

## Quantum Algorithms for Solving Hard Constrained Optimization Problems

Parfait Atchadé Adelmou

<http://hdl.handle.net/10803/674103>

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons: <http://creativecommons.org/licenses/by/4.0>

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons: <http://creativecommons.org/licenses/by/4.0>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license: <http://creativecommons.org/licenses/by/4.0>

## DOCTORAL THESIS

Title	Quantum Algorithms for solving Hard Constrained Optimization Problems
Presented by	Parfait Atchadé Adélomou
Centre	La Salle International School of Commerce and Digital Economy
Department	Departament d'Enginyeria, Animació i Arts Digitals
Directed by	Dra. Elisabet Golobardes Ribé Dr. Xavier Vilasís-Cardona

# Acknowledgements

*To Martina and Africa*

This PhD thesis research has resulted from numerous encounters, experiences, consultations, conferences, networking, friendships and sleepless nights. There are many people to whom I owe a high debt of gratitude for their invaluable inspiration, comments, critiques, corrections, and references. All of which makes for an incredible and unforgettable academic culmination that now, notably and sadly, reaches the end after three years of hard, challenging and sometimes lonely research.

For this, one of my most sincere and profound thanks goes to the IBMQ team, especially to the *aqua* division represented by *Steve Wood*. For two years of correspondence, comments and technical exchange, we have come to understand many concepts and, above all feel, we are wrapped in a community that gave us all the facilities to achieve our goal. To Simone Severini from AWS-Braket for his support to achieve our goal and to the quantum community and people I met during this quantum mechanics and computing journey.

I make a special dedication to Martina to find a door to the quantum world through this work. A hope to understand and correctly model our environment and to have a tool to understand some of the problems or diseases that today have no cure. To Jenny, tireless, loving, respectful and helpful for all the tangible and intangible details to achieve this result. It reminds me of Bell's inequality; Two entangled particles, before the measurement, and even at an immeasurable distance, modifying one, the other remains faithful to the effects of the first and vice versa.

To Africa, since it will be, I believe, one of the indisputable pillars of quantum computing.

This thesis work is not the end but the beginning of a long journey in the quantum world.

I want to be open to any reader, researcher who wishes to share their point of view with me at [pifparfait@gmail.com](mailto:pifparfait@gmail.com).

Thank you

Parfait Atchadé Adélomou  
Barcelona 17 August 2020

## Abstract

In this research, we examine the optimization techniques to solve constraint problems. We study the quantum era and its players as well as IBM, D-Wave, Google, Xanadu, AWS-Braket and Microsoft, learn over its community, framework and facilities and learn deeply the quantum mechanics' postulates that serve to create the most efficient quantum computing algorithms. To know if it is possible to solve *Constraint Search Problem* (CSP) problems more efficiently with quantum computing, we have defined a scenario for both classical computing and quantum computing to have a good benchmark. First, the proof of concept is focused on the scheduling problem of social workers and later in the batching and picking issue as a generalization of the *Social Workers' Problem* (SWP). The scheduling problem of social workers is a class of combinatorial optimization problems that can be solved in exponential time at best. Because it belongs to the type of problem known as NP-Hard, which has a huge impact on our society, nowadays, the focus on the quantum computer should no longer be just for its enormous computing capacity. But also, for the use of its imperfection in this *Noisy Intermediate-Scale Quantum* (NISQ) era to create a powerful machine learning device that uses the variational principle to solve the optimization problem by reducing their complexity's class. We propose a formulation (quadratic) to solve the social workers' schedule problem efficiently using *Variational Quantum Eigensolver* (VQE), *Quantum Approximate Optimization Algorithm* (QAOA), *Minimum Eigen Optimizer* and ADMM optimizer. The quantum feasibility of the algorithm has been modelled in QUBO form, with docplex simulated Cirq, Or-Tools and tested on IBMQ computers. After analyzing the results of the previous approach, a scenario has been designed to solve the SWP as case-based reasoning, both quantumly and classically. Thus, we could contribute with a quantum algorithm focused on artificial intelligence and machine learning. In the SWP definition, if instead of patients we have batches and instead of social workers we get mobile robots, and generalizing the objective function and the restrictions, we have proposed a proof of concept and a new formulation to solve the problems of picking and batching as qRobot. We did a proof of concept in this part of the project through a Raspberry Pi 4. We tested the integration capacity of quantum computing within mobile robotics, with one of the most demanded problems in this industrial sector: picking and batching problems. The results have been promising using different technologies. Moreover, in case of computational need, the robot will parallelize part of the operations in hybrid computing (quantum + classical), accessing CPUs and QPUs distributed in a public or private cloud. Furthermore, we developed a stable environment (ARM64) inside the robot (Raspberry) to run gradient operations and other quantum algorithms on IBMQ, Amazon Braket (D-Wave), and Pennylane locally or remotely. To improve the execution time of variational algorithms in this NISQ era and the next, we have proposed EVA: a quantum Exponential Value Approximation algorithm that speeds up the VQE, that is, to date, the flagship of the quantum computation. With all this, we can still propose several quantum algorithms and answer the hypothesis of this thesis work.



# Contents

<b>I</b>	<b>State of the Art</b>	<b>25</b>
<b>1</b>	<b>Introduction</b>	<b>27</b>
1.1	Motivation . . . . .	30
1.2	Work context . . . . .	30
1.3	Goals . . . . .	30
1.4	Hypothesis . . . . .	31
1.5	Development of the thesis . . . . .	32
1.6	Thesis structure . . . . .	33
<b>2</b>	<b>Combinatorial Optimization problems</b>	<b>35</b>
2.1	Introduction . . . . .	35
2.2	Travelling Salesman Problem . . . . .	35
2.2.1	Constructive heuristics . . . . .	36
2.3	Job Shop Scheduling Problems . . . . .	36
2.4	Vehicle Routing Problem . . . . .	37
2.5	Summary . . . . .	37
<b>3</b>	<b>Systems and useful concepts</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Hamiltonian path and cycles . . . . .	39
3.3	Ising model and the Hamiltonian system . . . . .	40
3.4	The Hamiltonian of a TSP . . . . .	41
3.5	The Hamiltonian of a VRP . . . . .	42
3.6	Summary . . . . .	43
<b>4</b>	<b>Solving combinatorial optimization problems</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.2	Linear Programming . . . . .	45
4.2.1	The Objective function, decision variables and the constraints . . . . .	46
4.2.2	Formulation of linear programming of binary integers . . . . .	46
4.3	Quadratic Programming . . . . .	47
4.3.1	Quadratic Formulation . . . . .	48
4.3.2	Change of variables to simplify the quadratic formulation . . . . .	49
4.4	Quadratic programming with quadratic constraints . . . . .	51
4.5	Summary . . . . .	53

