander Mendiburu Alberro

Summary

Quantum computing is one of the most researched areas in computer science and with one of the greatest future prospects thanks to the new discoveries and methodologies that can provide approaches of a different kind to tackle problems and find solutions. One of such methodologies is none other than quantum annealing, a metaheuristic focused on the qualities of quantum mechanics.

Combinatorial Optimization (CO) problems have always been a hassle to find solutions in large scale problems owing to their complexity and resource consumption. However, these can be transformed into an equivalent Quadratic Unconstrained Binary Optimization (QUBO) model that is constraint-free and its variables are (binary) decision variables. What is best, QUBO models can be efficiently solve with quantum annealing in a quantum computer.

The objective of the project will be the study of all the particular functionalities and properties of each aspect of all the transformation chain for the specific case study with the QAP problem. The implementation of this problem, its transformation into a QUBO model and the solution obtained through quantum annealing are the key points that lead to harness the potential of quantum computing and set its current limits.

Contents

List of Figures

List of Tables

1. CHAPTER

Introduction

In recent years, the rise of artificial intelligence, machine learning and a large number of enticing and highly cost effective approaches to solve problems, develop new methodologies or find groundbreaking discoveries is odd to none. However, despite still being outshone by the tendencies hogging the spotlight, quantum computation is slowly but steadily being researched and proving new alternatives to the already fixed classical methodologies.

The truly empowering quality of quantum computation lies in the underlying potential it possess. All in all, it is not a stretch to consider it a completely different and intriguing paradigm that will revolutionize the field of computation in years to come. Among the expectations for this design philosophy, new breakthroughs in science, enhanced machine learning approaches, development of medicines, never before seen technologies from the build material to the capabilities of the system and revitalized algorithms that fully use the possibilities offered by quantum computation are included.

In the quantum world, a single particle can represent two states simultaneously; two particles can even be so tightly ”tied” that they can’t be split in two, and can communicate immediately despite being light years apart; and the very act of observing a quantum system can irreversibly alter the system. All these are few of the many phenomenons that occur in quantum computation that deviate from the more understandable and intuitive perspective towards the behaviour of a classical system. After all, the laws imposed until now are shattered once one dives into the quantum world, only for new ones to emerge and exploit possibilities unattainable before.

The main concept behind quantum computing is to build a computer that represents bits not with transistors but with subatomic particles, such as electrons or photons. This shift in representation implies a huge leap in the grand scheme of things, since it means the abandonment of the laws of Newton's classical mechanics for the laws of quantum mechanics, completely defying the laws of physics established so far.

The reasons to build such computer are many: from an engineers perspective, the size of the microchips have become so small that quantum effects have taken its toll and affect their functionality; from a physicist standpoint, it is a natural step to learn from and simulate quantum systems; and for computer scientists, quantum computing is a new resource that can help and enable solutions to problems that were thought at first to be intractable. The last one is quite remarkable in the fact that impossible problems with classical known means become actually tractable with quantum mechanics.

Such is the case with Combinatorial Optimization (CO) problems, which consists on finding the best possible solution (optimization) from a pool of feasible solutions of a combinatorial problem. Unfortunately, these problems tend to escalate rather quickly in resources when it comes to the amount of time it takes to search for that solution or the space the model requires to hold. Consequently, despite the total amount of solutions being a finite discrete number, it could still be exceedingly large for an exhaustive search.

Many of the CO problems share the condition of not having solutions in polynomial time, hence the cost of producing desirable results, or, at the very best, there are algorithms that solve specific instances of that problem like when the amount of variables is limited or when the search space is fixed to specific solutions of only n elements. That is why most of the CO problems' complexity swings around the NP, $NP-Complete$ and $NP-hard$ complexity classes.vv

As an alternative, metaheuristics have become one remarkable method, given how they search for the best heuristic that produces the best guidelines towards finding good solutions. Even if these solutions are not the best choice of the objects of the CO problem necessarily, they are still quite good approximations that could still be practical in the context of the problem. In particular, quantum annealing is one metaheurisitc method based on quantum mechanics that serves this purpose.

QUBO models are perfect fit for the quantum annealing method, since they offer a constraint free problem of decision variables. The selling point, however, lies in the method to transform constrained CO problems into QUBO models under specific circumstances. One of such CO problems is the Quadratic Assignment Problem (QAP), the case study

of this project and the CO problem that will be transformed into a QUBO model. Once transformed, through the quantum annealing method the QUBO is solved which effectively solves the original problem at hand.

The framework for the quantum annealing is the quantum computers provided by D-wave systems which are available to use through the quantum leap initiative. Therefore, the QAP problem instance is first transformed into a QUBO equivalent and then it is sent to a D-wave quantum computer to solve with quantum annealing.

The nature is much more than a Turing machine and quantum mechanics are here to prove it beyond the glimpse of doubt and despite the difficulty of its realization.

2. CHAPTER

The aims of the project

The aims of the project are centered towards learning the fundamentals of quantum computing to find and understand new emerging possibilities to solve intractable problems with high complexity levels. Quantum computing has still a long way to go, however the recent discoveries have opened the path to build a bridge that joins both traditional methodologies and the ability to represent these problems following the principles of quantum mechanics.

The first objective is a deep study of the field of quantum computing and the laws of physics that define it, namely, quantum mechanics. From the basic grasp of linear algebra for quantum mechanics to understanding counterintuitive behaviours and phenomenons that make possible never before seen events and milestones in the story of computer science. At the core of quantum mechanics, the four pillars that build the ground are the four postulates of quantum mechanics, which are essential to comprehend the inner workings of quantum computing.

Secondly, a reformulation of CO problems as QUBO models is pursued. These real-world problems are troublesome for their complexity and high resource consumption in consequence. Generally, these problems present constraints, however with the methodology of transforming the problem into a QUBO model, a binary unconstrained quadratic problem is obtained.

The third step is to learn about how quantum annealing works, which is a metaheuristic method based on quantum mechanics that is highly compatible with QUBO models. For that end, the framework to be utilized is the quantum computers offered by D-wave

systems, which employ this technique to solve problems on QPUs.

And the last purpose of this project is to put together every insight acquired in a practical case study. The problem choice for the implementation of all of the above has been the QAP problem. The hard nature of the problem is a perfect fit to test the limits of current quantum technology available in the market and showcase a glimpse of what the future might have in store.

All in all, the goals of the project are the following ones:

- The study of the basic mathematical foundation required for quantum computing as well as the four postulates of quantum mechanics.
- The comprehension of the transformation of CO problems into QUBO, leaving aside constraints and creating an equivalent model to the original.
- The usage of quantum annealing to solve QUBO problems, which effectively solves the original CO problem.
- The implementation of the methodology for the particular case study with the QAP problem. Define the boundaries of quantum computing altogether.

3. CHAPTER

State of the Art

Quantum computing is based on the use of a quantum-mechanical approach to tackle problems and provide solutions through quantum computation. As such, the quantum computation field of study possess a large background of vital knowledge to understand how it performs the operations and on which systems it does so.

In this chapter, the main objective is to present the mathematical foundation required to perform different operations in the quantum space, present the architectures that support it and compare how substantially different the quantum computation and classic computation are. Altogether, the essential requirements to start diving into quantum computation are gathered here.

3.1 Mathematical foundation

The main mathematical field, from which quantum computation draws, is Linear Algebra. Although basic knowledge of Linear Algebra is understated, there are still many aspects to tie together and introduce them accordingly. They go from specific notation to particular applications that fit together with the quantum environment in which they are employed. Basic notation used through the chapter is covered here, and is progressively extended while the section advances:

- Sets of numbers: $\mathbb{C}$ (complex), $\mathbb{R}$ (real), $\mathbb{Z}$ (integer) and $\mathbb{N}$ (natural) respectively.

- $[m]$: Corresponds to the set $\{1,...,m\}$

In general, the notation that is used through the entirety of the chapter is the bra-ket notation, also known as the Dirac notation, which is used to describe quantum states. The symbols used in that notation include ψ, ϕ, $\langle$, $\rangle$ and $|$, which are commonly used in quantum mechanics.

3.1.1 Complex numbers and vectors

The mathematical object to work with are complex vectors $|\psi\rangle \in \mathbb{C}^n$, where n corresponds to the complex space's dimension n. Each element of the vector is denoted $\psi_i \in \mathbb{C}$. In particular, the terminology for $|\psi\rangle$ is called "ket ψ".

$$|\psi\rangle = \begin{pmatrix} \psi_i \\ \vdots \\ \psi_n \end{pmatrix} \tag{3.1}$$

A complex number $c \in \mathbb{C}$ can be represented with two different but equivalent forms. The first form is the rectangular form $c = a + bi$ for $a, b \in \mathbb{R}$ and $i = \sqrt{-1}$. The other form, known as polar form, is written as $c = re^{i\theta}$. They are both interchangeable and can be converted from one to another. Their relationship can be illustrated in the complex 2D plane, where the x-axis corresponds to the real part, while the y-axis corresponds to the imaginary part (see Figure 3.1).

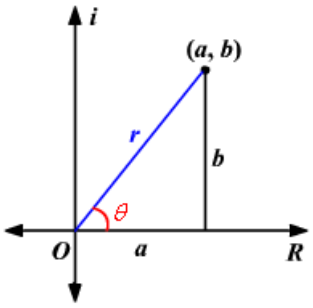

Figure 3.1: Graphical representation in a 2D plane of the terms composing each complex numbers' forms.

The conjugate of a complex number c is expressed c^*. It corresponds to $a - ib$ to the rectangular counterpart and $re^{-i\theta}$ to the polar form. The magnitude of a complex number c is calculated as $\sqrt{cc^*}$, which in case of each of the forms it can be computed this way:

- Rectangular form ($c = a + bi$):

$$|c| = \sqrt{(a+bi)(a-bi)} = \sqrt{a^2+b^2} \tag{3.2}$$

- Polar form ($c = re^{i\theta}$):

$$|c| = \sqrt{re^{i\theta}re^{-i\theta}} = \sqrt{r^2} = r \tag{3.3}$$

The utility of complex vectors resides in using them to represent quantum states. Therefore, some properties of the vectors are required to understand the different options available when working with them. As starters, the conjugate transpose of $|\psi\rangle$ is defined as $\langle\psi|$ and it refers to a row vector. Specifically, the conjugate transpose is the transposed column vector $|\psi\rangle$ where each of the elements $\psi_i \in \mathbb{C}$ is the conjugate of the original, as described below.

$$\langle\psi| = (\psi_1^*, \psi_2^*, ..., \psi_n^*) \tag{3.4}$$

The term $\langle\psi|$ is read as "bra ψ". With this established the overlapping of two vectors can be quantified through the inner product operation of two vectors, also known as dot product, which is presented below.

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \psi_i^* \phi_i \tag{3.5}$$

The inner product has the property of $(\langle\psi|\phi\rangle)^* = \langle\phi|\psi\rangle$. The magnitude of a vector $|\psi\rangle$ can now be computed as the inner product with itself, which is the Euclidean norm of the vector $|\psi\rangle$.

$$\||\psi\rangle\|_2 = \sqrt{\langle\psi|\psi\rangle} \tag{3.6}$$

After defining a norm, the measurement of distance between two vectors comes into the play with the Euclidean distance between two vectors: $\||\psi\rangle - |\phi\rangle\|_2$. The distance between two vectors helps in measuring how much two quantum states $|\psi\rangle$ and $|\phi\rangle$ can be differentiated. The Euclidean norm has two properties than can be of use:

- Positive scalability: For $\alpha \in \mathbb{C}$,

$$\|\alpha|\psi\rangle\|_2 = \alpha\||\psi\rangle\|_2 \tag{3.7}$$

- Triangle inequality: For any pair of vectors $|\psi\rangle$ and $|\phi\rangle$,

$$\||\psi\rangle + |\phi\rangle\|_2 \leq \||\psi\rangle\|_2 + \||\phi\rangle\|_2 \tag{3.8}$$

These two properties show that for all $|\psi\rangle \in \mathbb{C}^n$, $\||\psi\rangle\|_2 \geq 0$

3.1.2 Concepts of Linear Algebra

Orthonormal bases

In linear algebra, vectors are represented in a vector space of n dimensionality. A more common and natural way to achieve it consists in defining a basis. A basis is a set of vectors that are sufficient to define any other vector of the vector space of said dimensionality. For that to satisfy with the minimum amount of vectors (which corresponds to the dimensionality of the vector space), generally, orthonormal bases are used, which allow a practical and useful representation of the vector space owing to, mainly, the following characteristics for a set of vectors $|\psi\rangle_i \in \mathbb{C}^n$

- Orthogonality: Defines the concept of perpendicularity for vectors in a vector space in linear algebra. Two vectors $|\psi\rangle_i$ and $|\psi\rangle_j$ of the set that defines the basis are said to be orthogonal if $i \neq j$ and for their inner product they hold that $\langle\psi|_i|\psi\rangle_j = 0$. This ensures that if a basis is orthogonal, every pair of vector don't overlap each other.

- Unit vectors: Every vector of the set has length 1, in other words, the inner product with itself is $\langle\psi|_i|\psi\rangle_i = 1$. Another way to view this is to consider that for every vector $|\psi\rangle_i$ its Euclidean norm is $\||\psi\rangle_i\|_2 = 1$.

Therefore, an orthonormal basis can be defined using the Kronecker delta function where for every pair of vectors $|\psi\rangle_i$ and $|\psi\rangle_j$ the inner product $\langle\psi|_i|\psi\rangle_j$ is 1 if $i = j$ or 0 if $i \neq j$. This means that every vector is an unit vector of the vector space and, for every different pair of vectors of an orthonormal basis, no overlapping occurs.

A very common basis is the *computational basis* for the vector space with the set $\mathbb{C}^2$. It is built with the vectors $|0\rangle$ and $|1\rangle$ forming an orthonormal basis, which are illustrated below:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{3.9}$$

The *computational basis* $\{|0\rangle, |1\rangle\}$, as for every other orthonormal basis, can represent any vector $|\psi\rangle \in \mathbb{C}_2$ for any parameters $\alpha, \beta \in \mathbb{C}$ in the following manner:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.10}$$

In case the vector represented has length 1 ($\||\psi\rangle\|_2 = 1$), it is referred as normalized vector, which also fulfills that in such case $|\alpha|^2 + |\beta|^2 = 1$.

The computational basis can be used for every n dimensional sets. For building a basis that ensures the orthonormal requirements, for i in the integer interval $[0, n)$ a vector $|i\rangle$ is created and placed in the set of vectors that will form the basis of the corresponding vector space. Every vector $|i\rangle$ has the value 1 in the i-th position of the vector and 0 in the rest of the positions.

