ALMA MATER STUDIORUM · UNIVERSITY OF BOLOGNA

School of Science
Department of Computer Science and Engineering
Master's Degree in Computer Science

A clustering aggregation algorithm on neutral-atoms and annealing quantum processors

Supervisor:
Prof. Stefano Lodi

Submitted by:
Riccardo Scotti

Cosupervisors:
Dr. Gabriella Bettonte
Dr. Antonio Costantini

Session II
Academic Year 2023/2024

"The last few steps are always the darkest and most difficult."

Dale Cooper, *Twin Peaks*

UNIVERSITY OF BOLOGNA

Abstract

School of Science
Department of Computer Science and Engineering

Master's Degree in Computer Science

A clustering aggregation algorithm on neutral-atoms and annealing quantum processors

by Riccardo Scotti

This work presents a hybrid quantum-classical algorithm to perform clustering aggregation, designed for neutral-atoms and quantum annealing processors. Clustering aggregation is a technique that mitigates the weaknesses of clustering algorithms, an important class of data science methods to partition datasets, widely employed in many industries. By expressing the problem as a Maximum Independent Set (MIS) problem and as a Quadratic Unconstrained Binary Optimization (QUBO) problem with additional constraints, it was possible to submit it to Pasqal's Fresnel (neutral-atoms processor) and D-Wave's Advantage QPU (quantum annealer); additionally, the algorithm was also tested on an emulation of Fresnel running on the QuTiP framework. Results revealed technical limitations, such as the difficulty of adding additional constraints on the employed neutral-atoms platform, the necessity of better metrics to measure the quality of produced clusterings, and the slow performance on the emulator. However, findings suggest a promising potential for future advancements in hybrid quantum-classical pipelines, though further improvements are needed in both quantum and classical components.

UNIVERSITÀ DI BOLOGNA

Sommario

Scuola di Scienze
Dipartimento di Informatica – Scienza e Ingegneria

Laurea Magistrale in Informatica

A clustering aggregation algorithm on neutral-atoms and annealing quantum processors

di Riccardo Scotti

Questo lavoro presenta un algoritmo ibrido quantistico-classico per effettuare clustering aggregation, progettato per processori con tecnologie ad atomi neutri e quantum annealing. La clustering aggregation è una tecnica atta a mitigare le debolezze degli algoritmi di clustering, un'importante classe di metodi di data science capaci di partizionare dataset, ampiamente adoperati in numerosi campi industriali. Esprimendo il problema sotto forma di Maximum Independent Set (MIS) e di Quadratic Unconstrained Binary Optimization (QUBO) con vincoli addizionali, è stato possibile sottometterlo a Fresnel (processore ad atomi neutri), prodotto da Pasqal, e ad Advantage QPU (quantum annealer), prodotto da D-Wave; l'algoritmo è stato inoltre testato su un emulatore di Fresnel basato sul framework QuTiP. I risultati hanno rivelato limitazioni tecniche, come ad esempio la difficoltà nell'includere vincoli addizionali sulla piattaforma ad atomi neutri adoperata, la necessità di ricercare metriche più adatte per misurare la qualità dei clustering prodotti, e la lentezza dell'emulazione. Allo stesso tempo, i risultati suggeriscono un potenziale promettente per futuri sviluppi nelle pipeline ibride quantistico-classiche, sebbene siano necessari ulteriori miglioramenti sia nelle componenti quantistiche che in quelle classiche.

Acknowledgements

This work is the result of the collaboration of many people, without whom this endeavor would have not been possible.

First and foremost, I would like to thank my supervisor, Prof. Stefano Lodi, for his guidance and for always being available when I needed his help.

Dr. Gabriella Bettonte, my cosupervisor and tutor, accompanied me throughout my internship at CINECA and during the preparation of this thesis. I cannot thank her enough for everything she has done for me, and for all the advice she has given me.

I am also deeply grateful to Dr. Antonio Costantini, my cosupervisor, for helping me tirelessly and for everything he has taught me about quantum mechanics.

I would like to thank Dr. Daniele Ottaviani, Dr. Sara Marzella, and all the people from CINECA Quantum Computing Lab. They treated me as one of their peers, showed great interest in this research, and supported me constantly.

Dr. Simone Tibaldi kindly shared with me his experience with Fresnel and gave me important advice on the optimization of parameters, for which I am enormously thankful.

I want to thank my university colleagues and friends Daniela, Christian, Mike, Akira, Chiara, Cliff and Frederick, for working with me on various projects in these years, and for the nice moments spent together.

Special thanks to Daniele and Paolo, for proofreading the manuscript of this thesis, and to Antonio, for helping me with debugging.

Finally, I want to dedicate my biggest thank you to my family and my friends, for all the love they never cease to give me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum computing is a promising branch of computer science, offering a completely new paradigm of computation and a different perspective on how information is encoded and manipulated. Since its first proposal in 1982, in a seminal paper by Richard Feynman [1], different areas of research were able to produce remarkable discoveries, demonstrating that quantum technologies have the potential to revolutionize numerous fields, for example cryptography [2], machine learning [3], materials science [4] and many more.

Despite the numerous and groundbreaking theoretical results, quantum utility, or the usage of quantum computers to solve practical problems, is still far from reality, mainly due to technical limitations in the construction of quantum machines. At the same time, the presence of quantum technologies in industry is moving its first steps, with large companies beginning to include them as part of their environment, and large investments coming both from private and public organizations. As an example, the European High Performance Computing Joint Undertaking (EuroHPC JU), a major initiative by the European Union aimed at developing and supporting a large supercomputing infrastructure across Europe, has recognized the potential of quantum computing, and part of its strategic agenda is the integration of such technologies in the supercomputing ecosystems of the continent [5], not as replacements of the classical processors but as accelerators, similarly to the role that GPUs have taken in the HPC field. Given that the current technological landscape of quantum computing is uncertain and fragmented, with many available typologies of quantum machines but no clear superior one, EuroHPC is integrating a different technology in every supercomputing center, with the idea of allowing researchers and industries to choose the machine better suited for the problem to address.

Some of the most promising results in the practical application of quantum computing are coming from a class of quantum computers called quantum annealers, special purpose machines particularly well suited to solve optimization problems by leveraging concepts such as the adiabatic theorem and quantum tunneling, making them of great interest not only for academia, but also for industries where solving optimization problems is paramount, such as logistics, finance and chemistry.

A similar class of quantum machines that has emerged in the latest years is that of neutral-atoms based quantum computers, machines that can work both as general purpose, gate-based quantum computers (in the so called digital mode), as well as special purpose machine able to solve optimization problems (analog mode).

This work focuses on the development of an algorithm to perform clustering aggregation, a data analysis task, designed specifically for neutral-atoms and annealing processors. Clustering is a ubiquitous data science technique, with important applications in critical fields such as image processing, marketing and energy distribution; many clustering algorithms, however, present weaknesses when working on certain data distributions, which can be mitigated by clustering aggregation, a method that produces a unique and more robust solution from the results of multiple clustering algorithms. Apart from its industrial relevance and the possibility of improving its scalability, clustering aggregation is an ideal candidate for the design of a hybrid algorithm that employs neutral-atoms and annealing processors due to the possibility of expressing it as a Maximum Independent Set problem, an optimization problem that is naturally solved by neutral-atoms machines, as well as a QUBO problem, a class of optimization problems that are easily tackled by quantum annealers.

Being able to run an algorithm on multiple machines, built with different technologies, is not usually feasible in research, given the high costs of such systems, but the collaboration between CINECA (Italian University Consortium for Computing), the French company Pasqal and the Canadian company D-Wave, has allowed us to test the algorithm on real hardware, namely Pasqal's neutral-atoms based processor Fresnel, and D-Wave's quantum annealer Advantage.

Testing the algorithm on these machines serves not only as a demonstration of the utility of such machines within a hybrid pipeline, and possibly as a first step towards a larger scale usage in industry, but also as a tool to measure performances and compare the strengths and weaknesses of technologies from a practical point of view, especially considering how the benchmark of quantum computers is still an open research area with no single and well established methodology.

The thesis is structured as follows: Chapter 2 is a review of the fundamental theoretical notions necessary for this work, such as clustering, optimization problems, basics of quantum computing and quantum annealing, and an overview of the neutral-atoms technology. Chapter 3 describes the context within which this work was developed, reviewing scientific literature regarding clustering, role of quantum computing in industry and as accelerators, and benchmarking of quantum computers. Chapter 4 illustrates in detail the proposed algorithm for clustering aggregation. Chapter 5 contains a description of the experimental setup, of the chosen metrics, as well the experimental results and considerations on them. Chapter 6 contains a summary of the work, as well as possible improvements and future directions to follow.

Chapter 2

Background

2.1 Clustering

Clustering is an unsupervised data analysis technique that can be informally defined as the partitioning of a set of objects into groups called clusters; such partitioning must ensure that objects within a same cluster are more similar[1] to each other than to objects in other clusters [6].

Clustering is widely employed in numerous scenarios that require to group together similar data points, or to extract knowledge from a set of objects in the absence of any prior information. For instance, clustering is used in marketing and finance as a profiling tool [7]; in image processing and computer vision, as a segmentation technique [8], where it plays a pivotal role in various fields, such as remote image sensing [9] and digital forensics [10]; by energy distribution companies, to optimize the allocation of resources to end users [11].

Formally, a clustering of a set of objects (or *dataset*) $D = \{x_1, ..., x_n\}$ is defined as a collection of k elements $\mathcal{C} = \{c_1, ..., c_k\}$, with $\mathcal{C} \subseteq \mathcal{P}(D)$, such that the following properties hold:

$$D = \bigcup_{i=1}^{k} c_i, \tag{2.1}$$

and

$$c_i \cap c_j = \emptyset \quad \forall c_i, c_j \in \mathcal{C}. \tag{2.2}$$

Equation 2.1 translates to the fact that the all points of the dataset appear in at least one cluster, and therefore the union of all clusters yields back the original dataset; Equation 2.2 requires clusters to have an empty intersection, meaning that a point of the dataset cannot be put in two different clusters.

2.1.1 Clustering aggregation

Clustering aggregation, also known as clustering ensemble, is a technique that aims to improve the robustness and overall quality of clustering results, by combining multiple clustering solutions into a single one [12].

The motivation for utilizing clustering aggregation arises from the observation that no single clustering algorithm is universally optimal for all types of data or applications, as shown in Figure 2.1. Variability in initial conditions, distance metrics, and the inherent characteristics of the dataset can lead to different

[1] Provided that a binary ordering relation is defined on the set of objects.

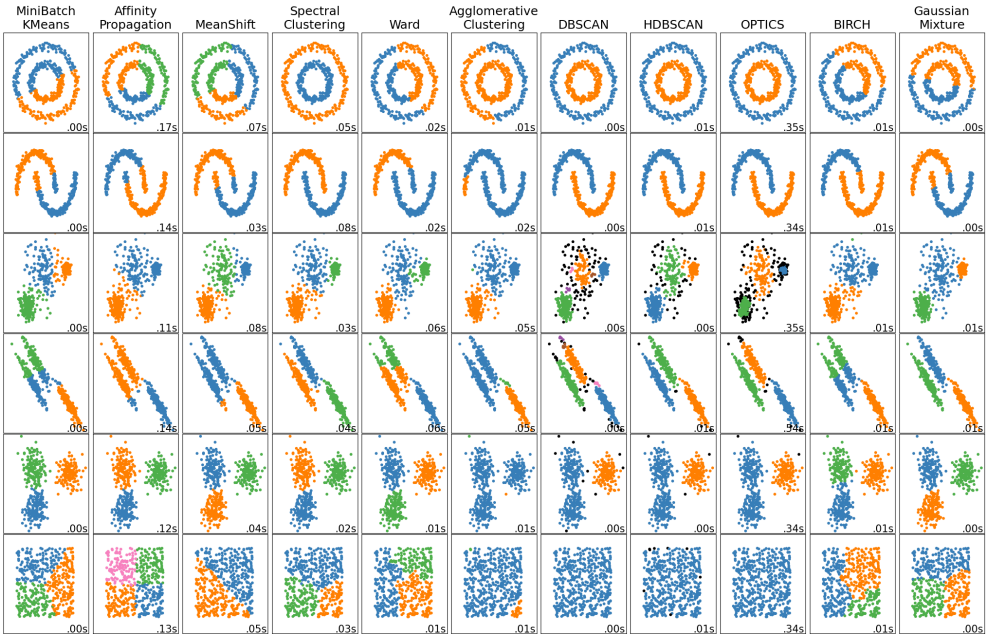


Figure 2.1: Comparison of clusterings produced by different algorithms on various data distribution. It is possible to notice how every algorithm tends to perform better on some distributions, and worse on others. Image taken from [14].

results, even when using the same algorithm. Aggregating these clustering solutions helps overcoming discrepancies and reduces the risk of selecting a sub optimal solution [13].

2.2 Optimization problems

An optimization problem consists in finding a value, or set of values, that maximizes or minimizes a given objective function, possibly under additional constraints. Optimization problems are extremely common in many fields of science, engineering and logistics, and there is an active research regarding methods to solve them, both analytically and numerically [15].

An important class of optimization problems is the class of Quadratic Unconstrained Binary Optimization, or QUBO for short; in a QUBO problem, the variables of the function to minimize, or maximize, are binary, meaning that they can assume value either 0 or 1. A function of a QUBO problem has the generic form

$$f(\mathbf{x}) = \sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j, \tag{2.3}$$

where x_i and x_j are binary variables, and a_i and b_{ij} are coefficients. Solving a QUBO problem means finding a binary vector $\bar{\mathbf{x}}$ that minimizes, or maximizes, the function defined in Equation 2.3; the binary vector can also be equivalently expressed as a string of zeros and ones, or bitstring. For example, the vector $\begin{bmatrix}1 & 0 & 1\end{bmatrix}$ has `101` as its corresponding bitstring.

Another way to express a QUBO problem is via an upper triangular matrix Q, where

- an element on the diagonal Q_{ii} corresponds to the linear coefficient a_i;
- an off-diagonal element Q_{ij} corresponds to the quadratic coefficient b_{ij}.

In matrix form, Equation 2.3 becomes then

$$f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x}. \tag{2.4}$$

The model of a QUBO problem, as the name suggests, does not allow to include additional constraints on the function. It is however possible to express in QUBO form many relevant optimization problems that require the use of constraints, by introducing quadratic penalties that enforce constraints indirectly [15]. This approach transforms constrained problems into unconstrained ones by incorporating penalties as additive terms in the objective function. Using a minimization problem as an example, a penalty term is zero when the constraints are satisfied, and grows quadratically as the violation increases, guiding the optimizer towards feasible solutions while minimizing the objective.

