Industrial ecology and quantum computing

Potential applications in the field and case study

Roberto Manzanaro

Leiden university and TU Delft

TU Delft

Universiteit Leiden
The Netherlands

Industrial ecology and quantum computing

Potential applications in the field and case study

by

Roberto Manzanaro

Roberto	Student Number
Manzanaro	s2728680

First supervisor:	H. Wang
Second supervisor:	S. Cucurachi
Project Duration:	March, 2022 - August, 2022
Faculty:	LIACS and CML, Leiden

Cover: Foto by Pat Whelen in Unsplash
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Acknowledgements

After two years of studying industrial ecology (with the first year online), this journey is getting to an end, and what a ride it has been. After I finished my bachelor's in mathematics, I knew that I wanted to move towards sustainability which I was really passionate about. This master's has given me the perfect opportunity to make a big shift in my education and get introduced in this wonderful world. It has also allowed me to make use of my technical skills while learning many other valuable concepts, for which I am very grateful.

I would like to thank my supervisors for their constant support these last months. To Hao Wang and Xavi Bonet for their extensive knowledge on the topic and their willingness to answering the doubts and making me think. To Stefano Cucurachi for his guide and being an essential link to IE. Also, to my family for their constant support and interest in the thesis. To my colleagues for all the helpful study sessions where we kept each other motivated and supported each other.

Roberto Manzanaro
Delft, August 2022

Abstract

The activities of humans are increasingly influencing the Earth's systems. The climate is changing and the ecosystems are being affected by our actions. If we stay in the same path, the consequences could be catastrophic. For this reason, there is a growing focus on how to minimise our negative impacts on the planets and develop in a more sustainable manner. To aid in this objective, computers are a useful tool. Computational techniques can offer solutions to complex problems in sustainability, which often involve uncertainty, optimisation and decision making. However, the potential of classical computers is expected to reach a barrier, since the size of computer chips is reaching its physical limit. This requires new solutions that can provide better computational results in the future. A promising paradigm that is emerging is quantum computing, which works with quantum bits (or qubits) to make calculations and produce new solutions.

Given the potential of quantum computers, it is interesting to consider how they could help industrial ecology. Up to date, there have been no studies that focus on these two scientific disciplines together. For this reason, the first aim of this work is to identify what industrial ecology problems can benefit from quantum computing. The second aim is to provide a practical case study to illustrate how to work with quantum computers. In particular, the case study focuses on how to optimise vehicle routings in order to minimise emissions. Although routing problems have been extensively studied from the sustainability point of view, there are no such studies done on quantum computing yet. Overall, the main research question is *"How can quantum computing benefit industrial ecology and how can it be applied to a relevant problem?"*

Regarding the methodology, it is separated into different parts. Firstly, a literature review is conducted to identify possible applications for industrial ecology. This is done by considering relevant industrial ecology problems and investigating if any gains can be obtained by using quantum computers. The second part of the methodology focuses on the more practical side of quantum computing. It starts by benchmarking two state-of-the-art variational quantum algorithms (Rosalin and LCB CMA-ES) to test the performance and evaluate the produced results. Following the benchmarking, a green routing problem will be encoded and optimised using the mentioned algorithms. All the coding is done in Python with the help of Cirq and Openfermion, which are two packages for quantum computing developed by Google.

After following the methodology, several results were obtained. The literature review highlighted a series of problems that can benefit from quantum computing. Examples are the optimisation of flows in an industrial park and the management of data centres to reduce energy needs. Moreover, the review illustrated different techniques to tackle these problems, such as quantum machine learning and quantum information. Secondly, as for the benchmarking, both algorithms perform well in the test cases and there is no clear advantage of one over the other. One is slightly better for some test cases and the other slightly better in the other cases. In the logistics problem, the problem is encoded into a hamiltonian that is more complex than the test cases. To minimise the expectation of the hamiltonian, the previously mentioned algorithms are tested with different quantum circuits and the results are compared. LCB CMA-ES with circuit of depth 1 gives the best results, while the other LCB CMA-ES experiments do not do as well. This is unexpected because all the LCB CMA-ES experiments are able to optimise successfully the expectation of the hamiltonian. This indicates that it is possible that the encoding of the routing problem is not optimal. Regarding the Rosalin experiments, they are not able to achieve a desirable result due to the adaptive nature of this algorithm, which does not allow a sufficient number of iterations of the optimisation process.

Concluding, although there is potential for using quantum computing in industrial ecology problems, it is clear that for the moment, classical computers produce the best results. However, quantum computing is evolving rapidly, and there may be a point where the tables are turned and quantum com-

puters become the norm. As for now, the obtained results are far from ideal, but these could improve in the coming years.

Contents

1

Introduction

1.1. Background information

The activities of humans are increasingly influencing the Earth's systems. These effects have are profound and range from altering the climatic conditions of the planet [1] to altering the natural ecosystems [2]. Only in the last century, human activities have increased tremendously putting more pressure to the natural systems. This risks irreversible environmental alterations that could have even catastrophic consequences for life on the planet.

Due to the severity of the situation, there is a growing focus on sustainability. According to the World Commission on Environment and Development [3], sustainable development is defined as *"development that meets the needs of the present without compromising the ability of future generations to meet their own needs"*. In order to achieve this, it is necessary to shift human activities so that they are more in sync with the natural ecosystems [4].

A powerful tool to aid in this search are computers, which are able to provide solutions to numerous real world problems. Particularly for sustainability, problems are usually highly complex, often involving uncertainty, optimisation and complex decision making [5]. Luckily, computational techniques such as big data and machine learning can aid in the process of finding solutions to these problems.

1.2. Why quantum computers?

The potential of classical computers is expected to reach a barrier, as the size of computer chips is reaching its physical limit ($\sim$ 10nm) [6]. Unfortunately, there are still numerous problems that are too difficult to tackle with the current classical computational techniques, for which the only option is to search for approximate solutions that are in occasions far from accurate [7]. One example is the difficulty to model how molecules evolve during a chemical reaction [8]. It is for some of these problems where quantum computers can offer an advantage with respect to classical ones.

The real potential of quantum computers is still undetermined, but for certain tasks they are believed to be superior to classical computers, for instance when dealing with quantum systems like molecules and materials [9]. The superiority of quantum computers for a defined task is known as quantum supremacy. According to [7], *"quantum supremacy is achieved when a formal computational task is performed with an existing quantum device, but the same task cannot be performed using any known algorithm running on an existing classical supercomputer in a reasonable amount of time"*. This definition indicates that the quantum supremacy depends on the state of the current quantum computers, as well as the best classical computers and algorithms. Up to date, there is little formal evidence about which problems will be solved better on a quantum device (i.e. show quantum supremacy) [10]. However, supremacy seems to be achievable for some specific problems, which is the main motivation behind the development of quantum computers [11].

1.3. Research gap

Given the potential of quantum computers, it is logical to ask how this new computing paradigm can be applied in industrial ecology related problems. At the present time, there are no papers that discuss both disciplines together. For this reason, the aim of this thesis is to give a brief introduction to quantum computing and to identify what type of solutions it could bring to industrial ecology. Moreover, a case study is also included to present how quantum computing works in the practice. The case study is separated in two parts. The first one focuses on two state-of-the-art quantum algorithms and how they perform when compared with one another. The second part focuses on solving the optimisation problem of minimising the carbon emissions of a series of vehicle routes. It is important to note that there are several papers that have applied quantum computing to vehicle routing problems [12][13][14]. However, all the papers focus on optimising costs, but none focus on minimising emissions, so this work is also novel in the sense that it studies vehicle routing problems with quantum computing techniques and from a point of view of sustainability and minimising emissions.

1.4. Research questions

Formally, the research question that is studied in this work is ***"How can quantum computing benefit industrial ecology and how can it be applied to a relevant problem?"***. In order to answer the main research questions, the following sub-questions are also formulated.

- *What industrial ecology problems can benefit from quantum computing?*
- *What is the state-of-the-art of variational quantum algorithms and how do they perform?*
- *How do these algorithms perform in the task of optimising a green logistics problem?*

2

Quantum mechanics and quantum computing: an introduction

2.1. Mathematical concepts

2.1.1. Vectors

A vector is a mathematical object which is identified by a tuple of components in relation to a vector basis. The basis is a set of vectors that allows to represent other vectors. For instance if we consider in $\mathbb{C}^2$ the basis of vectors {(1,0),(0,1)}, we can express all the vectors in the plane. This basis can be changed, causing the components of the vector to change, but the vector itself does not change.

In the context of quantum mechanics vectors are expressed using the Dirac notation. This notation calls row vectors as "bras" ($\langle v|$) and column vectors as "kets" ($|w\rangle$). It is also known as bra-ket notation. From now on this will be the notation used in the rest of the thesis.

2.1.2. Vector spaces

A vector space is a set of vectors which are closed under addition (if $|u\rangle$ and $|v\rangle$ are part of the vector space, then $|w\rangle = |u\rangle + |v\rangle$ is also part of the space) and are closed under multiplication by a constant (if $|u\rangle$ belongs to the space and c is a scalar, then $c \cdot |u\rangle$ is also part of the space). A particular type of vector spaces are inner product spaces, which are equipped with a bilinear form, $\langle \cdot \mid \cdot \rangle : V \times V \to \mathbb{R}$, such that $\langle x \rangle \geq 0$, which define the notion of vector multiplication. The inner product between two vectors $|a\rangle , |b\rangle$ is an operation denoted as $\langle a|b\rangle$ which returns a scalar. The notion of inner product allows to formally define geometric concepts such as angles, orthogonality and lengths. The particular case of a vector with length 1 is called a unit vector, and these are the vectors that are used in quantum mechanics, as will be later shown.

In the context of quantum mechanics and quantum computing, the vector spaces that are used are Hilbert spaces, which are complete and separable inner product spaces. The vectors of the Hilbert space represent quantum states and the unitary matrices that transform the vectors also belong to Hilbert spaces. The reason to use a Hilbert space is that quantum mechanics works with complex numbers in the vectors and the matrices.

An interesting concept is the possibility to construct a bigger Hilbert space from several smaller ones. The most important way to construct the bigger space is called the tensor product, resulting in a tensor product space which is able to present every multilinear map defined on the smaller spaces. Mathematically, the tensor product of two spaces is denoted as $\mathcal{H}_A \bigotimes \mathcal{H}_B$, given $\mathcal{H}_A$ and $\mathcal{H}_B$ Hilbert spaces. It will be explained in detail how this works in the case where the Hilbert spaces represent qubit states.

2.1.3. Linear operators

In the most general sense, operators map elements of a space into elements of another space. In quantum mechanics, the operators that are used are linear operators, which have the properties shown in equations 2.1 and 2.2. Moreover, the operators have to map elements from one vector space into itself.

$$O(|A\rangle + |B\rangle) = O(|A\rangle) + O(|B\rangle) \tag{2.1}$$

$$O(c\,|A\rangle) = cO(|A\rangle) \tag{2.2}$$

Linear operators that map between finite-dimensional vector spaces can be expressed as matrices, which are a powerful mathematical representation. An important concept of linear operators are eigenvalues and eigenvectors. An eigenvector of an linear operator A is a non-zero vector $|v\rangle$ such that $A\,|v\rangle = a\,|v\rangle$, with $a \in \mathbb{C}$. The value a is known as the eigenvalue associated to the eigenvector $|v\rangle$. In the case of quantum mechanics, eigenvalues represent quantities that can be measured (energy and velocity for instance), so in the operators have to be of a special type where all its eigenvalues are real.

In order to ensure that an operator with complex coefficients has real eigenvalues, it needs to be hermitian. A hermitian operator has the property shown in equation 2.3, where $A^\intercal$ is the transpose of A and A^* is the complex conjugate of A. By using hermitian operators it is ensured that the measurable quantities are real numbers, since it does not make sense to have a complex number expressing the measurable quantities.

$$A = A^\dagger, \text{with} A^\dagger = (A^\intercal)^* \tag{2.3}$$

2.2. Quantum mechanics

Quantum mechanics is a theory used to predict the behaviour of atomic and sub-atomic particles [15]. It is well understood mathematically, which allows physicists to understand the building blocks of the theory and put them together in a way that allows them to explain a phenomenon of interest.

At the basis, it defines a series of postulates about how the microscopic world influences the macroscopic world. It also describes what results the measuring instruments will produce when studying a certain system. In total there are four postulates [15].