<b>5</b>	<b>Introduction Quantum Mechanics</b>	<b>55</b>
5.1	Introduction . . . . .	55
5.2	Moore Law . . . . .	57
5.2.1	State of the Quantum Technologies . . . . .	57
5.3	Noisy Intermediate-Scale Quantum . . . . .	59
5.4	Postulate 1 . . . . .	60
5.5	Postulate 2 . . . . .	61
5.5.1	Postulate 2' . . . . .	62
5.6	Postulate 3 . . . . .	63
5.7	Postulate 4 . . . . .	66
5.8	Postulate 5 and 6 . . . . .	66
5.9	Summary . . . . .	66
<b>6</b>	<b>Complexity Class</b>	<b>69</b>
6.1	Introduction . . . . .	69
6.2	Quantum complexity . . . . .	70
6.3	The deterministic Turing machine . . . . .	70
6.4	The probabilistic Turing machine. . . . .	71
6.5	Quantum Turing Machine . . . . .	71
6.6	Study of complexity . . . . .	72
6.7	Summary . . . . .	74
<b>7</b>	<b>Quantum Gates</b>	<b>75</b>
7.1	Introduction . . . . .	75
7.2	General U-gate . . . . .	75
7.3	Single qubit gates . . . . .	76
7.4	Gate I . . . . .	76
7.5	Gate NOT . . . . .	77
7.6	Gate Y . . . . .	77
7.7	Gate Z . . . . .	78
7.8	The Pauli Operators . . . . .	79
7.9	Hadamard gate . . . . .	79
7.10	Gate $R_\phi$ . . . . .	82
7.11	Gate T . . . . .	82
7.12	Gate S . . . . .	83
7.13	More than one qubit system. . . . .	83
7.14	Toffoli gate . . . . .	86
7.15	Summary . . . . .	86
<b>8</b>	<b>Quantum Computers</b>	<b>87</b>
8.1	Introduction . . . . .	87
8.2	Simulated annealing . . . . .	88
8.3	Quantum annealing . . . . .	89
8.3.1	Quantum Tunneling . . . . .	90
8.4	Quantum Computers . . . . .	92
8.4.1	Quantum gate-based computers . . . . .	95
8.4.2	Quantum annealing-based computers . . . . .	98
8.5	summary . . . . .	99

<b>9</b>	<b>Quantum Computing</b>	<b>101</b>
9.1	Introduction	101
9.1.1	Quadratic Unconstrained Binary Optimization Problems	101
9.2	Variational calculation	103
9.2.1	Variational Quantum Eigensolver	105
9.2.2	Noisy VQE for the Optimization problem	107
9.2.3	Quantum approximate optimization algorithm	108
9.3	Summary	111
<b>II</b>	<b>Contributions</b>	<b>113</b>
<b>10</b>	<b>Research Design</b>	<b>115</b>
10.1	Introduction	115
10.2	Proposed problem	115
10.3	Experimentation tools	117
10.3.1	Programming environments	117
10.3.1.1	PennyLane:	119
10.3.1.2	Cirq	120
10.3.1.3	Qsharp	121
10.3.1.4	IBM Quantum Experience	123
10.3.2	Experimentation algorithms	125
10.3.2.1	SPSA	126
10.3.2.2	SLSQP	126
10.3.2.3	COBYLA	127
10.3.3	VQE Experimentation	127
10.3.4	CPLEX	130
10.3.5	DOCPLEX	131
10.4	Summary	132
<b>11</b>	<b>Approaches to solve the SWP</b>	<b>133</b>
11.1	Introduction	133
11.2	Social Workers' Problem based on new VRP formulation	133
11.2.1	Social Workers' Problem based on a QUBO approach	135
11.2.2	The algebraic approach of the Hamiltonian Formulation	149
11.2.3	Vectorization form of our formulation	150
11.2.4	Mapping the SWP's formulation into a list of Pauli operators	151
11.2.4.1	Mapping of quadratic variables of the Hamiltonian in the Z-basis.	152
11.2.4.2	Coding the Hamiltonian into the Pauli Operators	152
11.2.4.3	Quantum solution from the ground up and closer to the Quantum Native form.	152
11.2.5	Resolution of our formulation with docplex	153
11.2.6	Cplex solver	153
11.2.7	Solving our Quantum SWP with MinimumEigenOptimizer	153
11.2.8	Solving our quantum SWP by using the ADMM Optimizer	154
11.2.9	Backtracking	155
11.2.10	First approach of the SWP's generalization	155
11.3	Research Results	158
11.3.1	Experimentation	158
11.4	Discussions	160
11.5	Benchmarks of SWP	161



11.5.1	Comparing VQE solution with classical solver . . . . .	162
11.5.2	Backtracking . . . . .	162
11.5.3	Comparison of the algorithms' complexity . . . . .	164
11.5.4	OR Tool SAT . . . . .	164
11.5.5	Comparing 10 scenarios of SWP with VQE . . . . .	165
11.5.5.1	Comparing 5 scenarios of SWP with VQE and QAOA . . . . .	165
11.5.6	Comparing 10 scenarios of SWP on VQE and QAOA . . . . .	166
11.5.7	Comparing 10 banks of the Social Workers' schedules on QAOA . . . . .	167
11.5.8	Comparing the SWP on all quantum algorithms . . . . .	167
11.6	Conclusion and Further Work . . . . .	169
11.6.1	Conclusions . . . . .	170
11.6.2	Future directions . . . . .	172
11.7	Summary . . . . .	174
<b>12</b>	<b>The quantum CBR (qCBR)</b> . . . . .	<b>177</b>
12.1	Introduction . . . . .	177
12.1.1	Related Work . . . . .	178
12.2	Case-Based Reasoning . . . . .	179
12.3	Implementation . . . . .	179
12.4	Quantum classifiers . . . . .	180
12.5	Variational Quantum Classifier . . . . .	181
12.5.1	The Ansatz . . . . .	181
12.5.1.1	Fidelity cost function . . . . .	182
12.6	Details of qCBR . . . . .	185
12.6.1	Memory Structure . . . . .	187
12.7	Results . . . . .	189
12.8	Discussions . . . . .	191
12.9	Conclusions and further work . . . . .	194
12.10	Summary . . . . .	194
<b>13</b>	<b>The qRobot: generalizing the SWP</b> . . . . .	<b>195</b>
13.1	Introduction . . . . .	195
13.2	Work Context . . . . .	196
13.3	Implementation . . . . .	199
13.3.1	The problem's formulation . . . . .	199
13.3.2	Picking and Batching formulation . . . . .	200
13.3.3	Mapping the classical to quantum optimization . . . . .	202
13.4	qRobot Results . . . . .	203
13.5	Discussions . . . . .	205
13.6	Conclusions and further work . . . . .	207
13.7	Summary . . . . .	210
<b>III</b>	<b>Conclusions</b> . . . . .	<b>213</b>
<b>14</b>	<b>Results</b> . . . . .	<b>215</b>
14.1	Introduction . . . . .	215
14.2	Discussion . . . . .	218
14.3	Conclusions and Further Work . . . . .	220
14.3.1	Conclusions . . . . .	220
14.3.2	Further Work . . . . .	221

14.3.3	International contribution and talent pool . . . . .	222
14.3.4	Schedule/Work plan . . . . .	226
14.3.5	Pedagogical contribution . . . . .	227
14.3.6	Contribution to business vision . . . . .	227
14.3.7	Lessons learned (Hard Skills) . . . . .	228
14.3.8	Consolidated skills (Soft Skills) . . . . .	228

## List of Publications

- ◇ The results of this thesis have been made public in 7 papers.
  - \* Atchade-Adelomou P., Golobardes Ribé E., Vilasís Cardona X. (2020) Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver. In: de la Cal E.A., Villar Flecha J.R., Quintián H., Corchado E. (eds) Hybrid Artificial Intelligent Systems. HAIS 2020. Lecture Notes in Computer Science, vol 12344. Springer, Cham. [https://doi.org/10.1007/978-3-030-61705-9\\_21](https://doi.org/10.1007/978-3-030-61705-9_21)
  - \* Atchade-Adelomou P., Golobardes Ribé E., Vilasís Cardona X. Formulation of the social workers' problem in quadratic unconstrained binary optimization form and solve it on a quantum computer. *Journal of Computer and Communications*. 2020 Nov 5;8(11):44-68. DOI: 10.4236/jcc.2020.811004
  - \* Atchade-Adelomou P., Alonso-Linaje G, Albo-Canals J, Casado-Fauli D. qRobot: A Quantum computing approach in mobile robot order picking and batching problem solver optimization. *MDPI AG*, vol 14, number = 7, pages = 194, 2021 May 11. Algorithms, url = <https://doi.org/10.3390/a14070194>
  - \* Atchade-Adelomou P, Casado-Fauli D, Golobardes Ribé E, Vilasís Cardona X. quantum Case-Based Reasoning (qCBR). arXiv preprint arXiv:2104.00409. 2021 Apr 1.
  - \* Atchade-Adelomou P., Golobardes Ribé E., Vilasís Cardona X. Using the Parameterized Quantum Circuit combined with Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers' schedule problem solver. arXiv preprint arXiv:2010.05863. 2020 Oct 12.
  - \* Atchade-Adelomou P., Alonso-Linaje G. Quantum Enhanced Filter: QFilter. arXiv preprint arXiv:2104.03418. 2021 Apr 7.
  - \* Alonso-Linaje G, Atchade-Adelomou P. EVA: a quantum Exponential Value Approximation algorithm. arXiv preprint arXiv:2106.08731. 2021 Jun 16.