An equivalent form [15] to express a QUBO problem is in the so-called Ising model: introduced by Lenz and Ising around 1925 for the 1-dimensional case, it is a simplified description of the interaction among particle's spins arranged in a lattice. Equation 2.3 in Ising form becomes

$$H(\mathbf{s}) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j, \tag{2.5}$$

where s_i and s_j are spin variables and can assume values either -1 or 1, while the terms h_i are external field terms, and J_{ij} are coupling coefficients. The problem of finding the minimum of a function is then translated in finding the ground state for the system described by the Hamiltonian in Equation 2.5.

QUBO problems can be embedded into quantum machines by designing a system that represents the objective function; by letting the system evolve until it reaches its lowest energy state (ground state), it is possible to obtain a solution which minimizes the objective function.

2.3 Quantum computing

Quantum computing is a set of computational models and paradigms that combines concepts of computer science, physics and engineering. It offers an alternative perspective on computing to that of classical computing, both in terms of theoretical framework, as well as regarding the physical realization of machines.

The basic ideas of quantum computing were established by Richard Feynman in a 1982 paper, discussing the usage of computers to perform physics simulations [1]. Feynman observed that, due to their inherent complexity, quantum systems could not be feasibly simulated by classical computers, and introduced the idea of computing machines based on quantum principles [16]. A subsequent

work by Deutsch demonstrated, via the construction of a quantum Turing machine, that quantum computers are computationally equivalent to classical computers, and could therefore be useful for scopes beyond mere quantum simulation [17].

2.3.1 Qubits and quantum gates

A classical computer can be analyzed, from an abstract point of view, as a system of bits, which represent information, and logical gates, which manipulate information; analogously, a quantum computer is composed of quantum bits, or qubits for short, and quantum gates.

A quantum computer differs from a classical computer in the fact that its components and its behavior directly reflect the underlying quantum system it represents; as such, a quantum computer abides to the laws of quantum mechanics. Some of these laws, referred to as the postulates of quantum mechanics, give a synthetic description of qubits and quantum gates [18].

Postulate 1 (State space) *The state of a quantum system is described by a unit vector in a Hilbert space $\mathcal{H}$.*

Postulate 2 (Evolution) *The evolution of a system from one state to another is described by a unitary operator.*

Postulate 3 (Composition of systems) *When two systems, whose state spaces are $\mathcal{H}_1$ and $\mathcal{H}_2$, are treated as a unique, combined system, then the state space for the overall system is the tensor product of the single state spaces $\mathcal{H}_1 \otimes \mathcal{H}_2$.*

Postulate 4 (Measurement) *Given a system with state*

$$|\psi\rangle = \sum_i \alpha_i |\varphi_i\rangle , \tag{2.6}$$

performing a measurement on the system will yield label i as result with probability $|\alpha_i|^2$ and leave the system in the state $|\varphi_i\rangle$.

From postulate 1 follows one of the principal characteristics of qubits, which goes under the name of *superposition*. While the state of a classical bit can only have one of its possible values (i.e. either `0` or `1`), the state of a qubit is a linear combination, or superposition, of base vectors. Supposing that a qubit can only have two values $|0\rangle$ and $|1\rangle$, it assumes the general form

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle ,$$

where (α, β) is a unit vector of a Hilbert space $\mathcal{H}$, or alternatively $|\alpha|^2 + |\beta|^2 = 1$.

Postulate 2 gives a mathematical description of quantum gates, and from it it is possible to derive the property that all quantum gates are reversible.

Postulate 3 states that multiple qubits can be analyzed as a single system and implicitly introduces the notion of *entanglement*; a quantum state is said to be entangled if it cannot be separated into pure states of the subsystem 1 or 2.

Postulate 4 describes the operation of measurement; while the measurement of a classical bit does not, in principle, affect the bit itself, measurement in quantum mechanics is a destructive operation that breaks superposition and causes the qubit to collapse in one of the base states. Such operation reveals also the probabilistic nature of quantum mechanics, because qubits collapse in the state $|\phi_i\rangle$ with probability α_i, as per Equation 2.6.

Quantum computers are usually divided in two classes: gate-based and analog-based. In the former, qubit manipulation is obtained through quantum logic gates, represented formally by unitary operators [19]. The latter is the so called analog-based quantum computing [20], an approach that leverages the continuous evolution of a quantum system to solve specific problems. While gate-based quantum computers are, in principle, capable to perform any kind of algorithm [17], they currently face several practical challenges, including high error rates , limited qubit numbers, and difficulties in scaling. Analog-based quantum computers, on the other hand, are designed to solve specific classes of problems, typically optimization problems or those that can be framed in terms of minimizing an energy landscape. At the same time they have shown promise in delivering better results [21].

In this work, the proposed algorithm will be tested on quantum machines that work according to the analog paradigm.

2.3.2 Advantages of quantum computing

The peculiar characteristics of quantum computers, some of which mentioned in Section 2.3.1, make them better suited than classical computer at solving various tasks. A cornerstone in quantum computing is represented by the development of Shor's algorithm [2]. In this seminal paper it is presented a quantum algorithm for finding the prime factors of an integer. The problem is well-known and when addressed by a classical computer takes a non-polynomial time for its execution. Shor's algorithm resolve the factorization problem in polynomial time, taking quantum gates of order $O\big((\log N)^2(\log\log N)(\log\log\log N)\big)$ to factorize and integer N. This is significantly faster than the most efficient known classical factoring algorithm, which run in $O\big(\exp(c(\log N)^{1/3}(\log\log N)^{2/3})\big)$ time [22]. This is not only a groundbreaking result in computer science and mathematics, but has also important implications for cryptography and security . Other examples of remarkable results obtained by quantum computers are in quantum chemistry simulation [23] and in solving optimization problems for logistics and finance [24].

Current state of technology poses significant challenges in building reliable quantum machines, due to problems such as noise and decoherence [25]. Despite this, in recent years, an increasing number of quantum machines demonstrated results in real world application; currently, quantum computing is not considered anymore a purely theoretical field, but offers interesting perspectives, both in terms of academic research and in industry. For example, in the current post-Moore's law era, with classical computing facing the limitations of transistor miniaturization, quantum technologies may play an important role as accelerators for high performance computing [26].

2.4 Quantum annealing

Quantum annealing is a meta-heuristic optimization technique that exploits principles of quantum mechanics, like quantum tunneling and the adiabatic theorem, to find the global minimum of a given objective function [27]. It is mainly used as a solver for combinatorial optimization problems, in lieu of analogous classical algorithms, such as simulated annealing.

The key idea in quantum annealing is encoding the problem function into the energy states of a quantum system, so that the optimal solution corresponds to the lowest energy state of the system, or ground state. A quantum system can be mathematically represented by a Hamiltonian, an operator that describes the system's total energy and whose eigenvalues correspond to the possible energy level the system can assume.

The process starts by initializing the system in a simple, known ground state; then, the system is evolved gradually by slowly changing its Hamiltonian from the initial one to the final one, which represents the problem to be solved. Quantum tunneling allows this method to leap from one local minimum to another, while the quantum adiabatic theorem ensures that, if the perturbation acting on the system is slow enough, then it will remain in a ground state [28].

2.5 Neutral-atoms technology

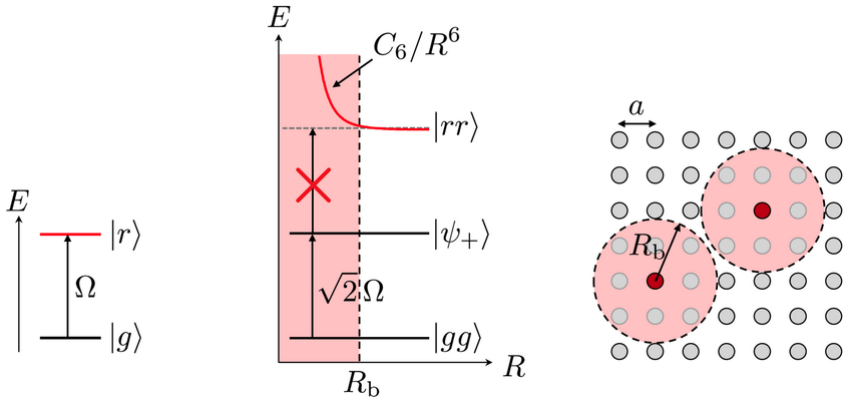


Figure 2.2: Exemplification of the Rydberg blockade mechanism. Left: an isolated atom has energy level splitting between ground state $|g\rangle$ and excited state $|r\rangle$ given by Ω. Center: when two atoms come closer, dipole-dipole interaction shifts the energy level to a state where both are excited, $|rr\rangle$. Right: if atoms are arranged in a lattice and one is in the excited state $|r\rangle$, atoms within the radius R_b are blocked from being excited in the same state. Figure is taken from [29].

Neutral-atoms quantum computers are quantum machines that operate by rearranging Rydberg atoms, trapped in optical tweezers, and inducing dipole-dipole interaction via optical or microwave pulses. This dipole-dipole interaction is better exploited using alkali atoms, such as Rubidium atoms.

In general, a Rydberg atom is an excited atom with one (or more) electrons with very high principal quantum number n. The higher the quantum number

n, the stronger the dipole generated by the atom. This is because, on average, the higher the value of n, the farther an electron is from the nucleus. Alkali atoms have only a single electron in the outermost level, so exciting this electron will create a large electron dipole moment. In the context of neutral-atoms quantum computers, a single Rydberg atom is not enough to create a qubit. In fact, if we place two excited Rydberg atom close together (closer than a distance known as the Rydberg radius R_b, which is $\mathcal{O}(\mu\text{m})$ for Rubidium), their electric dipole momenta start to interact with each other. This interaction will cause a splitting of the atom's energy levels in such a way that it prevents the two atoms to be in the same excited state [30, 31].

This is the fundamental mechanism exploited in neutral-atoms machines and is known as the Rydberg blockade. Rydberg blockade enables entanglement between qubits and, subsequently, the creation of multi-qubit gates, in which one atom can control the states of other atoms. A schematic example of the Rydberg blockade is shown in Figure 2.2. In an isolated Rydberg atom the energies of the ground state $|g\rangle$ and the excited state $|r\rangle$ differ by a value Ω. When two atoms are put close enough the doubly excited state $|rr\rangle$ is not accessible anymore due to the dipole-dipole interaction: only the state $|\psi_+\rangle = 1/\sqrt{2}(|gr\rangle + |rg\rangle)$ will be populated. If we consider the situation where many atoms are arranged in a lattice, an atom excited to the Rydberg level (red dot) prohibits the excitation of any atom within a sphere of radius R_b.

Chapter 3

Motivation and Context

The main goals of this work are designing a quantum algorithm to perform clustering aggregation, testing it on real hardware and comparing results across the different technologies. As a multidisciplinary research in the fields of data analysis algorithms and quantum computing, there are multiple reasons behind the choice of combining these different areas of interest, which are hereafter explained more in detail.

3.1 Clustering aggregation

Clustering is a widely employed machine learning method, with use cases ranging from image processing to finance. However, different clustering algorithms often yield divergent results due to differences in their underlying assumptions, biases or sensitivities to the input data. This variation presents a challenge when trying to derive a reliable clustering result using a single algorithm.

Clustering aggregation, or clustering ensemble, is a framework developed specifically to address this issue by combining multiple clustering solution into a single one, thereby improving the overall quality of the solution [13].

Since clustering aggregation is an NP-hard problem (see Section 4.1 for the proof), classical algorithms for this task often present scalability issues as the problem size increases, requiring algorithms with many trade-offs in terms of quality or of execution times [32].

For this reason, the exploration of alternative technologies such as quantum computers to address this task is a direction worth following, as it may lead to algorithms with improved performances, that produce solutions of higher quality.

3.2 Quantum processors usage in industry

There is a growing interest in the potential of quantum computing within industry, particularly for applications in optimization, machine learning [3], and materials science [4]. Rather than replacements of classical computers, quantum technologies are now viewed from the perspective of components that can complement classical systems, by solving specific types of problems more efficiently [26]. For example, the European Union, through the European High Performance Computing Joint Undertaking (EuropHPC JU), a major initiative of the European Union, is actively working on the integration of quantum computers to serve as accelerators in six of its largest supercomputers [5].

Thanks to the agreements between CINECA, the Italian University Consortium for Computing, and the companies D-Wave and Pasqal, it was possible to perform experiments on two different technologies: quantum annealers and neutral-atoms based processors.

Quantum annealers, such as systems produced by D-Wave, are particularly suited for combinatorial optimization tasks, because they naturally solve problems by finding global minima in an energy landscape, and thanks to quantum tunneling they can lead to a faster convergence compared to classical methods [33].

Neutral-atoms quantum processors are a relatively new technology, compared to the more consolidated quantum annealers. They present unique characteristics, such as allowing high connectivity between qubits and a flexible, programmable architecture, which can be reasonably beneficial for practical implementations of complex optimization problems [31].

This work presents an algorithm to perform clustering aggregation, a practical problem of great interest in industry, as a good candidate for investigating the integration between classical and quantum computers for industrial purposes, with the advantage of performing tests on different technologies.

3.3 Benchmarking of quantum computers

Measuring the performances of quantum computers is not only an essential step to assess how a single quantum computer compares to other quantum computers, but is also an important indicator of the extent to which these machines are capable of tackling practical problems, also referred to as *quantum utility* [34].

Benchmarking of quantum systems, a term used to indicate the systematic process of evaluation and measurement of performances, presents numerous challenges, particularly when comparing different quantum technologies such as gate-based, annealing and neutral-atoms processors. These systems operate on different physical principles, making it difficult to establish a universal metric for comparison. Another factor that affects the complexity of performing a benchmark of quantum processors is that certain technologies are suited to work only with a restricted set of tasks; designing a single algorithm to serve as a benchmark of a generic quantum computer would not take in account the differences in the various approaches. For example, quantum annealers, which rely on adiabatic evolution to solve optimization problems, excel at specific tasks but are hindered by issues like noise, qubit connectivity and problem embedding, which limits their scalability. Similarly, neutral-atoms quantum processors, which can work both in digital (gate-based) and analog mode [35], offer highly scalable architectures by manipulating individual atoms through optical tweezers, which mitigates the issue of problem embedding, but have still to face noise, decoherence and fidelity during quantum operations [31]. Gate-based systems are, on the other hand, more versatile and capable of working with a broader set of algorithms, but suffer from qubit decoherence and operational errors, which reduce gate fidelity in multi-qubit operations [36].