- The first postulate states that an isolated physical system is represented, at a certain time t, by a state vector $|\psi\rangle$, which belongs to a state space $\mathcal{H}$ [16]. $\mathcal{H}$ is a Hilbert space, and each unit vector in this space corresponds to a possible state of the system, and every possible state is represented by a vector in the space.
- The second postulate describes the physical quantities of the system that can be measured. These physical quantities are known as observables and can be position, energy or momentum, among others. In quantum mechanics, the observables are represented by Hermitian operators on the Hilbert space and the eigenvalues of these operators represent the possible outcomes obtained when measuring the observable. An important operator is the hamiltonian ($\hat{H}$), which represents the total energy of the system. The eigenvalues of the hamiltonian indicate what values of the energy can be measured for the system.
- The third postulate describes how composite systems are formed when different systems are put together. The Hilbert space of the composite system is obtained with the tensor product of the individual components ($\mathcal{H}_1 \bigotimes \mathcal{H}_2 \bigotimes ... \bigotimes \mathcal{H}_n$).
- The last postulate deals with the dynamics of a system, or how it evolves with time. Two contexts can be distinguished, how the system evolves on its own, and what happens when the system is measured. First, considering the state of a quantum system at time t_0, and knowing the constraints and forces which the system is subject to, Schrödinger's equation gives the state of the

system at any other time t. This equation is deterministic which means that a system in an initial state will always evolve in the same way. This is interesting because other aspects of quantum mechanics are probabilistic, like the measurement of observables. In the second context, when the measurement of an operator is performed and the result is a certain eigenvalue, the state of the system collapses to the associated eigenstate. Given a state $|x\rangle$, to which eigenstate it will collapse is given by probability. Given an observable of interest A, the probability of measuring the value b_i is shown in equation 2.4. Note that here the notation refers to the eigenvalue as b_i and the eigenvector $|b_i\rangle$.

$$P(b_i) = |\langle x|b_i\rangle|^2 \tag{2.4}$$

2.3. Quantum computation

Quantum computation is a new paradigm which takes advantage of different quantum effects to solve problems. The basic unit it the qubit.

2.3.1. What is a qubit?

A qubit is the basic unit of quantum information, and it is the quantum equivalent of bits in classical computing. Qubits are a two-level quantum mechanical system, and is among the simplest quantum systems. A famous example of these kind of system is the spin of an electron in which the two levels are spin up and spin down. The state of a qubit can be represented as in equation 2.5, with α and β complex numbers known as the amplitudes of the state. As explained before, qubits are a two-level system because upon measurement, one of two results will be measured. In the case of equation 2.5, the qubit is expressed in the basis $\{|0\rangle, |1\rangle\}$, and measuring in this basis will give either the results 0 or 1. It is possible to express the qubits in other basis too, as long as the vectors of the basis are orthonormal (unit vectors and orthogonal).

The difference between classical bits and qubits is that while a classical bit can have two different values (0 and 1), a qubit can be in infinitely many different states [17]. Note that it is only after measurement when the qubit will collapse to one of the two possible values.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C} \tag{2.5}$$

Interestingly, this state (α and β) cannot be known by the observer. What is possible is to measure the qubit, where the result will be either 0 or 1. Because the quantum world is probabilistic by nature, the measurement of a phenomenon will return different values when repeated several times. For a qubit with the state described in equation 2.5, Born's rule states that the probability of obtaining 0 is $|\alpha|^2$ and of obtaining 1 is $|\beta|^2$. Because there are two posibilities (0 or 1) and the total probability must add up to 1, $|\alpha|^2 + |\beta|^2 = 1$. For clarification, $|x|$ is the modulus of a complex number. If $x = a + ib$, the modulus is calculated as $\sqrt{a^2 + b^2}$, and is a real number.

Since the only option is to know the probabilities $|\alpha|^2$ and $|\beta|^2$, but not α and β, not all the information about the system is known. The reason why information is lost is because two different complex numbers can have the same modulus, for example $0.25 + 0.5i$ has the same modulus as $0.5 + 0.25i$.

2.3.2. Working with more than one qubit

In the case of having two bits, the possible results that can be measured are 00, 01, 10 and 11. Similarly, a two qubit system has four possible configurations $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$, and the state vector is defined as indicated in equation 2.6 (although it could be also defined in another basis). The probabilities of measuring each option are defined in the same way as in one qubit systems, with $|\alpha_x|^2$ being the probability of measuring x.

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2.6}$$

Generally speaking, a system of n qubits has 2^n amplitudes, each one associated to a certain outcome. The exponential increase of amplitudes with respect of the number of qubits is one of the

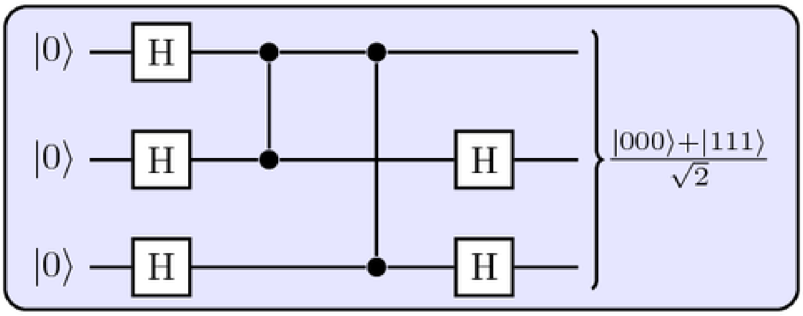

Figure 2.1: Example of a quantum circuit. Extracted from https://texample.net/tikz/examples/quantum-circuit/

most interesting properties of quantum computers, and the main difference with respect to classical computers, for which n bits can only store n bits of information, as is logical. The exponential increase of the amplitudes is something that occurs ”naturally” in the microscopic world and in quantum computers due to qubits being quantum mechanical systems. In contrast, when simulating a quantum computer on a classical computer, it is necessary to store the state vector, which grows exponentially with the number of qubits. Due to this exponential growth, the number of qubits that can be simulated in a classical computer is rather low. It has been calculated that 50 qubits is already impossible to simulate in any classical supercomputer [18]. Upon measurement of the qubits, the state collapses to the result that has been measured. For a system of n qubits, the measurement will return 0 or 1 for each of the qubits. Essentially, the result upon measurement looks identical to n bits, but the special part takes place in the preparation of the state, and how that can influence the output of 0s and 1s after measurement.

2.3.3. How do quantum computers work?

General functioning of quantum computers

Quantum computers, in the broadest sense, receive an input, perform a series of calculations and operations and return the output of interest. In order to achieve this, several computing models have been designed, such as quantum circuits, quantum annealing and the quantum Turing machine.

The models are theoretical frameworks on how to work with qubits, which have to be implemented later in a real quantum computer. It is important to do so in an accurate way in order to have high-quality qubits that work as expected by the theory. To do so, there are several technologies such as ion traps, topological quantum computers and transmons [19]. This is interesting for context on how quantum computers function, but this work will focus on the theoretical side, and not on the physical implementation of qubits.

Quantum circuit model

Currently, the most used model are quantum circuits where, by applying a series of gates to the qubits, the desired results are obtained. This model is similar to how classical computers work, and will be used from now on in the thesis.

Given that this work focuses on the quantum circuit model, a simple quantum circuit is showed in figure 2.1 to illustrate how this model works. This example has 3 qubits, one per line, which are initialised to the state $|0\rangle$ on the left (the measurement of this state is always 0). The circuit starts in the initial state and then goes through a series of gates from left to right. After all the gates act on the qubits, the resulting final state is $(|000\rangle + |111\rangle)/2$. Since it is a system of three qubits, there are 2^3 possible configurations ($|000\rangle, |001\rangle, ..., |111\rangle$). In this particular example, all the amplitudes are zero except for $|000\rangle$ and $|111\rangle$, which have amplitude $1/\sqrt{2}$. This means that the measurement of this quantum state will return $|000\rangle$ or $|111\rangle$, both with probability $(1/\sqrt{2})^2 = 1/2$. Note that the sum of both probabilities is 1, so it is a valid state for the 3-qubit system.

Regarding quantum gates, there are two types: single qubit gates, which act on one qubit and multiple qubit gates, which act on more than one. A gate can be expressed with an unitary matrix U

($UU^\dagger = I$). The unitary condition is the only one needed to define a valid gate and it ensures that the sum of probabilities of the different outcomes is still 1 after the transformation is made to a quantum state. For single qubits gates, these are $2x2$ matrices that act on the state of a qubit. The most important ones are the Pauli gates (I, X, Y and Z) and the Hadamard gate (H), all shown in equation 2.7. The Pauli matrices are of particular importance as they form an orthonormal basis of the Hilbert space of the operators that act on the qubits. As for the Hadamard gate it allows to create a uniform superposition between the states $|0\rangle$ and $|1\rangle$, and can be seen used in the circuit showed in 2.1. In the next chapters, these gates will be used to build circuits that prepare quantum states of interest to the problem that is trying to be solved. To give some context, for instance, an interesting state to calculate for certain chemistry problems is the ground state of a molecule.

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.7}$$

In the case of multiple qubit gates, they act at the same time on several qubits. The most famous two qubit gate is the CNOT gate or ”controlled NOT”. The effect of this gate is that if the first qubit is 0, the second qubit is maintained as it is. On the other hand, if the first qubit is 1, then the second one is flipped to the other value, as showed in equation 2.8. The CNOT gate is showed in matrix form in equation 2.9.

$$|00\rangle \rightarrow |00\rangle \, ; |01\rangle \rightarrow |01\rangle \, ; |10\rangle \rightarrow |11\rangle \, ; |11\rangle \rightarrow |10\rangle \tag{2.8}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.9}$$

Example of measurement

As it has been explained before, an observable is a hermitian matrix that changes the state of a qubit (the matrix multiplies the state vector and returns another state vector) and when that observable is measured, the state of the qubit will collapse to one of the eigenvectors of the operator.

To understand this in practice, let’s consider a 1-qubit system which is going to be measured qubit in the Z-basis to extract useful information. The Z-basis is the basis that has been used up until now ($\{|0\rangle, |1\rangle\}$). This basis receives that name because, as it can be seen in the Bloch-sphere in figure 2.2, the two components of the basis are placed in the poles of the z-axis. The operator associated to the measurement in the Z basis is the Z-Pauli matrix. In the circuit model, this measurement is done by adding a final Z gate at the end, for which the output will be measured (0 or 1).

$$Z \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \tag{2.10}$$

After multiplying by Z, the state of the qubit is $\alpha |0\rangle - \beta |1\rangle$, and the probabilities of obtaining $|0\rangle$ and $|1\rangle$ are $|\alpha|^2$ and $|-\beta|^2 = |\beta|^2$, respectively. The reason why the measurement can be done with this Pauli operator is because the qubit is expressed in terms of $|0\rangle$ and $|1\rangle$, which are exactly the eigenvectors of Z. Summarising, the operation that has been showed is what allows us to measure a qubit.

Current state of quantum computers

In theoretical works, qubits and quantum gates are considered to behave perfectly, but this is not the case in real life applications. For this reason, it is necessary to make a distinction between logical and physical qubits. Logical qubits are a theoretical construction used in quantum algorithms that behave exactly as planned, and the gates have the exact desired effect on the state of the qubits. On the other hand, physical qubits are physical representations of the logical qubits, and face several problems such as stability, decoherence and fault tolerance [20]. Fault tolerance describes how well

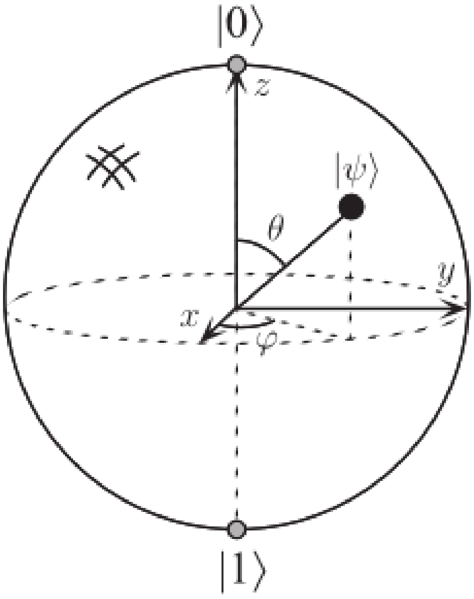

Figure 2.2: Visual representation of the Bloch sphere. The state of a qubit is expressed with two parameters, θ and ϕ. The states $|0\rangle$ and $|1\rangle$ are located in the north and south poles, respectively. Taken from [17]

the quantum circuit deals with the propagation of errors during the execution of the gates that form the quantum algorithm [21]. During this phase, the intermediate quantum states that are created by applying quantum gates need to be protected from errors. The reason for this is that if these errors are substantial, they will propagate and ruin the validity of the results.

Due to the aforementioned complications that quantum computers face, the current era in quantum computing has been named the Noisy Intermediate Scale Quantum era or NISQ era [22]. The term noisy refers to the fact that it is not currently possible to build a fault tolerant quantum computer that keeps the error propagation small. Intermediate scale refers to the size of the quantum computers available. Up to date, the biggest quantum computer has been developed by IBM, with 127 qubits [23]. The two factors described indicate that the present time is a developmental phase of quantum computing, still in the early stages.

Variational quantum algorithms

Given the current limitations on quantum computers, variational quantum algorithms (VQAs) are a promising method to obtain valuable results in the NISQ era [24]. VQAs offer an optimization- and learning-based approach that can be applied to a wide range of problems [25]. The idea behind it is not to design a specific circuit that produces the desired result because the error propagation would ruin the results but instead, the error is accepted as something that cannot be avoided, and has to be accepted instead. As for how these algorithms work, VQAs use parametrised circuits, where the parameters can vary to obtain different results. The objective is to minimise an objective function by using a classical optimiser to train the set of parameters, in order to generate a desirable result in the end. This type of algorithms are studied in more detail in the following chapters, as they offer a way to get meaningful results in the NISQ era. Moreover, it will be shown how to apply this type of algorithm to a relevant problem to the field of industrial ecology.

3

Potential applications of quantum computing to industrial ecology

This chapter focuses on how quantum computing can be useful for industrial ecology and what type of problems can benefit. It is important to note that some of the applications that are presented are out of reach in the current NISQ era. Nevertheless, these are included to give a general context of the possibilities that are offered by quantum computing.