Linear maps

A linear map is the transformation of a vector $|\psi\rangle \in \mathbb{C}^n$ in a vector space $\mathcal{X}$ to a vector of a vector space $\mathcal{Y}$. The set of linear maps of such characteristics are witten as $\mathcal{L}(\mathcal{X}, \mathcal{Y})$. If both vector spaces are the same ($\mathcal{X} = \mathcal{Y}$) this is called a linear endomorphism, which is a linear map of a vector space to itself and written as $\mathcal{L}(\mathcal{X})$ for a more compact notation. For example, the linear map $\Phi : \mathbb{C}^n \mapsto \mathbb{C}^n$ is an endomorphism of a vector space defined by the set $\mathbb{C}$ to itself.

Consider an arbitrary vector built by the addition of the vectors of the basis of the vector space, such as $\sum_{i=0}^{n-1} \alpha_i |\psi\rangle_i \in \mathbb{C}^n$, which allow to represent any possible vector in the terms of the vectors from the basis. The importance relies in the close relationship between the basis and the linear map itself, since a linear map $\Phi : \mathcal{X} \mapsto \mathcal{Y}$ with arbitrary vectors $|\psi\rangle, |\phi\rangle \in \mathcal{X}$ fulfill the following two constraints:

- Operation of addition: $\Phi(|\psi\rangle + |\phi\rangle) = \Phi(|\psi\rangle) + \Phi(|\phi\rangle)$
- Operation of scalar multiplication: $\Phi(\alpha|\psi\rangle) = \alpha\Phi(|\psi\rangle)$

Therefore, a linear map is said to preserve the operations of addition and scalar multiplication since they can be applied either before or after the linear mapping of the vector

without having any impact on the result. This is very useful for allowing the application of the linear mapping to the vector as the addition of the parameter by the mapping of the vector of the basis, as it is shown below:

$$\Phi(\sum_{i=0}^{n-1} \alpha_i |\psi\rangle_i) = \sum_{i=0}^{n-1} \alpha_i \Phi(|\psi\rangle_i) \tag{3.11}$$

These constraints, therefore, allow for a linear mapping in terms of the basis of the origin vector space to the destined vector space. In other words, the vectors of the basis are mapped to the corresponding ones in the other vector space. The idea is to change the perspective in which the vector space can be represented to fit from one to the next through the mapping of the vectors of the basis. This makes a linear mapping being specified by a matrix the natural form of expressing the mapping in mathematical terms.

A complex matrix A of dimensions $n \times n$ is a two dimensional array of complex numbers, where for every $i, j \in [n]$ a position of the matrix is denoted as (i, j)-th position and the value of that position corresponds to $A(i, j) \in \mathbb{C}$. A representation of the linear map $\Phi : \mathbb{C}^n \mapsto \mathbb{C}^n$ as a matrix A_Φ of dimensions $n \times n$ can be obtained observing the transformation of the basis of $\mathbb{C}^n$. For that purpose, the matrix is built as the linear map of the vectors from the standard basis of $\mathbb{C}^n$, where $\Phi(|i\rangle)$ corresponds to the i-th column of the matrix and $|i\rangle$ is the vector of the basis with value 1 in the i-th position and 0 in the other ones. This can be viewed as the following:

$$A_\Phi = [\Phi(|0\rangle), \Phi(|2\rangle), ..., \Phi(|n-1\rangle)] \tag{3.12}$$

The product of two matrices can be only obtained if the columns of the left side of the product and the rows of the right side of the product are equal. In case, the number of rows and columns of a matrix are the exact same one, the matrix is called a square matrix. After performing the product of a matrix, the amount of rows is the same as the amount of rows of the left side of the product's matrix, while the number of columns correspond to the number of columns of the right side of the product's matrix. Given the square matrices A and B of dimension $n \times n$ each, each position (i, j) of the product of the two matrices AB is calculated this way:

$$AB(i, j) = \sum_{k=1}^{n} A(i, k) B(k, j) \tag{3.13}$$

Unlike with scalars, the product of two matrices, if possible, doesn't ensure the same result when the matrices position in the product are swapped. However, if $AB = BA$, it is said that A and B commute.

Particularly, a linear map $A \in \mathcal{L}(\mathbb{C}^n)$ produces an output vector space of certain length. The entire set of the output vectors from the application of the linear map A is called the image of the linear map A:

$$Im(A) := \{|\psi\rangle \in \mathbb{C}^n \mid |\psi\rangle = A|\phi\rangle \ \wedge \ |\phi\rangle \in \mathbb{C}^n\} \tag{3.14}$$

The dimension of $Im(A)$ is called the rank of A ($rank(A)$). A dimension of a vector space is defined as the amount of vectors that form a basis of that vector space. Another thing to consider is the null space or the kernel of the linear map A, which consists in the set of vectors that are transformed into the null vector by A:

$$Null(A) := \{|\psi\rangle \in \mathbb{C}^n \mid A|\psi\rangle = 0\} \tag{3.15}$$

The dimension of the image of A and the dimension of the null space of A are connected with the dimension of the vector space $\mathbb{C}^n$ as follows:

$$dim(Null(A)) + dim(Im(A)) = n \tag{3.16}$$

Matrix operations

Matrices store the vectors to which different operations are performed. There are many operations available to matrices that allow their convenient manipulation and transformation, making linear maps even more versatile. Some of the basic operations are listed below:

- Complex conjugate: The complex conjugate of a matrix A, consists in replacing each of the elements $A(i, j)$ of the matrix by its complex conjugate $(A(i, j))^*$.
- Transpose: The transpose of a matrix is obtained by shifting the column vectors form left to right into the row vectors from top to bottom. Every element transposed $A^T(i, j)$ is placed in $A(j, i)$.

- Adjoint: This is a combination of the previous two operations, also known as conjugate transpose of the matrix, and written as $A^\dagger = (A^*)^T$

- Trace: The trace of a matrix is the linear map $Tr : \mathcal{L}(\mathbb{C}^n) \mapsto \mathbb{C}$ which is the sum of all the elements of the diagonal of the matrix calculated as $Tr(A) = \sum_{i=1}^{n} A(i,i)$. It fulfills the cyclic property which states that the trace is the same no matter in which order matrices are multiplied. For example, $Tr(CAB) = Tr(ABC)$. This is satisfied even if they don't commute with each other.

For the transpose and the adjoint the following is satisfied:

$$(AB)^\dagger = B^\dagger A^\dagger \qquad (AB)^T = B^T A^T \tag{3.17}$$

Another tool when operating with matrices is the outer product, which yields a description of matrices through applying these operations to vectors. Formally, given two vectors $|\psi\rangle, |\phi\rangle \in \mathbb{C}$, the outer product of these two vectors is a linear map $|\psi\rangle\langle\phi| \in \mathcal{L}(\mathbb{C}^n)$ which results in a $n \times n$ matrix. It is calculated with the rules of matrix multiplication as occurs with the examples below:

$$|0\rangle\langle 0| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad |1\rangle\langle 0| = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{3.18}$$

A general rule can be extracted for the outer product of the vectors of the computational basis. For a matrix resulting from the outer product of $|i\rangle\langle j| \in \mathcal{L}(\mathbb{C}^n)$, it has at position $(i+1, j+1)$ the value $|i\rangle\langle j|(i+1, j+1) = 1$ and for the rest of the positions the value 0. Therefore, a matrix $A \in \mathcal{L}(\mathbb{C}^n)$ can be written as the sum of the outer product of the vectors of the computational basis by the value taken in that position by the matrix A:

$$A = \sum_{i=0, j=0}^{n-1} A(i+1, j+1)|i\rangle\langle j| \tag{3.19}$$

There is also the possibility of returning the value $A(i+1, j+1)$ using the following expression:

$$\begin{aligned}\langle i|A|j\rangle &= \langle i|(\sum_{i'=0,j'=0}^{n-1} A(i'+1,j'+1)|i'\rangle\langle j'|)|j\rangle \\ &= \sum_{i'=0,j'=0}^{n-1} A(i'+1,j'+1)\langle i|i'\rangle\langle j'|j\rangle \\ &= \sum_{i'=0,j'=0}^{n-1} A(i'+1,j'+1)\delta_{ii'}\delta_{jj'} \\ &= A(i,j)\end{aligned} \tag{3.20}$$

The key part of the equation relies in understanding the implication of the third equality. Since $\{|i\rangle\}$ creates an orthonormal basis for $\mathbb{C}_n$, the vectors inner product is only 1 when the pair of vectors selected is the same and 0 otherwise, which is the Kronecker delta function.

Eigenvalues and eigenvectors

Through a linear transformation there are special non-zero vectors $|\psi\rangle \in \mathbb{C}^n$ that are only stretched by a scalar value $\lambda \in \mathbb{C}$ when a matrix $A \in \mathcal{L}(\mathbb{C}^n)$ is applied to perform the linear mapping of the vector to another vector space. If λ is positive the vector is stretched in the same direction, while if λ is negative the vector is stretched in the opposite direction. These vectors $|\psi\rangle \in \mathbb{C}^n$ are called eigenvectors and the scalar $\lambda \in \mathbb{C}$ associated to them is the eigenvalue that dictates the amount they stretch, which are related with the equation below:

$$A|\psi\rangle = \lambda|\psi\rangle \tag{3.21}$$

This is similar to the idea of the axis of a rotation. For example, if we were rotating a three dimensional cube, which is a linear transformation in reality, the vectors that form the cube would be changed. However, the vectors that are in the axis of the rotation would remain the same, which would be the eigenvectors of the rotation.

The equation above is equivalent to substracting the scalar λ from the diagonal of the matrix A, as shown by the following equation:

$$(A - \lambda I)|\psi\rangle = 0 \tag{3.22}$$

Nevertheless, for the linear transformation of a non-zero vector to produce a zero vector (matrix multiplication with a non-zero vector), the only possible way is for the linear transformation A to compress and squish the dimension of the vector space into a lower dimension. This can only be obtained if the determinant of the matrix is 0, resulting in the equation below:

$$det(A - \lambda I) = 0 \tag{3.23}$$

The determinant can be calculated recursively following this equation:

$$det(A) = \sum_{j=1}^{n} (-1)^{i+j} A(i,j) A_{i,j} \tag{3.24}$$

Here $A_{i,j}$ is the submatrix of A deleting the row i and column j. The base case of the recursion is a 1×1 matrix, for which $det([a]) = a$.

Among the matrices, diagonalizable matrices are particular cases in which the matrix is similar to a diagonal matrix. A matrix is considered to be diagonalizable if an invertible matrix P and a diagonal matrix D can be obtained satisfying the following equation:

$$P^{-1}AP = D \qquad A = PDP^{-1} \tag{3.25}$$

Any matrix A that fulfills that $AA^{\dagger} = A^{\dagger}A$, it is called a normal matrix. A normal matrix is a diagonalizable matrix, because, as the spectral theorem states, a matrix is considered to be normal if it is similar to a diagonal matrix.

Diagonalizable matrices are particularly interesting, since they can be represented by the eigenvectors and eigenvalues in a process called the spectral decomposition of a matrix. This shows how much a diagonalizable matrix A affects $\mathbb{C}^n$ owing to the eigenvectors $|\lambda_i\rangle \in \mathbb{C}^n$ forming an orthonormal basis for $\mathbb{C}^n$. Therefore, any vector $|\psi\rangle \in \mathbb{C}^n$ with some scalar $\alpha_i \in \mathbb{C}$ can be represented in terms of the eigenvectors of A as following:

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |\lambda_i\rangle \tag{3.26}$$

The spectral decomposition of a matrix A is obtained using outer products similarly to how it was achieved with the computational basis before:

$$A = \sum_{i=0}^{n-1} \lambda_i |\lambda_i\rangle\langle\lambda_i| \tag{3.27}$$

Through the spectral decomposition the $rank(A)$ can be extracted, that is, the total number of non-zero eigenvalues of A. Moreover, this also makes the spectral decomposition an easy way to calculate the trace of the matrix:

$$\begin{aligned} Tr(A) &= Tr(\sum_{i=0}^{n-1} \lambda_i |\lambda_i\rangle\langle\lambda_i|) \\ &= \sum_{i=0}^{n-1} \lambda_i \, Tr(|\lambda_i\rangle\langle\lambda_i|) \\ &= \sum_{i=0}^{n-1} \lambda_i \, Tr(\langle\lambda_i|\lambda_i\rangle) \\ &= \sum_{i=0}^{n-1} \lambda_i \end{aligned} \tag{3.28}$$

This occurs as a result of three conditions satisfying: First, the trace operation is a linear operation, second, the trace has the property of being cyclic and, last, the eigenvectors are orthonormal.

As a final note on diagonal matrices, it is straightforward to determine the eigenvalues and eigenvectors of such matrices. The values of the diagonal of the matrix correspond to the eigenvalues, while the eigenvectors correspond to the computational basis.

Hilbert spaces

A hilbert space is a vector space $\mathbb{H}$ over real or complex numbers with a defined inner product $\langle\cdot|\cdot\rangle$, which allows to measure the length and angle of a vector, and with a complete metric space through the inner product itself. This means that for every pair of vectors $|\psi\rangle, |\phi\rangle \in \mathbb{H}$ the inner product $\langle\psi|\phi\rangle$ is defined and there is a metric using the inner product whose results being within the limits of the vector space $\mathbb{H}$. This metric is the norm, which has already been explained in detail.

In fact, vector spaces for vectors representing quantum states are Hilbert spaces. All in all, Hilbert spaces model the quantum states in quantum systems.

3.2 The leap from classical to quantum

After having prepared the ground with the mathematical toolkit that is required to master in order to formally express the representation of a quantum system and the diverse operations for enabling new possibilities, in this section a broad perspective of the classical approach and the quantum approach is discussed. The principal choice of design and operations in each of them are addressed, and how the intuition that served as the logical conduct in a classical systems ends up distorted when a quantum system enters into the equation.

One of the discrepancies between the classical system and the quantum system is how the information is handled and the state of the system defined. In a classical approach, there is only one possibility for a state to hold, however this turns into a blurry perception when a quantum system introduces the concept of superposition that gives the possibility to introduce indeterministic quantum states that are unclear whether the description of the state corresponds to one situation or another.

Above all, this is the transformation of the classical systems with clear description of each state and notions following the common sense to the more sophisticated and hardly intuitive quantum systems. For this purpose, the particular approaches will be tackled through graph theory and matrix representation of each system's philosophy.

3.2.1 Classical systems

A classical system is equivalent to a graph representing the flow of the dynamics of the environment through deterministic transitions between states. That is to say, the nodes represent the states of the system and the edges the next state to which it will change after a time click has passed. The weighted nodes have a number associated to describe the state of the system, while the change between states is represented by a matrix.

The matrix M used in representing a transition between states, however, is a boolean matrix that encapsulates the graph as an adjacency matrix. As such, when there is an edge between two nodes (i, j) of the graph, the value in the matrix in that position is $M[i, j] = 1$. Otherwise, if there is no such edge the value is $M[i, j] = 0$.

The main feature of a classical system can be perceived by the intrinsic deterministic nature it holds which leads the graph into being a directed graph and with only one outgoing edge from each of the nodes. This way, it is always known how the system will behave for

each of the nodes, since from one node there is always only one path to next node. Hence the deterministic nature.