Different benchmark suites for quantum computers have been proposed (for example a quantum version of LINPACK, a benchmark algorithm used to rank

supercomputers in the TOP500 list [37, 38]), but because of the heterogeneity of quantum machines the metrics they offer do not reflect clearly how the processors performs on various classes of algorithms, and are strictly related to the underlying technology. For such reasons, better alternatives for assessing performances of quantum computers are application-oriented benchmarks, in which the quality judgment of the machine is given based on its practical capabilities at reaching a quantum advantage for specific tasks, mainly NP-hard problems like Maximum Independent Set (MIS) and Maximum-Weight Independent Set (MWIS). These two problems in particular will be discussed more in detail in Chapter 4, as solving them will be an important step of a hybrid quantum-classical algorithm.

Solving clustering aggregation may be a valid approach to the benchmarking of neutral-atoms and annealing quantum processors, two technologies for which benchmarking research is lacking, given that the majority of research in the field is focused on gate-based quantum computers [39].

Chapter 4

Proposed method

The proposed hybrid algorithm for clustering aggregation is described in this chapter. The key idea, introduced in [40], is reducing the clustering aggregation problem to an equivalent graph partitioning problem, in particular the Maximum-Weight Independent Set (MWIS). The possible use case for a quantum machine relies in the fact that such a problem is NP-hard, and with classical computers it can only be solved, in a reasonable amount of time, with approximate methods, such as simulated annealing; a quantum approach could potentially lead to quantum advantage.

Additionally, it is possible to demonstrate that the MWIS is equivalent to a QUBO problem with constraints, a class of problems that can be tackled by D-Wave's quantum annealer Advantage QPU. While this problem cannot be currently solved by Pasqal's Fresnel, this neutral-atoms based quantum processor is able to solve Maximum Independent Sets (MIS) problems; given that a MWIS is a MIS with additional constraints, Fresnel can be used to reduce the search space for finding an optimal clustering.

4.1 Clustering aggregation as a graph problem

Given a dataset $D = \{d_1, ..., d_n\}$, we consider m clusterings of D; each clustering $\mathcal{C}_i$ contains k_i clusters, which are subsets of D.

It is possible to build an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

$$\mathcal{V} = \bigcup_{i=1}^{m} \mathcal{C}_i = \{c_{11}, ..., c_{1k_1}, ..., c_{m1}, ..., c_{mk_m}\} \tag{4.1}$$

$$\mathcal{E} = \{(c_i, c_j) \in \mathcal{V}^2 | c_i \cap c_j \neq \emptyset\}; \tag{4.2}$$

in other words, the vertices of the graph are the clusters from all the clusterings, and an edge is between two of them if and only if they overlap. According to Equation 2.2, a necessary condition for a clustering to be valid is that its clusters do not overlap; in the graph representation, this translates to the fact that vertices representing clusters of a valid clustering must form an independent set.

In this way, the search space for the optimal aggregated clustering is reduced to the clusterings that form independent sets. In order to find the optimal one, it is necessary to choose a metric to maximize; for this algorithm, as proposed in [40], the metric chosen is the sum of the total silhouette score of the clusters composing

the clustering. The silhouette score will be more thoroughly illustrated in Section 5.3.1.

In terms of graph representation, this is equal to associate a weight (the silhouette score) to each vertex, and the problem of finding the optimal clustering becomes a search for a Maximum-Weight Independent Set, an optimization problem on graphs which will be formally defined in Section 4.2.

4.2 QUBO formulation of the MWIS

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be represented in matrix form via its associated adjacency matrix, which has the form $A = (a_{ij})_{m \times m}$, where

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Furthermore, a subset of the vertices $\mathcal{V}' \subseteq \mathcal{V}$ can be represented as a vector $\mathbf{x} \in \{0,1\}^m$, where

$$x_i = \begin{cases} 1 & \text{if } i \in V' \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Calling $\mathbf{w} \in \mathbb{R}^m$ the vector of weights associated to each vertex, finding a MWIS in $\mathcal{G}$ is equivalent to finding a vector $\overline{\mathbf{x}} \in \{0,1\}^m$ such that

$$\overline{\mathbf{x}}^T A \overline{\mathbf{x}} = 0, \tag{4.5}$$

which means that the vertices must be solution of another optimization problem on $\mathcal{V}$, the Maximal Independent Set problem (MIS), and also such that

$$\overline{\mathbf{x}} = \arg\max_{\mathbf{x}} \mathbf{w}^T \mathbf{x}, \tag{4.6}$$

which accounts for the maximization of the weights of the clusters.

Since the graph is non oriented, matrix A can be considered an upper triangular matrix, and Equation 4.5 has the same form of Equation 2.4, expressing therefore a QUBO (Quadratic Unconstrained Binary Optimization) problem. Equation 4.6 is an additional constraint on the QUBO.

Being able to reduce the clustering aggregation problem to a MWIS, which in turn can be reduced to a QUBO with constraints, is essential to submit the problem to the quantum platforms we are considering. While D-Wave's Advantage QPU can solve QUBO problems with constraints, Pasqal's Fresnel can, at the moment, only work in analog mode, and is only able to address MIS problems. Therefore, when testing the algorithm, Fresnel will only be able to find solutions that satisfy Equation 4.5.

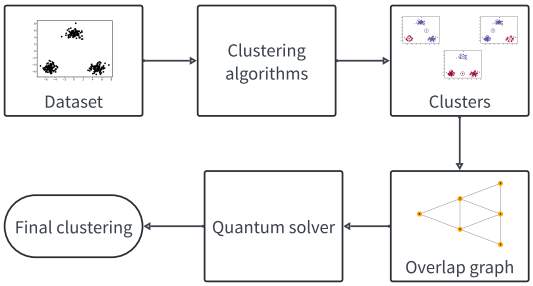


Figure 4.1: Steps of the proposed algorithm.

4.3 Description of the algorithm

The steps of the proposed method are listed below, and can also be visualized from Figure 4.1:

Step 1. Multiple clustering algorithms, either distinct algorithms such as DBSCAN and Spectral Clustering, or the same algorithm with different values of parameters (an example of algorithms with some of their parameters, from the `sklearn.cluster` implementation [41], are presented in Table 4.1), are run on the dataset; clusters produced by the various algorithms are then given unique labels and silhouette score is computed for each of them.

Step 2. Overlap graph is built according to Section 4.1, with vertices of the graph weighted according to the silhouette score of the cluster they represent, and edges having associated weight equal to 1.

Step 3. Associated adjacency matrix is built from the overlap graph, in such a way it is triangular and all elements on the diagonal having the same value (possible because the graph is undirected).

Step 4. The QUBO matrix, built in the previous point from the overlap graph and representing the MWIS problem, is solved using a quantum machine.

Algorithm	**Parameters**
K-means	**number of clusters**, number of iterations, tolerance, ...
DBSCAN	**epsilon**, minimum number of samples, leaf size, ...
Spectral Clustering	**number of clusters**, gamma, affinity, ...

Table 4.1: Algorithms used in this work with some of their corresponding parameters, in the `sklearn.cluster` implementation [41]. In bold the parameter that was tuned in the practical implementation. K-means and Spectral Clustering require to specify how many clusters the dataset should be partitioned into, whereas DBSCAN's parameter, called epsilon, is a distance value used to select points to add to the clusters.

The last step is fundamentally dependent on the technology and the actual quantum machine chose. For this work, two different approaches were chosen:

a neutral-atoms quantum computer in analog mode, developed by Pasqal, and a D-Wave quantum annealer.

As also said before, at the moment, Pasqal's neutral-atoms based Fresnel is only able to work in analog mode. It is only possible to control all qubits at the same time, but not individually; the only class of optimization problems addressable right now by this machine are those that can be reduced to MIS, while MWIS are not easily solvable. For this reason, the information of the silhouette cannot be included in problem embedding.

D-Wave's quantum annealer Advantage QPU, on the other hand, offers a framework to add constraints to the QUBO sent in input to the solver, making it possible to fully perform the intended algorithm and solving the MWIS problem.

4.4 Example

We present here a simple example of the algorithm, tested on a small dataset, shown in Figure 4.2. The dataset was generated using `make_blobs` function from `sklearn.dataset` module [42], and contains 300 points divided in three blobs, positioned at an equal distance from each other.

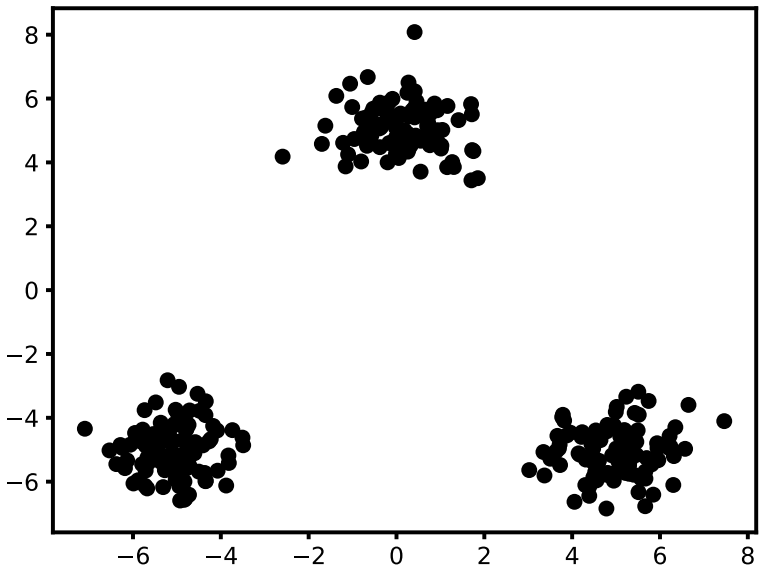
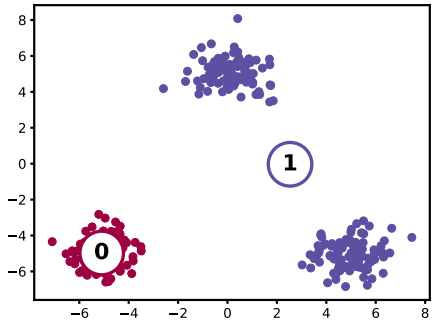


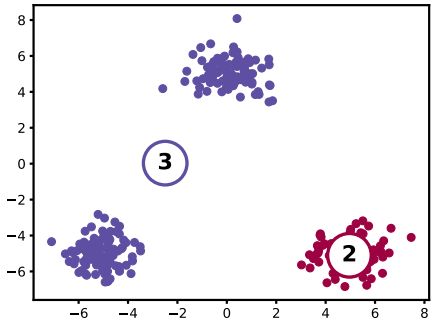
Figure 4.2: Example dataset, generated with function `make_blobs` from module `sklearn.dataset`, containing 300 points divided in three blobs.

The first step is running on the dataset multiple clustering algorithms, which can be different algorithms (such as DBSCAN or Spectral Clustering), or the same algorithm with different hyperparameters. In this case, three instances of K-Means were run on the dataset (Figure 4.3), all of them with number of clusters set to 2 and having different seed values, chosen in order to obtain three distinct

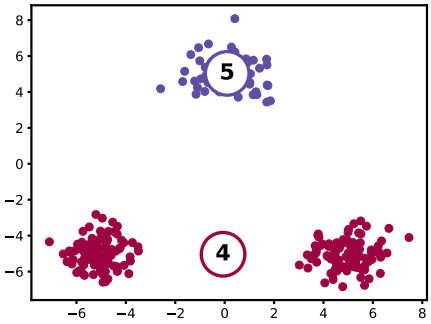
results. In this toy example, the seed is playing the role of a generic hyperparameter. Each clustering produces two clusters, which are uniquely labeled across the various clusterings; silhouette scores are then computed for each point, and to each cluster is assigned the sum of the silhouettes of the point it contains (Table 4.2).



(a) Clustering #1



(b) Clustering #2



(c) Clustering #3

Figure 4.3: Three clusterings of the original dataset, all of which obtained via K-Means with number of clusters equal to 2 and seed chosen in order to obtain different results. (a) contains clusters with labels 0 and 1; (b) contains clusters with labels 2 and 3; (c) contains clusters with labels 4 and 5.

Cluster	Silhouette (avg)	Silhouette (sum)
0	0.875	87.529
1	0.402	80.328
2	0.867	86.724
3	0.402	80.381
4	0.489	97.723
5	0.878	87.761

Table 4.2: Silhouette scores for the example clusterings.

Overlap graph is then built according to the points shared between clusters, as shown in Figure 4.4. Its associated adjacency matrix representation has the

form

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.7}$$

The only independent set that respects both Equation 2.1 and Equation 2.2 is vector $\mathbf{x} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$, representing the clustering containing clusters with labels 0, 2 and 5. In this particular case, being this the only solution, it is not necessary to check which solution maximizes the silhouette sum.

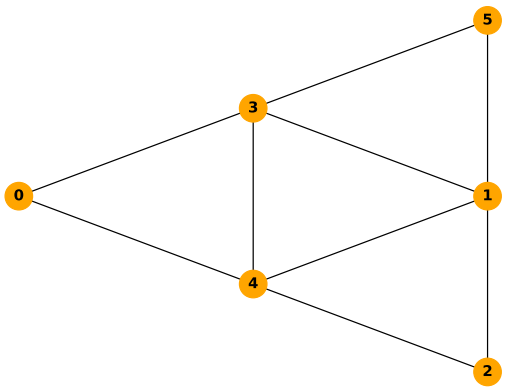


Figure 4.4: Adjacency graph for the example dataset.

Chapter 5

Experiments and Results

The algorithm introduced in Chapter 4 was tested on different quantum platforms, in order to show the feasibility of executing a hybrid quantum-classical algorithm for clustering aggregation, and also to measure the performances of said platforms. This chapter begins with a description of the quantum environments used to run the algorithm, of the dataset and the clustering algorithms chosen to generate the input clusters for the clustering aggregation, and of the metrics used to evaluate results; then, experimental results are presented and discussed.