3.1. Industrial ecology

According to the International Society for Industrial Ecology [26], *”industrial ecology is the study of systemic relationships between society, the economy, and the natural environment. It focuses on the use of technology to reduce environmental impacts and reconcile human development with environmental stewardship while recognising the importance of socioeconomic factors in achieving these goals”*. By studying how industrial systems work and how they interact with the natural ecosystems, it is possible to find ways to restructure them so that they are more aligned with the natural world.

This discipline deals with complex systems such as the economy and the natural ecosystems, so it is at the intersection of several disciplines. It has developed several quantitative methods such as Life Cycle Assessment (LCA) and Material Flow Analysis (MFA) that aim to study how materials and energy flow in society, and what the environmental and societal consequences are.

In a broader context, industrial ecology has influenced the perspective of economic systems towards resources efficiency. Before, companies aimed for efficiency for economic reasons, while now resource efficiency and waste minimisation are also desirable from the point of view of the environment [26]. These concepts are the basis of what is widely known as the circular economy, where waste flows are valuable and thus recovered for another purpose.

Given the potential of quantum computers over classical computers, it is interesting to identify which industrial ecology problems can benefit from the quantum option.

3.2. Potential applications to industrial ecology

3.2.1. Multiobjective optimization

In real-world optimisation problems, it is most common to have a series of objectives that have to be satisfied at the same time. Some examples of objectives are cost (to be minimised) and client satisfaction (to be maximised). It is common that some of these objectives will be in conflict with one another, meaning that improving one may have a trade-off in another. For this reason, multi-objective problems do not have one optimal solution, but several feasible solutions that better respect the trade-offs between all the objectives.

Problem size	Combinatorial	Polynomial
5	$5!$ = 120	5^5 = 3125
10	$10!$ = 3628800	10^5 = 100000
15	$15!$ = 1.308e+12	15^5 = 759375

Table 3.1: Comparison of polynomial and combinatorial complexity. Given a problem size, the number of needed operations is showed for both classes.

Currently, the state-of-the-art are multi-objective evolutionary algorithms (MOEAs). According to [27], quantum computing may offer better techniques to find suitable solutions for multi-objective problems. This is done by exploiting quantum properties such as entanglement, interference and superposition to develop new algorithms. Depending on the target system of the algorithms, they can be classified in two groups. The first is quantum metaheuristics (QM) and are designed to be used in a quantum computer. On the other hand, when the target is a classical computer, the name is quantum-inspired metaheuristics (abbreviated as QIMs) [28].

This type of optimisation is suitable for industrial ecology, since this discipline deals with complex systems where there are several factors to take into account. Practical examples include [29], which seeks to optimise the exchange flows in an industrial park. The objectives are to reduce pollution and waste and to maximise the shared resources. Another example is on industrial water management [30], where fresh water use is minimised while regenerated water is maximised. As it can be seen, multi-objective optimisation is very relevant for industrial ecology, and quantum computing offers an interesting opportunity to reach better solutions than those obtainable with current computational methods.

3.2.2. Combinatorial optimisation

The objective of quantum computers is to find algorithms that need less resources and time to reach a (near) optimal solution [31]. Many applied problems have combinatorial complexity. Combinatorial complexity is one of the complexity classes defined in computational complexity theory. The aim of this theory is to classify problems taking into account the resource use (time and memory). The fastest algorithm that exists to solve a problem determines what complexity class the problem belongs to (for instance there are different methods for matrix multiplication with different complexities). Roughly speaking, if the input size of a problem is n, then in the combinatorial case the number of operations required to solve it is $n! = n \cdot (n-1) \cdot ... \cdot 2 \cdot 1$. To give a comparison, another complexity class is polynomial, where for an input of size n, the required number of operations are n^m for some $m \in \mathbb{N}$. To see how these complexity classes are compared, lets suppose one problem has combinatorial complexity and another has polynomial complexity with $m = 5$. The comparison of both is shown in table 3.1, and it can be seen that the combinatorial case grows much faster than the polynomial. For this reason, when combinatorial problems arise in the real-world, it is in most cases impossible to find an exact solution. Instead it is more interesting to use heuristic methods that search for an valid solution as close as possible to the optimum. Examples of combinatorial problems are the Sudoku and routing problems like the one that will be studied in the a later chapter.

A big part of combinatorial problems can be formulated as a quadratic unconstrained binary optimization (QUBO), as it is done is done in the logistics chapter. For instances, QUBO permits the formulation of scheduling and resource allocation problems, as well as graph optimization tasks, relevant for industrial and urban problems, among others.

The surprising advantage of QUBO solvers is that in general, they find solutions that are better or similar to the best state-of-the-art specialised heuristics [32]. For this reason and for the wide applicability for industry problems, this makes this approach an attractive option. There is still no method available to solve the QUBO problem in polynomial time (finding a way of making the QUBO problem of the polynomial complexity class), either for quantum or classical computers. In the case of the former, certain families of the QUBO in a more efficient way when compared to classical computers [33] [34].

In the context of industrial ecology, many interesting problems have combinatorial complexity. For instance, the problem of where to place a facility to optimise reverse logistics to recover products after the use phase. This problem requires to work with routes to find a good location of the facility that works well for all the involved actors [35].

3.2.3. Machine learning for industrial ecology

Machine learning (ML) is an umbrella term that encompasses a series of techniques that extract patterns in sets of data [36]. In order for the model to work, it needs to be trained with some input data, so it can later extract its own deductions from new data. It is being used in the field of industrial ecology in a variety of applications. For instance, to improve industrial symbiosis by identifying waste flows that could be used in new applications within the context of an industrial park [37]. This is done by analysing the data in academic articles and patent databases. Another application is to improve the energy efficiency of data centres [38], which is relevant topic for the high amount of energy consumption and the projection in growth that is expected in the future. They work by developing novel control strategies for the processes that take place, like the cooling of the infrastructure.

In the context of quantum computing, the question that arises is how these new computing paradigm can improve automatic data mining. According to [39], our ability to implement machine learning algorithms is constrained by the available computational power. While the availability of data has greatly increased, the performance of computers is staying behind. For this reason, quantum computing appears as an interesting option for ML, considering the higher computational abilities when compared to classical computers. With powerful quantum computers, it would be possible to study massive amounts of data coming from smart cities and meteorological sensors, among others.

3.2.4. Quantum information for industrial ecology

In order for an adaptive system to prosper, it is necessary that the components of the system interpret signals from the environment and respond in an adequate manner [40]. To adapt more successfully, the use of complex strategies can be beneficial, but this requires the actors to remember a high number of past experiences. This requires large amounts of memory and, interestingly, quantum information can offer a offer a solution to this problem. Quantum information is discipline related to quantum computing that studies how to store information in a quantum state and how to do operations on that information. By using quantum information, the information stored in the memory can be compressed more effectively than when using classical memory. This offers scaling advantages for quantum computers, which are specially useful when the agents need to recall events far into the past, allowing for the design of more sophisticated and complex agent-based models. On the downside, [40] presents a theoretical model which has still to be executed in a real quantum computer. This step is crucial as it could present added complications that were not foreseen in the theoretical work.

Even though the technology is still not available in the practice, there are numerous problems of interest for industrial ecology that could benefit from quantum information. For instance, modelling how the markets will behave when a new product is introduced or how citizens will respond to a new recycling strategy [41]. Also, how companies interact with one another in order to make a sector more sustainable.

4

Benchmarking of VQAs

This chapter introduces two state-of-the-art variational quantum algorithms (VQAs). Also, it tests tests the performance of both to see the quality of results and how they compare with one another.

4.1. Benchmarking and why is it used

With the number of optimisation algorithms increasing constantly, it is important to conduct experiments to see which one is more suitable for a certain problem [42]. This experiments are widely known as optimisation benchmarking and, in detail, benchmarking is to compare different products over a collection of performance metrics. In this case, the implementation of several algorithms constitute the products that are being compared and the performance metrics are obtained after evaluating the implementations on a series of test problems.

Common performance metrics are memory use, final distance to the optimum (an algorithm may not reach the optimum but can stay sufficiently close), and running time (two algorithms can reach the same value in very different times). The objectives of benchmarking are twofold:

- This process can reveal strengths and weaknesses of a certain algorithm, which allows researchers to improve the performance.
- It also allows end-users to choose an appropriate algorithm for a problem of interest.

4.2. How do VQAs work?

As explained in chapter 2, VQAs train a parameterised quantum circuit by using a classical optimiser. The circuit is parametrised by a series of angles $\theta = \{\theta_1, \theta_2, ..., \theta_n\}$ that rotate the qubits in order to produce different quantum states. These rotations are on one of the three directions of the Bloch sphere ($R_x(\theta), R_y(\theta)$ and $R_z(\theta)$. Depending on the rotations that are applied to the initial state of the qubits, the final quantum state will be different. In equation 4.1, it is shown that the final state $\left|\psi(\vec{\theta})\right\rangle$ is obtained by applying the parametrised circuit $U(\vec{\theta})$ to the initial state $|\psi\rangle$.

$$\left|\psi(\vec{\theta})\right\rangle = U(\vec{\theta})\,|\psi\rangle \tag{4.1}$$

The idea is to use a method to find better values for θ that give better results for a cost function. In quantum computing, the cost function is defined as the expectation value of a certain observable [43]. It is necessary to take the expectation value because measuring an observable will yield different results each time. The cost function is represented in equation 4.2. $\langle\hat{O}\rangle$ is the expectation, and is calculated by multiplying the state by the operator matrix and again by the state.

$$C(\vec{\theta}) = \langle\hat{O}\rangle = \left\langle\psi(\vec{\theta})\middle|\,\hat{O}\,\middle|\psi(\vec{\theta})\right\rangle \tag{4.2}$$

Instead of working directly with the observable $\hat{O}$, it is common to write it as a linear combination of Pauli operators, which are easy to measure, as indicated in equation 4.3. $\hat{P}_i \in \{\mathbf{I}, X, Y, Z\}^{\bigotimes N}$.

$$\hat{O} = \sum_i c_i \hat{P}_i \tag{4.3}$$

Combining both equations 4.2 and 4.3, the calculation of the cost in terms of the expectation of single Pauli operators is shown in equation 4.4.

$$C(\vec{\theta}) = \langle \hat{O} \rangle = \sum_i c_i \langle \hat{P}_i \rangle \tag{4.4}$$

There are several techniques to find the θ, such as gradient descent and gradient free methods. This section will describe one optimiser of each group, namely Rosalin and LCB CMA-ES.

Now that the parametrised circuit and the cost function have been introduced, it is possible to explain the general functioning of a VQA. Essentially, the quantum side evaluates the cost function on a quantum computer, while new candidates to improve the cost function are generated on the classical side [24]. On the quantum side there have been important advances, such as resilience to certain kinds of noise [44], and reducing the necessary measurements by searching for subsets of commuting operators [45]. Apart from the quantum side, it is necessary to develop powerful optimisers in the classical side in order to exploit the potential of VQAs. Due to the unique workings of quantum computers, it is believed that off-the-shelf classical optimisers might not be ideal to solve VQAs. For this reasons, new optimisation algorithms are being developed specifically for optimising quantum results.

In a real setting, there is no access to the state vector, so it is not possible to calculate the expectation value exactly [43]. For this reason, the idea is to measure the observable and record the result several times to get an approximate value of the expectation value. Each of the times that the state is prepared (with the circuit) and the observable is measured is called a shot. Due to Heisenberg's uncertainty principle, not all observables can be measured together. Essentially, if two observables do not commute, the standard deviation of the measurements will be greater than a certain value (expressed in equation 4.5). Not all observables can be measured at the same time with precision, that is the reason why the Paulis are measured separately.

$$\Delta(C)\Delta(D) \geq \frac{|\langle\psi| [C, D] |\psi\rangle|}{2} \tag{4.5}$$

4.3. Quantum circuit for the VQA

Given the usual representation of a qubit shown in 2.5, and given the fact that $|\alpha|^2 + |\beta|^2 = 1$, this equation can be rewritten as shown in equation 4.7 by writing the complex numbers α and β in polar form ($e^{i\theta}r$)[17]. The new way to express the state if a qubit is now a function of two angles, θ and ϕ. Equation 4.6 describes the usual representation of a qubit that has been used up until now, and equation 4.7 represents the equivalent version expressed in terms of θ and ϕ.