# List of Figures

3.1	Shows a graph of 26 vertices in which a Hamiltonian cycle is highlighted . . . . .	40
3.2	A Lattice in the Euclidean plane [240] . . . . .	41
3.3	Five Lattices in the Euclidean plane [240] . . . . .	41
4.1	Variable change to simplify quadratic programming . . . . .	49
5.1	TBluefors dilution refrigerators used in the MIT Engineering Quantum Systems group to operate superconducting qubits at near-zero temperatures [235] . . . . .	56
5.2	The areas of quantum information science (QIS) . . . . .	56
5.3	Moore law Roadmap: Quantum Computing Stack Exchange [187] . . . . .	57
5.4	The number of research papers published per year in quantum computing and technologies [214] . . . . .	58
5.5	Quantum Computing is becoming the talk of the town [4] . . . . .	58
5.6	The number of qubits achieved by date and organization. Roadmap by MIT [137] . . . . .	59
5.7	Metrics and milestones to help monitor the development of quantum computing. Roadmap by MIT [214] . . . . .	60
5.8	Companies and startups involved in the 2018 quantum computing ecosystem. Roadmap by Quantum World Association . . . . .	61
5.9	Observability of a classical system . . . . .	67
5.10	Observability of a quantum system . . . . .	67
6.1	The conjectured evolution of the quantum complexity of the operator $e^{iHt}$ [38] . . . . .	72
7.1	One-qubit circuit of quantum gate association to illustrate unit transformation matrix $U$	75
7.2	Circuit of the Quantum Identity gate $I$ . . . . .	77
7.3	The Circuit of the quantum Pauli's first gate if we do not consider gate I . . . . .	78
7.4	Circuit of the quantum gate Y. The second gate of Pauli . . . . .	78
7.5	Circuit of the quantum gate Z. Third gate of Pauli . . . . .	78
7.6	Hadamard Gate Circuit. A significant gate to get the superposition . . . . .	80
7.7	A circuit in Qiskit of the operation equivalent to $HZH = X$ . . . . .	80
7.8	$R_\phi$ Gate of a qubit. This gate is significant for rotational operations . . . . .	82
7.9	$T$ Gate of one qubit. Remember that the inverse of this gate is not itself . . . . .	82
7.10	$S$ Gate of one qubit. Remember that the inverse of this gate is not itself . . . . .	83
7.11	A 3-qubit system where we apply a Hadamard gate to each qubit. With this application, we assign each of the input qubits with the same probability . . . . .	84
7.12	The two qubits $C_{NOT}$ gate . . . . .	85
7.13	The Toffoli gate implemented with 2 $C_{NOT}$ gate, 2 $R_{\phi \phi=\pi}$ gates and another gate $R_{\phi \phi=-\pi}$ . . . . .	86

8.1	The quantum computing ecosystem [76] . . . . .	88
8.2	Simulated annealing concept. The motion described by the particle but viewed uphill is described as a thermal jump. This is because Simulated annealing is a physics algorithm that mimics the behaviour of materials as they cool slowly. The "walker" is like an atom in a metal driven by temperature to reconfigure itself. These changes are random but are more likely to move to lower energy settings than higher energy configurations. That is why it is said that the "walker" follows a biased random movement. . . . .	89
8.3	Advantage of quantum tunnelling over thermal jump taking advantage of the Simulated Annealing concept . . . . .	89
8.4	one-dimensional potential barrier . . . . .	90
8.5	Full-Stack execution [46] . . . . .	93
8.6	compiler infrastructure [46] . . . . .	93
8.7	An example of general-purpose quantum microarchitecture [46] . . . . .	94
8.8	Example of experimental implementation of microarchitecture for (real) superconducting qubits [46] . . . . .	94
8.9	Example of the new microarchitecture for the quantum genome sequencing accelerator [46] . . . . .	95
8.10	Microsoft Quantum Computer [105]. . . . .	95
8.11	IBM Quantum Computer [18]. . . . .	96
8.12	Xanadu Quantum Computer [54]. . . . .	96
8.13	Inside D-Wave Quantum Computer [104]. . . . .	96
8.14	The challenge of simulating transmon quantum processors [143] . . . . .	97
8.15	Microsoft Quantum Development Kit. Provides an end-to-end scalable quantum development environment and leverages the Q# quantum programming language, which enables users to design, compile, and simulate quantum algorithms [167]. . . . .	98
9.1	Adaptative Variational Quantum Eigensolver . . . . .	105
9.2	Architecture of the Variational Quantum Eigensolver. The architecture of the quantum-variational eigensolver. Algorithm 1: Quantum states that have been previously prepared are fed into the quantum modules, which compute $\langle H_i \rangle$ , where $H_i$ is any given term in the sum defining $H$ . The results are passed to the CPU, which computes $\langle H \rangle$ . Algorithm 2: The classical minimization algorithm, run on the CPU, takes $\langle H \rangle$ and determines the new state parameters, which are then fed back to the QPU . . . . .	106
9.3	Minimization principle used in Variational Quantum Eigensolvers to empower Quantum Machine Learning . . . . .	107
9.4	From Annealing to QAOA. These graphs conceptually show us the differences/evolution between quantum annealing (left) that follows this adiabatic time evolution path. Simulated annealing (middle) follows the same way in discrete steps and QAOA that follows adiabatic time evolution path in $p$ steps. These define the accuracy of the solution. . . . .	109
9.5	QAOA algorithm operation. The system is initialized along the y-direction in the Bloch sphere in the $ +\rangle^{\otimes N}$ state. The unitary evolution under $H = X_0 + Y_1 + Z_2$ is implemented for $\gamma_i \beta_i$ angles for $p$ times. At the end of the algorithm, global measurements in the x and the y basis are performed to compute the average energy $\langle H \rangle = E(\vec{\gamma}, \vec{\beta})$ , which is compared to the theoretical ground state energy $E_{groundstate}$ . . . . .	110
9.6	QAOA's diagram process to perform it experimentally with quantum devices. In our study, 1000 samples are used to estimate the value of the cost function $E(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta}   H   \vec{\gamma}, \vec{\beta} \rangle$ at each variational iteration. . . . .	111
10.1	My research's path over these 3 years . . . . .	116