5.1 Quantum platforms

The platform used to test the clustering aggregation algorithm are two quantum machines with different technologies: Pasqal's Fresnel, a neutral-atoms based quantum computer, and D-Wave's Advantage QPU, a quantum annealer. The two machines are compared in Table 5.1. Additionally, the algorithm was also run on a classical emulator of Fresnel based on the QuTiP software framework; testing on the emulator was useful to have a better understanding of the feasibility of the algorithm, due to the novelty of Pasqal's machine.

5.1.1 Pasqal's Fresnel

Fresnel is a neutral-atoms quantum computer, developed by the French company Pasqal.

The main component of Fresnel is the register, which contains Rubidium atoms [43] (in this context also called qubits) trapped into optical tweezers; while, in principle, it is possible to rearrange atoms freely in the register, currently they can only be placed in fixed, predetermined layouts [44]. At the moment, Fresnel can work with at most 100 qubits.

Another important set of components of Fresnel are the channels, which are able to manipulate the atoms in the register by inducing transitions between their electronic levels, via sequences of laser pulses. Channels are organized, according to their addressing, in local and global, depending on whether they target individual atoms or the entire system, respectively [45].

Fresnel can, in principle, work both in digital (gate-based) mode and in analog mode; at the moment, however, only the latter is possible [44]. This means that only global channels can be used, and the entire system is described by the Hamiltonian

$$H = \sum_i \left(\frac{\hbar\Omega(t)}{2}\sigma_i^x - \frac{\hbar\delta(t)}{2}\sigma_i^z + \sum_{j<i} U_{ij} n_i n_j \right), \tag{5.1}$$

where $\Omega(t)$ is called Rabi frequency, and $\delta(t)$ detuning. These two parameters can be continuously manipulated via sequences of pulses, thereby controlling the dynamics of the system. In analog mode, the only class of problem that can be addressed are optimization problems reducible to the Maximal Independent Set (MIS).

Through Pulser, a Python framework developed by Pasqal, it is possible to define the configuration of qubits in the register and design sequences to manipulate them [45].

5.1.2 D-Wave's Advantage QPU

The Advantage QPU is a quantum annealer produced by the Canadian company D-Wave, representing the first commercially available quantum annealer. It was specifically designed as a solver for combinatorial optimization problems, with a particular focus on usage in industry as a tool to work with business problems [46], being able to solve QUBO problems with constraints.

The Advantage QPU offers over 5000 qubits and employs the Pegasus topology, which has a 15-way qubit connectivity, meaning that each qubit is connected to 15 other qubits, a major improvement from the previous machine manufactured by D-Wave, the D-Wave 2000Q QPU (now discontinued), which had over 2000 qubits in the Chimera topology, with a 6-way connectivity [46]. The increased number of qubits, paired with the higher degree of connectivity, allows for a more efficient embedding of optimization problems on the quantum processors. Despite the improvements, however, the QPU has not been able to show a significant advantage over classical methods for practical applications [47].

	Fresnel	**Advantage QPU**
Technology	Neutral atoms	Quantum annealer
Number of qubits	100	over 5000
Connectivity	Dependent on the layout	15-way
Solvable problem type	MIS	QUBO with constraints

Table 5.1: Comparison of the main characteristics of Pasqal's Fresnel and D-Wave's Advantage QPU.

5.1.3 QuTiP

The Quantum Toolbox in Python, also known as QuTiP, is an object oriented Python framework designed to represent and simulate the dynamics of open and closed quantum systems [48].

The core of information representation in QuTiP is the `Qobj` class, which is used to represent both quantum states (such as pure states or density matrices) and quantum operators (such as Hamiltonians or measurement operators). The

`Qobj` class offers methods to perform matrix operations like addition, multiplication and tensor product, which are essential to manipulate quantum systems; sparse matrix encoding of objects allows these operations to be executed relatively efficiently.

To perform simulations using QuTiP, it is necessary to specify the initial conditions of the system, for example ket vectors or density matrices, and a Hamiltonian or other operators, which describe the time evolution and possible dissipations. It is then possible to select different kinds of solvers, such as the Monte Carlo wave function method for a Schrödinger equation describing evolution of pure states, or a numerical approach for a Lindblad master equation in the case of open systems.

Results are presented in the form of evolved states at fixed intervals throughout the simulation time windows. Additionally, QuTiP allows to perform the calculations of expectation values for observables (such as spin, position, or energy) over time.

Since QuTiP does not directly support neutral-atoms systems, Pulser contains a library that acts as an interface to QuTiP, in order to simplify the definition of neutral-atom arrays and the simulation of experiments with laser pulses [45].

5.2 Dataset

The dataset used to test the hybrid clustering aggregation is shown in Figure 5.1. It is specifically designed to contain different shapes and configurations of points, in such a way that most clustering algorithms fail to produce a correct clustering. Its 788 points are labeled from 1 to 7, reflecting the clustering considered optimal, which will be used as a reference to measure the quality of the results of the hybrid clustering aggregation algorithm.

The dataset was first introduced by Gionis, Mannila and Tsaparas to test classical clustering aggregation algorithms [32]; it was then used by Li and Latecki to test another classical clustering aggregation algorithm that uses simulated annealing [40]. In [32] various classical cluster algorithms were tested over the dataset: Single Linkage, Complete Linkage, Average Linkage, Ward's Clustering and K-means, all with parameter number of clusters set to 7.

5.3 Evaluation metrics

This work employs different metrics to evaluate the quality of single clusters and of clusterings as a whole. Silhouette score is used as a factor to weigh clusters in the clustering aggregation algorithm; adjusted Rand index will be used to evaluate the overall quality of solutions (i.e. clusterings returned by the clustering aggregation algorithm).

The adjusted Rand index, however, cannot be used if the clustering aggregation algorithm returns a set of clusters that either overlap, or do not contain all the points of the entire dataset, because the index can be computed only between data collections that share the exact same points. For this reason, the number of returned clusters will be used as an additional metric, considering of higher

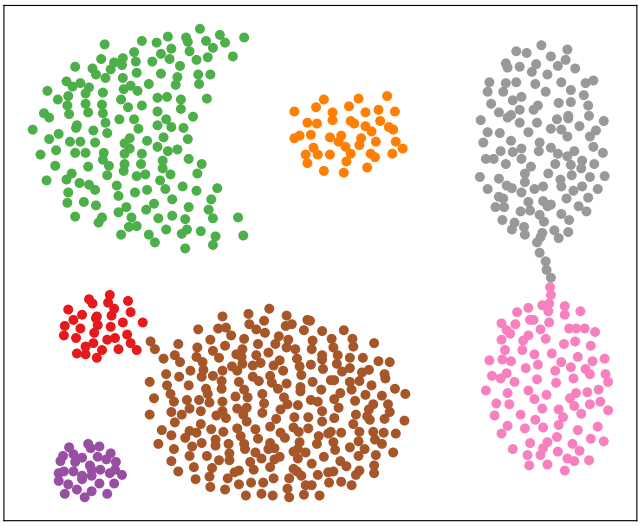


Figure 5.1: Plot of the dataset used to test the hybrid clustering aggregation algorithm, made up of 788 points. The different colors indicate the 7 clusters of the clustering considered optimal.

quality solutions that return the exact number of intended clusters (in this case, 7).

The silhouette score and the adjusted Rand Index will be illustrated more in detail in the following sections.

5.3.1 Silhouette score

The silhouette score is a metric used to evaluate clustering quality, introduced by Rousseeuw in 1987 [49]. The score is computed for each point in the clustering and reflects both its cohesion with points in the same cluster, as well as the separation with respect to points in different clusters. Specifically, the silhouette score of a point p is defined as

$$S_p = \frac{b_p - a_p}{\max(a_p, b_p)}, \tag{5.2}$$

where a_p is the average distance from the point to other points in the same cluster (intra-cluster distance), and b_p is the average distance to points in the nearest different cluster (inter-cluster distance). The score ranges between -1 and 1, where values close to 1 indicate that points are well-clustered, with high cohesion and good separation, while negative values suggest points may have been assigned to the wrong cluster.

A quantification of the quality for a single cluster can be obtained by computing the average of the silhouette score of the point it contains; more formally, for

a cluster c_i, the formula for its average silhouette (AS_i) is

$$AS_i = \frac{\sum_{p \in c_i} S_p}{|c_i|}. \tag{5.3}$$

5.3.2 Adjusted Rand index

The Rand index (RI) is a metric that quantifies the similarity between two data clusterings, first proposed by Rand in 1971 [50]. It is computed by considering all possible pairs of points in the dataset and assessing whether they are assigned to the same or to different clusters in the two clusterings being compared. The index assumes values between 0 and 1, with 1 indicating perfect agreement between the two clusterings.

Given two clusterings $\mathcal{C}_1$ and $\mathcal{C}_2$, their Rand index is computed as

$$RI = \frac{a+b}{a+b+c+d}, \tag{5.4}$$

where:

- a is the number of point pairs that are put in the same cluster in both clusterings;
- b is the number of point pairs that are put in different clusters in both clusterings;
- c is the number of point pairs that are put in the same cluster in $\mathcal{C}_1$, but in different clusters in $\mathcal{C}_2$;
- d is the number of point pairs that are put in different clusters in $\mathcal{C}_1$, but in the same cluster in $\mathcal{C}_2$.

The adjusted Rand index (ARI) is a more robust refinement of the Rand index, introduced by Hubert and Arabie in 1985 [51]. An issue of Rand index is that it does not take in account chance: even if two clusterings are randomly generated, the Rand index could assume positive values, potentially leading to misleading conclusions about the similarity of clusterings. The adjusted Rand index, on the other hand, adjusts for this by subtracting the expected similarity of two random clusterings and then normalizing the result. As a consequence, the ARI has a value of 0 for random clusterings and a maximum value of 1 for perfectly identical clusterings.

Given two clusterings $\mathcal{C}_1$ and $\mathcal{C}_2$, their adjusted Rand index is computed as

$$ARI = \frac{RI - E(RI)}{\max(RI) - E(RI)}, \tag{5.5}$$

where RI is the Rand index of $\mathcal{C}_1$ and $\mathcal{C}_2$, and $E(RI)$ is the expected value of RI.

5.4 Description of experiments

A total of four experiments were performed to test the hybrid clustering aggregation algorithm: one on real hardware Fresnel, one on QuTiP emulated Fresnel, and two on Advantage QPU. The classical part of the algorithm, namely running the classical clustering algorithms on the input dataset introduced in Section 5.2, evaluating silhouettes, and building the overlap graph and associated adjacency matrix, was executed only once, and the results then sent to the different quantum platforms. The experiment was repeated 1000 times on each of the four quantum platforms.

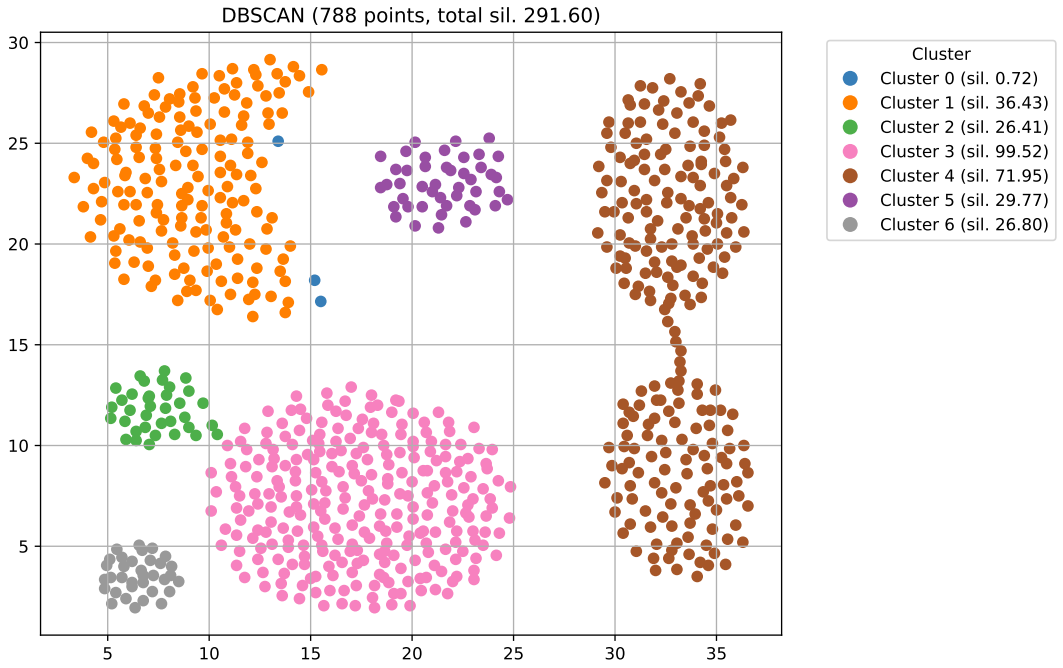
The classical algorithms chosen for the first step of the proposed method are DBSCAN, with parameter epsilon equal to 1.2, and Spectral Clustering, with parameter number of clusters equal to 5, which produced clusterings with 7 and 5 clusters, respectively, for a total of 12 clusters (we label these clusters with an increasing number, from 0 to 11, as shown in Figure 5.2). Since the total number of clusters corresponds to the size of the overlap graph and, subsequently, the number of required qubits, the amount of clustering algorithms and their relative parameters were chosen with the aim of preventing this number to grow too large, because of the number of qubits available on Fresnel. While Fresnel can work with up to 100 qubits [44], at the moment they can only be placed on a fixed layout; due to intrinsic limits of Rydberg atoms (namely, that they have to be at a distance larger than the Rydberg radius to prevent unnecessary entanglements, but cannot be closer than $5\mu m$), it was simpler to consider a smaller, yet significant, instance of the problem, which required 12 total qubits.

The clusterings returned by the quantum platforms are in the form of 12-digit bitstrings: if, for example, the found solution is completely aligned with the one found by DBSCAN, the bitstring will be 111111100000. On the contrary, if the found solution contains the same clusters obtained by running Spectral Clustering, the bitstring will be 000000011111. A generic solution, e.g. with bitstring 001010011001, corresponds to a situation where clusters labeled 2 and 4 are taken from DBSCAN, whereas clusters labeled 7, 8 and 11 come from Spectral Clustering.