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \tag{4.6}$$

$$|\psi\rangle = \cos\frac{\theta}{2} |0\rangle + e^{i\phi} \sin\frac{\theta}{2} |1\rangle \tag{4.7}$$

This new mathematical expression allows to visually represent all the possible states of a qubit in what is know as the Bloch sphere, as shown in figure 4.1. Notice that the states $|0\rangle$ and $|1\rangle$ appear in the top and lower poles of the sphere. As it is a three-dimensional sphere, the axis x, y and z are also shown in the figure. This sphere can be used to track how the state of a qubit changes when a gate is applied to that qubit. One intuitive way of changing the state (which is symbolised as a point in the

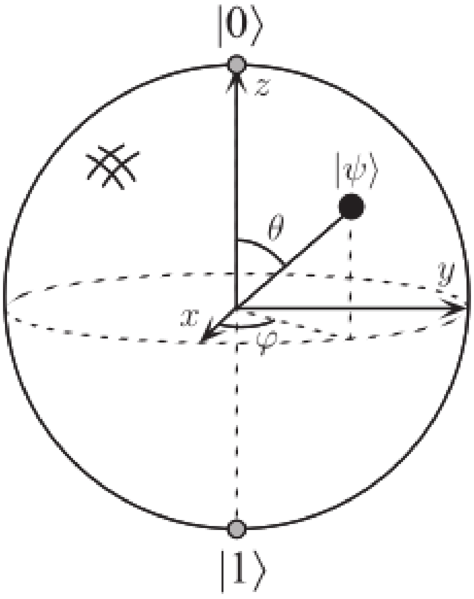

Figure 4.1: Visual representation of the Bloch sphere. The state of a qubit is expressed with two parameters, θ and ϕ. The states $|0\rangle$ and $|1\rangle$ are located in the north and south poles, respectively. Taken from [17]

Bloch sphere) is to rotate around the x, y and z axis of the sphere (see the figure for clarification). These three rotations are expressed in equations 4.8, 4.9 and 4.10 [17]. It can be seen that these rotations are built using the Pauli matrices (X, Y, Z and I). The I is the identity and does not change the state, and the matrices X, Y and Z generate 180° rotations around the x, y and z axes, respectively. For this reason, these new rotation matrices give more flexibility than the regular Pauli matrices, as they allow rotations of any degrees around the axes.

$$R_x(\theta) = e^{(-i\theta X/2)} = cos\theta/2 = cost\theta/2I - i\sin\theta/2X = \begin{pmatrix} \cos\theta/2 & -i\sin\theta/2 \\ -i\sin\theta/2 & \cos\theta/2 \end{pmatrix} \tag{4.8}$$

$$R_y(\theta) = e^{(-i\theta Y/2)} = cos\theta/2 = cost\theta/2I - i\sin\theta/2Y = \begin{pmatrix} \cos\theta/2 & -\sin\theta/2 \\ \sin\theta/2 & \cos\theta/2 \end{pmatrix} \tag{4.9}$$

$$R_z(\theta) = e^{(-i\theta Z/2)} = cos\theta/2 = cost\theta/2I - i\sin\theta/2Z = \begin{pmatrix} e^{(-i\theta/2)} & 0 \\ 0 & e^{(i\theta/2)} \end{pmatrix} \tag{4.10}$$

The newly defined rotation matrices $R_x(\theta), R_y(\theta)$ and $R_z(\theta)$ can be used as the components of the circuits used in VQAs. These circuits are characteristic for having parameters that are changed to produce different quantum states for the qubits.

For all the experiments done in this thesis the circuit that is used is defined in figure 4.2. It can be seen that the circuits is composed of a main building blocks represented with curly braces ({}). This indicates that the gates contained inside the brackets can be repeated several times, one in front of the other. Depending on the number of repetitions, the final size of the circuit will vary, as well as the number of total parameters. It will be useful to test different sizes when solving a problem. The U gates in the circuit are formed by several rotation gates (R_y and R_z) and three CNOT gates (the parts that contain the $\bigoplus$ symbol). The composition can be seen in the lower part of the figure 4.2. After the part in curly braces, the circuit ends by doing a R_y rotation gate on all of the qubits.

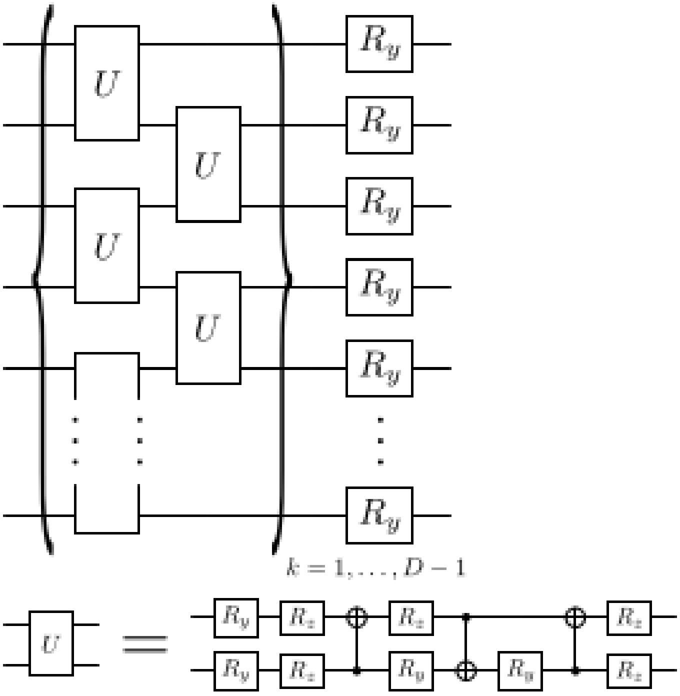

Figure 4.2: Parametrised circuit used in the experiments of this thesis. The portion in between the curly braces can be repeated several times to create circuits of different size which can fit better different problems. Taken from [46]

4.4. Shot allocation strategies

Given a number of shots to estimate the expectation value of an observable s_{tot}, it is important to find a way to distribute the shots among the Pauli operators that form the observable. There are three main ways to estimate $\langle H \rangle$, according to [24], and different algorithms use different strategies. Three possible strategies are described:

- Uniform deterministic sampling (UDS): this sampling method allocates the same number of shots to each term. If the hamiltonian H has N terms, the number of shots per term is n_{tot}/N.
- Weighted deterministic sampling (WDS): this method looks at the coefficients of each term, and allocates more shots the bigger the absolute value of the coefficient is. It follows the formula $s_i = s_{tot}\frac{|c_i|}{M}$. For instance, if $H = 5X + 2Y - Z$ and $s_{tot} = 100$, then $s_{5X} = 100\frac{5}{8} = 62.5$, $s_{2Y} = 100\frac{2}{8} = 25$, $s_{-Z} = 100\frac{1}{8} = 12.5$.
- Weighted random sampling (WRS): the choice of where to allocate each shot is now done based on probability, with a bigger probability for the biggest coefficients in absolute value. Each term will be chosen with probability $p_i = \frac{|c_i|}{M}$. In this alternative the allocation of shots will be similar to the WDS if s_tot is big enough, but there will be some variation due to the stochastic nature of this sampling method.

4.5. Rosalin

When designing an optimiser for VQAs, it should be able to handle hardware noise as well as shot noise. Hardware noise is due to the imperfect gates and circuits of the NISQ era, and shot noise comes from the fact that a limited number of shots cannot calculate the exact expectation of a hamiltonian to calculate the cost value. One of the options is to design a gradient-based optimiser, which use the information of the gradient to generate the new angles of the circuit θ. For clarification, the gradient is the multidimensional equivalent to the derivative, and it indicates how a function varies locally. Calculating or approximating the gradient requires a high number of shots when there is a high number of partial derivatives or the cost function is the sum of a high number of non-commutable observables (if they commuted, they could be grouped, as explained in the uncertainty principle at the start of this chapter) [47]. To reduce the number of shots, the focus is on shot-frugal gradient descent, which adapts the number of shots to maximise the expected gain (or improvement in the value of the cost function) for a partial derivative per shot.

This algorithm is known as iCANS (which stands for individual Coupled Adaptive Number of Shots) [46], and uses WDS as the shot allocation strategy. The authors argue that this choice needs at least one shot per term of the hamiltonian, so if there a lot of terms, the minimum number of shots needed to estimate the observable without bias is quite high. On the other hand, the WRS strategy can produce an unbiased estimate of the observable with only one shot. In this thesis, the focus is on another algorithm called Rosalin. This algorithm is identical to iCANS with the only difference that it uses WRS instead of WDS.

In order to create new parameters for the next iteration of the algorithm, Rosalin estimates the gradient and changes the parameters in the opposite direction to the gradient. The reason for this is that the gradient shows the direction of biggest increase in the cost function, and the opposite direction is the biggest descent, which is of interest in a minimisation problem. The reason why the gradient must be estimated is because the cost function does not have an analytic expression. If the cost function had an analytic expression (if it could be written down mathematically like $f(x) = x^2$), then to calculate the gradient it would only be necessary to use the derivation rules. In this case, the analytic form of the cost function is unknown so estimating it is the only option. This is done with the finite differences method, shown in equation 4.11.

$\hat{E}_j^+$ and $\hat{E}_j^-$ are single shot estimates of the cost function with a small change in the input parameters. In the case of $\hat{E}_j^+$, the parameters are increased by a small quantity, and for $\hat{E}_j^-$, the parameters are diminished also by a small quantity. By combining both, $(\hat{E}_j^+ - \hat{E}_j^-)$ is used as an estimate of one of the partial derivatives, which then are used to estimate the gradient.

$$g_l = \frac{\sum_{j=1}^{s_{tot}} (\hat{E}_j^+ - \hat{E}_j^-)}{2 s_{tot}} \tag{4.11}$$

Rosalin uses an adaptive number of shots by calculating how many shots should go to estimate each of the components of the gradient. After the gradient is estimated, the parameters θ are changed as it is showed in equation 4.12.

$$\theta_l^{t+1} = \theta_l^t - \alpha g_l \tag{4.12}$$

With the new set of parameters, a new iteration starts again and the gradient is recalculated. This is done several times with the idea to improve the cost function repeatedly and arrive at a result close to the optimum.

4.6. LCB CMA-ES

LCB CMA-ES stands for Lower Confidence Bound Covariance Matrix Adaptation Evolution Strategy. It is a variation of CMA-ES (Covariance Matrix Adaptation Evolution Strategy) adapted to take into account the randomness of quantum computers and works to reduce the variance of the solutions [48].

Evolution strategies are used to optimise real-parameter non-linear functions in a continuous domain. This is the case of quantum observables, which are often non-linear functions (e.g. $H = X_1 X_2 Y_4$) with real values (the measurement of an observable is a real quantity).

An evolutionary strategy starts with some random candidates in the parameter space and evolves them based on the laws of selection and recombination that guide how species change over time. The candidates are evaluated with respect to a cost function (in this case the expectation of the hamiltonian) and the ones with the lowest value (good for minimisation problem) are maintained and will play a role in the next population. This is known as ”survival of the fittest”. The next step is to recombine the parents to create new offsprings. There are different ways of doing this, but the logic behind is similar in all cases. The special feature of CMA-ES with respect to other evolution strategies is that the selection of new candidates is done in a way that adapts to the contour lines of the function. This is done by using the multivariate normal distribution which requires the covariance matrix. The covariance matrix is estimated with each new generation of the population to generate the next generation more accurately.

Normally, when the candidates of the evolution strategy are evaluated, the cost function is a fixed number. However, due to the uncertainty of measurement in quantum computers, the candidates' cost function will have some sampling noise (noise=uncertainty). An option is to take enormous numbers of shots and then the uncertainty will be minimal, but this is not always possible due to limited resources. In a real-world setting, quantum hardware can prepare and measure between $10^6 - 10^9$ shots, which must be cleverly used to find a good set of parameters (each member of the population is a set of parameters) and to measure the cost function to a certain accuracy.

Taking the randomness of the measurement into mind, it is possible that a candidate with a not-so-good real cost value appears as a much better candidate due to the sampling noise. For this reason, the lower confidence bound method is used to choose which candidates to reevaluate by using more shots. In 4.13, the equation for the lower confidence bound (LCB) is shown. In the equation, N is the number of shots, $\|\vec{c}\|_1$ is the sum of all the coefficients, and the c_i's are the coefficients. $\bar{C}(\vec{\theta}, N)$ is the approximation of the cost function with N shots, and $C(\vec{\theta}, N)$ is the real cost function. By calculating the LCB, we can assure that the real cost value $C(\vec{\theta}, N)$ will be outside of the confidence interval $[LCB(\vec{\theta}, N, t), \infty]$ with a very small probability. For this reason, if one candidate has a lower LCB than another, it is possible that the real cost function of the former is smaller than that of the latter, so it is a more promising solution.

.

$$LCB(\vec{\theta}, N, t) = \bar{C}(\vec{\theta}, N) - \varepsilon \sqrt{\|\vec{c}\|_1 \sum_i \sum_j \sqrt{|c_i c_j|}/N} \tag{4.13}$$

4.7. Benchmarking experiments

The two optimisation algorithms will be tested on four different hamiltonians that are decomposed into Pauli matrices to improve the results. The hamiltonians are the Ising model, the Heisenberg model, the Ghz and a random one. The mathematical expression of these hamiltonians is:

- Random: $0.549I + 3.182X1 - 2.953Z_1Y_2 + 0.462Z_1Z_3$
- Heisenberg: $-2.0Z_0Z_1 - 2.0Z_1Z_2 - 2.0Z_0Z_3 - 2.0Z_2Z_3 + 1.0X_0 + 1.0X1 + 1.0X_2 + 1.0X_3$
- Ising: $1.0X_0X_1 + 1.0X_0X_3 - 3.0Y_0Y_1 - 3.0Y_0Y_3 + 2.0Z_0Z_1 + 2.0Z_0Z_3 + 1.0X_1X_2 - 3.0Y_1Y_2 + 2.0Z_1Z_2 + 1.0X_2X_3 - 3.0Y_2Y_3 + 2.0Z_2Z_3$
- GHZ: $2.0Z_0Z_1 + 2.0Z_0Z_3 + 2.0Z_1Z_2 + 2.0Z_2Z_3$

Going in detail into these quantum mechanical models is out of the scope of this work, but a brief description is included. The Ising model is used to study ferromagnetism. It is formed by discrete variables (the σ) which represent the nuclear magnetic moment that can be in the states +1 or -1 [49]. The Heisenberg model is similar to the Ising model but there not only the z-component of the spin interacts but also the x and y-components. The random hamiltonian is, as the name indicates, a random sum of possible products of X, Y and Z Pauli matrices acting on the qubits.