10.2	Using the Variational-Quantum-Eigsolver (VQE) as Quantum Machine Learning to creates an Intelligent social workers schedule problem solver . . . . .	117
10.3	Overview of all published quantum computing programming environments [96] . . . .	118
10.4	Detailed characteristics of all published quantum computing programming environments. Data, August 2018 [96] . . . . .	119
10.5	Xanadu bets with its PennyLane platform [44]. . . . .	120
10.6	Google bets with its Cirq platform. . . . .	120
10.7	Bet on Microsoft with its Qsharp platform [168]. . . . .	121
10.8	Azure Quantum Computing [167] . . . . .	121
10.9	The heat map on the results of analysis and documentation . . . . .	122
10.10	Results of the evaluation for the community analysis. For each project, we indicate whether there is a public development roadmap and whether the software is published in the form of releases. Besides, we report the GitHub community profile score, the total number of contributors, the type of discussion channel focused on the user and the developer, and the type of public code review process, specifically if it applies to internal (I) external (E) taxpayers. Data were obtained in August 2018 [96] . . . . .	123
10.11	Evaluation results for the static analysis of each project and its source code [96] . . . .	124
10.12	Components of Qiskit’s framework [241] . . . . .	124
10.13	IBMQ User Profile . . . . .	125
10.14	IBMQ Circuit Composer . . . . .	125
10.15	IBMQ Laboratory Profile . . . . .	126
10.16	IBMQ Jupyter Notebook Files view . . . . .	126
10.17	IBMQ Jupyter Notebook view . . . . .	127
10.18	From IBMQ’s Jupyter Notebook, the Qiskit version . . . . .	127
10.19	From IBMQ’s a code view from Jupyter Notebook . . . . .	128
10.20	VQE experiment with three qubits on linear and full entanglement . . . . .	129
10.21	IBMQ ILOG CPLEX Optimization Studio uses decision optimisation technology to create optimal plans, and schedules [160]. . . . .	130
11.1	Quantum Optimization in Qiskit. A high-level modelling language, a modular set of state-of-the-art quantum optimisation algorithms for different problem classes, leveraging Qiskit’s fundamental quantum algorithms as well as core circuit functionality; combined with a uniform and flexible interface for easy testing, benchmarking, and validation of models and algorithms [191]. . . . .	154
11.2	Generated 10 Social Workers’ schedules . . . . .	159
11.3	Generated 10 Social Workers’ Distance Matrix . . . . .	159
11.4	Generated 10 Social Workers’ Distance Nodes . . . . .	160
11.5	Outcomes from the SWP by the Backtracking algorithm. From Monday to Wednesday	161
11.6	Outcomes from the SWP by the Backtracking algorithm. From Thursday to Friday . .	162
11.7	Outcomes from the SWP by the Quantum Exact solver algorithm. From Monday to Wednesday . . . . .	163
11.8	Outcomes from the SWP by the Quantum Exact solver algorithm. From Thursday to Friday . . . . .	164
11.9	Outcomes from the SWP by the Quantum ASM Simulator from IBMQ. From Monday to Wednesday . . . . .	165
11.10	Outcomes from the SWP by the Quantum ASM Simulator from IBMQ. From Thursday to Friday . . . . .	166
11.11	Outcomes from the SWP by ibmq_16_melbourne v2.1.0 from IBMQ. . . . .	166
11.12	Outcomes from the SWP by ADMM optimizer from IBMQ. . . . .	167
11.13	Outcomes from the SWP by Minimum Eigen Optimizer from IBMQ. . . . .	167

11.14	Here we present the results by Monday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver. . . . .	168
11.15	Here we present the results by Tuesday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver. . . . .	168
11.16	Here we present the results by Wednesday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.	169
11.17	Here we present the results by Thursday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.	169
11.18	Here we present the results by Friday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver. . . . .	170
11.19	Here we present the results by analysing Friday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.	170
11.20	The structure of the <i>ibmq_melbourne</i> quantum device from IBM. This computer has fifteen superconducting qubits, which are connected by the $C_{NOT}$ gate. The bidirectionality of arrows reflects that each of the qubits can be both a control and a target.	171
11.21	Backtracking's behaviour depending on the execution time and the number of patients.	171
11.22	VQE's behaviour depending on the execution time and the number of patients and number of trials . . . . .	173
11.23	SWP solved by OR-Tool-SAT from google . . . . .	174
11.24	Results of the experimentation by comparing ten scenarios of the combinations of Social Workers' schedules for a fixed patient's location (for Monday). We plot the error bar of the median distribution of the location of each patient and their schedules concerning the number of repetitions of the quantum's instance. . . . .	175
11.25	Results of the experimentation by comparing six scenarios of the combinations of Social Workers' Problem on VQE and QAOA. By changing the number of patients, the social worker shot configuration intending to analyze the quantum cost to meet the optimal solution. We realize that the QAOA with the same settings as VQE finds the optimal solution with the little sample. We must indeed consider the high values of the seed parameter. . . . .	175
11.26	Results of the experimentation by hanging the number of patients, social worker shot configuration intending to analyze the quantum cost to meet the optimal solution. We realize that the QAOA with the same settings as VQE finds the optimal solution with the little sample. . . . .	176
11.27	The Standard Deviation expected total anneal time for 98% per cent success for each mapping value, with the best $\varepsilon$ for each shoot is shown. Our optimal case is for $\varepsilon = 0, 7$ . Our most representative cases are for $\varepsilon = 0, 4$ , $\varepsilon = 0, 7$ and $\varepsilon = 1, 1$ . . . . .	176
11.28	Results of the experimentation by comparing ten scenarios of the combinations of Social Workers' schedules for a fixed patient's location. We plot the error bar of the median distribution of the location of each patient and their schedules concerning the number of repetitions of the quantum's instance. . . . .	176
12.1	Case-Based Reasoning block diagram. For the standard CBR, two essential blocks are distinguished in this work. The classifier and the synthesiser. The classifier is made up of the retrieve and retain blocks and the re-use and revise blocks to make up the synthesis system. . . . .	178
12.2	VQE working principle based on the quantum variational circuit in Qiskit . . . . .	180
12.3	Quantum case-based system block diagram. In this scheme, to convert the classic CBR into quantum one, it is proposed to change the retrieve and retain blocks for a quantum variational classifier and the re-use and revise blocks for a synthesis system based on VQE with initial_point. . . . .	181

12.4	Bloch sphere, where the infinite Hilbert vector space resides. It is worth remembering that Hilbert space is the immense place where the states that describe a quantum system live. Let $N$ be the number of qubits and let $2^N$ be the space dimension. Given the 65-qubit IBM processor, it will be $3.6893510^{19}$ dimensions. Many works of literature lead us to use or, at least, to think about using the quantum computer in Machine Learning (ML). By definition, the Hilbert vector space has defined these operations (internal and external products and mapping inputs in a large space) that a quantum computer performs natively and very easily [130]. . . . .	182
12.5	$R_y$ and $R_z$ Ansatzes without entanglement used in qCBR experimentation. . . . .	183
12.6	$R_y$ and $R_z$ Ansatzes with $CRZ$ entanglement used in qCBR experimentation. . . . .	183
12.7	$R_y$ and $R_z$ Ansatzes with $CNOT$ entanglement used in qCBR. . . . .	183
12.8	Three-qubit quantum classifier circuit without entanglement. . . . .	184
12.9	Three-qubit quantum classifier circuit with entanglement by using $CZ$ or $CNOT$ gates	184
12.10	Representation of the generation of the $n$ weekly schedules of the SWP. The last plane, the one to the right of everything, represents all the SWP classes. The overlapping effect generated by the social workers' problem's characteristics and experimentation scenarios can be observed. This experimentation leads us to use the ICA technique to have a resulting dataset regardless of the schedules. . . . .	185
12.11	Fundamental processes to apply ICA to SWP dataset. The first thing that is done is to centre the data $x$ by subtracting the mean, balancing the data $x$ removing its variance, and calculating the unmixing matrix of $W$ . Then, the new value of $w$ is calculated. Then $w$ is normalised before checking if, with the said value, the algorithm converges or not. If it does not converge, a new $w$ must be recalculated, and if it converges, calculate the scalar product of $\langle x, y \rangle$ to obtain the independent weekly schedules. . . . .	186
12.12	qCBR's Memory structure. The use the cross-validation technique helps to improve the quality of the classifier training[59] . . . . .	187
12.13	Memory structure used for the Training System. A label identifies each class. And in this case, $A$ is one of the labels. For more detail, refer to the code [27]. . . . .	188
12.14	The SWP is represented in vector form to take advantage of the Hilbert vector space's characteristics within quantum computing. It is seen that each patient represents a dimension, and the points that make up the dataset are a dimension of $2n$ coordinates, $n$ being the total number of patients. In this figure, to simplify the understanding, we use two patients, therefore, two dimensions . . . . .	188
12.15	Block diagram proposed for the resolution of qCBR, considering a real dataset with an overlap problem between the data components. This block diagram contemplates the treatment of the input data and the use of ICA[188] and PCA[248] before training and classifying the data. In this version, a classifier based on the re-uploader has been designed to handle the classification tasks. And for the synthesis tasks, a decision tree passed in the classifier predictions have been used together with the VQE plus the <i>Initial_point</i> . . . . .	189
12.16	Comparative graphs between different ansatzes, taking into account the classifier's accuracy as a function of depth. Represents the evolution of the ansatzes of dimension two and eight . . . . .	190
12.17	Energy comparison between VQE algorithm without using or not the initial point . . . . .	193
13.1	Distribution of considered order picking problems based on the percentage of publications. . . . .	196
13.2	Technical Specifications of the Quantum Hardware Technologies (Gate-based superconducting qubits, Gate-based ion traps and Quantum annealing) available in Amazon Braket. . . . .	197