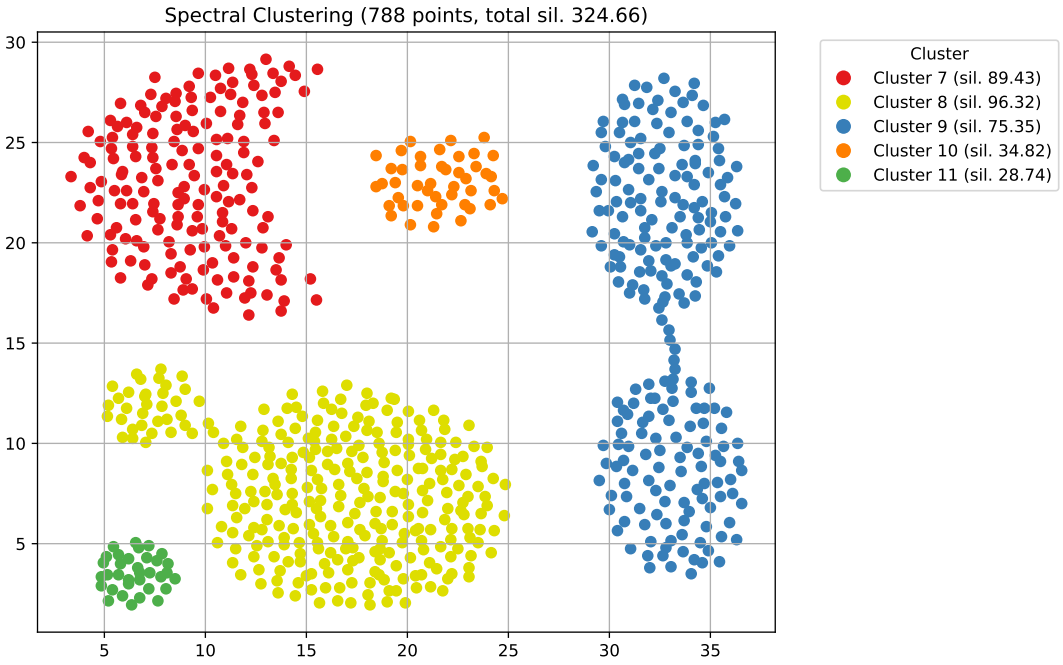
5.4.1 Fresnel

Running the experiments on Fresnel requires to define a qubit configuration that encodes the problem in the register, and specify a sequence of pulses to act on the system of qubits in order to make it reach its ground state. Both the register configuration and the sequence are the same for the experiment on QuTiP emulator and the one on real hardware.

The register configuration for Fresnel, shown in Figure 5.3, was built in such a way that qubits corresponding to overlapping clusters are at a Rydberg radius distance from one another, therefore entangled, recreating in the overlap graph obtained from the previous step of the algorithm.



(a)



(b)

Figure 5.2: Results of the clustering algorithms run on the input dataset, whose clusters will be selected by the clustering aggregation algorithm. Clustering in (a) was obtained through DBSCAN, with epsilon equal to 1.2, and contains a total of 7 clusters labeled from 0 to 6; clustering in (b) was obtained through Spectral Clustering, with number of clusters equal to 5, and contains 5 clusters labeled from 7 to 11.

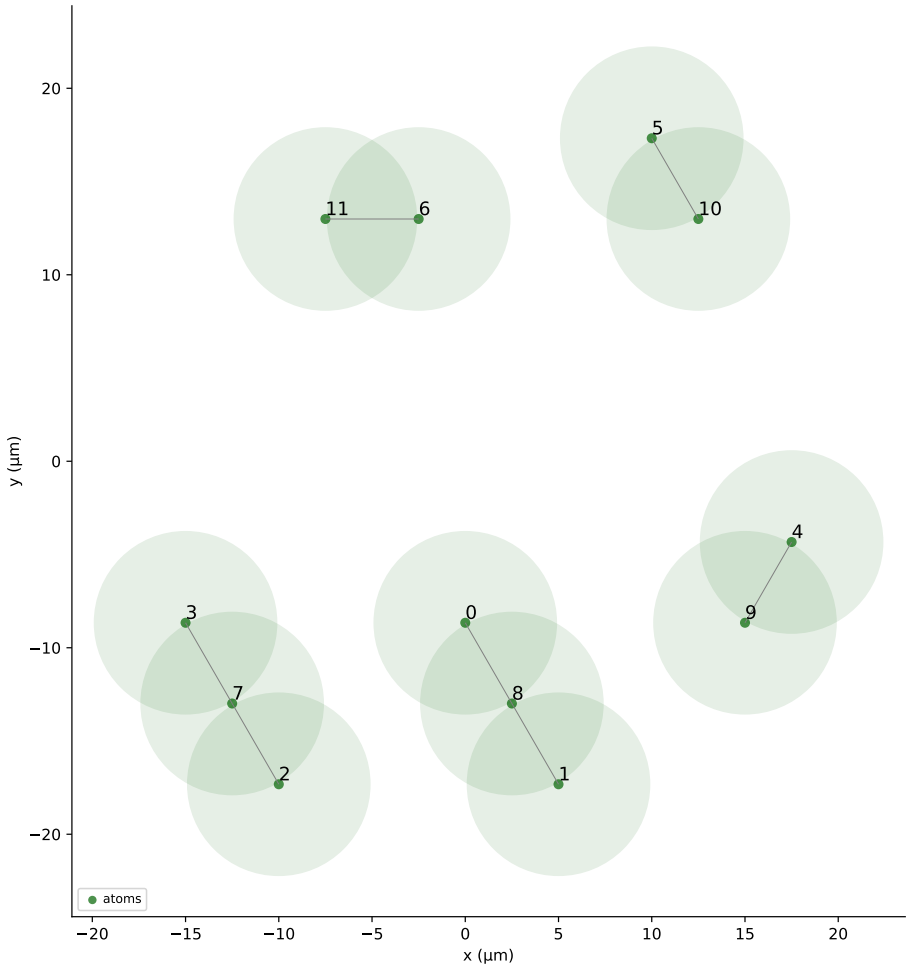


Figure 5.3: Plot of the qubits configuration in the register of Fresnel, with labels corresponding to clusters with the same label. This configuration was used both in the QuTiP emulated Fresnel experiment, as well as the real hardware Fresnel experiment, and reflects the exact position of atoms. The green circles around qubits represent the Rydberg radius, while the arcs between qubits indicate that the qubits are entangled.

Due to the impossibility, at the moment, of controlling each qubit independently, there is no way of adding the silhouette score information to weigh vertices of the graph; the algorithm can therefore only solve a MIS problem (and not a MWIS, which is equivalent to clustering aggregation as showed in Section 4.1).

The sequence used, showed in Figure 5.4, was built with the Pulser framework [45], and is made up of a single pulse; the Rabi frequency function $\Omega(t)$ is obtained with the Pulser object `InterpolatedWaveform`, constructed with values `T, [1e-9, Omega, 1e-9]`, while the detuning function $\delta(t)$ with the same object, but with parameters `T, [delta_0, 0, -delta_0]`. The values of these parameters are displayed in Table 5.2.

Values of the parameters used to generate the sequence, namely `Omega`, `delta_0` and `T`, were obtained through Bayesian optimization. In particular, the function used was `gp_minimize`, contained in the package `skopt` [52], which found the parameters of the sequence that yielded clusterings with the highest total silhouette

Parameter	Value
Omega	0.9552
delta_0	-3
T	3873

Table 5.2: Parameters used to generate the Rabi frequency and detuning functions of the pulse contained in the sequence used for the experiments on Fresnel.

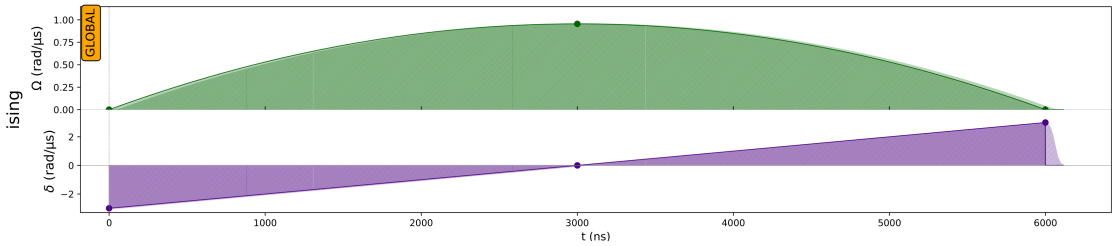


Figure 5.4: Sequence used in the the Fresnel experiments, both on QuTiP emulator and on real hardware.

score, after a total of 10 iterations.

Figure 5.5 shows the probabilities (in percentage) of reading a given bitstring from the output of the experiments on Fresnel, while Figure 5.6 and Figure 5.7 show a visualization of the clusterings corresponding to the three most likely bitstrings of QuTiP emulator and real hardware Fresnel experiments, respectively.

Figure 5.8 shows a comparison of the adjusted Rand index of DBSCAN and Spectral Clustering algorithms, used at the beginning of the aggregation algorithm, and of the clusterings corresponding to the three most likely bitstrings.

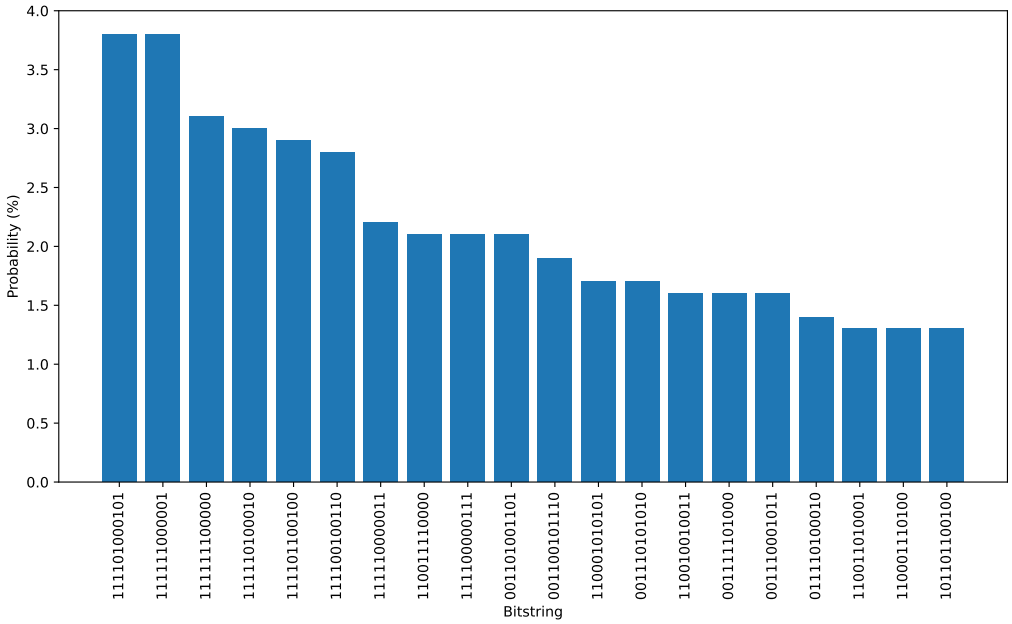
Due to the fact that the adjusted Rand index can only be computed when the two clusterings contain the same points, it was not possible to evaluate it for the three most likely solutions of real hardware Fresnel, since they all contain more points than the original dataset.

Figure 5.9 shows the probabilities (in percentage) of reading a bitstring with a given number of clusters, both in the QuTiP simulated Fresnel experiment, as well as in the real hardware Fresnel one.

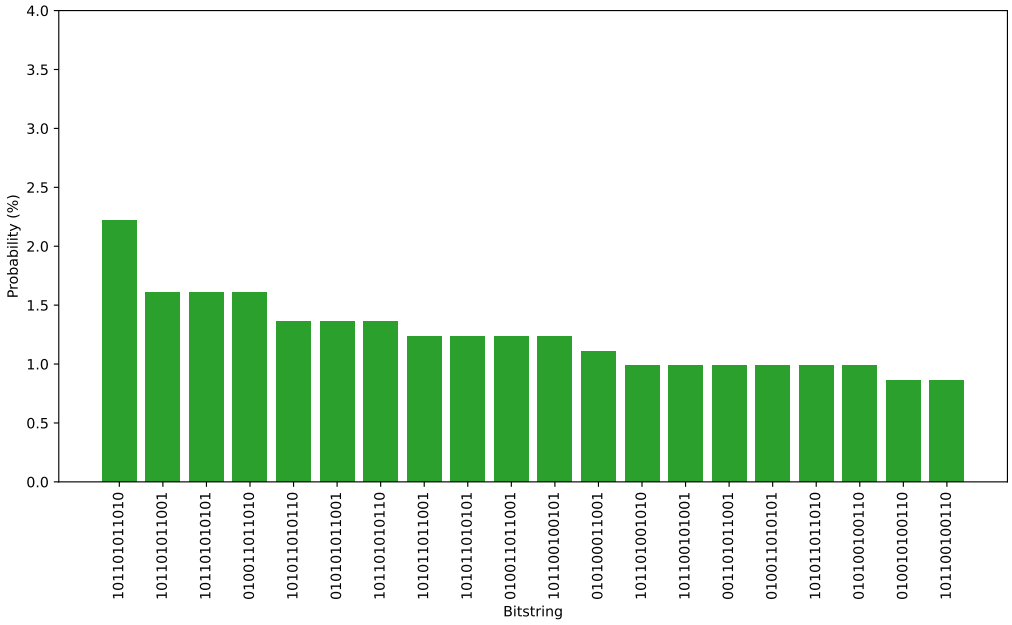
5.4.2 D-Wave's Advantage QPU

Experiments on D-Wave's Advantage QPU did not require the same manual register preparation of Fresnel, given that the Advantage QPU has a fixed topology; on the other hand, the embedding of the graph onto the machine's topology is not a trivial taks and represents a huge bottleneck of the process. Thanks to the possibility of adding constraints to the problem, it was possible to test the algorithm both without providing the information of silhouettes, in order to compare it to the experiments on Fresnel with equal conditions, and with the silhouette values as well. Both experiments were performed 1000 times, and results gathered in the form of bitstrings of length 12.

Figure 5.10 shows the probabilities (in percentage) of reading a given bitstring from the output of the experiments, while Figure 5.11 and Figure 5.12 show a



(a) QuTiP emulated Fresnel.