For instance, consider the following state $X = [0,5,7,4]^T$ and the dynamics of the system are defined by a matrix M. One time step changes the current state X to Y, which can be obtained by matrix multiplication as showed below:

$$Y = MX = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 5 \\ 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 7 \\ 4 \\ 5 \end{pmatrix} \tag{3.29}$$

As mentioned before, the deterministic nature is reflected by only allowing one possible edge in each column of the matrix or, in other words, with only one outgoing edge existing for each of the nodes. If more than one time step want to be taken simultaneously, it is enough to multiply the matrix k times, where k is the total amount of steps that want to be taken one after another.

$$Y = M^k X = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}^k \begin{pmatrix} 0 \\ 5 \\ 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 7 \\ 4 \\ 5 \end{pmatrix} \tag{3.30}$$

3.2.2 Quantum systems

Before entering into quantum territory, probabilistic systems have to be considered. In this systems, there is indeterminacy about the physical state in which the system is at, which is expressed with vectors that meet two conditions. The first condition is to ensure that the sum of all values (real numbers) of the vector is 1, inducing the probability regarding to in which state the system currently is. The second condition is that when the state is changed the resulting state must satisfy the first condition once again.

Although the dynamic of the system is still represented by a matrix as it occurred with a deterministic system, the transition are based in probabilities and therefore a position of the matrix $M[i, j]$ indicates the likelihood to go from node i to j. As a result, the system isn't deterministic anymore and the values of for each column of the matrix must sum up to one.

The world of quantum systems shifts the common understanding of probabilities expressed with real numbers and uses complex numbers instead. Each weight of a state is a complex number c such that $|c|^2$ is a real number between 0 and 1. Similarly to probabilistic systems, each weight represents a probability, however, unlike with real numbers, probabilities given with complex numbers can cancel each other.

This phenomenon is called quantum interference and implies that the probability of summed weights doesn't need to be necessarily higher than the probability of each weight separately. Therefore, an order relationship can't be established and the probabilities may cancel each other out in the process. In order to illustrate this, consider the following two complex numbers c_1 and c_2 for which either one of the next equations can be held depending on the respective values of the complex numbers:

$$\begin{aligned} |c_1|^2 &> |c_1 + c_2|^2 \\ |c_2|^2 &> |c_1 + c_2|^2 \end{aligned} \tag{3.31}$$

$$\begin{aligned} |c_1|^2 &\leq |c_1 + c_2|^2 \\ |c_2|^2 &\leq |c_1 + c_2|^2 \end{aligned} \tag{3.32}$$

The transition and modifications to a state are performed similarly through a matrix, nevertheless, with an unitary matrix, which is covered in the next section. The reason behind choosing unitary matrices lies in how they can be reversed to undo steps taken, achieved though the Hermitian adjoint (a generalization of conjugate transpose in Hilbert spaces for matrices of any dimension) of the unitary matrix. That is why quantum gates are said to be reversible, the transformation of the quantum state can be restored back to a previous state with the Hermitian adjoint.

The final and, maybe, greatest selling point of quantum systems is that a quantum state can be overlapped with many basic states to create a superposition. As a result, one quantum state may refer to many basic states through a probability system based on complex numbers.

3.3 The four postulates of quantum mechanics

Quantum computation is based on using devices operated by using quantum mechanics. These are mathematical rules and theorems that describe the behaviour of subatomic particles such as protons and electrons. However, grasping the manner in which they occur goes sometimes against the common sense, which is a critical factor to understand how and why quantum mechanics work. For that end, quantum mechanics principles can be gathered in four postulates that describe the inner workings of how it performs. These will be covered in this section.

3.3.1 First postulate: Single quantum system

In a classical approach, a bit can only take on value, either 0 or 1. However, a quantum bit, also known as a qubit, is not limited by this constraint of only taking one value between them. A quibit can take both values simultaneously, so it would essentially be both 1 and 0, quite contradictory and many times conflicting. The uncertainty of the statement is what makes the difference and its an attribute of quantum mechanics. This indeterministic nature can be represented mathematically by encoding bits 0 and 1 with the computational basis $|0\rangle, |1\rangle \in \mathbb{C}^n$ respectively, called a superposition:

$$|0\rangle + |1\rangle \tag{3.33}$$

In order to decide how close the qubit is from one state to another, parameters known as amplitudes $\alpha, \beta \in \mathbb{C}$ are used. The result is a vector $|\psi\rangle \in \mathbb{C}^2$ with the constraint of being an unit vector. It can be concluded that every unit vector $|\psi\rangle$ corresponds to a representation of the state of a single qubit. These statement can be presented with the following two equations:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad |\alpha|^2 + |\beta|^2 = 1 \tag{3.34}$$

This can be generalized for a d dimensional single qubit, where the vector $|\psi\rangle \in \mathbb{C}^d$ is an unit vector that describes the quantum state of the single qubit. The way to compute the unit vector is, once again, using the computational basis vectors $|i\rangle \in \mathbb{C}^d$ and parameters $\alpha_i \in \mathbb{C}$:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + ... + \alpha_{d-1}|d-1\rangle = \sum_{i=0}^{d-1} \alpha_i|i\rangle \qquad \sum_{i=0}^{d-1} |\alpha_i|^2 = 1 \tag{3.35}$$

3.3.2 Second postulate: Quantum operations

A vector representation allows a qubit to be transformed applying a linear map. However, there is a key limitation for what kind of linear maps can be used to transform a vector $|\psi\rangle$ representing the state of the qubit. In particular, the matrix $U \in \mathcal{L}(\mathbb{C}^n)$ is an unitary matrix, which means that it satisfies $UU^\dagger = U^\dagger U = I$. Therefore, both the inverse of the matrix and the conjugate transpose of the matrix are the same. The matrix or linear map U in the quantum environment is the quantum gate that transforms the qubit's state and can be always reversed, so that quantum gates are reversible.

The most basic quantum gates when working with only a single qubit state are Pauli gates:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.36}$$

The X matrix is a quantum gate with the same effect as a NOT operator would have, the elements of the vector are swapped as a result:

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \qquad X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \tag{3.37}$$

The gate Z, on the other hand is set apart from the parallelism established for the X quantum gate. This gate introduces a relative phase to the vector. In particular, the phase is relative because it's only applied to the second element of the vector and the quantity phased is -1. Therefore only the amplitude of the second vector of the basis $|1\rangle$ is affected, as it is shown below:

$$Z|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \qquad X|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1\rangle \tag{3.38}$$

This introduces an important distinction to be made between global phase and relative

phase. In the former, all the amplitudes are phased by the same amount, while the later only is applied to the second vector of the basis. Relative phasing is a crucial element in quantum mechanics that allows to go beyond what is possible to achieve with a classical system.

The fourth quantum gate corresponds to the Hadamard gate. It performs a linear map to create a superposition of the quantum state, as well as reversing the superposition to the original quantum state. Consequently, the matrix H is a self-inverse as well as the Pauli gates presented before ($H^2 = I$). The matrix H is the following:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.39}$$

3.3.3 Third postulate: Composite quantum systems

Until now, only single quantum systems have been considered with quantum states $|\psi\rangle \in \mathbb{C}^2$. The next natural step is to consider quantum systems that work with many qubits simultaneously. The operation to join two qubit states consists on applying the tensor product $\otimes$ between two vectors $|\psi\rangle, |\phi\rangle \in \mathbb{C}^2$ to create a larger vector $|\psi\rangle \otimes |\phi\rangle \in \mathbb{C}^{2\times 2}$ that stores at the same time both quantum states by enlarging the dimension of the vector. The tensor product is calculated with the multiplication of the pair elements of between the vectors, that is, for indices $i, j \in \{0, 1\}$ the elements of each of the vectors are multiplied in the following way:

$$(|\psi\rangle \otimes |\phi\rangle)(i, j) := \psi_i \phi_j \tag{3.40}$$

Every pair of indices (i, j) corresponds to an element of the result of the tensor product vector with indices $x \in \{0..n \times n-1\}$. Another aspect to the tensor product is that it creates the orthonormal basis of the new dimension of the vector space if the tensor product is calculated for every pair of computational basis vectors of each of the vector spaces. As an example, take the orthonormal bases $B_1 = \{|\psi_0\rangle, |\psi_1\rangle\}$ and $B_2 = \{|\phi_0\rangle, |\phi_1\rangle\}$ for $\mathbb{C}^2$, a basis for $\mathbb{C}^4$ can be created through the tensor product operation for each pair of vectors of the bases $\{|\psi_0\rangle \otimes |\phi_0\rangle, |\psi_0\rangle \otimes |\phi_1\rangle, |\psi_1\rangle \otimes |\phi_0\rangle, |\psi_1\rangle \otimes |\phi_1\rangle\}$.

The application of the tensor product for two qubit systems can be extended for n qubit systems. For $|\psi\rangle \in \mathbb{C}^{n_1}$ and $|\phi\rangle \in \mathbb{C}^{n_2}$ with indices (i, j) where $i \in \{0..n_1 - 1\}$ and $j \in \{0..n_2 - 1\}$, the tensor product of the vectors $(|\psi\rangle \otimes |\phi\rangle)(i, j) = \psi_i \phi_j$ corresponds to the

vector space $\mathbb{C}^{n_1 \times n_2}$. Every time a new qubit is added, the dimension of the vector space grows by a factor of 2, which makes the n qubit systems have exponential growth ($\mathbb{C}^4 \otimes \mathbb{C}^2 = \mathbb{C}^8$). Generally, a n qubit system will have states represented with a vector $|\psi\rangle \in \mathbb{C}^{2^n}$ and, coincidentally, this exponential growth is one of the limitation factors for classical computers to handle quantum system simulations.

Given vectors $|a\rangle, |b\rangle \in \mathbb{C}^{n_1}$ and $|c\rangle, |d\rangle \in \mathbb{C}^{n_2}$, the tensor product has the following properties for vectors:

$$\begin{aligned}
(|a\rangle + |b\rangle) \otimes |c\rangle &= |a\rangle \otimes |c\rangle + |b\rangle \otimes |c\rangle \\
|a\rangle \otimes (|c\rangle + |d\rangle) &= |a\rangle \otimes |c\rangle + |a\rangle \otimes |d\rangle \\
c(|a\rangle \otimes |c\rangle) &= (c|a\rangle) \otimes |c\rangle = |a\rangle \otimes (c|c\rangle) \\
(|a\rangle \otimes |c\rangle)^\dagger &= |a\rangle^\dagger \otimes |c\rangle^\dagger = \langle a| \otimes \langle c| \\
(\langle a| \otimes \langle c|)(|b\rangle \otimes |d\rangle) &= \langle a|b\rangle\langle c|d\rangle
\end{aligned} \tag{3.41}$$

Quantum entanglement

There is a special phenomenon when describing a quantum state that can't be decomposed into two separate lower dimension quantum states. Such condition is called quantum entanglement, where two qubits q_0, q_1 have such a strong connection that separating them is not possible. Formally speaking, the joint state of q_0 and q_1 can be represented with a vector, but there are no pair of vectors $|\psi_1\rangle, |\psi_2\rangle \in \mathbb{C}^2$ whose tensor product $|\psi_1\rangle \otimes |\psi_2\rangle$ can represent the joint state of q_0 and q_1. This is a vital resource for quantum computers to achieve a higher performance than classical computers. These states are called Bell state, and there are four so far:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \tag{3.42}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle - \frac{1}{\sqrt{2}}|1\rangle \otimes |1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \tag{3.43}$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |1\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \tag{3.44}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes |1\rangle - \frac{1}{\sqrt{2}}|1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \tag{3.45}$$

This is a mind blowing quantum state, how could you only represent the joint state of two qubits but not the qubit itself, which would seem to be the obvious case in more classical approach. Despite being spatially separate between each other, there is no representation for each of them individually.

The four Bell states all together form an orthonormal basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ for $\mathbb{C}^4$ called the *Bell* basis.

Two qubit quantum gates

After presenting two qubits state using unit vectors in $\mathbb{C}^4$ obtained form the tensor product of vectors in $\mathbb{C}^2$ for doing so, two qubit quantum gates become a possibility and are, as presented in the second postulate, unitary matrices $U \in \mathbb{C}^4$. These can be obtained from two different sources, namely, the first option are quantum gates obtained from the tensor product of one qubit gates, while the second option is using truly specific two qubit gates that are created from scratch together.

The two qubit gates created from one qubit gates are tensor product of one qubit gates such as $X \otimes Z$ or $H \otimes H$. The tensor product can be used similarly for both vectors and

matrices. Given two matrices $A \in \mathcal{L}(\mathbb{C}^{n_1})$ and $B \in \mathcal{L}(\mathbb{C}^{n_2})$, the tensor product $A \otimes B$ yields a $n_1 n_2 \times n_1 n_2$ complex matrix with indices $([n_1] \times [n_2], [n_1] \times [n_2])$ where $[n] = \{0, ..., n-1\}$:

$$(A \otimes B)((i_1, j_1), (i_2, j_2)) := A(i_1, i_2)B(j_1, j_2) \tag{3.46}$$

For a better understanding of what is meant with the equation above, for two arbitrary matrices $A, B \in \mathcal{L}(\mathbb{C}^2)$, their tensor product builds a new complex matrix with 4×4 dimension for which each element $a_i \in A$ becomes a scalar multiplying each time the matrix B. Each element $b_i \in B$ is multiplied for every copy of matrix B by the scalar a_i:

$$A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \qquad B = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \tag{3.47}$$

$$A \otimes B = \begin{pmatrix} a_1 * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} & a_2 * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \\ \\ a_3 * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} & a_4 * \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \end{pmatrix} \tag{3.48}$$

The properties of vectors for the tensor product remain the same, with two new additions for matrices:

$$\begin{aligned} (A \otimes B)(C \otimes D) &= AC \otimes CD \\ Tr(A \otimes B) &= Tr(A)Tr(B) \end{aligned} \tag{3.49}$$

The second option is to consider two qubit gates designed to fulfill with a specific purpose, instead of simply generate one from a tensor product of single qubit gates. A perfect example for this is the remarkable controlled-NOT gate, called CNOT. The idea behind the CNOT quantum gates is based on assigning two different roles to the first and second qubit, which are the control qubit and the target qubit respectively. Therefore, if there are two qubits q_0 (control qubit) and q_1 (target qubit), the CNOT gate will apply the Pauli X gate to the target qubit if and only if the control qubit is set to $|1\rangle$. The effect of the CNOT gate for a two qubit basis can be seen below:

$$\begin{aligned} CNOT|0\rangle \otimes |0\rangle &= |0\rangle \otimes |0\rangle \\ CNOT|0\rangle \otimes |1\rangle &= |0\rangle \otimes |1\rangle \\ CNOT|1\rangle \otimes |0\rangle &= |1\rangle \otimes |1\rangle \\ CNOT|1\rangle \otimes |1\rangle &= |1\rangle \otimes |0\rangle \end{aligned} \tag{3.50}$$

The CNOT gate corresponds to the following matrix, where X and I are the Pauli X gate and the identity matrix:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \tag{3.51}$$

An interesting possibility is that the CNOT gate can be used the obtain the Bell states from the two qubit basis $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$ in conjunction with the Hadamard gate H and the identity matrix I. For instance, if we were to obtain the Bell state $|\Phi_+\rangle$, the following equation would yield the Bell state:

$$\begin{aligned} CNOT(H \otimes I)|0\rangle \otimes |0\rangle &= CNOT(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}} CNOT(|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle) \\ &= |\Phi_+\rangle \end{aligned} \tag{3.52}$$

3.3.4 Fourth postulate: Measurements

The act of measuring a quantum system consists on mathematically modeling the observation or measurement of the state a quantum system represents. That is, extracting from a quantum system a measurement of what particularly corresponds to when observed. In quantum mechanics, the very simple action of observing a quantum system conditions its state permanently, which implies that a quantum system can be only defined when a measurement is taken for it.