As shown in table 4.1, the experiments are done with 4 qubits. The number of shots on each optimisation run is 100000. This means that, given the task of finding the ground state of one of the hamiltonians, the quantum circuit is prepared and measured a total of 100000 times to find the best θ possible with the lowest expectation value. Due to the random initial θ used at the start of Rosain and LCB CMA-ES, it is possible that one of the algorithms finds a near-optimal solution very quickly, and the other has more difficulty. For this reason, and to make if fair, the optimisation is repeated 50 times for each hamiltonian, changing the initial parameters for each run. By doing this and taking the average result from the 50 runs, we are assured that it is not luck in one special run and we can see how the algorithms perform in a consistent manner. To set the ”random” initial parameters for each run in a controlled manner, a seed is used. A seed is a number that is utilised to start a pseudorandom

Characteristics	Value
Algorithms	LCB CMA-ES and Rosalin
Test functions	Random, Ising, Heisenberg, GHZ
# qubits	4
# shots	100000
# runs	50
Seed	Equal for both algorithms on each run
Types of simulation	Noisy and noiseless
Types of measure	Absolute and relative
Type of circuit	Hardware efficient circuit

Table 4.1: Table showing the design of the benchmarking experiments.

number generator. By defining a seed, we will get the same pseudorandom sequence every time. The sequence is defined as pseudorandom because it is defined by the initial value (seed) but the properties of these sequences approximate those of real random sequences [50]. This is needed because it is not possible to generate truly random sequences.

Both Rosalin and LCB CMA-ES return the values of the cost function as noisy and noiseless. The noiseless result is the exact expectation value and is calculated as $\langle\psi| H |\psi\rangle$. To calculate this value, it is necessary to know the state of the qubits $|\psi\rangle$. In a real experiment with a quantum computer this is impossible because the state is unknown. However, as the quantum computer is being simulated in a classical computer, the python packages (Cirq and Openfermion by Google) do keep track of the state, so it is possible to calculate the noiseless value. The noisy value is calculated by assigning a number of shots that measure the cost function and averaging the results to get an estimate of the true expectation.

To measure how the algorithms are performing as the optimisation process advances (the shots are used), two different metrics are used. These metrics are the absolute result 4.14 and the relative energy error 4.15. As for the first one, because the ground state of the different hamiltonians is different, it can be confusing to look at plots of different hamitonians at the same time. For this reason, the value of the cost function is translated by the optimal value so that all the plots show the optimisation towards zero. Regarding the relative measure, C_{opt} is the ground state (optimal result) and c_0 is the biggest coefficient in the hamiltonian.

$$C_0(\vec{\theta}) = C(\vec{\theta}) - C_{opt} \tag{4.14}$$

$$\Delta_r E = \left| \frac{C(\vec{\theta}_{opt}) - C_{opt}}{C_{opt} - c_0} \right| \tag{4.15}$$

4.8. Results

The final plots of the benchmarking are shown in figures 4.3 and 4.4. Both figures use the absolute metric.

It can be observed that both the noiseless and the noisy plots are quite similar. This is due to the fact that the plots show the average results of the 50 runs. Normally, the noisy plots for a single run will be less smooth with more ups and downs, but with 50 runs the curves are smoother. In the figures, the results are plotted as dark green and blue lines. The ”wrapping” around these lines represents the

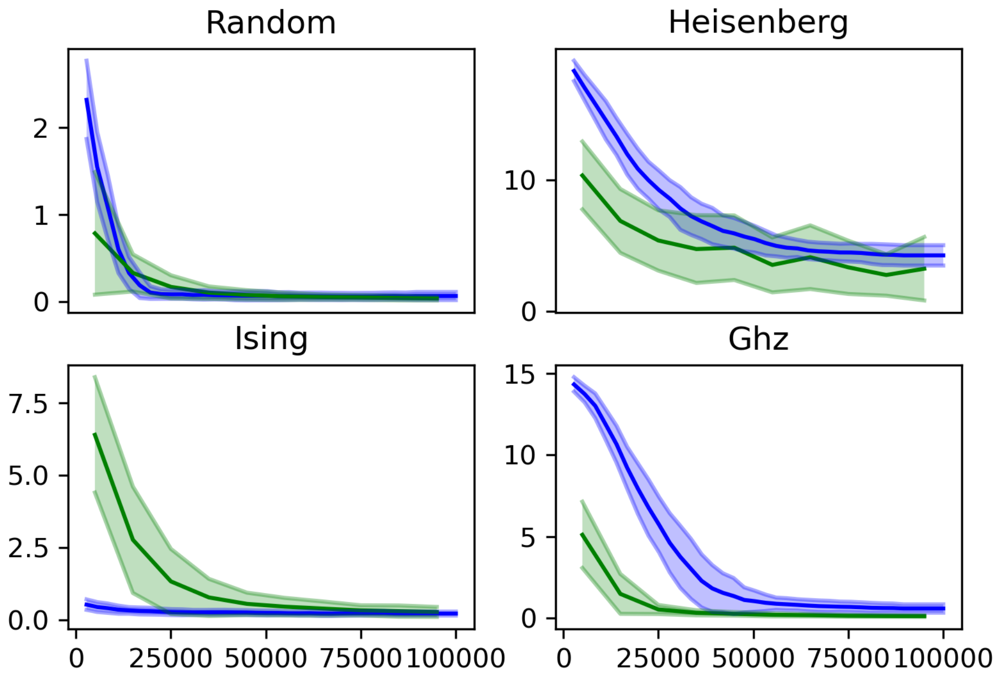

Figure 4.3: Noiseless plots of the four hamiltonians. The green plot corresponds to Rosalin and the blue one to LCB CMA-ES. In these plots, the exact cost function is calculated on every step of the algorithms, making it noiseless. The graphs obtained are an average of 50 runs of the algorithms, to increase the reliability of the results

standard deviation of the results over the 50 runs of the algorithms. Given a certain iteration of the algorithm, if the standard deviation of the results of all runs is small, the ”wrapping” will be narrower, and if the standard deviation is large, the ”wrapping” will be wider.

For the noiseless results, in the random hamiltonian both algorithms get close to the optimum in the first shots. As it can be seen in the graph, the curve decreases very steeply at the start and then it stabilises close to the optimum. As for the Heisenberg hamiltonian, Rosalin (green plot) gets close to the optimum sooner than LCB (blue plot), which also gets close to the optimum but more slowly. It is interesting to note that Rosalin finds better results consistently in the first half of the shots, but then fails to converge towards a value in the last shots. This is due to the adaptive shots allocation, which in the case of this hamiltonian, it requires a vast amount of shots per iteration (to find each new θ, a lot of shots are required). Despite all the used shots to calculate the gradient, the resulting new θ does not guarantee a better result. On the other hand, because LCB maintains the best solutions for each generation, it can only obtain better solutions in each iteration. Overall, it seems that even though Rosalin varies the results greatly, it manages to produce better results that LCB, which seems to stabilise around 5. The Heisenberg hamiltonian seems like a difficult hamiltonian to optimise and it would be interesting to investigate why this happens. For the Ising hamiltonian, both algorithms get close to the minimum, although LCB finds near-optimal solutions from the start. Overall, LCB finds a slightly better result. Lastly, for the GHZ hamiltonian, it can be observed that Rosalin finds a better result and faster than LCB.

Regarding the standard deviation, this measure gives an idea of how the different runs of the algorithms vary among themselves. In the cases of the random, Ising and GHZ hamiltonians, it is observed that the standard deviation is reduced as the optimisation process advances. This behaviour happens because for these hamiltonians, the optimisers reach a close point to the optimum on the average of the 50 runs. Because the average is close to the optimum and the noiseless result gives the real result which can never be below the optimum, it means that the standard deviation between the different runs will be smaller as more shots are used. In contrast, the standard deviation is larger

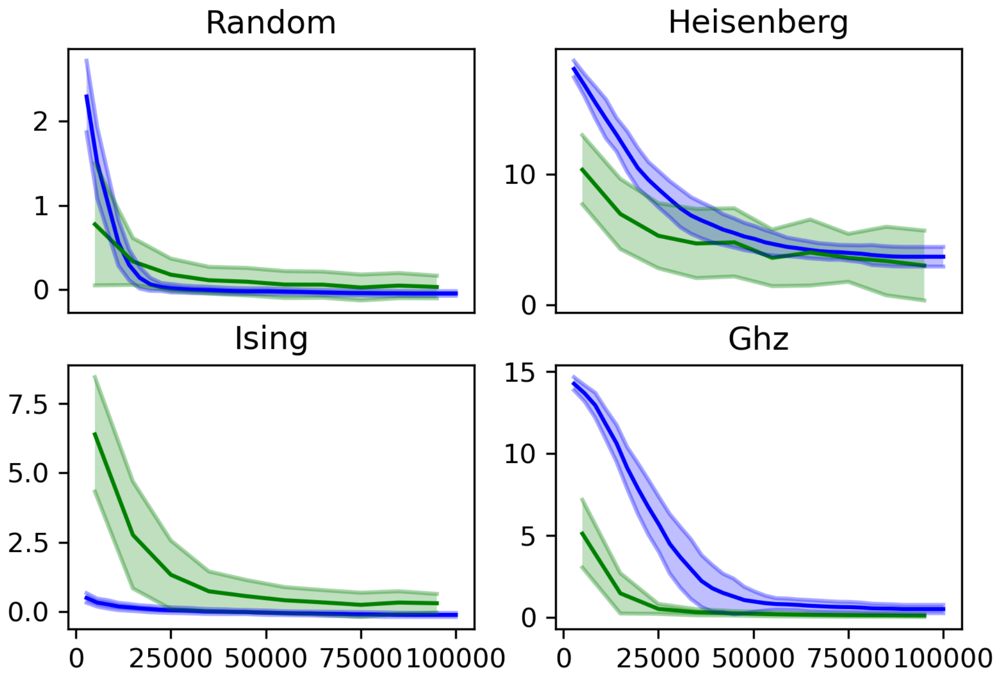

Figure 4.4: Noisy plots of the four hamiltonians. The green plot corresponds to Rosalin and the blue one to LCB CMA-ES. In these plots, the cost function is approximated on every step of the algorithms, thus the noise. The graphs obtained are an average of 50 runs of the algorithms, to increase the reliability of the results.

for the Heisenbeg hamiltonian, even also as the optimisation process advances. The reason for this behaviour is that the optimisers do not get as close to the optimum as in the other cases, and this creates more variation among the results of the different runs.

It is interesting to note that none of the algorithms reach the optimum solution (0). The reason for this may be that the variational circuit is not complex enough and more parameters would be needed. It is very probable that given a certain parametrised circuit, none of the possible combinations of θ produces the exact state for the qubits that corresponds to the ground state of the hamiltonian. Possibly by expanding the number of parameters of the circuit it is possible to get results closer to the ground state. Depending on the application, the current results may be sufficiently good. However, if a better result is needed, other circuits can be tried, as well as improving the number of shots can also improve the result.

Regarding the noisy plots in figure 4.4 the results are quite similar to the noiseless. The main differences are that Rosalin gets a better result for the random hamiltonian than LCB, when in the noiseless the results were identical. Moreover, for the Ising hamiltonian LCB performs much better than Rosalin. For the other two hamiltonians the results are very similar. Overall, the results are satisfactory, but could be further improved for the Heisenberg hamiltonian. In the case of the standard deviation of the plots, the observed results are similar to the noiseless case. The only main difference is that, as the noisy evaluations of the cost function can be below the optimum, even if the average of the 50 runs gets close to the optimum, there is no guarantee that the standard deviation will be small. This can be seen for instance in the random and Ising hamiltonians, where the standard deviation of the Rosalin plot (green) stays wide as the optimisation process advances, when compared to the noiseless plots. For other plots, it does become smaller, as it can be seen in the LCB CMA-ES plots (blue) for the random, Ising and GHZ hamiltonians.

5 Logistics

This chapter focuses on the implementation on a quantum computer of a problem relevant to industrial ecology. The chosen problem is the pickup and delivery problem with a focus on minimising emissions.