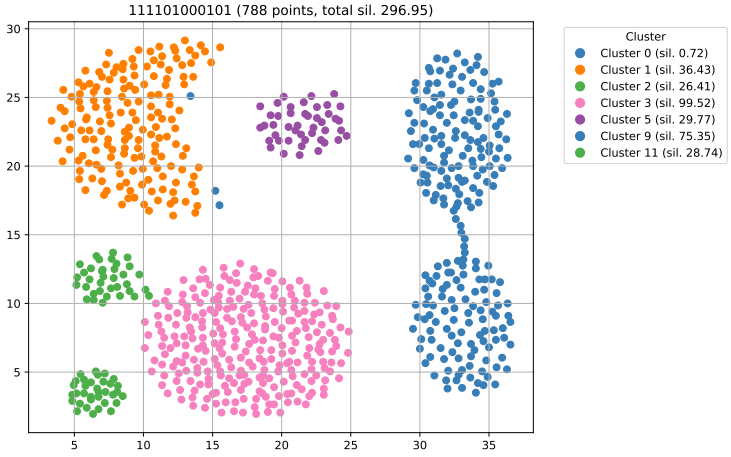


(b) Real hardware Fresnel.

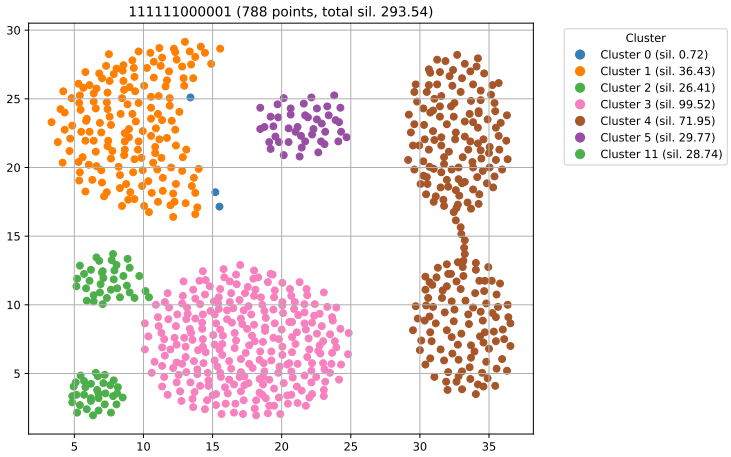
Figure 5.5: Probability (in percentage) of reading a certain bistring after measuring the experiments on QuTiP emulated and real hardware Fresnel. (a) shows the 20 most likely bitstrings from the experiment on the emulator, while (b) the 20 most likely bitstrings from the real hardware experiment.

visualization of the clusterings corresponding to the three most likely bitstrings of the experiments without and with the silhouette constraints, respectively.

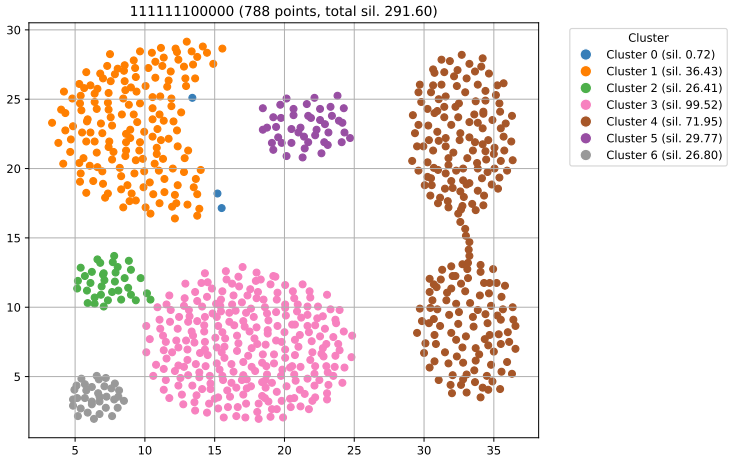
Similarly to the results of the experiment on real hardware Fresnel, the clusterings corresponding to the three most likely bitstrings do not contain the same points the original dataset, and therefore it is not possible to compute the adjusted



(a) Clustering corresponding to bitstring 111101000101.

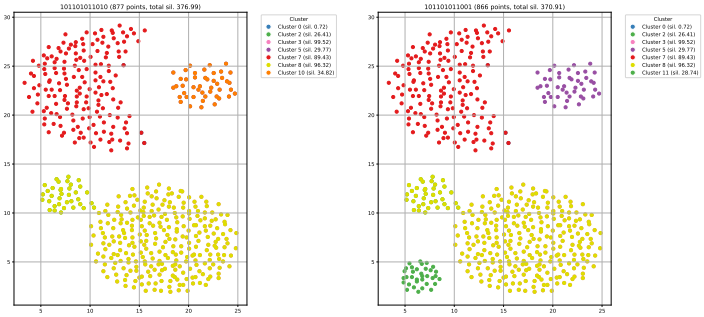


(b) Clustering corresponding to bitstring 111111000001.



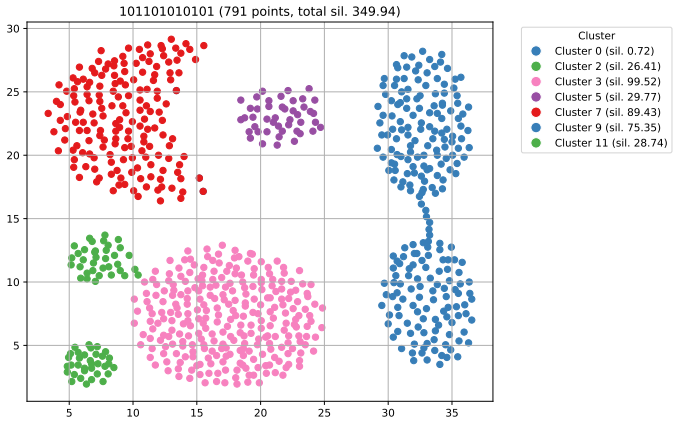
(c) Clustering corresponding to bitstring 111111100000.

Figure 5.6: Visualization of the clusters represented by the three bitstrings with highest probability for the experiment on the QuTiP emulated Fresnel; (a) corresponds to the most likely bitstring, while (b) and (c) to the second and third ones, respectively.



(a) Clustering corresponding to bitstring 101101011010.

(b) Clustering corresponding to bitstring 101101011001.



(c) Clustering corresponding to bitstring 101101010101.

Figure 5.7: Visualization of the clusters represented by the three bitstring with highest probability for the experiment on real hardware Fresnel; (a) corresponds to the most likely bitstring, while (b) and (c) to the second and third ones, respectively.

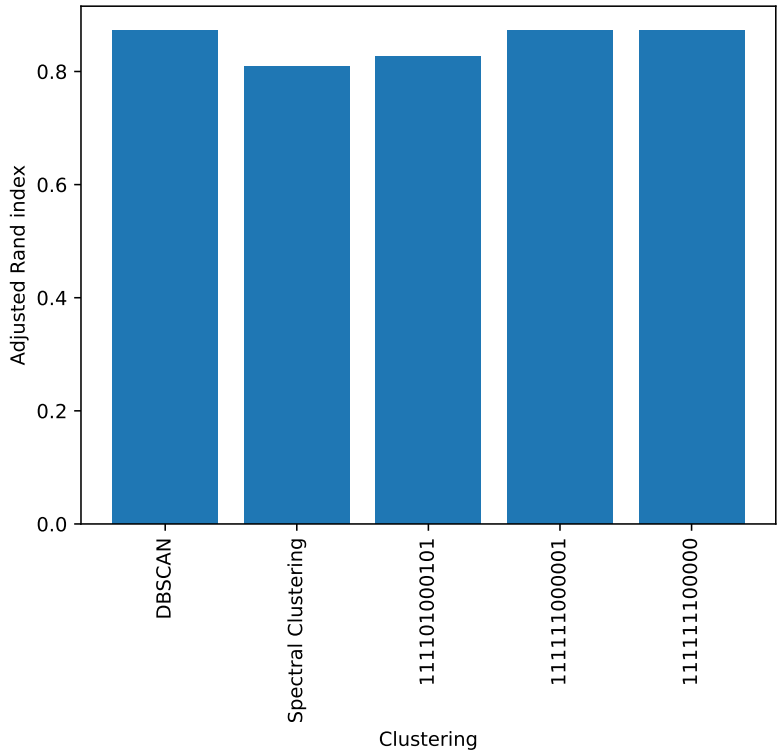


Figure 5.8: Comparison of the adjusted Rand index of DBSCAN and Spectral Clustering algorithms, and of the clusterings corresponding to the three most likely bitstrings.

Rand index.

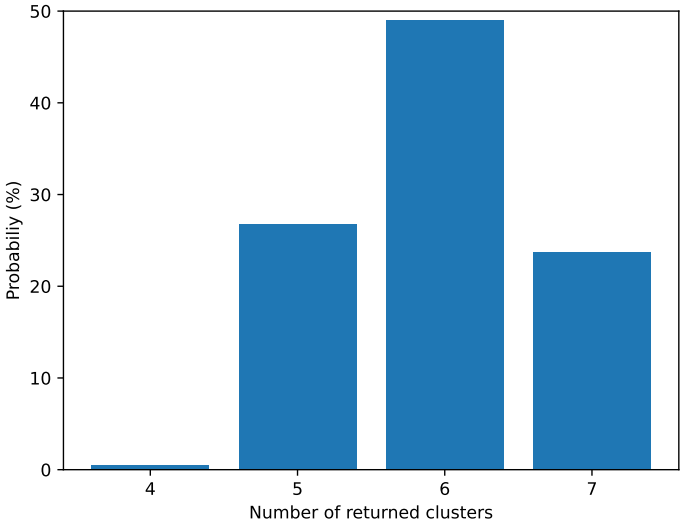
Figure 5.13 shows the probabilities (in percentage) of reading a bitstring with a given number of clusters, both with and without silhouette constraints.

5.5 Results discussion

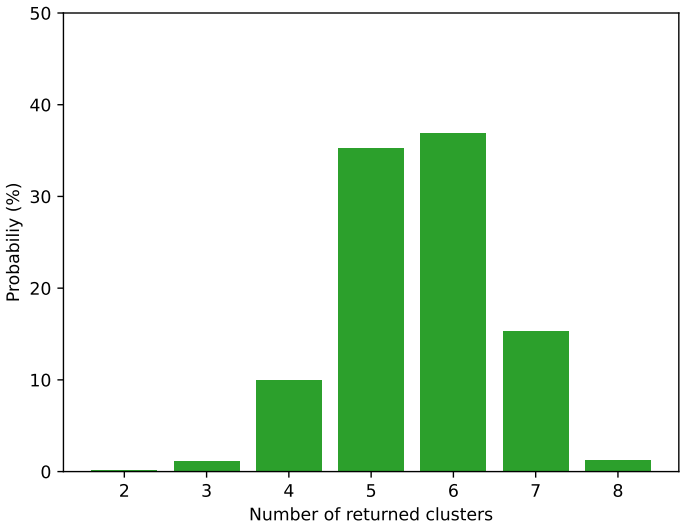
An important detail to consider when comparing the efficiency of machine is the total time necessary to execute the algorithm. Due to the inherent differences of the quantum platforms used to perform the experiments, defining how to measure the execution time is not obvious, and requires a more thorough discussion.

The processing cycle for Pasqal's Fresnel, shown in Figure 5.14, is made up of the following steps: register loading, sub-register rearranging, and quantum processing. Unfortunately, at the moment the only disclosed information regarding timing is that the quantum processing step in a processing cycle takes at most $100\mu s$. Such information concerning the time execution on the Fresnel QPU has been provided by the Pasqal Team via private communication.

On D-Wave's Advantage QPU, the time necessary to actually execute a job (in this context known as QMI, or Quantum Machine Instruction) is referred to as *access time*, which does not take in account the time necessary for data to go back and forth to the cloud, or for the QMI to wait in queue for the QPU, but only the time effectively used by the QPU to solve the problem. Access time can be further broken up in several phases, as shown in Figure 5.15 [53], namely

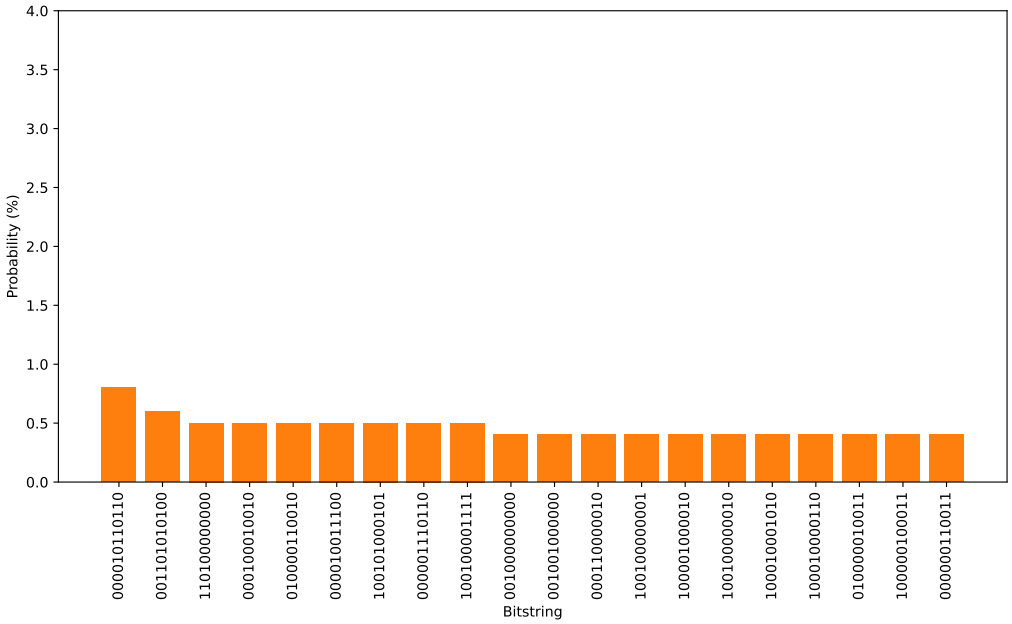


(a) QuTiP emulated Fresnel.