In order to model this occurrence, the idea of projective or von Neumann measurement is used. This requires to define three levels of linear operators, more restrictive in ascendant order:

- *Hermitian operators*: A linear operator $M \in \mathcal{L}(\mathbb{C}^n)$ is said to be a Hermitian operator if the adjoint $M^\dagger$ is the same operator, thus, $M = M^\dagger$ is satisfied for a linear operator to be Hermitian. One of their properties is that all of the eigenvalues of the linear operator are real and, therefore, a Hermitian operator is considered to be a higher dimensional generalization of the real numbers. Examples of Hermitian operators are Pauli gates (X, Y and Z).

- *Positive semi-definite operators*: It's a subset of Hermitian operators in which only the linear operators with real positive eigenvalues are taken. This results in the higher dimensional generalization of the positive real numbers.

- *Orthogonal projection operators*: For a Hermitian matrix $\Pi \in \mathcal{L}(\mathbb{C}^n)$ to be an orthogonal projection operator, the only possible eigenvalues of the linear operator are either 0 or 1. This means that $\Pi = \Pi^2$ is fulfilled for the Hermitian operator. A way to showcase why the eigenvalues can be only either 0 or 1 for orthogonal projection operators is by the spectral decomposition of the Hermitian matrix:

$$\begin{aligned} \Pi &= \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i| \\ &= \Pi^2 = \left(\sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|\right)\left(\sum_j \lambda_j |\lambda_j\rangle\langle\lambda_j|\right) \\ &= \sum_i \lambda_i^2 |\lambda_i\rangle\langle\lambda_i| \end{aligned} \tag{3.53}$$

  The last inequality is achieved since $\{|\lambda_i\rangle\}$ is an orthonormal basis. In consequence, every pair of vectors $(|\lambda_i\rangle, |\lambda_j\rangle)$ of the basis is independent (orthogonal) between each other, which leads the eigenvalues of the Hermitian operator to satisfy $\lambda_i = \lambda_i^2$. The only non-negative real values that fulfill that constraint are either 0 or 1, which concludes that $\lambda_i \in \{0, 1\}$. Altogether, the spectral decomposition of a projector for a given orthonormal basis $\{|\psi_i\rangle\}$ is $\Pi = \sum_i |\psi_i\rangle\langle\psi_i|$.

  If the spectral decomposition of a projector can be written in terms of the outer product of some vector $|\psi\rangle \in \mathbb{C}^n$ in this form $\Pi = |\psi\rangle\langle\psi|$, then the rank of the

matrix is 1. This occurs as a result of the rank of a projector Π being the number of non-zero eigenvalues (because is Hermitian) and only having one non-zero eigenvalue since its spectral decomposition is $\Pi = |\psi\rangle\langle\psi|$.

In order to understand the effect of a projector $\Pi = \sum_i |\psi_i\rangle\langle\psi_i|$ when a state $|\phi\rangle$ is being measured by it, observe that the following occurs:

$$\Pi|\phi\rangle = \left(\sum_i |\psi_i\rangle\langle\psi_i|\right)|\phi\rangle = \sum_i |\psi_i\rangle(\langle\psi_i|\phi\rangle) = \sum_i (\langle\psi_i|\phi\rangle)|\psi_i\rangle \in Span(\{|\psi_i\rangle\}) \tag{3.54}$$

The inner product $\langle\psi_i|\phi\rangle \in \mathbb{C}$ yields a complex number and, as a result, the projector Π projects down onto the span of vectors $\{|\psi_i\rangle\}$.

Projective measurements

A projective measurement is a set of projectors $B = \{\Pi_i\}_{i=0}^m$ that satisfies the completeness relation ($\sum_{i=0}^m \Pi_i = I$). In case every projector Π_i is rank one, that is, their spectral decomposition is $\Pi_i = |\psi_i\rangle\langle\psi_i|$, the projective measurement B is said to model a measurement in basis $\{|\psi_i\rangle\}$. More generally, the measurement is performed in the computational basis which forms the projective measurement $B = \{|i\rangle\langle i|\}_{i=0}^{n-1}$ for $\mathbb{C}^n$, for example, for $\mathbb{C}^2$ the projective measurement in the computational basis is $B = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$.

A projective measurement $B = \{\Pi_i\}_{i=0}^m \in \mathbb{C}^n$ is applied to a quantum state $|\psi\rangle \in \mathbb{C}^n$ to obtain the probability of each outcome $i \in \{0, ..., m\}$ when $|\psi\rangle$ is measured with B:

$$P(i) = Tr(\Pi_i|\psi\rangle\langle\psi|\Pi_i) = Tr(\Pi_i^2|\psi\rangle\langle\psi|) = Tr(\Pi_i|\psi\rangle\langle\psi|) \tag{3.55}$$

The second equality is thanks to the cyclic property of the trace operation and the third one since a projector holds that $\Pi = \Pi^2$. A direct analysis over modeling the probability distribution from the obtained outcomes shows that the vector $|\psi\rangle \in \mathbb{C}^n$ has to be an unit vector ($\sum_i |\psi_i|^2 = 1$) for all the outcome probabilities to sum to 1. Another thing to note is that the projective measurements are probabilistic, therefore the final outcome can't be predicted.

As mentioned at the beginning, the very act of observing a quantum state creates some disturbance in the system. When measuring with B for the obtained outcome Π_i, the system comes down to

$$\frac{\Pi_i|\psi\rangle\langle\psi|\Pi_i}{Tr(\Pi_i|\psi\rangle\langle\psi|\Pi_i)} \tag{3.56}$$

The denominator of the above division is a scalar indicating the probability associated with outcome i. The output is a quantum state $\Pi_i|\psi\rangle$, conditioned by the outcome Π_i. Nevertheless, the new quantum state is represented by a matrix, instead of a vector as up until now. This is called the density operator formalism and a vector $|\psi\rangle$ can be transformed into a density matrix by $|\psi\rangle\langle\psi|$. This is a more general approach to describe a quantum state, and is particularly important when working with individual subsystems of larger composite quantum states.

After the projective measurement is completed, the quantum state needs to be normalized once again, since the operation modifies part of the original vector $|\psi\rangle$. For that purpose, the measured quantum state $\Pi_i|\psi\rangle$ is divided by the Euclidian norm as follows:

$$|\psi'\rangle = \frac{\Pi_i|\psi\rangle}{\|\Pi_i|\psi\rangle\|_2} = \frac{\Pi_i|\psi\rangle}{\sqrt{\langle\psi|\Pi_i\Pi_i|\psi\rangle}} = \frac{\Pi_i|\psi\rangle}{\sqrt{\langle\psi|\Pi_i|\psi\rangle}} \tag{3.57}$$

Having obtained the outcome i, the state $|\psi'\rangle$ is the new state of the system after the projective measurement has been conducted.

The last part is to consider the implications of a projector's $\Pi = \Pi^2$ property has over the projective measurements carried out. In quantum systems, if there are no modifications through gates or noise to its state, a measurement taken at any point in time should return the same result. For a projective measurement $B = \{\Pi_i\}$ and consecutive measurement obtaining i and j for measurement 1 and measurement 2 respectively, the static result obtained can be modeled as presented below:

$$P(j|i) = Tr\left(\Pi_j \frac{\Pi_i|\psi\rangle\langle\psi|\Pi_i}{Tr(\Pi_i|\psi\rangle\langle\psi|)}\Pi_j\right) = \frac{Tr(\Pi_j\Pi_i|\psi\rangle\langle\psi|\Pi_i\Pi_j)}{Tr(\Pi_i|\psi\rangle\langle\psi|)} \tag{3.58}$$

Through the application of the completeness relation ($\sum_i \Pi_i = I$) the expression can be reduced, since then $\Pi_j\Pi_i = \delta_{ij}\Pi_i$ with δ_{ij} being the Kronecker's delta function. Therefore, if $i \neq j$ the equation above equals to 0, while if $i = j$ the following occurs:

$$\frac{Tr(\Pi_j\Pi_i|\psi\rangle\langle\psi|\Pi_i\Pi_j)}{Tr(\Pi_i|\psi\rangle\langle\psi|)} = \frac{Tr(\Pi_i^2|\psi\rangle\langle\psi|\Pi_i^2)}{Tr(\Pi_i|\psi\rangle\langle\psi|)} = \frac{Tr(\Pi_i|\psi\rangle\langle\psi|\Pi_i)}{Tr(\Pi_i|\psi\rangle\langle\psi|)} = \frac{Tr(\Pi_i|\psi\rangle\langle\psi|)}{Tr(\Pi_i|\psi\rangle\langle\psi|)} = 1 \tag{3.59}$$

In conclusion, measuring once again the quantum state yields the same outcome i every time. For that reason, even if the first observation introduces a disturbance in the system, subsequent observations will produce the same result constantly.

4. CHAPTER

Quantum architectures

Quantum computation requires certain aspects regarding its hardware to operate as expected. The heart of the quantum system is its Quantum Process Unit (QPU) which, in a similar fashion to how a standard CPU handles different tasks, enables the quantum computation by incorporating quantum mechanics to its arsenal.

The QPU is constructed with tiny loops that act as qubits. Each loop corresponds to a qubit whose current flows both forward and backwards, and, as a result, the qubit takes simultaneously both the value 0 and 1, effectively achieving a superposition. In order to become the loops superconductors that replicate quantum mechanical behaviour, the QPU temperature is reduced down to low temperatures, almost absolute zero. The qubit falls into either 1 or 0 when a request to solve a problem reaches its conclusion, which means that the superposition terminates for that specific task or problem into one value.

The qubits are connected through couplers (also superconductor loops) to create a structure that is controlled by a control circuitry that manages magnetical fields. All in all, programmable quantum devices are created to take advantage of the possibilities of quantum computation. Once a problem is solved, the superposition of each qubit yields either the value 0 or 1.

Every time quantum computation is used to handle a problem, two layers have to be distinguished, the classical layer and the quantum layer. Since the result can't be interpreted in terms of quantum mechanics owing to the superposition of the qubits, in the classical layer the description of the problem and its specification are passed onto the quantum layer to solve applying quantum mechanics. Once finished, the result is returned once again to

classical terms as the solution to the problem, fixing the values of the superposition.

4.1 Quantum annealing

Quantum annealing is a metaheuristic used to find the global optimum of a particular objective function from a given set of candidate solutions (represented as quantum states) through the use of quantum fluctuations[1]. Thanks to this methodology, a method or procedure to find the most optimal global solution from candidate solutions can be found in a finite search space. Common applications include problems where the search space is discrete with many local optimum.

In particular, quantum annealing strives to return the lowest possible solution in terms of energy, also known as the ground state. Therefore, Quantum annealing minimizes the objective function for that purpose. Whether the problem is an optimization problem that looks for the real energy level or a probabilistic sampling problem that yearns for low enough solutions in terms of energy, the search for low energy states remains in both cases and it is the main idea behind Quantum annealing.

In other words, Quantum annealing provides a method through which low energy states can be obtained to find the best possible approach to the global optimum of the problem. It follows the same tendency shown in physics and, in turn, Quantum annealing transforms the problem into an energy minimization problem in reality.

4.1.1 Practical application in D-Wave systems

A D-Wave system is formed by qubits that work as the lowest energy states of the system's superconductor loops. These are located in the QPU and, as explained before, a current flows in both directions within the loop of each qubit causing a superposition, which later, after the Quantum annealing process has been concluded, is fixed to one value.

The physics of the Quantum annealing can be divided in three parts for each qubit:

1. Only one minimum is set, a valley shaped diagram where the states are in superposition (Figure 4.1).

[1] A momentary change in energy amount in a point of the space.

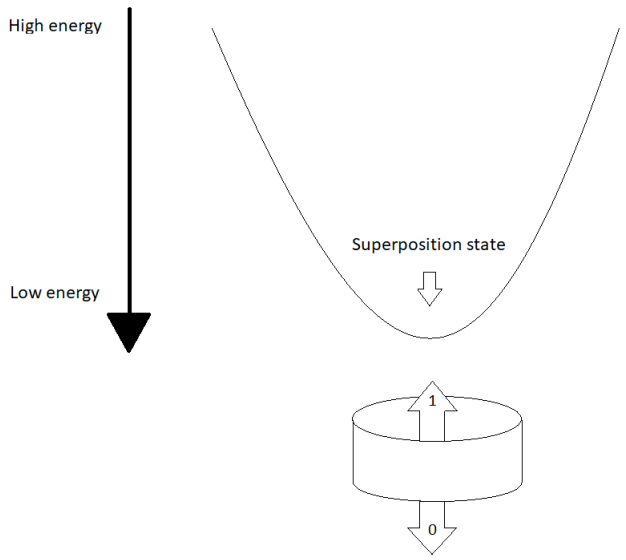

Figure 4.1: Initial state of superposition in the qubit at the beginning of the Quantum annealing.

2. After the Quantum annealing process finishes, the energy diagram is transformed into a double-well potential. The state 0 and state 1 are represented as two different low energy states in the process, which the Quantum annealing chooses between at the end (Figure 4.2).

3. An external magnetic field can be used to drop by a certain amount (bias) the energy level of a qubit. This increases the odds for the lowest energy state to be chosen between the both, instead of having equal chance of being either state 1 or state 0 (Figure 4.3).

However, the bias is not that meaningful and truly groundbreaking as qubits interacting with each other is. The qubits connected with couplers influence each other to a certain extent determined by a weight parameter. This can lead to two qubits having the same state or distance them as much as possible. When two or more qubits joined by couplers influence each other and the quantum state is defined by the three together, the qubits are said to be entangled.

The choice of bias and couplers' weight is called the energy landscape of the system. This influences how each qubit will be affected by the external magnetic force and how strong will the influence between qubits through couplers be. The amounts of qubits influencing each other when increased, it also increases the number of possibilities by a factor of two.