5.1. General pickup and delivery problem

The pickup and delivery problem (PDP) deals with the logistics of how to build a series of routes, each one satisfying a transportation request. A transportation request indicates what customers have to receive a certain good, and all of these customers have to be visited in the same route in the most efficient way. The mathematical formulation of the pickup and delivery problem is shown in equation 5.1. This formulation corresponds to the most general version of the pickup and delivery problem, and is explained in detail in [51]. In this general version, each vehicle has a starting and ending point (which can be different points), and a certain capacity that it can carry. For each transportation request, it specifies the the total load to be carried, the origins where the load is picked up (one or several locations) and the destinations (again one or several locations) where the load is to be delivered in order to complete the transportation request.

$$
\begin{array}{lrl}
\underset{x}{\text{minimize}} & f(x) & \\
\text{subject to} & \sum_{k\in M} z_i^k = 1, & i \in N, \\
& \sum_{j\in V\cup W} x_{lj}^k = \sum_{j\in V\cup W} x_{jl}^k = z_i^k, & i \in N, l \in N_i^+ \cup N_i^-, k \in M, \\
& \sum_{j\in V\cup\{k^-\}} x_{k^+j} = 1, & k \in M, \\
& \sum_{i\in V\cup\{k^+\}} x_{ik^-} = 1, & k \in M, \\
& D_{k^+} = 0, & k \in M, \\
& D_p \leq D_q, & i \in N, p \in N_i^+, q \in N_i^-, \\
& x_{ij}^k = 1 \Rightarrow D_i + t_{ij} \leq D_j, & i,j \in V \cup W, k \in M, \\
& y_l \leq \sum_{k\in M} Q_k z_i^k, & i \in N, l \in N_i^+ \cup N_i^-, \\
& x_{ij}^k = 1 \Rightarrow y_i = q_i = y_j, & i,j \in V \cup W, k \in M, \\
& x_{ij}^k \in \{0,1\}, & i,j \in V \cup W, k \in M, \\
& z_i^k \in \{0,1\}, & k = 0, \dots, N-1
\end{array}
\tag{5.1}
$$

After introducing the general pickup and delivery problem, it is now important to choose an objective function (what is to be minimised) that is adequate for the goal of minimising environmental impacts. In [52], the authors studied the pickup and delivery problem from a standpoint of minimising both cost and emissions. They found that the optimisation algorithms created different routes depending if the objective was to optimise for cost or for emissions. This suggests that current logistics routes could be made more environmentally friendly. In particular, the objective function that was used to measure the greenhouse gas emissions in [52] is defined in equation 5.2, and will be used in the case study presented next.

$$\sum_{(i,j)\in A} d_{ij}x_{ij}(\varepsilon_{CO_2} + \varepsilon_{NH_4} + \varepsilon_{N_2O}) \tag{5.2}$$

5.2. Case Study

The problem that will be studied is a specific case of the general pickup and delivery problem in described previously. It is defined as follows.

$$\begin{aligned} \min_{x_{ij}} \quad & \sum_{(i,j)\in A} d_{ij}x_{ij}(\varepsilon_{CO_2} + \varepsilon_{NH_4} + \varepsilon_{N_2O}) \\ \text{s.t.} \quad & \sum_{j\in V\cup W} x_{lj} = \sum_{j\in V\cup W} x_{jl}, \;\; l \in N^+ \cup N^-, \\ & \qquad \sum_{j\in V\cup\{1\}} x_{1j} = 1,, \\ & \qquad \sum_{i\in V\cup\{1\}} x_{i1} = 1, \end{aligned} \tag{5.3}$$

- The objective function estimates the amount of greenhouse gases that will be produced based on the total distance travelled. This equation is obtained from [52].
- d_{ij} is the distance between vertices i and j of the graph.
- x_{ij} is 1 if the vehicle goes from vertex i to vertex j in its route. If it does not, then the value is 0.
- ε_X refers to the environmental impacts of gas X.
- N^+ is the set of all pickup vertices (the warehouses) and N^- is the set of all delivery vertices (the customers).
- V is the set of all the pickup points and all the customer locations. V = $N^+ \cup N^-$.
- W is the set of all departure and arrival point of the trucks.
- Vertex 1 is the departure and arrival point of the truck.

The small problem instance that will be modelled is shown in Figure 5.1. Vertex 1 is the start and end point of the truck. In between it has to visit two warehouses (vertices 2 and 3) and then deliver the products to two customers (vertices 4 and 5). Regarding the edges that connect the vertices, each has an associated value. The values in the edges describe the distance between the vertices and, by knowing the distance that has been travelled for a certain route, the emissions can be calculated as in equation 5.2. In the figure, the optimal route is showed with green arrows ($1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 5 \Rightarrow 4 \Rightarrow 1$).

5.3. How is the problem solved with VQAs

To be able to work with the pickup and delivery problem on a quantum computer, it is necessary to transform it into an equivalent hamiltonian. In particular, the objective is to encode the problem as an Ising hamiltonian. With this objective in mind and following the steps described in the paper [13], the process of building an adequate hamiltonian is summarised here.

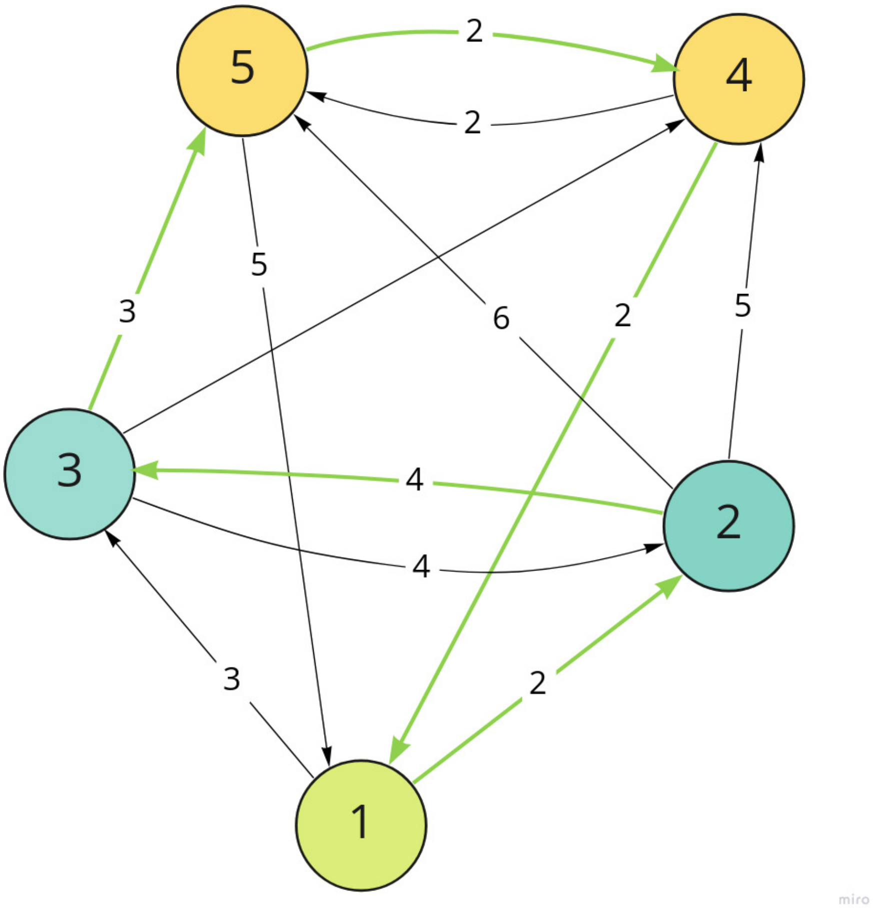

Figure 5.1: Routes for the pickup and delivery problem. Vertex 1 is the starting and ending point, 2 and 3 are the warehouses and 4 and 5 are the clients. The edges of the graph indicate the distance between the vertices. Overall, the optimal route is $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 1$.

To begin with, it is necessary to encode the logistics problem as a quadratic unconstrained binary optimisation problem (QUBO). This problem is of the form showed in 5.4, where the objective is to find a vector of binary variables that minimises the objective function, with M being a matrix and c a constant.

$$\min_{x} \quad \mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x} + c \quad \text{s.t.} \quad \mathbf{x} \in \{0,1\}^{n} \tag{5.4}$$

The reason why the formulation of the QUBO problem is interesting is because has the same form as the Ising Hamiltonian, which can be effectively studied using quantum computers. In the case of the Ising hamiltonian, the elements of the x vector will be qubits and the objective is to vary the state of these qubits so that the hamiltonian (or the QUBO problem) is minimised. To find these states of interest one can use algortihms such as LCB or Rosalin.

$$\min_{x}\{\mathbf{c}^{\mathsf{T}}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \{0,1\}^{n}\} \tag{5.5}$$

Now another question arises, which is how the logistics problem can be converted into a QUBO problem. According to [13], because the mathematical formulation showed in equation 5.3 is of the type 5.5, where all the constraints are linear and have equalities ($=$ rather than $\geq$ or $\leq$), these constraints can be encoded as a penalty function in the Hamiltonian. This is shown in the second term of 5.6, where the penalty term ($\rho\|Ax - b\|^2$) will be bigger the more the vector x deviates from the constraints, thus increasing the value of the hamiltonian and moving it further away from the minimisation objective.

$$H : \mathbf{x} \mapsto \mathbf{c}^{\mathsf{T}}\mathbf{x} + \rho\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^{2} \tag{5.6}$$

It is also important to balance the weight of the cost function ($\mathbf{c}^{\mathsf{T}}\mathbf{x}$) and the penalty term in the hamiltonian. This is done with the constant ρ and it has to be big enough such that in the case that we find an infeasible solution that does not match all the equality constraints, then the penalty must be bigger than the possible gain in $\mathbf{c}^{\mathsf{T}}\mathbf{x}$ provided by that infeasible solution. This is guaranteed by choosing a ρ such that $\rho > \sum_i |c_i|$. With this value of ρ, minimisng the hamiltonian is equivalent to minimising the logistics problem.

The last step is to write this hamiltonian is the QUBO form, which is done by using M as defined in equation 5.7.

$$\mathbf{M} = \rho\mathbf{A}^{\mathsf{T}}\mathbf{A} + \rho\mathbf{Diag}(-2\mathbf{A}^{\mathsf{T}}\mathbf{b}) + \mathbf{Diag}(\mathbf{c}) \tag{5.7}$$

5.4. Experiments and results

The steps explained in the previous section show how to calculate the matrix M for the test case depicted in figure 5.1. This matrix is shown in equation 5.8. With this matrix, the next step is to create the hamiltonian that contains the information of the logistics problem. The resulting hamiltonian is shown in equation 5.9. It can be seen that the shape of this hamiltonian has linear (Z_i) and quadratic (Z_iZ_j) terms, so it follows the QUBO formulation $x^{\mathsf{T}}Mx$ (when the decision variables x are multiplied by matrix M and then again by x, the quadratic terms are obtained). Now that the QUBO formulation is obtained, it can be transformed into the Ising model by changing the decision variables x in QUBO with a vector of Z-Paulis in Ising. The difference between both is that the decision variables in QUBO are binary (0 or 1) whereas the decision variables in Ising can be a number in the interval [-1,1] (because the expectation of the Z operator is somewhere between 1 and -1).

$$\begin{pmatrix}
-84 & 43 & 0 & 43 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 \\
43 & -83 & 43 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 \\
0 & 43 & -82 & 0 & 43 & 43 & 0 & 0 & 0 & 0 \\
0 & 0 \\
43 & 0 & 0 & -82 & 0 & 0 & 43 & 43 & 0 & 0 \\
0 & 0 \\
0 & 0 & 43 & 0 & -81 & 43 & 43 & 0 & 0 & 43 \\
0 & 0 \\
0 & 0 & 43 & 0 & 43 & -80 & 0 & 43 & 43 & 0 \\
0 & 0 \\
0 & 0 & 0 & 43 & 43 & 0 & -82 & 43 & 0 & 43 \\
0 & 0 \\
0 & 0 & 0 & 43 & 0 & 43 & 43 & -83 & 43 & 0 \\
0 & 0 \\
0 & 0 & 0 & 0 & 0 & 43 & 0 & 43 & -84 & 0 \\
43 & 0 \\
0 & 0 & 0 & 0 & 43 & 0 & 43 & 0 & 0 & -84 \\
0 & 43 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 43 & 0 \\
-84 & 43 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 43 \\
43 & -81
\end{pmatrix} \tag{5.8}$$

$$\begin{aligned}
&- Z_0 - 1.5Z_1 - 23.5Z_2 - 23.5Z_3 - 45.5Z_4 - 46Z_5 - 45Z_6 - 44.5Z_7 - 22.5Z_8 - 22.5Z_9 - Z_{10} \\
&- 2.51Z_{11} + 21.5Z_0Z_1 + 21.5Z_0Z_3 + 21.5Z_1Z_2 + 21.5Z_2Z_4 + 21.5Z_2Z_5 + 21.5Z_{3Z_6} + 21.5Z_3Z_7 \\
&+ 21.5Z_4Z_5 + 21.5Z_4Z_6 + 21.5Z_4Z_9 + 21.5Z_5Z_7 + 21.5Z_5Z_8 + 21.5Z_6Z_7 + 21.5Z_6Z_9 + 21.5Z_7Z_8 \\
&+ 21.5Z_8Z_{10} + 21.5Z_9Z_{11} + 21.5Z_{10}Z_{11}
\end{aligned} \tag{5.9}$$

Before switching the problem from classical (QUBO) to quantum (Ising), the problem was understood as if a decision variable $x_i \in$ **x** is 0, then the vehicle does not travel along that edge of the graph, and if it is 1 then it does travel. Now that it is translated to Ising, if the expectation of a Pauli Z_i is close to -1, then the vehicle travels along that edge, and if the expectation is close to -1, then it does not travel along the edge. This leaves some doubt about what to do with those expectation values that are far away from -1 and 1 (for instance close to 0), but this will be discussed later in the results.

$$x_i = \frac{1 - z_i}{2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{5.10}$$

The Z Pauli matrix, as defined in chapter 3, has 1 and -1 as eigenvalues, which are respectively associated to the eigenvectors $|0\rangle$ and $|1\rangle$. Because the decisions variables x_i can only take values 0 and 1, the Z Paulis are not equivalent because the eigenvalues are different. For this reason, the transformation shown in equation 5.10 is needed to create an operator with eigenvalues 0 and 1.