13.3	Technical Specifications of Quantum Simulators systems where we can see it state vector simulator (34 qubits) and tensor network simulator (50 qubits). . . . .	197
13.4	We propose a robot that prepares batches and increases the efficiency of picking in a warehouse, taking advantage of the classic Machine Learning experience and leveraging hybrid computing (classical + quantum) in the cloud and distributed. This robot uses the Optimal routing strategy to calculate the shortest route, regardless of the layout or location of the items. . . . .	198
13.5	VQE working principle based on the quantum variational circuit. . . . .	199
13.6	Scenario 1, Independent lots. The robot receives the orders and calculates which order is the most optimal according to the coordinates in which each product is found. In this example, lot 4 is the most optimal. . . . .	200
13.7	Scenario 2, Collecting products in the same route from different batches. The robot will calculate a path that includes all the products to optimize their collection in a single journey. For example, if the product from Lot 2 is next to one from Lot 1, the robot will pick it up and store it in the basket from Lot 2. . . . .	201
13.8	Structure of our warehouse with pick locations. The warehouse has a rectangular layout with no unused space. We use all the parallel corridors. This proof of concept contemplates a single warehouse used to take the order and deliver it, and it is also divided into blocks, which contain slots for storing products, and transverse aisles separate them. The cross aisles do not have any products but allow the collector to navigate the warehouse. We base our picking strategy on minimising the route and optimising batch preparation. We do not contemplate shelving of different levels. . . . .	203
13.9	qRobot operation diagram. This diagram shows all the necessary blocks and processes that allow modelling the picking-batching problem and its proper functioning on the Raspberry Pi 4. . . . .	204
13.10	In these graphs, we can observe the results of the algorithm in different scenarios. A different colour represents each qRobots; qRobot number 1 is red, next is blue, and the third is yellow, so on. While the depot is the 0 node in yellow color and the rest of the nodes are represented in blue. The weights of each item (not normalized) in kg are $w_0 = 0, w_1 = 8, w_2 = 8, w_3 = 3$ and $w_4 = 3$ . The maximum capacity of each qRobots is 45. In this case, we have 4 items and the possibility of using up to 3 qRobots. Reading the images from left to right, we see that the nodes and their respective distances are shown in the first image. The second image shows the result of the algorithm having a qRobot. In the third and fourth images, we can see two different cases solved by two qRobots. And finally, in the fifth and sixth images, we can see two other issues solved by three qRobots. It is important to highlight that our algorithm in this proof of concept minimises the distance travelled and optimises the number of qRobots necessary to solve the cases presented. If it judges that the task can be performed with a single qRobot, it will not send 2 qRobots. . . . .	206

13.11	In these graphs, we can observe the results of the algorithm in different scenarios. A different color represents each qRobots; qRobot number 1 is red, the next is blue, the third is yellow, and so on, while the depot is the 0 node in yellow color, and the rest of the nodes are represented in blue. The weights of each item (not normalized) in kg are $w_0 = 0, w_1 = 8, w_2 = 8, w_3 = 3, w_4 = 3, w_5 = 1, w_6 = 2,$ and $w_7 = 4$ . The maximum capacity of each qRobots is 45. In this case, we have 7 items and the possibility of using up to 4 qRobots. Reading the images from left to right, we see that the nodes and their respective distances are shown in the first image. The second image shows the result of the algorithm having a qRobot. In the third and fourth images, we can see two different cases solved by two qRobots. And, finally, in the fifth and sixth images, we can see two other issues solved by three qRobots. It is important to highlight that our algorithm in this proof of concept minimises the distance travelled and optimises the number of qRobots necessary to solve the cases presented. If it judges that the task can be performed with a single qRobot, it will not send 2 qRobots. . . . .	207
13.12	This figure shows that we judge important environments to carry out quantum computing to robotics and beyond. We can see the correct installation of TensorFlow 3.2.1 as required for all gradient operations; see the installation of PennyLane version 14.1, the installation of the latest version of Amazon Braket, and all the packages of the newest version of Qiskit 0.25 minus the qiskit-machine-learning package. . . . .	209
13.13	In this figure, we can see the correct installation of the Jupyter package and the Jupyter notebook that has been our environment of proof of concept. With this, everything is ready to import or write code in the different frameworks mentioned above (IMBQ, AWS-Braket, PennyLane and D-Wave). . . . .	209
13.14	This figure shows the files window through the CyberDuck client SSH [122] viewer with the directory and file structure. And on the left, you can see the terminal that gives access to the qRobot. To access the qRobot by SSH, the username and password are required. Everything is configurable [1]. . . . .	210
13.15	. . . . .	211
14.1	In this figure, we can observe a workflow of a quantum algorithm. First, let us remember that a quantum computer or hardware only executes a quantum circuit. But we can use classical algorithms to generate a description of our quantum circuit efficiently, so it would be smart to use it. Our quantum algorithm is born to execute or answer problems/approaches very expensive for classical computing. A good quantum algorithm would be the polynomial scale with qubits. This figure shows that we can clarify it polynomially given a problem and follow these guidelines. . . . .	217
14.2	We have installed the following frameworks successfully (Qiskit, PennyLane, AWS-Braket) on our Raspberry Pi 4 under the ARM64 operating system. More information about the qRobot platform can be found at Ref. [26] . . . . .	224