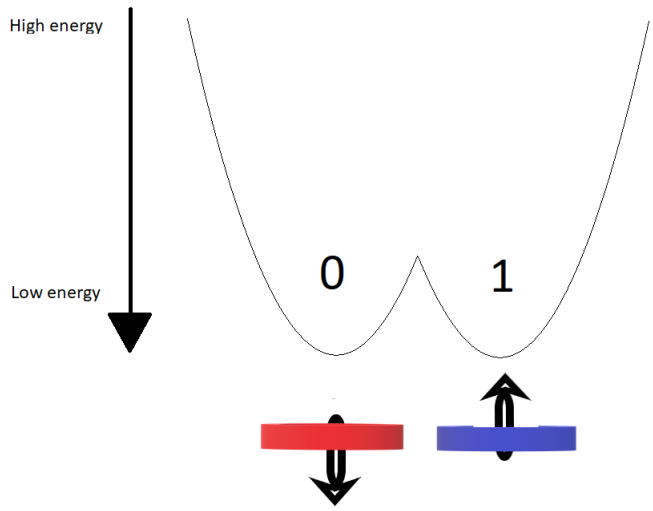

Figure 4.2: Double-well potential of the two states in a qubit after Quantum annealing has been performed.

4.1.2 Essential Quantum physics

Behind the scenes, valuable quantum physics define the guidelines of a quantum system's logic and operation. One of the most relevant aspects of Quantum physics is the Hamiltonian. Unlike the classical meaning of defining mathematically a physical system with variables of position and momentum, thus obtaining a description of the physical system through its energies; in quantum systems, a Hamiltonian (H) is a Hermitian operator of a Hilbert space.

The eigenvectors $|\alpha\rangle$ of H form an orthonormal basis in the Hilbert space. The spectrum of allowed energy levels is defined with the eigenvalues $\{E_\alpha\}$ and is the solution of the following equation:

$$H|\alpha\rangle = E_\alpha|\alpha\rangle \tag{4.1}$$

There are some specific states of the quantum system that the Hamiltonian ties with energy, called eigenstates. When a system is in one of these eigenstates, the energy associated to it is called eigenenergy. The set of pairs of eigenstates and their associated eigenenergy is known as eigenspectrum.

In a D-Wave system, for instance, the Hamiltonian can be expressed as following, where $\hat{\sigma}_{x,z}^{(i)}$ is the Pauli gate acting on the i-th qubit, q_i is the i-th qubit, h_i is the bias and $J_{i,j}$ is

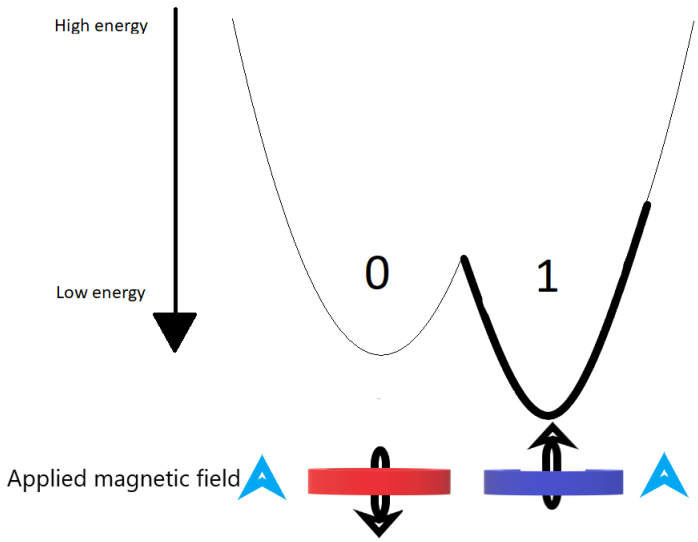

Figure 4.3: External magnetic field affecting the energy level of each state of the qubit.

the coupler's strenght between qubit i and j:

$$\mathcal{H}_{ising} = \underbrace{-\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right)}_{\text{Final Hamiltonian}} \tag{4.2}$$

At the beginning of the annealing, the eigenstate with the lowest eigenenergy is chosen for the initial hamiltonian. However, in the process the importance of the initial Hamiltonian is reduced considerably and the spotlight is redirected towards the final Hamiltonian. By the end, the final hamiltonian will have arrived to the best eigenstate possible for the lowest eigenenergy. The qubits at the end return a classical output solution to the problem.

5. CHAPTER

The QUBO approach to tackle combinatorial optimization problems

Combinatorial Optimization (CO) is one of the most relevant fields of optimization with practical applications active in each industry. Meanwhile, CO is also one of the hot spots in research among the Operation Research, Computer Science and Analytics communities in the pursue of finding good solutions to real world CO problems.

In most cases, CO problems are built around a set of binary choices where an objective function indicates the consequences of those choices, which are usually the cost, loss or profit. Therefore, in order to find the best possible solution a good selection of choices must be accomplished to maximize or minimize the objective function depending on the problem. This is incredibly hard and taxing, due to the vast number of combinations and time constraints straining the search of the desired solution.

Unlike a traditional solution which exploits the particular conditions and features of a CO problem and can only be used for that specific problem, a more general approach which can be applied to a wide range of CO problems without limitations is going to be used. The method consists in a mathematical reformulation of the CO problem called Quadratic Unconstrained Binary Optimization (QUBO) that wraps a large variety of well-known CO problems into an equivalent mathematical model. This way, solving the QUBO model effectively solves the CO problem.

The utility of the QUBO model lies in the compatibility with quantum annealing and, as a result, with the experimentation in quantum computers developed by D-Wave systems.

The potential of the QUBO model is being explored by both the traditional computation and quantum computation communities as a new methodology and framework that presents an effective and different approach to the more traditional models and methodologies.

QUBO problems are part of a class of problems considered to be NP-hard. Therefore, a exact solver for the problem will be a poor choice to make for a large number of instances owing to the long duration of the search being unfeasible. Nevertheless, modern metaheuristic methods, which provide good quality solutions to the problem without necessarily being optimal, but in a relatively short amount of time in comparison, have been a tremendous hit for these problems. This is further closing the gap between traditional and quantum computation.

5.1 Standard QUBO problem formulation

When a CO problem is transformed into a QUBO model, the QUBO formulation maintains the same objective as it did prior to the conversion, that is, the optimization of the problem as a QUBO model. The QUBO model is given as the following optimization problem:

$$\text{QUBO: } \min_{x_i \in \{0,1\}} \; / \; \max_{x_i \in \{0,1\}} y = x^t Q x \tag{5.1}$$

where x is a vector that contains binary decision variables $x_i \in \{0,1\}$ and Q is a square matrix of constant values corresponding to the constant values of the cost function. For the latter part, the Q matrix is commonly either in symmetric form or in upper triangular form, which is obtained with the corresponding modification for each one of the alternatives:

- Symmetric form: Except for the elements in the diagonal of the matrix, for every pair of positions $i, j \in \{1...n\}$ each element $q_{i,j}$ is replaced with $\frac{q_{i,j}+q_{j,i}}{2}$.

- Upper triangular form: For every pair of positions $i, j \in \{1...n\}$ under the condition that $j > i$, each element $q_{i,j}$ is replaced with $q_{i,j}+q_{j,i}$. After that, every value under the main diagonal of the matrix is set to 0.

5.2 Adding penalties to QUBO models

Unconstrained problems (apart for the decision variables being binary) require nothing more to be represented by a equivalent QUBO model. However, a large group of relevant and interesting problems contain constraints that must be satisfied to consider a solution feasible in the first place. This can be achieved by introducing quadratic penalties in the objective function that penalize unfeasible solutions in return.

The penalties are introduced so that the optimizer can avoid unfeasible solutions by considerably distancing the value of the objective function from feasible solutions. This way, the optimizer prevents not satisfying the constraints of the problem as it would normally be the case in a more traditional approach. For a QUBO model that means that the penalty value is added when a solution is detected to be unfeasible for not satisfying at least a constraint, while the penalty value is suppressed when a solution satisfies every constraint of the problem.

Naturally, the more constraints that are not met, the greater the penalty value. In a minimization problem, the penalty value increases the objective function; for maximization problems the penalty value subtracts the objective function. Therefore, the optimizer will look for solutions that restore the penalized objective function to the original function by choosing solutions where there is no penalization in the objective function, that is, feasible solutions to the problem.

There are already a set of known constrained penalties, presented in Table 5.1, that can be used to penalize an objective function when transforming the constrained problem into a QUBO model. The variables of the penalty terms are binary and the parameter P is a positive scalar value. The parameter P needs to be large enough to ensure that the original constraint is satisfied so that the optimizer yields a feasible solution.

Classical constraint	Equivalent penalty
$x+y \leq 1$	$P(xy)$
$x+y \geq 1$	$P(1-x-y+xy)$
$x+y = 1$	$P(1-x-y+2xy)$
$x \leq y$	$P(x-xy)$
$x_1+x_2+x_3 \leq 1$	$P(x_1x_2+x_1x_3+x_2x_3)$
$x = y$	$P(x+y-2xy)$

Table 5.1: Common binary variables' constraints as penalties for QUBO models.

The scalar penalty P can take different values according to the constraint's nature or im-

portance in the context of the problem, which means that the domain knowledge is essential when making the choice for a particular problem. Although the same penalty can be used for different constraints, if there is some kind of priority or demand on fulfilling a constraint over other constraints owing to specific circumstances of the problem, the penalty parameter P can be set accordingly to represent this necessity.

Nevertheless, a large parameter P value can affect the search for the solution, since the objective function is too affected for not satisfying a constraint, hindering the distinction of the quality between solutions. If the opposite occurs, and the parameter P is set to a small value, the optimizer might have difficulties to discern feasible solutions from unfeasible ones.

5.3 A general purpose approach to create QUBO models

Certain CO problems are difficult to directly represent as QUBO models or the penalties needed to satisfy the constraints are not known from the start. In these cases a more general methodology is employed to create the QUBO model corresponding to that CO problem. With the following procedure it is possible to find the penalties of the CO problem that have to be introduced in the objective function to ensure that feasible solutions are obtained in the process. TO that end, the following optimization problem of binary variables is considered:

$$\min y = x^t C x \tag{5.2}$$

$$Ax = b, x \in \{0,1\} \tag{5.3}$$

This model contains both linear and quadratic functions. The linear case particularly corresponds to a diagonal C matrix, owing to the binary nature of the variables ($x_i = x_i^2$ if $x \in \{0,1\}$). When stumbling into a problem with inequality constraints, considering always that A and b are both composed of integer components, the inequalities can be represented by adding slack variables and representing them as a binary expansion.

For instance, given the following constraint of binary variables $x_1 + 3x_2 - 2x_3 \leq 3$ a slack variable is introduced in the inequality $x_1 + 3x_2 - 2x_3 + s = 3$ whose boundaries $s \leq 5$ can be represented by the binary expansion $s_1 + s_2 + 3s_3$ where $s_1, s_2, s_3 \in \{0,1\}$. These

constrained quadratic optimization models are transformed into QUBO models (unconstrained) by introducing the constraints $Ax = b$ as penalties into the objective function, as explained in Section 5.2. Particularly, the quadratic penalty $P(Ax-b)^t(Ax-b)$ is added into the objective function resulting in the following model:

$$y = x^t Cx + P(Ax-b)^t(Ax-b) \tag{5.4}$$
$$= x^t Cx + x^t Dx + c \tag{5.5}$$
$$= x^t Qx + c \tag{5.6}$$

The matrix D and the additive constant c are the result of the matrix multiplication of the penalty terms. Leaving the additive constant c aside, the now unconstrained QUBO problem of the originally constrained problem is:

$$QUBO : \min x^t Qx,\ x \in \{0,1\} \tag{5.7}$$

5.4 QAP's background

Quadratic Assignment Problem (QAP) is one of the most notable CO problems in the field of operation research in mathematics with many applications showcased in [Commander, 2005]. It falls into the category of facilities location and it was introduced by Koopmans and Beckmann in [Koopmans and Beckmann, 1957]. The optimization problem is based on dealing with economic activity that has to be regulated in the optimal way to manage the resources available. The optimization problem is the following:

There are n different locations on a map with a distance between each one of them known and defined by the $d_{i,j}$ parameter for each pair of locations i and j. Meanwhile another n facilities have been selected and must be placed in different locations of the map, so that all the locations have one facility assigned. Between facilities, there is a flow that determines how much traffic is generated for each pair (i, j) of facilities and specified by the parameter $w_{i,j}$. The goal is to assign the facilities to different locations so that the sum of the multiplication of the flows by distance for each pair of locations is minimized.

The objective function gives an important first glance on how could the assignment work as efficient as possible, which is by grouping the pairs of facilities with high flow in close

locations. This should help specially when the distribution of the flows has high variance, so that this principle should hold. Nevertheless, the opposite can also happen, namely, that the distribution of the flows could have very little variance, making the search much more intricate in return.

The crucial step to bring QAP to quantum computation and use the quantum annealing methodology lies in a mathematical reformulation of combinatorial optimization problems known as Quadratic Unconstrained Binary Optimization (QUBO). In consequence, the QAP is transformed into a QUBO model that can be solved efficiently with QUBO solution methods to obtain high quality solutions to the problem (even if not optimal) in a relatively brief amount of time.

5.5 Formal definition of the QAP

Formally, the quadratic assignment problem consists on being given two sets of elements F for facilities and L for locations of equal length, and given a weight function $W : F \times F \to \mathbb{R}$ that represents the flow between facilities and a distance function $D : L \times L \to \mathbb{R}$ that determines the distance between two locations. The main purpose is to decide the best possible assignment $A : F \to L$ so that the objective function is minimized:

$$\min_{A \in S_n} \sum_{x,y \in F} W(x,y)\, D(A(x),A(y)) \tag{5.8}$$

The best assignment A from all possible permutations of n locations or facilities S_n is searched. Furthermore, one facility is only assigned to one location and all locations must have one facility assigned, as expressed with the next assertion with the Kronecker delta function:

$$\forall k \in L \left(\sum_{x \in F} \delta(A(x),k) = 1 \right) \tag{5.9}$$

Generally, the weight function and distance function are given in matrix form, W and D respectively. Another addition to the problem is to consider another weight function $C : F \times L \to \mathbb{R}$ that represents the cost of assigning a facility from F to a location from L, which can also be represented as a matrix. With this new incorporation, the objec-

tive function slightly changes to consider also the cost of performing the corresponding assignment:

$$\min_{A \in S_n} \sum_{x,y \in F} W(x,y)\, D(A(x),A(y)) + \sum_{x \in F} C(x,A(x)) \tag{5.10}$$

Another perspective to view the QAP is through the traces of the matrices W and D. For that end, a squared permutation matrix $X = [x_{i,j}]$ of size n is used where every element fulfills that

$$x_{i,j} = \begin{cases} 1 & \text{if } A(i) = j \\ 0 & \text{if } A(i) \neq j \end{cases} \tag{5.11}$$

and, therefore, if a facility i is in location j, $x_{i,j} = 1$; otherwise, $x_{i,j} = 0$. The trace formulation is stated as follows:

$$\min_{X \in \Pi_n} Tr(WXD^tX^t) \tag{5.12}$$

where Π_n is the set of squared matrix permutations of size n that stick to the constraints of the permutation matrix X.