When the QUBO problem ($x^T Mx$) is expanded, i.e. the multiplication of vectors and matrices is done, the x_i cannot be directly switched with Z_i, and instead need to be transformed as indicated in equation 5.10 to represent the QUBO as a hamiltonian. When x_i is transformed into the variable $z_i \in 1, -1$, this variable can then be substituted by the Pauli Z matrix.

Decision variable x_{ij}	Optimal x_{ij}	Equivalent Pauli Z_i	Optimal Z_i
x_{12}	1	Z_0	-1
x_{13}	0	Z_1	1
x_{23}	1	Z_2	-1
x_{32}	0	Z_3	1
x_{24}	0	Z_4	1
x_{25}	0	Z_5	1
x_{34}	0	Z_6	1
x_{35}	1	Z_7	-1
x_{45}	0	Z_8	1
x_{54}	1	Z_9	-1
x_{41}	1	Z_{10}	-1
x_{51}	0	Z_{11}	1

Figure 5.2: Table showing the optimal solution for the pickup and delivery problem. The x_{ij} decision variables represent if the optimal route goes from vertex i to j (1 yes, 0 no). To calculate the x_{ij} decisions, first they are expressed in terms of Pauli Zi, which can be optimised by the VQAs.

$$\begin{aligned}
&\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \\
&= x_1^2 + x_2^2 + 4x_1x_2 = \\
&= x_1 + x_2 + 4x_1x_2 = \\
&= \frac{1-z_1}{2} + \frac{1-z_2}{2} + \frac{1-z_1}{2}\frac{1-z_2}{2} = \\
&\frac{5}{4} - \frac{3z_1}{4} - \frac{3z_2}{4} + \frac{z_1z_2}{4}
\end{aligned} \tag{5.11}$$

An example of the transformation from QUBO to Hamiltonian is shown the equalities shown in equation 5.11. The calculations start with an example of a simple QUBO problem (x^TMx) which is expanded into a quadratic equation. It is important to notice that from the second to the third expression the only change is the change of x_i^2 is changed to x_i to simplify the expression. This can be done because the values of x_i can be 0 and 1, and when these two values are squared, the result is the same ($1^2 = 1$ and $0^2 = 2$). Then, x_i is changed to $\frac{1-z}{2}$. This is equivalent because if $z_i = 1$, then $\frac{1-z_i}{2} = 0$ and if $z_i = -1$, then $\frac{1-z_i}{2} = 1$. This shows that both are equivalent. The final expression in the equation can be optimised with a VQA.

To test the solving method that have been described, several experiments are created. Since the results of the benchmarking did not show a clear advantage between the two tested algorithms, both will be tested on the simple instance of the problem. In total there are six experiments, showed in table 5.1. Three of them using LCB CMA-ES and the other three with Rosalin. The biggest difference of the experiments is on the number of parameters in the circuit that have to be optimised. The same structure of circuit is used (shown in section 4.3), with the only difference being the number of repetitions of the main block. As in figure 4.2, the main block is indicated between brackets ({}) and if several are used, they are placed one after the other in the circuit. The different experiments are tested on circuits with 1, 3 and 5 repetitions of the main block..

The results are expressed in figures 5.3 and 5.4. The first figure indicates how the objective

Experiment	Algorithm	# shots	Circuit depth
LCB-1	LCB CMA-ES	1.000.000	1
LCB-3	LCB CMA-ES	1.000.000	3
LCB-5	LCB CMA-ES	1.000.000	5
Ros-1	Rosalin	1.000.000	1
Ros-3	Rosalin	1.000.000	3
Ros-5	Rosalin	1.000.000	5

Table 5.1: Design of experiments for the pickup and delivery problem.

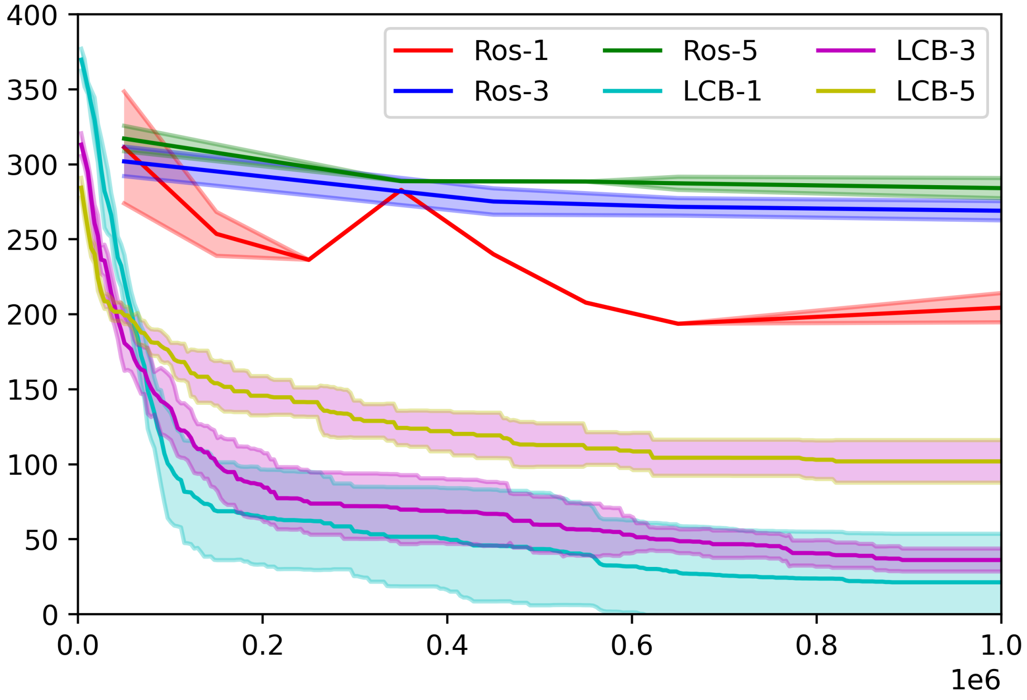

Figure 5.3: Evolution of the cost function for the different experiments. Each of the graphs is calculated by averaging five runs of the same experiment. The band around each graph corresponds to the standard deviation of the obtained results.

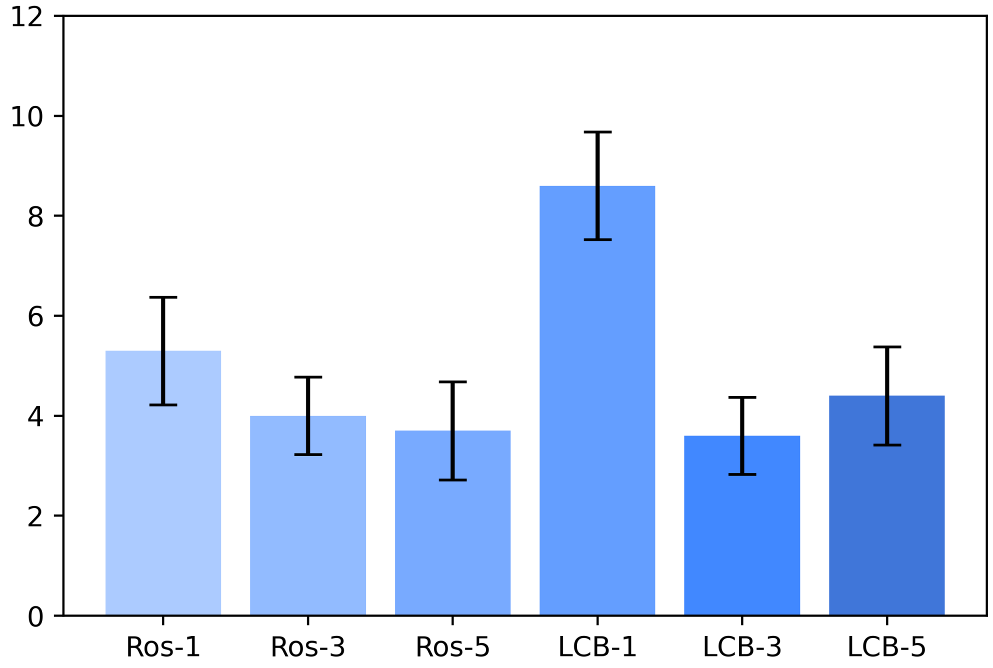

Figure 5.4: Bar graph indicating the average number of correct decisions made for the pickup and delivery problem. With a total of 12 binary decisions ($x \in \{0, 1\}$), the best solution is the experiment LCB-1 with an average result of 8.6.

function is optimised with respect to the number of shots used. It can be seen that Ros-1, Ros-3, Ros-5 reach a plateau early on and stay there for the rest of the evaluation. There may be several reasons for this. In the case of the Rosalin experiments, the adaptive nature of the algorithm is useful in some cases where it estimates the cost function with a low amount of shots, which may cause some uncertainty in the evaluations but it allows for more iterations. This adaptive strategy worked well for the hamiltonians in the benchmarking section, but with this more complex hamiltonian (equation 5.9), the adaptive strategy is consuming a high number of shots per iteration, which can be seen in table 5.3. The Rosalin experiments do less than 5 iterations on average with the given budget of one million shots. With this few iterations, it is not possible find the optimum parameters for such a big circuit, which explains why none of the Rosalin experiments arrive to a low value for the cost function. Moreover, it can be seen that the last iteration of all the Rosalin experiments explodes in terms of required shots. For instance, the last iteration in one of the Ros-3 experiments requires almost 4 million shots to find the new set of parameters, as shown in red in table 5.1.

On the other hand, this issue does not occur with the experiments that use the LCB CMA-ES algorithm, since the number of shots per iteration is fixed, guaranteeing a high number of iterations, which increase the possibility of obtaining a better value for the cost function. It can be observed in figure 5.3 that the LCB CMA-ES experiments perform significantly better than those of Rosalin. Of the three LCB CMA-ES experiments, the best results are obtained in LCB-1 and LCB-3. In LCB-1, the average of the results gets closer to the optimum than the average of LCB-3, but the difference is not large. As for the standard deviations, it is considerably larger for LCB-1 than LCB-3. This may indicate that the circuit used in LB-3 yields more similar results, whereas the results for LCB-1 are more scattered around. However, this cannot be confirmed because the number of runs per experiment is rather low (only 5). It is possible that with more runs per experiment the standard deviations vary. Overall, with the available results it is clear that the smaller circuits of LCB-1 and LCB-3 provide the best results when compared with the bigger circuit of LCB-5. With respect to the optimal solution, LCB-1 and LCB-3 get relatively close to the optimum and, because the graphs seem to converge, this indicates that further

Difference	Value added
$\left\|x - \frac{I-Z}{2}\right\| < 0.25$	1
$0.25 < \left\|x - \frac{I-Z}{2}\right\| < 0.5$	0.5
$\left\|x - \frac{I-Z}{2}\right\| > 0.5$	0

Table 5.2: Design of experiments for the pickup and delivery problem.

improvement is improbable even if more shots were allocated. Given that the hamiltonian to optimise is quite complex, the results achieved can be considered satisfactory, but there is still possibility for improvement.

After the optimisation process finishes, the algorithms return a final set of parameters for the parametrised circuit. By running the circuit with the final set of parameters, a final state $|\psi\rangle$ is obtained. It is with this state that the measurements can be made to obtain the results. As explained in chapter 3, operators are used to measure observables from a quantum state. Because the decision variables are binary, the observable that is used is $\frac{I-Z}{2}$, which has eigenvalues 0 and 1. In table **??**, it is explained which decision variables x are represented by which Z operators. However, because the eigenvalues of Z (1, -1) are different from the values of x (0, 1) it is necessary to make the transformation showed in equation 5.10.

With a total of 12 decision variables for the pickup and delivery problem, figure 5.4 shows the average number of correct decisions for the decision variables. Because the expectation of the operator $\frac{I-Z}{2}$ can be anywhere between 0 and 1, a criteria to determine when the expectation is correct is needed. For this work, the following criteria is followed: if $\left|x - \frac{I-Z}{2}\right| < 0.25$, then the answer provided by the expectation of the observable is considered to be correct. If $0.25 < \left|x - \frac{I-Z}{2}\right| < 0.5$, then the distance beging to be considerable and the answer is ”half-correct”. Finally, if $\left|x - \frac{I-Z}{2}\right| > 0.5$, it is considered that the solution is wrong. This criteria is showed in table 5.2.

From this table, it is clear that if all the expectation values of the Z operators showed in table 5.2 are close to the corresponding decision variable x, then the maximum result is 12 if all the decisions are correct. Figure 5.4 shows the average correct results of each of the experiments. Regarding the Rosalin experiments, the low results all below 6 are expected because the cost function was not optimised close to the optimum in any of the cases. However, the case of the LCB CMA-ES experiments is different from what could be expected. Because the average plot of all the experiments get close to the optimum, it would be logical that this in turn shows positive results in the bar graph of high numbers close to 12. However, this is not the case. Overall, the best performing experiment is LCB-1, with an average result of 8.6 correct decisions out of 12. This result is better than the rest but still not of the quality that would be desired.

A possible explanation is that the logistics problem is not perfectly encoded in the matrix M, and thus the hamiltonian that is being optimised is not exactly the correct one. It would be useful to see other ways to encode the logistics problem into a matrix and see if this would improve the results.