# List of Tables

9.1	Comparison between Ising and QUBO representation and related terminology. . . . .	103
10.1	Schedule of patient visits without any association with social workers. Where $U_1$ to $U_2$ are the patients (users) and equal to the variable $i$ or $j$ of the mathematical formulation	116
10.2	Status and release of the modules towards Qiskit 1.0 . . . . .	128
11.1	QUBO $Q$ Matrix for the Social Workers' Problem . . . . .	142
11.2	$J_{ij}$ Interaction forces between grid neighbours. We assume that $J_{ij} = 0$ for $i$ and $j$ are not adjacent. . . . .	143
11.3	Calculated values of the external force $h_i$ . . . . .	148
11.4	This is <code>ibmq_16_melbourne_calibrations</code> . . . . .	172
11.5	Experiment scenario using the Backtracking technique with 2, 3, 4 and 5 patients . . .	173
11.6	VQE evolution of trials and number of patients . . . . .	173
12.1	Table of the NN brute force's, k-d tree's, and Ball tree's complexity method. Where $d$ , is the data dimensionality, $N$ is the number of points in the training dataset and $K$ is the algorithm's neighbours' number . . . . .	178
12.2	The result of the qCBR with a variational classifier and using the VQE and the Initial_point with some decision trees as a synthesizer (12.15). This table shows the different studies made as a function of the quantum circuit's depth (layer). number of the patients and the social workers. The accuracy of the classifier is the maximum with the number of layers equal to 10. SW denotes Social Workers. . . . .	190
12.3	The result of the qCBR for many patients and social workers fixed at 5 and 4, respectively. The better behaviour of qCBR can be observed for some cases greater than 240. To have a good functioning of the qCBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 92%. The "accuracy" value is the percentage of the number of correct solutions found by the qCBR. A hyphen (-) denotes no solution found within the 20 cases. These tests were done for the quantum circuit depth (layers) equal to 8. . . . .	191
12.4	The result of the qCBR for many patients and social workers fixed at 4 and 3, respectively. The better behaviour of qCBR can be observed for some cases greater than 240. To have a good functioning of the qCBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 91%. The "accuracy" value is the percentage of the number of correct solutions found by the qCBR. A hyphen (-) denotes no solution found within the 20 cases. These tests were done for the quantum circuit depth (layers) equal to 8. . . . .	191

12.5	The classical CBR result on KNN classifier for many patients and social workers fixed at 5 and 4, respectively. The better behaviour of this CBR can be observed for some cases greater than 240. To have a good functioning of the CBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 52.6%. The "accuracy" value is the percentage of the number of correct solutions found by the CBR leveraged on KNN, applying a 10-KFold cross-validation process. A hyphen (-) denotes no solution found within the 20 cases. All these tests were done for the layer equal to 1. . . . .	192
12.6	The classical CBR result on KNN classifier for many patients and social workers fixed at 4 and 3, respectively. The better behaviour of this CBR can be observed for some cases greater than 100. To have a good functioning of the CBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 70.3%. The "accuracy" value is the percentage of the number of correct solutions found by the CBR leveraged on KNN, applying a 10-KFold cross-validation process. A hyphen (-) denotes no solution found within the 20 cases. All these tests were done for the layer equal to 1. . . . .	192
12.7	CBR with a neural network classifier and a backtracking algorithm as a synthesizer. SW denotes Social Workers. . . . .	192
12.8	CBR with a KNN classifier and a backtracking algorithm as a synthesizer. SW denotes Social Workers. . . . .	193
12.9	Table of the complexity of qCBR counting the PCA cases and the ICA complexity. In this case, K, the number of shots of the VQE, is fixed to 50 . . . . .	193
13.1	In this experimentation, both the number of qRobots and their capacities (maximum load) are fixed and are worth 1 and 45 respectively. We compare the execution time of our algorithm in the different public access simulators in the market, solving the picking and batching problem. We see that for issues of this nature, and especially for the number of qubits required in each scenario, the behaviour of the D-Wave is the desired one at the temporal level, comparing it with technologies based on quantum gates. However, it should be noted that for the experiments on numbers of qubits less than 20, the behaviour of these simulators is equated with the D-Wave. This experimentation helps to have a clear vision about the feasibility of this proof of concept.	205
13.2	Table of the computational results of our picking and batching algorithm on only 1 qRobot. The value of time is an average and includes the waiting time, queue, execution and return of the solution. In the case of K is equal to 2 for 9 items with the qRobot capacity equal to 15, the number of qubits is 188. The execution time is on AWS Braket and on the D-Wave Advantage_system1 quantum computer. We can realize that there is a latency time in executing the algorithm from the qRobot to the real quantum computer. In the tests we've done, the quantum computer is in the US West (Oregon). Of all the tests that we have done, we have had an average latency time of about 2 plus all the work management processes that rises more or less to about 8 seconds. For the number of qubits exceeding 30, it is very convenient to use AWS-Braket (Advantage_system1.1)[247] instead of Qiskit or Pennylane. By the number of qubits and the execution time, that is differentially better. This scenario makes the use of quantum in robotics very viable. For the tests with a value of $M$ lower than those in the table, the number of qubits is relaxed, and the execution time is improved. This leads us to normalize the weights of the batches. Since the number of qubits follows the formula $K(n + 1)(n + 2) + K \lceil \log_2 M \rceil$ . . . . .	208
13.3	In this table, we only consider the running time of the quantum algorithm on the real quantum computer from the qRobot (Advantage_system1.1[247]), not counting latency time, job creation, and job return time. . . . .	208

14.1 PhD Tasks development . . . . . 226



# Acronyms

**CBR** Case-Based Reasoning.

**EVA** quantum Exponential Value Approximation.

**JSP** Job Shop Problem.

**JSSP** Job Shop Scheduling Problem.

**NISQ** Noisy and Intermediate Scale Quantum.

**QAOA** Quantum Approximate Optimization Algorithm.

**qCBR** quantum Case-Based Reasoning.

**QEC** Quantum Error Correction.

**QECC** Quantum Error Correction Concept.

**QFilter** quantum Filter.

**QML** Quantum Machine Learning.

**qRobot** quantum Robot.

**QSVM** Quantum Super Vector Machine.

**QUBO** Quadratic Unconstrained Binary Optimization.

**SVM** Super Vector Machine.

**SWP** Social Workers' Problem.

**TSP** Travelling Salesman Problem.

**VQA** Variational Quantum Algorithm.

**VQC** Variational Quantum Classifier.

**VQE** Variational Quantum Eigensolver.

**VRP** Vehicle Routing Problem.





## Part I

# State of the Art



# Chapter 1

## Introduction

One of the most frequent problems in our daily lives is optimisation problems, and for a long time, mathematics has always looked for a way to solve them. The differential calculus technique helps solve these problems for certain purposes that fulfil certain properties. However, many functions that appear when trying to solve or model these tasks do not satisfy the hypotheses of the differential calculus theorems or have additional restrictions that nullify their usefulness.

One of the successes of the last century in the field of mathematics was the development of the simplex algorithm [81] for solving optimisation problems. This finding allowed and continues to find the global optimum of a linear function with linear constraints. However, some limitations of this technique are observed when one wants to generalise the algorithm or tend to solve more complicated problems that are not only summarised in finding the minimum of a linear function with linear restrictions [80]. As a result, several numerical calculation tools were born to approximate optimisation tasks under certain regularity conditions. Still, they are not enough to try to solve all the problems. Most of the techniques to solve these optimisation problems require continuity and differentiability conditions. However, many of the functions that appear in these tasks to be solved do not meet these conditions, and in some cases, it does not even make sense to talk about these properties because we are dealing with a discrete set. The field that deals with these last problems are known as combinatorial optimisation.

The *Travelling Salesman Problem* (TSP) [150] belongs to this kind of problem where the brute force algorithms, due to their computational impracticality cost, grows factorially as a function of  $N$ , with  $N$  being the number of cities, making it difficult to have an efficient optimisation algorithm.

Due to this fact, there is then the need to develop specific algorithms for each problem, making an in-depth study of this field and each of the available techniques essential.

Some of these techniques are what are known as heuristic algorithms. Well, the main objective of these heuristic algorithms is to approach the global minimum of a function, but without guaranteeing that it will be achieved. Instead, they find a method of finding good solutions for a certain process (task). As we discussed earlier, the same argument that prevents the impossibility of finding global minimum search algorithms for any function makes it not easy to choose the appropriate heuristic algorithm for each problem either. But, again, the operation of this will depend on the nature of the function with which we are working, making it necessary to adapt it.