5.6 Representing the QAP as a QUBO model

The QAP can be represented as a QUBO model to solve this challenging problem with quantum annealing. Since the variables are binary, the objective function will be slightly tweaked into an equivalent form. The flow function or traffic between i and j is represented by the constant f_{ij}, and similarly, the distance between location i and j is represented by d_{ij}. The objective is to find the best location for each facility, so that the flow between facilities is minimized based on distance, as explained before. The decision variables x_{ij} corresponds to facility i being assigned to location j, which means that $x_{ij} = 1$ if facility i is assigned to location j and $x_{ij} = 0$ otherwise. With all this settled, the QAP can be formulated as:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij}d_{kl}x_{ik}x_{jl} \tag{5.13}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j \in S_n \tag{5.14}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i \in S_n \tag{5.15}$$

$$x_{ij} \in \{0,1\} \quad i,j \in S_n \tag{5.16}$$

In practical applications, QAP problems generate large models as a direct consequence of harboring n^2 variables. As it can be observed in Section 5.3, the model has the same general structure required for the transformation into a QUBO model and, therefore, any QAP problem can be converted to an equivalent QUBO model.

In order to observe how it can be reformulated as a QUBO model, consider the following example for $n = 3$ where the flow matrix and distance matrix are given as follows:

$$F = \begin{bmatrix} 0 & 5 & 2 \\ 5 & 0 & 3 \\ 2 & 3 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 8 & 15 \\ 8 & 0 & 13 \\ 15 & 13 & 0 \end{bmatrix} \tag{5.17}$$

With flow and distance matrices the QAP model turns into:

$$\begin{aligned} \min y = {} & 80x_{11}x_{22} + 150x_{11}x_{23} + 32x_{11}x_{32} + 60x_{11}x_{33} + 80x_{12}x_{21} + 130x_{12}x_{23} + 60x_{12}x_{31} \\ & + 52x_{12}x_{33} + 150x_{13}x_{21} + 130x_{13}x_{22} + 60x_{13}x_{31} + 52x_{13}x_{32} + 48x_{21}x_{32} + 90x_{21}x_{33} \\ & + 78x_{22}x_{33} + 78x_{23}x_{32} \end{aligned} \tag{5.18}$$

With constraints:

$$\begin{aligned}
x_{11}+x_{12}+x_{13} &= 1\\
x_{21}+x_{22}+x_{23} &= 1\\
x_{31}+x_{32}+x_{33} &= 1\\
x_{11}+x_{21}+x_{31} &= 1\\
x_{12}+x_{22}+x_{32} &= 1\\
x_{13}+x_{23}+x_{33} &= 1
\end{aligned} \tag{5.19}$$

Introducing the constraints as quadratic penalties into the objective function, the unconstrained quadratic model is achieved:

$$\begin{aligned}
\min y = {} & 80x_{11}x_{22}+150x_{11}x_{23}+32x_{11}x_{32}+60x_{11}x_{33}+80x_{12}x_{21}+130x_{12}x_{23}+60x_{12}x_{31}\\
& +52x_{12}x_{33}+150x_{13}x_{21}+130x_{13}x_{22}+60x_{13}x_{31}+52x_{13}x_{32}+48x_{21}x_{32}+90x_{21}x_{33}\\
& +78x_{22}x_{33}+78x_{23}x_{32}+P(x_{11}+x_{12}+x_{13}-1)^2+P(x_{21}+x_{22}+x_{23}-1)^2\\
& +P(x_{31}+x_{32}+x_{33}-1)^2+P(x_{11}+x_{21}+x_{31}-1)^2+P(x_{12}+x_{22}+x_{32}-1)^2\\
& +P(x_{13}+x_{23}+x_{33}-1)^2
\end{aligned} \tag{5.20}$$

Choosing an appropriate penalty value of P=200, the problem is successfully transformed into the standard QUBO problem

$$\text{QUBO} : \min y = x^t Q x \tag{5.21}$$

with the additive constant being 1200 and the Q matrix being the following:

$$Q = \begin{bmatrix}
-400 & 200 & 200 & 200 & 40 & 75 & 200 & 16 & 30\\
200 & -400 & 200 & 40 & 200 & 65 & 16 & 200 & 26\\
200 & 200 & -400 & 75 & 65 & 200 & 30 & 26 & 200\\
200 & 40 & 75 & -400 & 200 & 200 & 200 & 24 & 45\\
40 & 200 & 65 & 200 & -400 & 200 & 24 & 200 & 39\\
75 & 65 & 200 & 200 & 200 & -400 & 45 & 39 & 200\\
200 & 16 & 30 & 200 & 24 & 45 & -400 & 200 & 200\\
16 & 200 & 26 & 24 & 200 & 39 & 200 & -400 & 200\\
30 & 26 & 200 & 45 & 39 & 200 & 200 & 200 & -400
\end{bmatrix} \tag{5.22}$$

The solution to the QUBO model is $y = -982$ with $x_{11} = x_{22} = x_{33} = 1$. The original objective function can be obtained by adding the additive constant of 1200 to the result $1200 - 982 = 218$.

6. CHAPTER

Implementing the QUBO approach for the QAP

Once having established the steps and the knowledge required to understand how to elaborate an implementation of the QUBO approach for the QAP, it is time to cover the details and important aspects related to the constraints and capacity of the physical machines running the program. At the moment, the QPU powered quantum computers are labeled as D-Wave 2000Q QPU, which essentially means that the quantum computer is built with 2000 qubits in its circuits.

In order to communicate with these quantum computers, D-Wave's Ocean Software[1] has been used. This allows to send a QUBO, build binary quadratic models immediately and even tweak the configuration of the request with several important parameters. The implementation has been uploaded to a repository[2] for public access.

Instances of the problem have been extracted from QAPLIB[3], a library of information and research around the QAP problem. Therefore, the instances of the problem are real scenarios in which researchers had to make use of classical approaches to solve it, unlike in this one which is centered around quantum annealing.

Although, the transformation of the standard QAP model to the QUBO model has already been explained thoroughly and, as a result, the implementation is quite straightforward, it can't be said the same about the whole communication with the server and how the QUBO model is exactly processed. The underlying question is to determine how to fit the

[1] https://readthedocs.org/projects/dwave-systemdocs/downloads/pdf/latest/
[2] https://github.com/Totx/QuantumQAP
[3] http://anjos.mgi.polymtl.ca/qaplib/

QUBO model into the physical QPU which is known as the embedding of the problem into the QPU.

6.1 Topologies in the D-Wave QPU architecture

A CO problem, even in an equivalent QUBO model of the problem, is still not embedded into the QPU. The mapping between the original QUBO model and the QPU is called the embedding of the problem. In order to perform this, the binary quadratic problem is represented as a graph, which will later be mapped into the graph of the QPU. These abstract graph structures of the QPU are the topologies of the system, which have physical limitations and constraints. It is a necessary step to find the embedding to solve the problem, since, otherwise, the problem can't be processed by the QPU.

Even if the D-Wave's Ocean software can already create the embedding automatically, without proper understanding of the implications of such decisions, namely, the original graph's size and the quality of the solution obtained; unexpected results might be returned. All of this is directly related to how the topologies are built around and the interaction between qubits.

D-Wave's Quantum Process Unit (QPU) can be represented as a grid of interconnected qubits. The qubits are connected to one another with couplers. However not every qubit is connected with every other qubit, which arises an important issue, that is to say, D-Wave's QPU's qubits are not fully connected. The topology followed by the D-Wave 2000Q QPU and previous generations is the Chimera topology and QPUs modeled with a more advanced topology follow the Pegasus topology.

6.1.1 Chimera topology

The qubits are organized in groups of eight qubits that form one unit cell all together. In each unit cell, there are four vertical qubits and another four horizontal qubits. The QPU is composed of a matrix of unit cell blocks. For instance, the qubits interact as can be noted in Figure 6.1 for four unit cells.

In a Chimera topology two main coupler types can be highlighted to understand the underlying graph of the topology better:

- Internal couplers: Internal couplers connect pairs of orthogonal qubits among them.

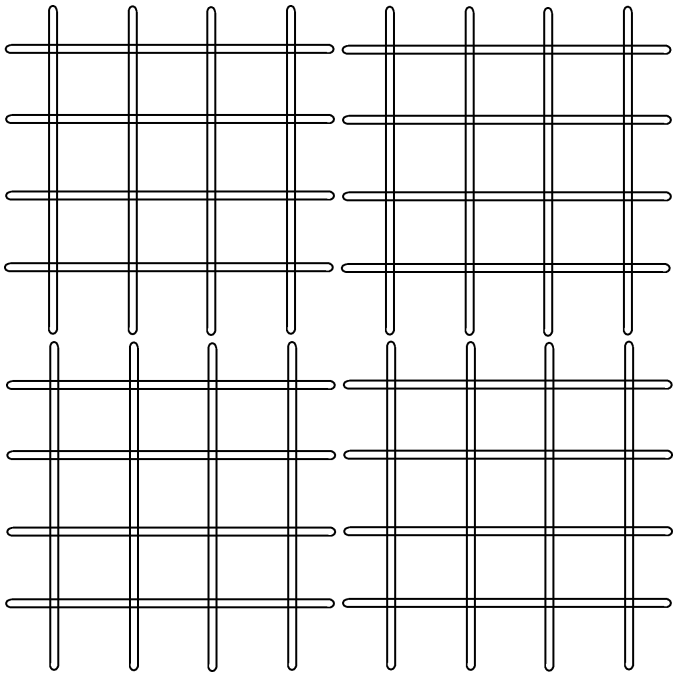

Figure 6.1: Subset of four unit cells of a QPU with Chimera topology

They enable connections between qubits inside each unit cell (interconnected four vertical qubits with other four horizontal qubits), hence they are considered to be internal as shown in Figure 6.2. The unit cell structure with the internal couplers corresponds to a bipartite graph of four nodes in each side, as show in Figure 6.3.

- External couplers: External couplers connect parallel qubits from different unit cells. This includes both row qubits and column qubits as shown in Figure 6.4.

In a Chimera topology, from what is being explained above, these two conditions are met:

- Nominal length: Internal couplers connect each qubit to other four orthogonal qubits inside an unit cell.
- Degree: Each qubit is coupled to other six qubits in total, four internal and two external.

A common notation to refer a Chimera topology of $n \times n$ unit cells is CN. For the D-Wave 2000Q QPU, a Chimera graph of 16 unit cells can be fitted, therefore the QPU is capable of embedding a C16 Chimera graph. As a result, the QPU's qubits are mapped to a 16×16 matrix where each element corresponds to 8 qubits.

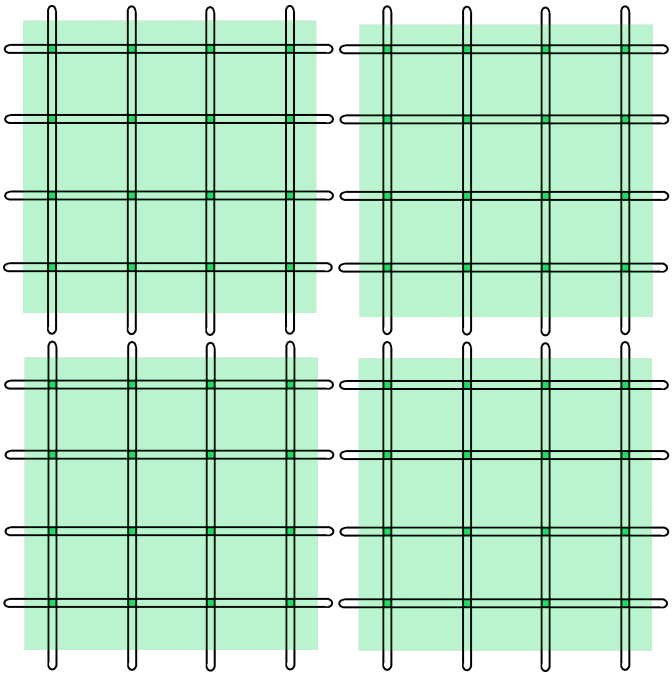

Figure 6.2: Internal couplers highlighted in deep green and the unit cell module block in light green covering all the internal couplers

6.1.2 Pegasus topology

In this more sophisticated topology, the orientation of the qubits is still limited to either vertical or horizontal, nevertheless, the alignment of unit cells is no longer satisfied. A cut part of the topology is shown in Figure 6.5.

The Pegasus topology's couplers belong to three different classes, that is to say, they are internal, external or odd.

- Internal couplers: Internal couplers connect orthogonal qubits between them, so that each qubit is internally coupled with other twelve qubits as shown in Figure 6.6.
- External couplers: With external couplers, qubits in the same orientation (both vertical or horizontal) are coupled with qubits with same orientation as can be observed in Figure 6.7.
- Odd couplers: Odd couplers connect qubits with similar alignment but not in the same parallel line as shown in Figure 6.8.

Pegasus topology has a nominal length of 12 (each qubit is connected to other 12 qubits

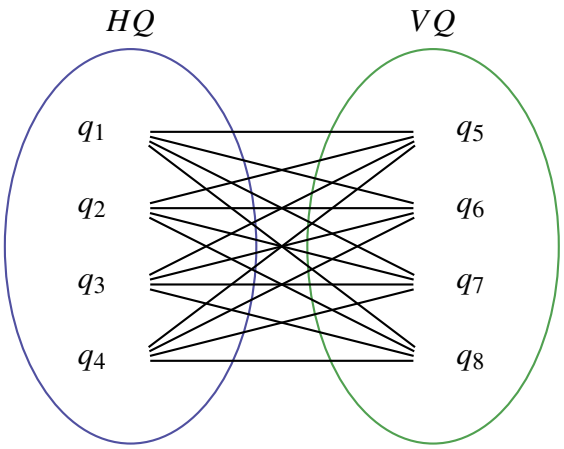

Figure 6.3: A graph representation of an unit cell with a bipartite graph. HQ stands for horizontal qubits and VQ stands for vertical qubits.

with internal couplers) and a degree of 15. It is a common notation, as it was the case for the Chimera graph (CN), to refer to a Pegasus graph of size N as P_n.

6.2 Chains and minor embeddings

The original QUBO model needs to be translated to the qubits and couplers in the QPU to apply quantum annealing. However, each of the logical qubits (decision variables of the binary quadratic unconstrained problem) could be represented by one or more physical qubits in order to fit the original graph into the topology's graph, which in this case corresponds to the Chimera graph. The process through which logical qubits are assigned one or more physical qubits is known as minor embedding.

This is a crucial part to set the boundaries of the current limit in quantum computing. If a QUBO problem can't be embedded into the QPU, the quantum annealing can't be carried out for that instance of the problem. Particularly, the limit in this case is the amount of qubits of the QPU (approximately 2000 qubits) and the particular density of the graph which correlates to the amount of edges of the QUBO model's graph. If more variables are connected with each other with a non-null weight, the graph will be more dense and the minor embedding will require to consider all those connections. The consequence is that even if two instances of the problems have the same amount of nodes, if their density is different, the lower dense graph might be embedded into the QPU, but not the higher dense graph.