Experiment	Example shots of a run	Average # iterations
LCB-1	3600, 7200, 10800...	250
LCB-3	3600, 7200, 10800...	250
LCB-5	3600, 7200, 10800...	250
Ros-1	2220, 335780, 2137648	4.4
Ros-3	6180, 621146, 4203748	3
Ros-5	10140, 390320, 2958712	3.6

Table 5.3: Shot distribution and number of iterations for the six experiments. Because LCB CMA-ES has a fixed number of shots per iteration, the shots are distributed equally on all runs (3600 shots per iteration). In the case of Rosalin, the number of shots is adaptive and it can be seen how it increases very fast. This can be seen in the average number of shots for the 5 runs of a given experiment (4.4, 3 and 3.6). This low number of iterations difficults the optimisation results, as more iterations are needed to improve the solution.

6

Conclusions and recommendations

6.1. Conclusions

This work has focused on the current state of quantum computers and what kind of results relevant to industrial ecology can be obtained in the present. In order to answer the main research question *"How can quantum computing benefit industrial ecology and how can it be applied to a relevant problem?"*, first the three sub-research questions will be answered.

Firstly, for Sub-RQ 1 *"what industrial ecology problems can benefit from quantum computing?"*, a series of common industrial ecology problems that can benefit from quantum computing have been found. Examples are the optimisation of flows in an industrial park, the management of data centres to reduce energy needs. Moreover, the literature review illustrated different techniques to tackle these problems, such as quantum machine learning and quantum information theory. The possibilities are broad, and it is possible that there are numerous other potential applications. The downside is that, given that the possibilities of quantum computing are still limited in terms of computational capacity, classical computers are a better option currently. However, it is an idea to keep in mind and perhaps start experimenting.

Secondly, for sub-RQ 2 *"what is the state-of-the-art of variational quantum algorithms and how do they perform?"*, two VQAs are presented: Rosalin and LCB CMA-ES. The former is a gradient-descent based method whereas the latter is a population-based method. Both algorithms work on a classical computer to optimise the parameters of a parametrised quantum circuit. For this reason, both algorithms need to be equipped with tools to handle the noise from the quantum side. Rosalin does this by distributing shots in an adaptive way, in the most efficient way considered by the algorithm. On the other side, LCB CMA-ES handles the quantum noise by calculating the lower confidence bound for every candidate solution. On the practice, four test problems were presented to compare the performance of both algorithms. The results found that both algorithms perform well in the test cases and there is no clear advantage of one over the other. One is slightly better for some test cases and the other slightly better in the other cases.

As for sub-RQ 3 *"how do these algorithms perform in the task of optimising a green logistics problem?"*, the chosen problem is the pickup and delivery problem, where a delivery truck has to visit a series of warehouses to pick up goods and then distribute those over a series of clients in the most efficient way. The objective is to find the optimal route so that it minimises the total emissions. In order to solve the problem, it is first necessary to encode it into a hamiltonian which can be optimised with a VQA. After encoding the problem, the resulting hamiltonian is considerably more complex than those of the test cases. There are six different experiments done, three with each algorithm. Of those three, the difference is the size of the parametrised circuit to be optimised. Among all experiments, LCB CMA-ES with circuit of depth 1 gives the best results, while the other LCB CMA-ES experiments do not do as well. This is unexpected because all the LCB CMA-ES experiments are able to optimise successfully the expectation of the hamiltonian. This indicates that it is possible that the encoding of the routing problem is not optimal. Regarding the Rosalin experiments, they are not able to achieve a desirable result due to the adaptive nature of this algorithm, which does not allow a sufficient number of iterations of the optimisation process.

At this point, it is possible to answer the main RQ *“How can quantum computing benefit industrial ecology and how can it be applied to a relevant problem?”*. The results have found that there are numerous industrial ecology problems where quantum computing can be helpful in the future. Moreover, a real case study is provided which illustrates how to solve a problem with a VQA in the practice. Even though the results are not as good as it would be desired, it is possible that the encoding does not exactly translate the routing problem into a hamiltonian. The reason for this is that LCB CMA-ES is optimising the expectation of the hamiltonian successfully, but this is not seen later when obtaining the results to the problem.

6.2. Recommendations

- Since quantum computing can be combined with machine learning and other types of algorithms, there will be numerous applications relevant for industrial ecology different from the ones that have been found here. Looking for possible linkages between industrial ecology and quantum computing may create novel results in the future to improve sustainability.
- A main takeaway from the results is that given a problem to solve, the results will depend on the algorithm used and the structure of the circuit. For this reason, it is useful to compare several options as it is done in the benchmarking experiments to get an idea of the performance of an algorithm.
- Regarding the process of simulating a quantum computer, the number of qubits that can be used in a classical computer is rather low. By having a restricted maximum number of qubits, the size of the problems that can be studied is small too. In spite of the size limits, it is useful to start getting familiar with quantum computing to be able to get the most benefit when powerful quantum computers are more accessible.

References

[1] A IPCC. *IPCC Fifth Assessment Report—Synthesis Report*. 2014.

[2] Marcus Lee and Salif Diop. “Millennium ecosystem assessment”. In: *An Assessment of Assessments: Findings of the Group of Experts Pursuant to UNGA Resolution 60/30* 1 (2009), p. 361.

[3] SPECIAL WORKING SESSION WCED. “World commission on environment and development”. In: *Our common future* 17.1 (1987), pp. 1–91.

[4] Paulo Savaget et al. “The theoretical foundations of sociotechnical systems change for sustainability: A systematic literature review”. In: *Journal of cleaner production* 206 (2019), pp. 878–892.

[5] Carla Gomes et al. “Computational sustainability: Computing for a better world and a sustainable future”. In: *Communications of the ACM* 62.9 (2019), pp. 56–65.

[6] Robert Colwell. “The chip design game at the end of Moore’s law”. In: *2013 IEEE Hot Chips 25 Symposium (HCS)*. IEEE Computer Society. 2013, pp. 1–16.

[7] Sergio Boixo et al. “Characterizing quantum supremacy in near-term devices”. In: *Nature Physics* 14.6 (2018), pp. 595–600.

[8] Thomas E O’Brien et al. “Calculating energy derivatives for quantum chemistry on a quantum computer”. In: *npj Quantum Information* 5.1 (2019), pp. 1–12.

[9] Emanuel Knill. “Quantum computing”. In: *Nature* 463.7280 (2010), pp. 441–443.

[10] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. “Average-case complexity versus approximate simulation of commuting quantum computations”. In: *Physical review letters* 117.8 (2016), p. 080501.

[11] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. “Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2126 (2011), pp. 459–472.

[12] Utkarsh Azad et al. “Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm”. In: *IEEE Transactions on Intelligent Transportation Systems* (2022).

[13] Stuart Harwood et al. “Formulating and solving routing problems on quantum computers”. In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 1–17.

[14] Fernando Hernández et al. “Application of quantum optimization techniques (QUBO method) to cargo logistics on ships and airplanes”. In: *2020 IEEE Congreso Bienal de Argentina (ARGENCON)*. IEEE. 2020, pp. 1–1.

[15] Jenann Ismael. “Quantum Mechanics”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2021. Metaphysics Research Lab, Stanford University, 2021.

[16] JC Solem and LC Biedenharn. “Understanding geometrical phases in quantum mechanics: An elementary example”. In: *Foundations of physics* 23.2 (1993), pp. 185–195.

[17] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[18] *The Value of Classical Quantum Simulators*. `https://ionq.com/posts/november-12-2021-the-value-of-quantum-simulators`. Accessed: 2022-08-25.

[19] Engineering National Academies of Sciences, Medicine, et al. “Quantum computing: progress and prospects”. In: (2019).

[20] Reinier W Heeres et al. “Implementing a universal gate set on a logical qubit encoded in an oscillator”. In: *Nature communications* 8.1 (2017), pp. 1–7.

[21] Rami Barends et al. “Superconducting quantum circuits at the surface code threshold for fault tolerance”. In: *Nature* 508.7497 (2014), pp. 500–503.

[22] John Preskill. “Quantum computing in the NISQ era and beyond”. In: *Quantum* 2 (2018), p. 79.

[23] *IBM’s 127-Qubit Eagle Is the Biggest Quantum Computer Yet*. `https://singularityhub.com/2021/11/22/ibms-127-qubit-eagle-is-the-biggest-quantum-computer-yet/`. Accessed: 2022-08-29.

[24] Andrew Arrasmith et al. “Operator sampling for shot-frugal optimization in variational algorithms”. In: *arXiv preprint arXiv:2004.06252* (2020).

[25] Marco Cerezo et al. “Variational quantum algorithms”. In: *Nature Reviews Physics* 3.9 (2021), pp. 625–644.

[26] *What is industrial ecology?* `https://is4ie.org/about/what-is-industrial-ecology`. Accessed: 2022-07-09.

[27] Cynthia Olvera, Oscar Montiel, and Yoshio Rubio. “Quantum-inspired evolutionary algorithms on continuous space multiobjective problems”. In: *Soft Computing* (2022), pp. 1–22.

[28] Oscar H Montiel Ross. “A review of quantum-inspired metaheuristics: going from classical computers to real quantum computers”. In: *IEEE Access* 8 (2019), pp. 814–838.

[29] Chao Gu et al. “Multi-objective optimization for industrial ecology: design and optimize exchange flows in an industrial park”. In: *Proceedings of the 2013 international conference on applied mathematics and computational methods (AMCM 2013)*. 2013, pp. 109–116.

[30] Marianne Boix et al. “Industrial water management by multiobjective optimization: from individual to collective solution through eco-industrial parks”. In: *Journal of Cleaner Production* 22.1 (2012), pp. 85–97.

[31] Ehsan Zahedinejad and Arman Zaribafiyan. “Combinatorial optimization on gate model quantum computers: A survey”. In: *arXiv preprint arXiv:1708.05294* (2017).

[32] Gary Kochenberger et al. “The unconstrained binary quadratic programming problem: a survey”. In: *Journal of combinatorial optimization* 28.1 (2014), pp. 58–81.

[33] Davide Venturelli et al. “Quantum optimization of fully connected spin glasses”. In: *Physical Review X* 5.3 (2015), p. 031040.

[34] Helmut G Katzgraber et al. “Seeking quantum speedup through spin glasses: The good, the bad, and the ugly”. In: *Physical Review X* 5.3 (2015), p. 031026.

[35] Abdelkader Sbihi and Richard W Eglese. “Combinatorial optimization and green logistics”. In: *Annals of Operations Research* 175.1 (2010), pp. 159–175.

[36] Priya L Donti and J Zico Kolter. “Machine learning for sustainable energy systems”. In: *Annual Review of Environment and Resources* 46.1 (2021).

[37] Chris Davis and Graham Aid. “Machine learning-assisted industrial symbiosis: Testing the ability of word vectors to estimate similarity for material substitutions”. In: *Journal of Industrial Ecology* 26.1 (2022), pp. 27–43.

[38] Zhen Yang et al. “Increasing the energy efficiency of a data center based on machine learning”. In: *Journal of Industrial Ecology* 26.1 (2022), pp. 323–335.

[39] Yao Zhang and Qiang Ni. “Recent advances in quantum machine learning”. In: *Quantum Engineering* 2.1 (2020), e34.

[40] Thomas J Elliott et al. “Quantum adaptive agents with efficient long-term memories”. In: *Physical Review X* 12.1 (2022), p. 011007.

[41] Robert L Axtell, Clinton J Andrews, and Mitchell J Small. “Agent-based modeling and industrial ecology”. In: *Journal of Industrial Ecology* 5.4 (2001), pp. 10–13.

[42] Vahid Beiranvand, Warren Hare, and Yves Lucet. “Best practices for comparing optimization algorithms”. In: *Optimization and Engineering* 18.4 (2017), pp. 815–848.

[43] Xavier Bonet-Monroig et al. “Performance comparison of optimization methods on variational quantum algorithms”. In: *arXiv preprint arXiv:2111.13454* (2021).

[44] Kunal Sharma et al. “Noise resilience of variational quantum compiling”. In: *New Journal of Physics* 22.4 (2020), p. 043006.

[45] Andrew Jena, Scott Genin, and Michele Mosca. “Pauli partitioning with respect to gate sets”. In: *arXiv preprint arXiv:1907.07859* (2019).

[46] Jonas M Kübler et al. “An adaptive optimizer for measurement-frugal variational algorithms”. In: *Quantum* 4 (2020), p. 263.

[47] Aram W Harrow and John C Napp. “Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms”. In: *Physical Review Letters* 126.14 (2021), p. 140502.

[48] Nikolaus Hansen. “The CMA evolution strategy: A tutorial”. In: *arXiv preprint arXiv:1604.00772* (2016).

[49] Giovanni Gallavotti. *Statistical mechanics: A short treatise*. Springer Science & Business Media, 1999.

[50] Elaine Barker et al. *Recommendation for key management: Part 1: General*. National Institute of Standards and Technology, Technology Administration …, 2006.

[51] Martin WP Savelsbergh and Marc Sol. “The general pickup and delivery problem”. In: *Transportation science* 29.1 (1995), pp. 17–29.

[52] Siwaporn Kunnapapdeelert and Ratchaphong Klinsrisuk. “Determination of green vehicle routing problem via differential evolution”. In: *International Journal of Logistics Systems and Management* 34.3 (2019), pp. 395–410.