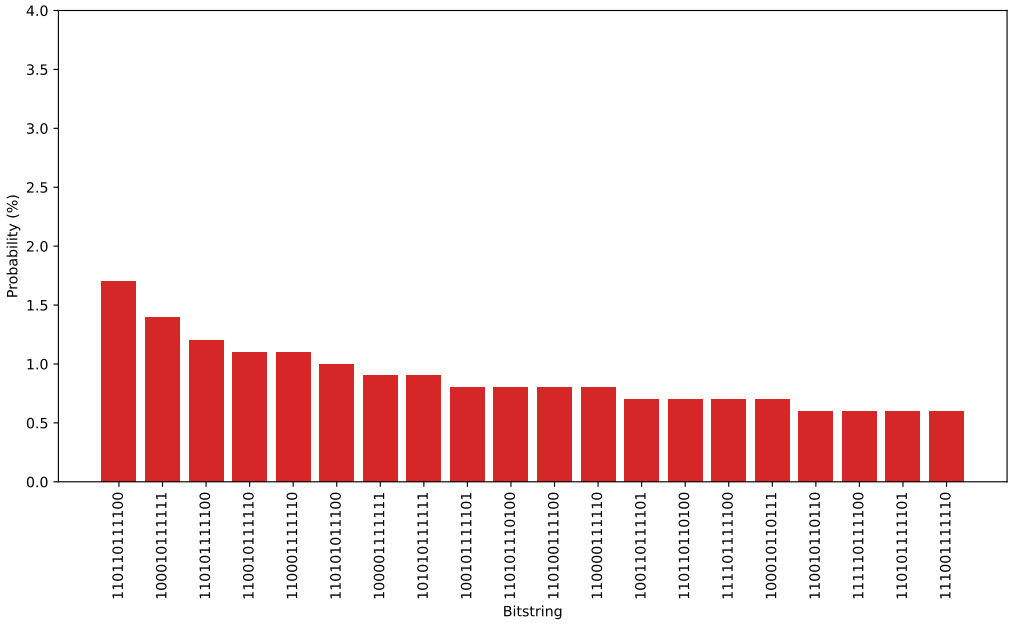


(b) Real hardware Fresnel.

Figure 5.9: Probability (in percentage) of reading a bitstring with a given number of clusters for the QuTiP emulated and real hardware experiments.



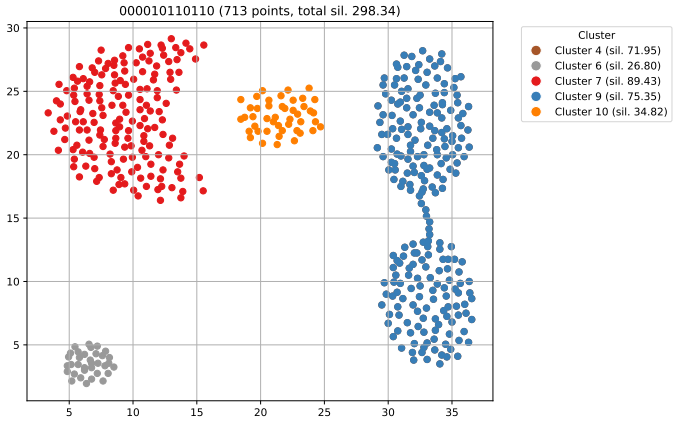
(a) Advantage QPU (without silhouette).



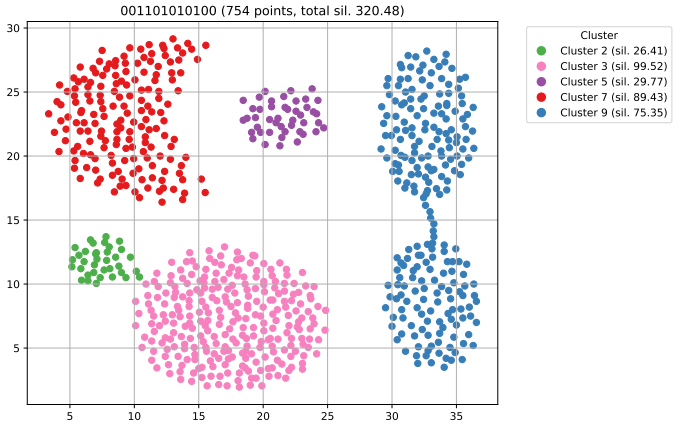
(b) Advantage QPU (with silhouette).

Figure 5.10: Probability (in percentage) of reading a certain bistring for the experiments on Advantage QPU, both without and with the silhouette constraint. (a) shows the 20 most likely bitstrings from the experiment that does not use the silhouette information, while (b) the 20 most likely bitstrings from the experiment that uses the silhouette information as a constraint.

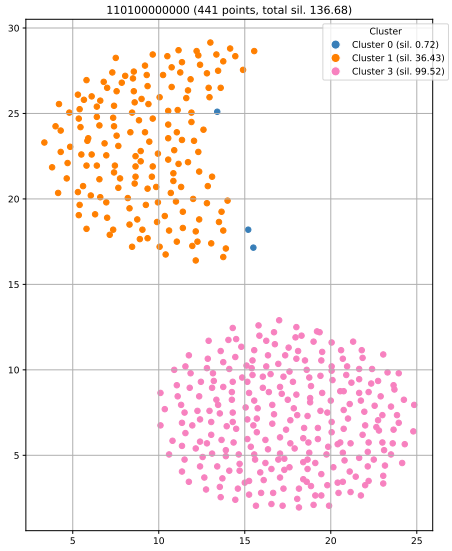
programming time, necessary to embed the problem onto the machine, plus sampling time, which is the time required to execute the algorithm and read results. In order to compare similar phases with Fresnel, the only time information taken in consideration will be the sampling time.



(a) Clustering corresponding to bitstring 000010110110.

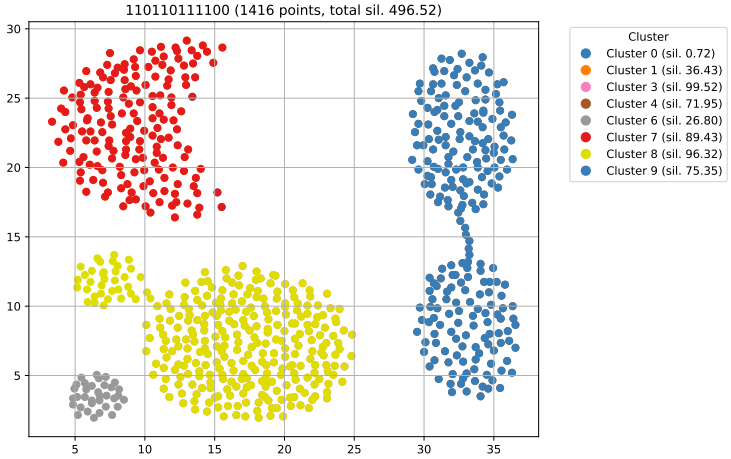


(b) Clustering corresponding to bitstring 001101010100.

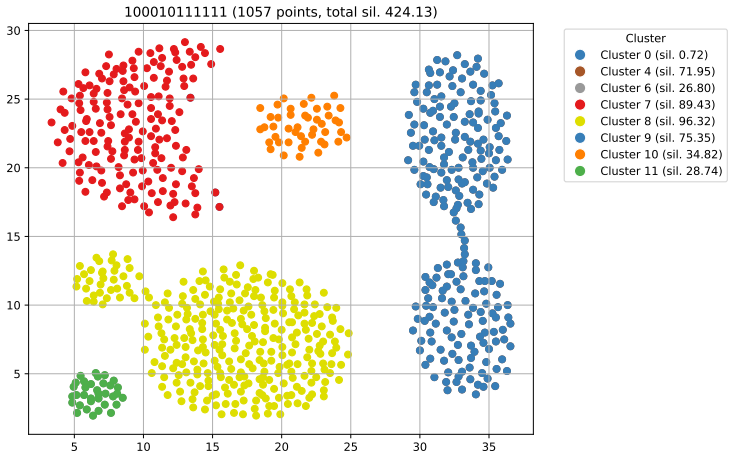


(c) Clustering corresponding to bitstring 110100000000.

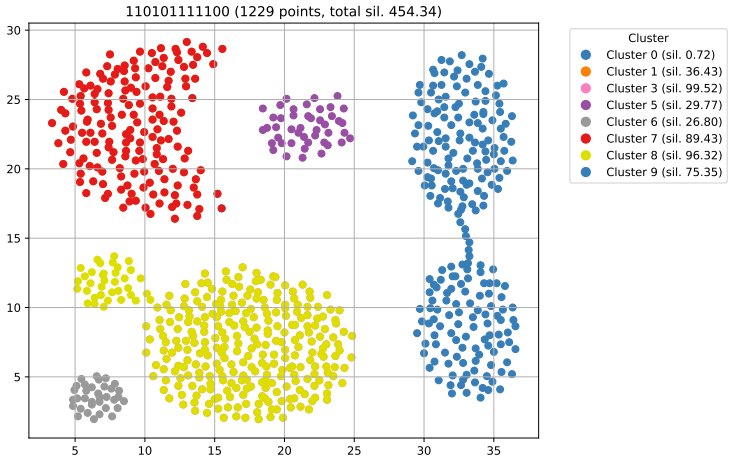
Figure 5.11: Visualization of the clusters represented by the three bitstrings with highest probability for the experiment on Advantage QPU without the silhouette constraints; (a) corresponds to the most likely bitstring, while (b) and (c) to the second and third ones, respectively.



(a) Clustering corresponding to bitstring 110110111100.

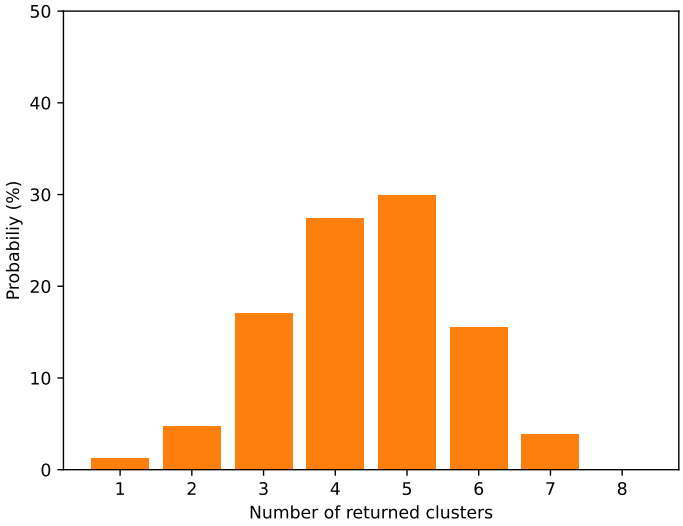


(b) Clustering corresponding to bitstring 100010111111.

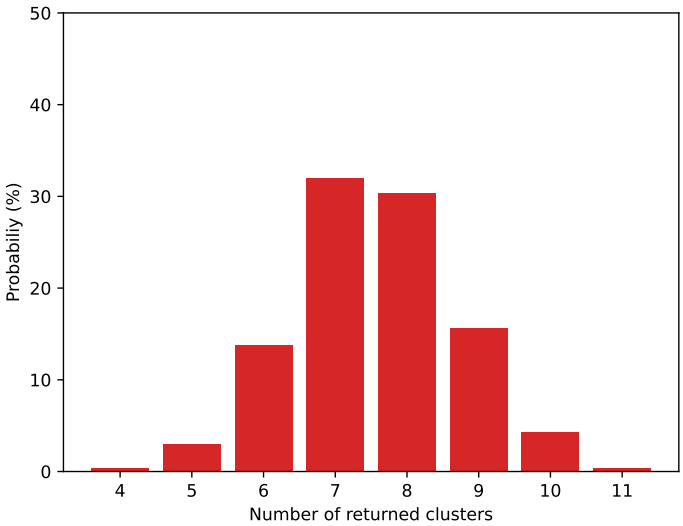


(c) Clustering corresponding to bitstring 110101111100.

Figure 5.12: Visualization of the clusters represented by the three bitstrings with highest probability for the experiment on Advantage QPU with the silhouette constraints; (a) corresponds to the most likely bitstring, while (b) and (c) to the second and third ones, respectively.



(a) Probability vs. number of clusters, without silhouette constraints.



(b) Probability vs. number of clusters, with silhouette constraints.

Figure 5.13: Probability of reading a bitstring with a given number of clusters for the Advantage QPU experiments; (a) refers to the results of the experiment without silhouette constraints, while (b) on the one with silhouette constraints.

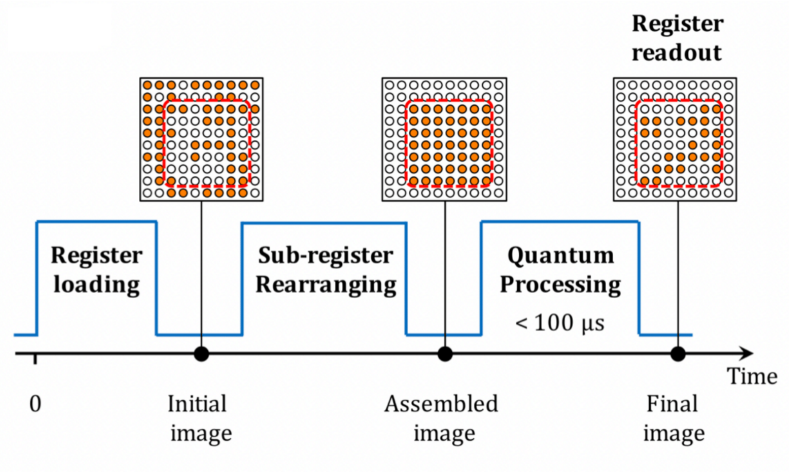


Figure 5.14: Steps of the QPU processing; at the moment, only quantum processing time has been disclosed, with a maximum value of $100\mu s$.

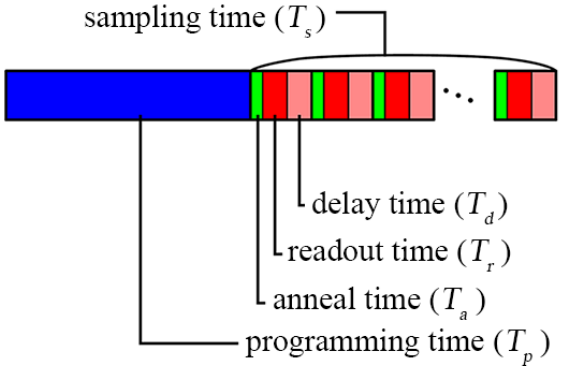


Figure 5.15: Detail of phases in QPU access time for D-Wave's Advantage QPU. Figure is taken from [53].

For the emulated version of Fresnel on QuTip, the execution time was obtained by calling Python's function `time.time()`, which returns the current UTC time, before and after the execution of the quantum simulation; execution time was then computed by taking the difference between the two timestamps.