Combinatorial optimisation problems in which social workers visit their patients in their respective homes and attend to them at a specific time, called *Social Workers' Problem* (SWP), are similar to TSP due to its NP-Hard characteristic require another focus if patient numbers grow considerably. The

problem of optimising the schedules of social workers who visit their patients at home is a scheduling and routing problem of the NP-Hard class [78]. These difficulties will be observed with an increasing number of patients where the number of possible solutions grows exponentially with the size of the problem. The tasks subject to the labour of the social workers are *combinatorial optimisation problems* and require an organisation based on time and its execution. Moreover, because of the importance of these problems, we usually find them in our daily life. These tasks may be subject to a finite set of resources and restrictions due, for instance, to the physical characteristics of the environment, the temporary relationships or the labour regulations. In addition to conditions, the goal is to optimise one or more criteria represented by an objective function and are usually related to cost, benefit, or execution time.

The nature of the SWP invites us to analyse the main generalisations of the TSP in search of a suitable formulation for our problem. To do this, we will analyze some generalizations of TSP such as the *Vehicle Routing Problem* (VRP) [72] [145] [229] and the *Job Shop Scheduling Problem* (JSSP) [55].

The generic VRP, seen as the generalisation of the TSP, purposes of finding a set of routes at a minimal cost. Usually, the path is defined from the beginning to the endpoint by passing through the depot to achieve the demand of all nodes.

VRP is an NP-hard combinatorial optimisation problem that can be exactly solved only for small cases of traditional computation problems. An approach that allows better results in practice is the heuristic approach. However, it does not guarantee optimisation. Thus, in the latest decades, meta-heuristics have emerged as the most hopeful research focus for the VRP family of problems.

Even with the best restrictions, the VRP (and without the scheduling adaptation) continues to have exponential complexity depending on the growth in the number of input data. This concept leads us to explore new approaches to solve problems, such as large-scale social workers. *Quantum computing* [82] could be one of these approaches.

Since Richard Feynman suggested that computing could be done more efficiently by exploiting the power of *quantum parallelism*, [82][202], Today, there is evidence that makes quantum computing one of the most prominent candidates to replace conventional silicon-based systems partially. During these last three decades, one finds a series of quantum algorithms more efficient than the better-known classical counterparts. The most renowned algorithms are the *Shor* [216] and *Grover* [21][52] algorithms. However, there is deep literature on quantum computing and techniques [179].

Quantum computers use *qubits*. Like a bit, a qubit can be in state 0 or state 1. Furthermore, like a classical computer, the first program for a quantum computer is just a string of zeros and ones. However, while a quantum computer runs, qubits can also be infinitely many superpositions between 0 and 1.

We can build complex designs up to quantum computers by combining one-qubit systems.

There are two dominant techniques for *quantum computing*. *Continuous-Time Quantum Computing* [136] used by D-Wave in which the problem to solve is mapped in quantum hamiltonians and the natural dynamics of physical systems. The *Gate based Quantum Computing* [163, 225] led by IBM, in which the computation is made through a series of discrete gate operations. The Ref.[136] argues how *Quantum Walk* (QW), *Quantum Annealing* (QA), and *Adiabatic Quantum Computing* (AQC) are related. QW and AQC are pure quantum evolutions (unitary), while QA involves external cooling.

The *Adiabatic Quantum Computing* proposed by Farhi [144, 93], is based on the *adiabatic theorem* [163]. This theorem affirms that if a quantum system is driven by a progressively evolving Hamiltonian, which grows from one starting point of the Hamiltonian ( $H_{init}$ ) to the final point  $H_{final}$ . Thus, if the systems start in the ground state of  $H_{init}$ , the system will finish up in the ground state of  $H_{final}$ . Thus, adiabatic computing was the first quantum computing technique [163].

Quantum Annealing, based on the adiabatic quantum computing paradigm, was initially introduced by Kadowaki and Nishimori [131]. Since its proposal, the QA technique has been a light for solving combinatorial optimisation problems. This technique tries to solve problems similar to how optimisation problems are solved using the classical simulated annealing [163]. From a multivariate

function formed from an energy landscape, the ground state corresponds to the optimal solution of the problem. The QA process must be repeated until finding the optimal solution to the problem. The most significant advantage of quantum annealing is its high parallelism over classical code execution. Because it analyses all possible inputs in parallel to find the optimal solution, this is very useful when we want to reduce the complexity of the NP-complete problems.

QA has confirmed its ability to solve a broad range of combinatorial optimisation problems and also in other fields, such as quantum chemistry (One of the fields that are taking great advantage of capacity and the era in which quantum computing is right now) [163], bioinformatics[163] and routing [145], to mention a few.

We can categorise combinatorial optimisation problems into several groups. Where the need for adequate techniques for solving such problems. One of the standardised optimisation problems is Quadratic Binary Optimisation Problems (QUBO)[163, 131, 46].

QUBO, as NP-hard, refers to a pattern matching technique that, among other applications, can be used in machine learning and optimisation, which involves minimising a quadratic polynomial on binary variables[163]. QUBO has demonstrated its potential in solving some standard combinatorial optimisation problems such as the colouring of graphics, workshop planning, vehicle routing and programming, neural networks, the partition problem, 3-SAT, and machine learning where the parameters of the problem can be expressed as Boolean variables [163, 131, 46]. Only to remember that Adiabatic quantum annealing techniques are also used to solve multi-objective optimisation problems[171]. The QUBO formulation is suitable for running a D-Wave architecture; nevertheless, QUBO can be mapped on the Ising model[163].

Advances in quantum computing offer a way forward for efficient solutions to many cases of substantial eigenvalue problems unsolvable in a traditional way [183]. Quantum approaches to finding eigenvalues previously relied on the Quantum Phase Estimation (QPE) algorithm. The QPE is one of the essential subroutines in quantum computation. It serves as a central building block for many quantum algorithms and offers exponential acceleration compared to classical methods, and requires several quantum operations  $O\left(\frac{1}{p}\right)$  to obtain an estimate with precision  $p$  [183, 112].

Variational Quantum Eigensolver (VQE) proposed by Peruzzo[183] based on the variational principle and form, estimates the ground state energy of the Hamiltonian of the problem [84]. The VQE is a hybrid quantum/classical algorithm originally proposed to approximate the ground state of a quantum system (the state attaining the minimum energy).

Quantum Approximate Optimization Algorithms (QAOA), based on the principles of adiabatic quantum computation [163, 112, 6], is used to solve QUBO problems. Farhi and Harrow showed the advantages of QAOA compared to classical approaches [144, 93]. While Reberntrost [200] just debated the problems of constrained polynomial optimisation using adiabatic quantum computation methods. Other scientists such as Vyskocil and Djidjev [237] worked on how to apply restrictions in QUBO systems to avoid the use of large numbers of the coefficients so, thus more qubits, resulting from the use of quadratic penalties, they proposed a new combinatorial design which involved solving problems of linear programming of mixed integers to adapt applications restitution. The following work [159] investigated and solved the Hamiltonian cycle problem in computational frameworks such as quantum circuits, quantum walks, and adiabatic quantum computing.

These advances in quantum computing have been applied to routing and scheduling techniques. For example, the Ref. [94] contributed an expansive vision and discussions on Ising formulations for various NP-complete and NP-hard optimisation problems, emphasising using as few as possible qubits. In the same way, there have been many works of literature on the VRP and its variants.

We will take advantage of all these related works to define an appropriate strategy for our thesis project in this NISQ era, going from the formulation of the SWP, its implementation, experimentation, and the various comparisons and the quantum machine learning approach to solve the SWP to its generalisation with the qRobot.

## 1.1 Motivation

Quantum computing is born with enormous potential at the computational level, being able to break the internet encryption keys based on the RSA technique, improve the best classical algorithms reducing their computational cost, and be used for the search for new drugs reducing the time to market of these drugs or, simply, to use as a logistics management tool for the optimal management of the medical care of patients who are at home. In addition, my motivation is also based on analysing quantum algorithms that promote my group's research to contribute and provide tools to a set of African scientific groups and engineers to reduce the gap that exists today between the third and the first world. Suppose I, or the Africans in my situation, cannot make this feasible. In that case, the difference between the so-called first world and the third world may be even greater since the first world is already using this technology strategically and commercially. Within the third world countries, there are several sub-Saharan African countries, such as Benin, where I was born; Cameroon, where I spent my adolescence; and Ivory Coast, where I lay the first stone to help Africa, through education and technology using the network of La Salle schools.