On the other hand, chains are the group of physical qubits that correspond to one logical

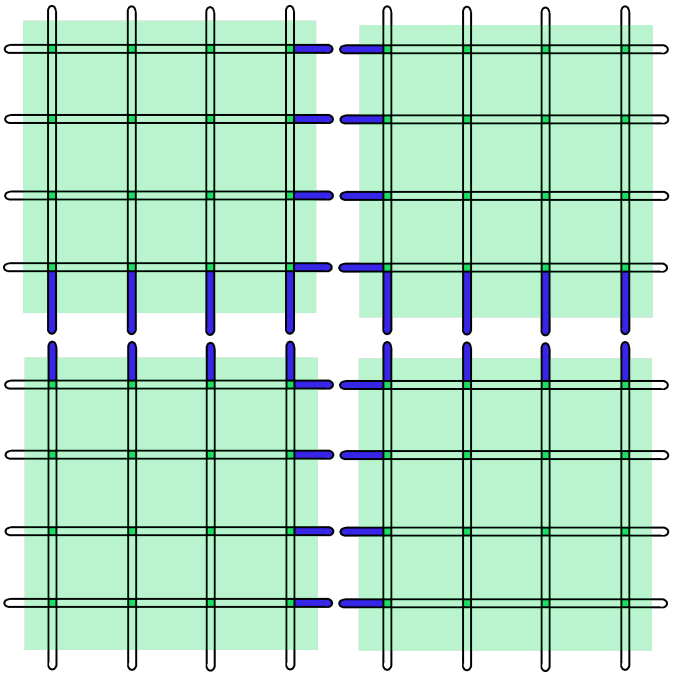

Figure 6.4: External couplers indicated with blue color connect adjacent horizontal and vertical qubits in parallel with each other.

qubit. Therefore, the strength of a chain, which indicates how strongly related are two physical qubits, has to be strong enough to ensure that the final value of these two physical qubits is the same, since they correspond to the same logical qubit. Adjusting the chain strength accordingly is necessary to prevent a chain break, which occurs when a chain of physical qubits disagree at the end, falling into different values despite referring to the same logical qubit.

For example, consider the graph on the left as the original model's graph that needs to be embedded into the Chimera graph on the right in Figure 6.9. In order to translate the closed loop to the Chimera graph a chain that joins two different physical qubits as one logical qubit will be created. The assignment of the physical qubits to logical qubits results in the minor embedded graph in Figure 6.10.

The next step is to set up the biases and strength relations between physical qubits accordingly. Aforementioned qubits belonging to a chain require to have strong relationship so that the chain doesn't break and there is no disagreement at the end as a result. With the knowledge of which physical qubits correspond to a logical qubit and the biases of the original graph, the relationships and biases to be determined correspond to the physical qubits q_0 and q_5. Table 6.1 below describes the current situation.

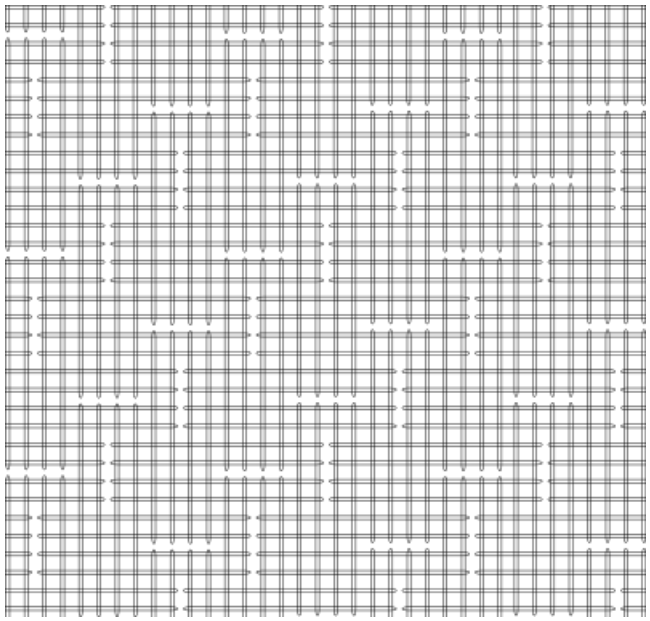

Figure 6.5: A trimmed view of the pegasus topology. The qubits aren't aligned anymore and are stretched.

Qubits	q_0	q_5	q_4	q_1
Variable	B	B	A	C
Bias			-1	-1

Coupler	(0,4)	(5,0)	(4,1)	(1,5)
Strength	2		2	2

Table 6.1: The minor embedding information before considering the chain implications.

For the qubits q_0 and q_5 forming a chain further readjustments must be carried out. Since both qubits represent one logical qubit, they need a strong negative coupler strength between them. The other coupling strength also need to be considered and modified to compensate for adding a negative coupling that wasn't present in first place. The process follows the next steps:

- The bias of the variable B is split equally for each of the qubits forming a chain. In this case the bias -1 of the variable B becomes -0.5 for q_0 and q_5.
- A strong coupling strength for the qubits q_0 and q_5 forming a chain must be chosen. For this particular case, a chain strength of -3 ensures that both qubits will fall into the same value at the end, since the coupling strength is higher than any other and is also negative.
- The compensation for adding a coupling strength of -3 in the previous step is to add

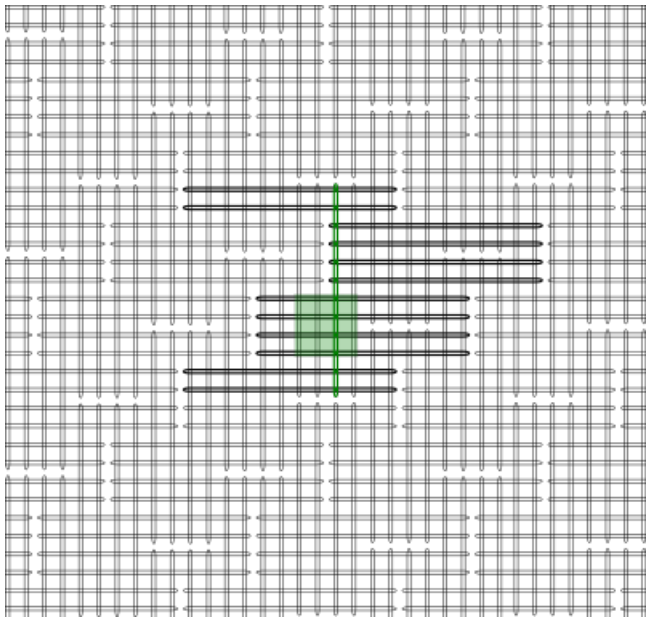

Figure 6.6: A vertical qubit (highlighted in green) coupled with other 12 horizontal qubits in bold. A Chimera unit cell is represented covering it with translucent green square.

1.5 to each bias of the chain, in this case to q_0 and q_5. The bias for the qubits in the chain is now 1.

- The final step involves normalizing the graph to fit the values within the physical boundaries of the system, which is $[-1, 1]$. Therefore, all values are divided by 3.

The result of the minor embedding can be seen in Table 6.2 where chained qubits have a strong negative coupling. The problem is now ready to be sent to the QPU.

Qubits	q_0	q_5	q_4	q_1
Variable	B	B	A	C
Bias	0.33	0.33	-0.33	-0.33

Coupler	(0,4)	(5,0)	(4,1)	(1,5)
Strength	0.667	-1	0.667	0.667

Table 6.2: The minor embedding after the chains have been properly set with a chain strength.

Some reflection about choosing a suitable chain strength can be viewed as follows: if the chain strength were to be too large, the coupling strength and biases of the rest of the qubits could be heavily reduced in the final step to the point where they become insignificant and hard to discern; meanwhile, a small value for the chain strength could cause chain breaks where the qubits will disagree and ambiguity is introduced in the solution. In consequence, the coupling strength of a chain has to be relatively higher than

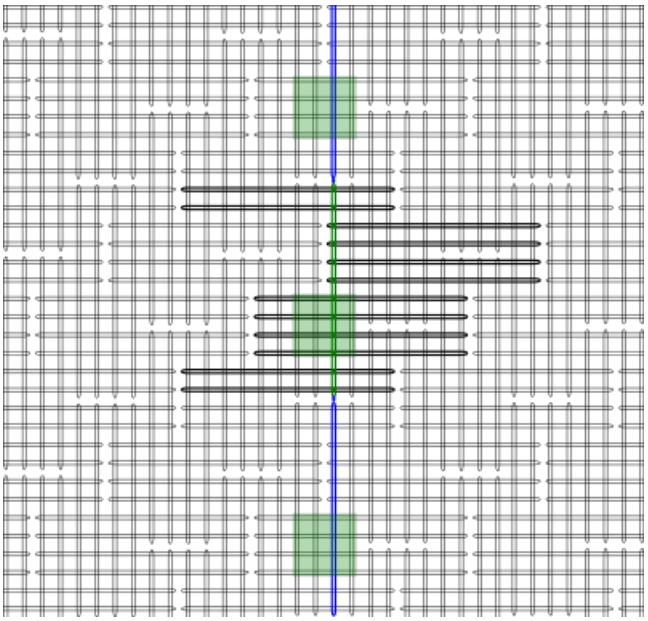

Figure 6.7: A vertical qubit (green qubit) is coupled with two similarly aligned qubits (blue qubits) with external couplers.

any other coupling strength preventing chain breaks, but without exceeding it to the point where the other coupling strengths are too influenced.

Finally, the QPU is ready to solve and return a sample of solutions for the problem. However, these solutions need to be unembedded from the physical qubit values to logical ones. In case a chain break occurs the D-Wave software handles the chain break fixing the different values of qubits in a chain to one only value. For the example presented in this section, one possible solution could be

$$q_0 = 1 \tag{6.1}$$
$$q_1 = 0 \tag{6.2}$$
$$q_4 = 0 \tag{6.3}$$
$$q_5 = 1 \tag{6.4}$$
$$\tag{6.5}$$

whose reverted values to the logical qubits are

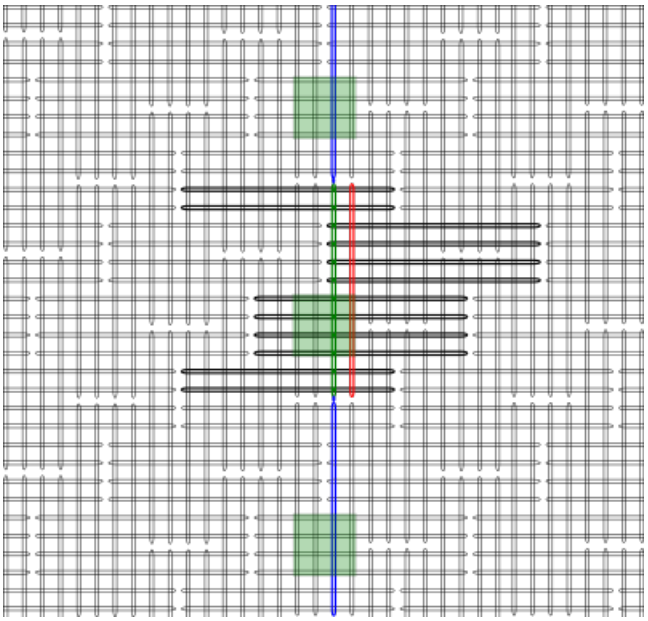

Figure 6.8: Odd couplers connecting the green qubit with the red qubit. The alignment is the same but they are not in the same vertical line.

$$A = q_4 = 0 \tag{6.6}$$

$$B = q_0 = q_5 = 1 \tag{6.7}$$

$$C = q_1 = 0 \tag{6.8}$$

$$\tag{6.9}$$

6.3 Experiments

The experiments carried out consist on creating a workflow where a instance of QAPLIB is read, transformed into a QUBO model, sent to the QPU and received for post processing of the sample solutions.

The Chimera topology to which a QUBO model needs to be adjusted has caused major problems owing to the current technical limitations. For a QAP problem of size n, the transformation of the standard constrained QAP model into the QUBO model raises the number of variables drastically from n variables to n^2 variables. In the experiments, one of the major setbacks has ocurred when trying to pass instances of the problem with too many variables or high density relationships.

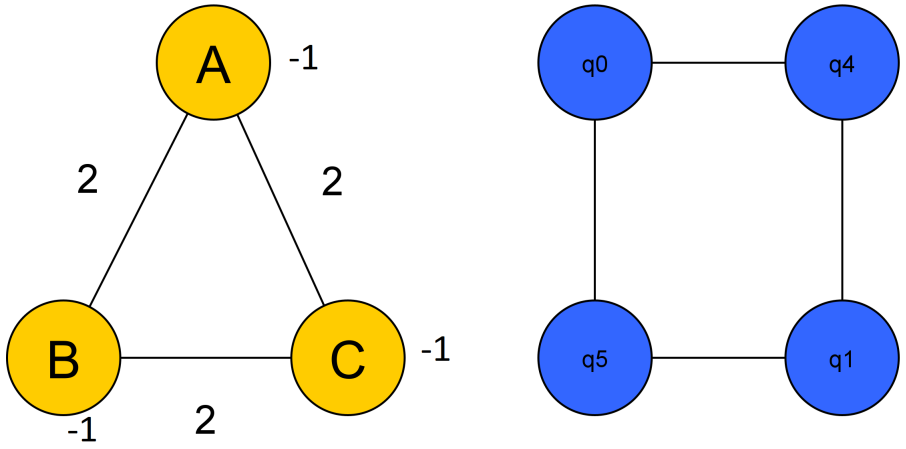

Figure 6.9: On the left, a QUBO model graph; on the right, the Chimera graph where it has to be minor embedded.

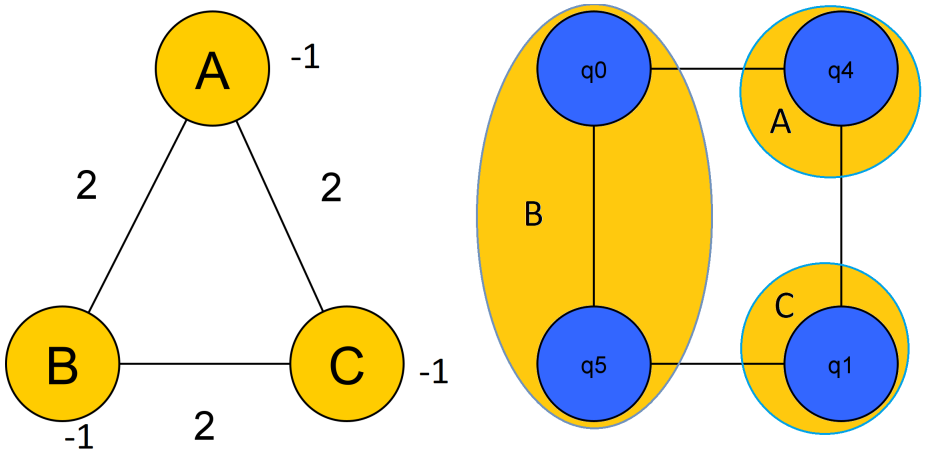

Figure 6.10: The minor embedding of the graph on the left to the Chimera graph on the right.