A comparison of the execution times on the different platforms is shown in Table 5.3. It is possible to observe how the times of experiments run on real quantum hardware have comparable orders of magnitude (with the experiment on Advantage QPU without silhouette constraint being slightly faster, in the order of $10^4\mu s$, than the other two, in the order of $10^5\mu s$), while the time of the QuTiP emulated Fresnel experiment is significantly slower, at around $4,62 \cdot 10^9\mu s$.

In terms of the most likely clusterings returned by the execution of the algorithm, the experiment on QuTiP emulated Fresnel yielded better results than the other three platforms. The three most likely clusterings of QuTiP emulated Fresnel (Figure 5.6) are all made up of 7 clusters, like the optimal clustering, and contain all the points of the original dataset. The adjusted Rand index of these solutions have all values above 0.8, better than the ARI of the clustering obtained with the classical Spectral Clustering and comparable to the ARI of the clustering obtained with the classical DBSCAN. The three most likely solution returned

Platform	Total time
QuTiP	$4.62 \cdot 10^9 \mu s$
Fresnel	$1.00 \cdot 10^5 \mu s$
Advantage QPU (without silhouette)	$8.82 \cdot 10^4 \mu s$
Advantage QPU (with silhouette)	$1,30 \cdot 10^5 \mu s$

Table 5.3: Comparison of the execution times for the clustering aggregation algorithms on the different quantum platforms. The times of experiments run on real quantum hardware have comparable orders of magnitude (with the experiment on Advantage QPU without silhouette constraint being slightly faster), while execution on QuTiP emulated Fresnel is significantly slower.

by the experiments on the other three quantum platforms, on the other hand, returned a number of clusters different from the optimal 7, with a different number of points compared to the original dataset, which made it impossible to evaluate their adjusted Rand scores.

In terms of most likely number of clusters in the solutions returned by the execution of the algorithm, the experiment on Advantage QPU with the silhouette constraints yielded better results than the other three platforms, returning clusterings with 7 clusters in more than 30% of the repetitions. Both experiments on Fresnel returned clusterings with 7 clusters around 25% of the times (for QuTiP emulated) and 15% of the times (for real hardware), and the experiment on Advantage QPU without the silhouette constraints around 5% of the times.

Chapter 6

Conclusions

Neutral-atoms and annealing quantum technologies have emerged as two of the most promising approaches in the path towards practical usage of quantum computing, particularly in their ability to address complex optimization problems. These technologies are, however, far from being mature and well integrated in industry, and research is still required both to demonstrate the feasibility of employing them to solve industry relevant tasks, as well as to evaluate their performances.

In this thesis, we have proposed a hybrid algorithm to perform clustering aggregation, a technique in data science that aims to mitigate the issues of clustering algorithms (which tend to perform poorly on certain data distributions), by combining the results of multiple clustering algorithms in a more robust consensus clustering. The algorithm leverages the possibility of expressing clustering aggregation as a Maximum-Weight Independent Set (MWIS), where the ideal clustering must contain non-overlapping clusters that maximize the sum of weights attributed to them, in this case the sum of their silhouette scores. The next step was reducing the MWIS, completely or only partially, to problems addressable by quantum machines, in particular QUBO with constraints and MIS.

Tests of the algorithm were performed on Pasqal's Fresnel, a neutral-atoms based quantum computer, and D-Wave's Advantage QPU, a quantum annealer; additionally, the algorithm was also run on an emulator of Fresnel provided by Pasqal, which uses the QuTiP framework. Two clustering algorithms, DBSCAN and Spectral Clustering, were run on a toy dataset, producing a total of 12 clusters; after converting the problem to its corresponding graph representation, it was submitted to the platforms and solutions gathered after running the problem 1000 times on each platform.

Due to current technical limitations, it was not possible to run the exact algorithm on the Fresnel platforms (both real and emulated), namely the silhouette information could not be taken in consideration; on the Advantage QPU, the algorithm was tested both with and without the silhouette information. While it is possible to observe a higher degree of maturity of D-Wave's Advantage QPU if compared to Pasqal's Fresnel, the perspectives of flexibility offered by a neutral-atoms based processor are interesting and represent a new technology worth exploring. Additionally, technical advancements in Fresnel, allowing to control each qubit individually, may lead to the inclusion of the silhouette information in future tests of the algorithm.

The total running times for the whole 1000 repetitions were approximately similar, in the order of $10^5 \mu s$, with the exception of the experiment on emulated

Fresnel which ran around 10^4 times slower than the other three experiments.

The experimental results indicate that the most likely solutions yielded by QuTiP emulated Fresnel were better than those of the other platforms; specifically, all three most likely clusterings from QuTiP emulated Fresnel comprised 7 clusters, aligning with the optimal clustering and including all data points. The adjusted Rand Index (ARI) for these solutions exceeded 0.8, surpassing the ARI obtained through classical Spectral Clustering and matching that of classical DBSCAN. Conversely, the other three quantum platforms produced clusterings with a number of clusters different from the optimal 7 and did not maintain all original dataset points, precluding the evaluation of their ARI.

Regarding the most likely number of clusters, the experiment on Advantage QPU with silhouette constraints yielded the best results, achieving 7 clusters in over 30% of solutions, better than the percentages of the experiments on Advantage QPU without silhouette constraints (around 5% of solutions) and on Fresnel (25% for QuTiP emulated, 15% real hardware).

Results show that there is still a lot of room for improvement; even in the experiment on Advantage QPU that included the silhouette information, only 30% of the returned solution contained the correct number of clusters. At the same time, these results have demonstrated that the design of hybrid algorithms that include neutral-atoms and annealing technologies is a viable path to follow, and that the development of such algorithms could be used as a way to benchmark machines based on different technologies, in terms of the quality of results they produce.

Refinements of the algorithm, both in its classical and quantum part, could lead to future improvements. For example, an issue that was observed both in the experiments on real hardware Fresnel and on Advantage QPU is that the algorithm may return solution with repeated or overlapping clusters. This problem could be partially resolved by performing a classical post processing phase, that unifies repeated clusters or prunes overlapping clusters.

While it was not possible to demonstrate a quantum advantage neither on the neutral-atoms based processor, nor on the quantum annealer, this work was able to show the feasibility of addressing practical, industry relevant problems with quantum technologies, representing a first step towards the integration of quantum computers as accelerators for high performance computing.

Bibliography

[1] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6–7 (June 1982), 467–488. ISSN: 1572-9575.

[2] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), 1484–1509. ISSN: 1095-7111.

[3] Sandeep Kumar Sood and Monika Agrewal. "Quantum Machine Learning for Computational Methods in Engineering: A Systematic Review". In: *Archives of Computational Methods in Engineering* 31.3 (Dec. 2023), 1555–1577. ISSN: 1886-1784.

[4] Yuri Alexeev et al. "Quantum-centric supercomputing for materials science: A perspective on challenges and future directions". In: *Future Generation Computer Systems* 160 (Nov. 2024), 666–710. ISSN: 0167-739X.

[5] *One step closer to European quantum computing: The EuroHPC JU signs hosting agreements for six quantum computers — eurohpc-ju.europa.eu*. `https://eurohpc-ju.europa.eu/one-step-closer-european-quantum-computing-eurohpc-ju-signs-hosting-agreements-six-quantum-computers-2023-06-27_en`. [Accessed 12-10-2024]. 2023.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: a review". In: *ACM Comput. Surv.* 31.3 (1999), 264–323. ISSN: 0360-0300.

[7] Mahmoud SalahEldin Kasem, Mohamed Hamada, and Islam Taj-Eddin. "Customer profiling, segmentation, and sales prediction using AI in direct marketing". In: *Neural Computing and Applications* 36.9 (Dec. 2023), 4995–5005. ISSN: 1433-3058.

[8] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, Global Edition*. en. 4th ed. London, England: Pearson Education, Sept. 2017, pp. 770–772.

[9] Anne Solberg, Torfinn Taxt, and Anil Jain. "A Markov Random Field Model for Classification of Multisource Satellite Imagery". In: *Geoscience and Remote Sensing, IEEE Transactions on* 34 (Feb. 1996), pp. 100 –113.

[10] Francesco Marra et al. "Blind PRNU-Based Image Clustering for Source Identification". In: *IEEE Transactions on Information Forensics and Security* 12.9 (2017), pp. 2197–2211.

[11] Jinwei Li et al. "A comparative study of clustering algorithms for intermittent heating demand considering time series". In: *Applied Energy* 353 (Jan. 2024), p. 122046. ISSN: 0306-2619.

[12] Salvador Vega-Pons and Jose Ruiz-Shulcloper. "A survey of clustering ensemble algorithms". In: *International Journal of Pattern Recognition and Artificial Intelligence* 25.03 (2011), pp. 337–372.

[13] Alexander Strehl, Joydeep Ghosh, and Claire Cardie. "Cluster ensembles—A knowledge reuse framework for combining multiple partitions". In: *Journal of Machine Learning Research* 3.Dec (2002), pp. 583–617.

[14] *Comparing different clustering algorithms on toy datasets — scikit-learn.org*. `https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html`. [Accessed 20-10-2024].

[15] Fred Glover, Gary Kochenberger, and Yu Du. "Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models". In: *4OR* 17.4 (Nov. 2019), 335–371. ISSN: 1614-2411.

[16] Richard P. Feynman. "Quantum Mechanical Computers". In: *Between Quantum and Cosmos*. Princeton University Press, Dec. 2017, 523–548.

[17] David Deutsch. "Quantum theory, the Church–Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[18] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Nov. 2006, pp. 38–51. ISBN: 9780191916724.

[19] Adriano Barenco et al. "Elementary gates for quantum computation". In: *Phys. Rev. A* 52 (1995), p. 3457.

[20] Vivien M. Kendon, Kae Nemoto, and William J. Munro. "Quantum analogue computing". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1924 (Aug. 2010), 3609–3620. ISSN: 1471-2962.

[21] Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayral. "Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks". In: *Phys. Rev. A* 102 (5 2020), p. 052617.

[22] Carl Pomerance. "A Tale of Two Sieves". In: *Notices of the AMS* 43.12 (Dec. 1996), 1473–1485.

[23] Ivan Kassal et al. "Simulating Chemistry Using Quantum Computers". In: *Annual Review of Physical Chemistry* 62.1 (May 2011), 185–207. ISSN: 1545-1593.

[24] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014.

[25] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X.

[26] Kathleen E. Hamilton et al. "Accelerating Scientific Computing in the Post-Moore's Era". In: *ACM Trans. Parallel Comput.* 7.1 (Mar. 2020). ISSN: 2329-4949.

[27] Atanu Rajak et al. "Quantum annealing: an overview". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 381.2241 (2023), p. 20210417.

[28] Satoshi Morita and Hidetoshi Nishimori. "Mathematical foundation of quantum annealing". In: *Journal of Mathematical Physics* 49.12 (Dec. 2008). ISSN: 1089-7658.

[29] Antoine Browaeys and Thierry Lahaye. "Many-body physics with individually controlled Rydberg atoms". In: *Nature Physics* 16.2 (Jan. 2020), pp. 132–142.

[30] M. D. Lukin et al. "Dipole Blockade and Quantum Information Processing in Mesoscopic Atomic Ensembles". In: *Physical Review Letters* 87.3 (June 2001). ISSN: 1079-7114.

[31] Loïc Henriet et al. "Quantum computing with neutral atoms". In: *Quantum* 4 (Sept. 2020), p. 327. ISSN: 2521-327X.

[32] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. "Clustering aggregation". In: *ACM Trans. Knowl. Discov. Data* 1.1 (2007), 4–es. ISSN: 1556-4681.

[33] M. W. Johnson et al. "Quantum annealing with manufactured spins". In: *Nature* 473.7346 (May 2011), 194–198. ISSN: 1476-4687.

[34] Timothy Proctor et al. *Benchmarking quantum computers*. 2024.

[35] Karen Wintersperger et al. "Neutral atom quantum computing hardware: performance and end-user perspective". In: *EPJ Quantum Technology* 10.1 (Aug. 2023). ISSN: 2196-0763.

[36] Nikolaj Moll et al. "Quantum optimization using variational algorithms on near-term quantum devices". In: *Quantum Science and Technology* 3.3 (June 2018), p. 030503. ISSN: 2058-9565.

[37] *The Linpack Benchmark*. `https://top500.org/project/linpack/`. [Accessed 17-10-2024].

[38] Yulong Dong and Lin Lin. "Random circuit block-encoded matrix and a proposal of quantum LINPACK benchmark". In: *Physical Review A* 103.6 (June 2021). ISSN: 2469-9934.

[39] Frédéric Barbaresco et al. *BACQ – Application-oriented Benchmarks for Quantum Computing*. 2024.

[40] Nan Li and Longin Jan Latecki. "Clustering aggregation as Maximum-Weight Independent Set". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 782–790.

[41] *scikit-leran – cluster module*. `https://scikit-learn.org/1.5/api/sklearn.cluster.html`. [Accessed 20-10-2024].

[42] *scikit-learn – datasets module*. `https://scikit-learn.org/stable/api/sklearn.datasets.html`. [Accessed 17-10-2024].

[43] Mauro D'Arcangelo et al. "Leveraging analog quantum computing with neutral atoms for solvent configuration prediction in drug discovery". In: *Physical Review Research* 6.4 (Oct. 2024). ISSN: 2643-1564.

[44] *Pasqal Cloud Services Documentation*. `https://docs.pasqal.cloud/fresnel-job`. [Accessed 18-10-2024].

[45] Henrique Silvério et al. "Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays". In: *Quantum* 6 (Jan. 2022), p. 629. ISSN: 2521-327X.

[46] D-Wave Systems. *The Advantage Quantum Computer | D-Wave*. 2023.

[47] Sheir Yarkoni et al. "Quantum annealing for industry applications: introduction and review". In: *Reports on Progress in Physics* 85.10 (Sept. 2022), p. 104001. ISSN: 1361-6633.

[48] J.R. Johansson, P.D. Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (Aug. 2012), 1760–1772. ISSN: 0010-4655.

[49] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[50] William M. Rand. "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical Association* 66.336 (1971), pp. 846–850.

[51] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of Classification* 2.1 (Dec. 1985), 193–218. ISSN: 1432-1343.

[52] *scikit-optimize – Sequential model-based optimization in Python*. `https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html`. [Accessed 19-10-2024].

[53] *Operation and Timing | D-Wave System Documentation*. `https://docs.dwavesys.com/docs/latest/c_qpu_timing.html`. [Accessed 14-10-2024].