A closer inspection to the QPU reveals that the D-Wave 2000Q QPU can at most hold a Chimera graph for fully connected graphs of 65 nodes. This means that in the worst case scenario where an instance has high density, the maximum size of the QAP problem is 8 due to the transformation of the QAP model into the QUBO model increasing the number of variables to 64.

In order to circumvent the limitation of the current quantum computers, instead of trying to solve the entire instances of the QAPLIB library, a subset of eight facilities and eight locations will only be considered to ensure that an embedding of the problem into the QAP is actually possible. The subset corresponds to the first eight rows and columns of the QAP instances.

Another thing to consider in the results obtained is the quality of the solutions, since

the results are intended towards having good quality and being close to the optimal but they don't ensure it. To that end, the results obtained with the quantum annealing will be directly compared to those obtained by a QAP solver to analyze how good the solutions really are.

6.4 Results

The results obtained from quantum annealing are influenced by the chain strength chosen heavily as explained previously, as well as the Lagrange parameter P introduced as a penalty in the objective function when performing the transformation of the QAP model into the QUBO model. As a result, for each of the instances different combinations of chain strengths and Lagrange parameters will be considered that suit the instance the most.

After conducting different tests, six instances will be used where the chain strength is set to 30000 and the Lagrange parameter P is set to 12000. These values are the same for all the instances since the range of values in the matrices are similar, as well as the density of the graphs. Therefore, these two values balance the chain strength to prevent chain breaks and ambiguous solutions, while the penalty term contributes to force the constraints to be satisfied in order to sample feasible solutions. The results are gathered in the table 6.3.

Instance	Energy	Objective function	Chain break frequency	Best solution
'chr12a.dat'	-181345.0	10655.0	0.078125	[6, 0, 4, 7, 2, 3, 5, 1]
'chr15b.dat'	-180251.0	11749.0	0.0	[2, 4, 6, 0, 3, 7, 1, 5]
'chr12c.dat'	-181422.0	10578.0	0.109375	[7, 5, 1, 3, 6, 0, 4, 2]
'chr15c.dat'	-181543.0	10457.0	0.0625	[2, 0, 1, 4, 6, 7, 5, 3]
'chr12b.dat'	-176403.0	15597.0	0.09375	[2, 3, 6, 4, 5, 1, 7, 0]
'chr15a.dat'	-181529.0	10471.0	0.03125	[1, 4, 2, 6, 7, 0, 5, 3]

Table 6.3: Results obtained with a chain strength of 30000 and a Lagrange parameter P of 12000 for six instances from QAPLIB.

The energy corresponds to the lowest energy level found, the objective function is the energy added the additive constant (offset), the chain break frequency indicates how frequently qubits of the same chain disagree and the best solution is a permutation of the facilities over locations.

The amount of time available for use of the QPU is limited to one minute per user. Owing to this, fine tuning of the Lagrange parameter and chain strength has been reduced to

a moderate number of attempts, which in return, leaves room for improvement in this aspect. Once the token expires because the amount of available time is depleated, the QPU is no longer accessible.

A QAP solver has been used to contrast the results obtained with quantum annealing and assess the quality of the solutions by comparing them to the best solutions found by traditional means. The QAP solver is a branch and bound algorithm, a heuristic approach to find solutions for the QAP. These have been the results obtained from the QAP solver, shown in Table 6.4.

Instance	**Objective function**	**Best solution**
'chr12a.dat'	7638	[6, 4, 3, 1, 0, 2, 5, 7]
'chr15b.dat'	6974	[6, 4, 1, 0, 7, 3, 2, 5]
'chr12c.dat'	9078	[2, 4, 3, 6, 5, 0, 7, 1]
'chr15c.dat'	5142	[1, 7, 6, 4, 0, 2, 3, 5]
'chr12b.dat'	10556	[6, 4, 3, 2, 1, 0, 5, 7]
'chr15a.dat'	6672	[6, 4, 0, 1, 7, 5, 3, 2]

Table 6.4: QAP solver results using the branch and bound algorithm.

Comparing the quantum annealing results with the branch and bound algorithm results, the results obtained with quantum annealing look quite promising. The instances with $n = 12$ produce closer results than those with $n = 15$, despite only using a subset of the first eight rows and columns in either case. Moreover, the permutations seem completely different between them for every case, which could be a sign of falling into a local minimum.

7. CHAPTER

Project management

The planning of the project through its different stages and possible setbacks is a necessary step in order to identify the risks involved with the project and the potential phases of the development, and to manage the most valuable resource in a large scale project of this sort, that is, time. The objective is to expose every aspect of the project of significance that must be committed in due time with deadlines in check and to find how to efficiently distribute the tasks in that regard, so that potential setbacks and delays can be prevented and the project can be successfully completed in time.

The management work will be divided in three main categories, namely: Project management, project development and documentation of the project. These three main phases cover the major aspects and the crucial decisions, as well as how difficulties have been confronted through the entire process. The modules of each phase constitute its focus points that have to be dealt with for the most adequate transition and evolution towards milestones and goals.

7.1 Description of the phases and their features

The different phases of the project have been gathered here to show which has been the dynamic of the project, how each part alone provides useful guidance and, all in all, a watchful observation about how the entire process works as a whole.

7.1.1 Management phase

The principal focus in this phase is to estimate the cost of different tasks and how the project could potentially evolve across its duration. Since this is an early phase of the project, estimating the time, resources and potential risks is of huge importance to be able to track afterwards if the project is going according to plan or there are some (major) deviations with regard to how it was planned in the beginning.

After choosing the specific tasks that have to be carried out, the most volatile part consists on placing them in the correct moment and for the right amount of time. Additionally, the project has to be tracked periodically to ensure that the milestones set for the project are being met or if there is need of some readjustments that allow to better handle tasks for the best possible outcome.

There are three main modules that define the management of the project to take into account:

- **Planning**: The main points of the project have been covered estimating their possible duration and the resources needed for their realization, as well as searching for when these objectives can be fulfilled and the order in which they are completed. The result is a set of activities ordered and placed across the time with their respective milestones and with a risk management plan to be ready for risks with prior knowledge on how to avoid them. Finally, the scope of the project is determined.

- **Tracking**: In order to check whether the objectives are being completed under the given time and to counter the present risks that can cause unexpected delays, the project is analyzed during its progression. Finding new risks, modifying the milestones and creating new objectives or replacing older ones is the main purpose.

- **Communication**: This parts joins the previous two modules and an assessment is conducted to evaluate how is the project progressing and to identify which tasks are going according to plan and which others are not. All of this is explained in detail to the directors of the project so that they can be up to date and informed of any kind of alterations in the plans. This issues are handled in periodical meetings when certain milestones have been finished or when a new risk has emerged. In the end, the tasks until the next meeting are decided, which are usually intentions for a short term.

7.1.2 Development phase

The project is heavily oriented towards learning the in and outs of quantum computing and how to apply quantum annealing to solve Combinatorial Optimization (CO) problems with a focus in the QAP problem. Most of the development phase is centered on learning and understanding the fundamentals of quantum computing, the main difficulties and complexity of CO problem with focus on QAP, grasping how the quantum annealing can be used to solve CO problems, following the steps of the methodology to transform the CO problem into a QUBO and managing the different topologies the QPU works with.

Once all of this is studied, the practical application that includes all those steps for the QAP is developed to see in action all the process in a direct communication with a D-Wave quantum computer and the results obtained are compared with traditional approaches.

7.1.3 Documentation of the project

The last part involves the report of the project, where the most relevant knowledge and information is gathered about the research conducted, both for the theoretical part and for the practical application. Since quantum computing requires a good background, the document relies on specific theoretical details and provides examples that further help to view how it actually works.

The amount of research in this phase is quite long to ensure that every aspect of the methodologies working with quantum annealing can be fully understood and to provide enriching and valuable insight about quantum computing in the process.

7.2 Estimations

After covering the three phases that compose the project, the initial estimation of time and the finally needed amount of time to complete the tasks will be displayed. In bold, the estimation for a phase is registered, while below, for each phase, each specific task of the project is broken down belonging to that phase. The estimated time and the final time of each task are summed for finding out how much time each phase has required in table 7.1.

	Estimated time	Final time
Management phase	65	67
Planning	30	25
Tracking	20	30
Communication	15	12
Developement phase	40	59
Managing QAP instances from QAPLIB	10	8
QUBO transformation	15	25
Quantum annealing in D-Wave environment	5	20
Comparison with classical QAP solvers	10	6
Documentation phase	230	221
Study basic mathematical foundation for quantum computing	60	55
Analyze the leap from classical computing to quantum computing	30	28
Learn about the four postulates of quantum computing	50	40
Acquire background about quantum annealing in D-Wave systems	30	25
Research on QAP	20	15
Learn how to transform CO problems into QUBO	40	30
Find out the topologies for the embeddings in a QPU	0	28
Total amount of time	335	347

Table 7.1: Estimation of tasks and their final required time.

7.3 Deviations

The deviations occurred have altered the initial planning, requiring some modifications in the task and even including a new task that wasn't intended in the first place. Some other minor deviations are also addressed.

The major deviation corresponds to finding out the limitation of the QPU to achieve a minor embedding of a QAP instance of size $n > 8$ and therefore heavily reducing the possibility to apply the methodology completely to a real world problem. This discover lead to gathering information about how the minor embedding worked and the topologies available in the D-Wave's QPU to justify the newly found setback.

Moreover, since QAPLIB instances couldn't be completely executed (only a subset of the instance), some extra research was carried out in the hope of finding a way to either manually minor embedding an instance into the QPU or look for an alternative that would allow for a hybrid implementation to compensate the extra size of the problem. Nevertheless, those alternatives were ultimately discarded for not being a generalized method in the first case and for straying too far from the focus in quantum computing for the second case.

Minor deviations come mainly for other tasks in the documentation phase that required more time for the extensive amount of information they included. They were quite distanced from the time consumption thought in the planning, a direct consequence of the inexperience in the beginning of the project around quantum computing.

7.4 Risk management plan

Despite identifying risks at the beginning of the project, unexpected events can cause deviations and negative impact on the development of the project. Therefore, the risk management plan has been a progressive plan to prevent risks, including new ones, swapping other ones or discarding some others. Therefore, the risk management plan has evolved to adapt to each moment. Particularly, the risks haven been identified in the tracking process and later added or modified after a meeting with the directors to discuss the potential consequences and how to prevent them.

One of the major risks of this project has been the quarantine derived from the COVID disease spreading rapidly, which caused certain physical resources to be out of reach and direct communication being cut. In order to address these issues, documentation from unavailable sources was delayed until gaining access again and change in the order in which certain tasks would be developed was introduced. The communication has been online through voice chat.

The risk management plan also considered the vast amount of background necessary for the project as a risk, and the estimated time for the documentation was increased in order to prevent unexpected deviations. However, even with that in mind, it took even longer, which required to adjust tasks in the management phase to reconsider the amount of time they would require.

8. CHAPTER

Conclusions

Despite the promising efforts and improvement over the years of the quantum computing field of research, CO problems are still a challenge due to their complexity and current limitations of the QPU. In particular, the high dimensionality of a QUBO model based on a QAP problem drags the minor embedding process to a point where a Chimera graph corresponding to that QUBO model far exceeds the capacity of current QPU.

On the other side, however, these are but the first steps and methodologies tested using quantum annealing that are still far from finished and developed. In the near future, moreover, D-Wave systems has intention of releasing the D-Wave 5000Q QPU which will hardly be their last leap in increasing the qubit number to be able to minor embed more complex and denser problems.

Another aspect to consider is that the Chimera topology used is quite old already and that there are already alternatives, such as the Pegasus topology, that not only change the layout of the qubits but also add new couplers and connect more qubits than before. The contribution of different topologies relies on the possibility to harbor much denser and complex structures.

In the practical setting of the case study of this project, the QAP problem, the results of the experiments with quantum annealing approximate adequately those obtained with a branch and bound algorithm. However, while both rely on heuristics, for the quantum annealing part, the QUBO model's Lagrange parameter and chain strength are two dependant factors of the methodology that can drastically impact the result. In consequence, in order to approximate the best solution for a QAP instance, a fine tuning of these parame-

ters is of huge importance.

All these breakthroughs will surely improve the current methods and give birth to new ones that have the potential to transform the classical means through which the current problems are tackled with hybrid methods that combine both traditional and quantum approaches alike. Therefore, the evolution of the quantum technology combined with new topologies and methodologies are sure to pave the way towards new heights.

9. CHAPTER

Future work

The objectives of the project can be further extended to not only focus in the QAP problem, but also create a library that solves a set of CO problems using quantum annealing and the QUBO approach. In this way, producing good quality results with quantum annealing for a set of CO problems could be implemented.

Apart from this, another extension to the project could be testing CO problems with the Pegasus topology as well as with the new QPUs that will be available in the future. As a result, the project can be extrapolated to other topologies and environments, while QUBO models still remain at its core.

As a final note, a better parameter optimization without the limit of only one minute use of the QPU could be devised. The instances and their respective Lagrange parameter and chain strength could be adjusted accordingly for yielding better results while maintaining the feasibility.

Bibliography

[D-W,] D-Wave systems documentation. https://docs.dwavesys.com/docs/latest/index.html.

[Ausiello et al., 2012] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., and Protasi, M. (2012). *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media.

[Burkard et al., 1997] Burkard, R. E., Karisch, S. E., and Rendl, F. (1997). Qaplib–a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403.

[Commander, 2005] Commander, C. W. (2005). A survey of the quadratic assignment problem, with applications.

[Kaye et al., 2007] Kaye, P., Laflamme, R., Mosca, M., et al. (2007). *An introduction to quantum computing*. Oxford university press.

[Koopmans and Beckmann, 1957] Koopmans, T. C. and Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, pages 53–76.

[Lawler, 1963] Lawler, E. L. (1963). The quadratic assignment problem. *Management science*, 9(4):586–599.

[Lewis and Glover, 2017] Lewis, M. and Glover, F. (2017). Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. *Networks*, 70(2):79–97.

[Williams, 2010] Williams, C. P. (2010). *Explorations in quantum computing*. Springer Science & Business Media.

[Yanofsky and Mannucci, 2008] Yanofsky, N. S. and Mannucci, M. A. (2008). *Quantum computing for computer scientists*. Cambridge University Press.

Glossary

CO Combinatorial Optimization. i, 2, 3, 5, 6, 39, 40, 42, 43, 50, 65, 69, 71, 74

QAP Quadratic Assignment Problem. i, iv, ix, 2, 3, 6, 43–47, 49, 50, 52, 54, 56, 58–61, 65, 66, 69, 71, 74

QPU Quantum Process Unit. iv, 49–51, 53, 56–61, 65, 66, 69, 71, 74

QUBO Quadratic Unconstrained Binary Optimization. i, iv, viii, ix, 2, 3, 5, 6, 39–50, 52–54, 56, 58–60, 65, 69, 71, 74