In short, my motivation is to research, study, design and implement an NP-complete / NP-Hard quantum algorithm. Furthermore, I contribute to the QC combinatorial optimisation algorithms for the DS4DS research group and Africa.

## 1.2 Work context

To develop this PhD thesis consisting of a theoretical and a practical part, I have had to take some courses in the foundations of quantum physics. I have had to develop and deepen optimisation mathematics, visit a quantum laboratory, and work in the Boston ecosystem. I also had the need, and I am collaborating in varied workgroups such as IBMQ, Xanadu, google, Quantum World, AWS Braket, local groups such as Hispanic Computation, especially with the PhD Alberto Enciso from advanced mathematics (CSIC). I have had to take some courses on quantum computing, work on the Qiskit textbook, Pennylane tutorial, share various results and research topics with the team led by Steve Wood and Jan Rainer Lahmann, the CTO of Lufthansa and IBMQ, work with several different frameworks of payment, free, online and locally (Qiskit, Pennylane, Cirq, Qibo, AWS\_Braket, Raspberry Pi, etc.). I have participated in various conferences, forums, hackathons and classes. I have advised on the writing of several quantum computing curricula. To finish, I have met, collaborated with several renowned scientists of the different computer techniques in marking, and led collaborative projects and final degrees at Valladolid and La Salle - Ramon Llull university on algorithms and quantum approaches I designed within the framework of this thesis work.

## 1.3 Goals

This thesis aims to study art and design a series of quantum algorithms on combinatorial optimisation problems. To do this, we will define and create a combinatorial optimisation problem with restrictions and solve it in two different approaches. Conversely, we will formulate the problem mathematically in a top-down approach, both classically and quantum. On the other hand, we will pose the problem statistically where we will base its resolution on the machine learning technique of the base case reasoning algorithm. Later, we will generalise our formulation of the SWP to carry out efficient management of robots by substituting social workers for robots and patients for pick-up orders. We call this generalisation qRobot. Finally, we set another objective to design a collaborative didactic framework in quantum university students with my DS4DS group.

In the path to attain its primary objective, this PhD research will also have in mind the following goals:

- ◇ Updated state of the art concerning Combinatorial Problem, Linear and Quadratic Constraints Solvers, Quantum Mechanic, Quantum Computers, Quantum Gates, Quantum Circuits, Quantum Computing, Quantum Machine Learning.
- ◇ State of the art of TSP, JSP and VRP.
- ◇ State of the art of heuristics for solving the combinatorial problems.
- ◇ State of the art of the concept of Hamiltonian of a system, Ising Hamiltonian Model, The Hamiltonian of a TSP, the Hamiltonian of a JSP, the Hamiltonian of a VRP.
- ◇ State of the art of the complexity class and Quantum Complexity class.
- ◇ Design and Model, mathematically the Social Workers' Problem (SWP) as CSP according to state of the art and the ones proposed in the scope of the PhD work
- ◇ Study of the players involved in SWP, namely quantum computing frameworks, quantum computers, and quantum research groups.
- ◇ Study and map the SWP in quantum computing.
- ◇ Developing and implementing the SWP in this NISQ era.
- ◇ Solve the SWP with QML. This methodology will be based on clustering and classification techniques, resulting in a rule base concerning the SWP. An extensive set of simulation results will be used as the basis for applying the proposed QML method.
- ◇ Designing and implementing a quantum Case-Based Reasoning (qCBR).
- ◇ Designing and implementing a Quantum Enhanced Filter: QFilter.
- ◇ Designing and implementing qRobot: A Quantum computing approach in mobile robot order picking and batching problem solver optimisation.
- ◇ Designing and implementing EVA: a quantum Exponential Value Approximation algorithm.
- ◇ Testing and validating the developed models and approaches.
- ◇ Pass on the knowledge acquired to the group of DS4DS collaborators.

## 1.4 Hypothesis

In this era of quantum computing with few numbers of qubits, we know that quantum optimisation algorithms based on the variational principle offer a good approximation to the optimal solution, taking advantage of the fundamentals of quantum mechanics such as superposition or quantum parallelism plus interference, reducing in some cases the computing's cost exponentially compared to classical computing. However, due to this era's technical problems, NISQ (few useful qubits, decoherence and noise), to what extent: Could we use quantum computing to solve an optimisation problem with hard restrictions? Also, is it possible to solve quantum combinatorial optimisation problems in Top and Down's philosophy and case-based reasoning?



## 1.5 Development of the thesis

This section will briefly present the technical results obtained in the thesis. After reviewing state of the art, techniques to solve combinatorial optimization problems, quantum fundamentals, current quantum technologies, and the leading companies in these fields, we have focused on defining combinatorial optimization problems to respond to our thesis hypotheses. First, we mathematically design and formulate the *Social Workers Problem* (SWP)[29, 31, 30, 178, 25, 9, 67, 10, 7, 8] and define a heuristic to allow this problem, which includes inequality and time constraints, to be implemented in current quantum computers with very few qubits. To implement this problem, we have had to analyze and study the *Ansatzes* and various variational algorithms such as VQE and QAOA.

Once the results were published at the *16th International Conference on Hybrid Artificial Intelligence Systems (HAIS'21)*, we began to solve the same problem (SWP) with the Machine Learning approach, and we called this solution *quantum Case-Based Reasoning* (qCBR) [33] inspired in classic *Case-Based Reasoning*. The qCBR was plated as the sum of two large blocks (a variational classifier and a synthesizer). The qCBR uses a data representation model from a multidimensional vector in Hilbert space. That is, it creates a vector subspace, where each vector has the information that defines the SWP on the Hilbert vector space and, when predicting whether an input vector (new case) corresponds to a previously analyzed class, the qCBR calculates the probability that each type corresponds to the new vector from the proximity of each of the vector subspaces generated from each category. The qCBR, with its synthesizer block, refines the retrieved data in the case of not being the optimal result since the qCBR has the function of "generating" a new outcome based on the retrieved information.

To scale the algorithms developed and to be able to compare them in various technologies and quantum environments on the market (mainly *Quantum Annealing* and *Quantum Gate Based computing*), a problem was defined along the lines of SWP, replacing social workers with mobile robots and, instead of patients, batches. We call the result *qRobot: A Quantum computing approach in mobile robot order picking and batching problem solver optimization* [32]. We developed a new formulation, turned a Raspberry Pi 4 into a quantum "computer," and implemented the solution. Finally, we studied the comparisons in AWS-Braket (D-wave), Pennylane and Qiskit. The results were promising; the article was published and on the cover of Algorithms magazine.

In parallel, we have been working to improve the VQE algorithm, the flagship of quantum computing, and we publish *EVA: a quantum Exponential Value Approximation algorithm* [15]. We proposed a different way to VQE to calculate the expected value given a quantum state. It should be remembered that the largest cloud services platforms for quantum computers charge by the number of shots and number of circuits. The VQE calculates the expected value by decomposing the Hamiltonian into Pauli operators and obtains this value for each of them, making simulations on cloud servers more expensive. After completing our work, we designed an algorithm capable of calculating this value using a single circuit. Finally, we carried out a cost-time study and verified that it was possible to obtain a good performance in certain more complex Hamiltonians compared to current methods. Also, to consolidate our vision of hybrid computing, we contributed a publication on CNNs with the following article: *Quantum Enhanced Filter, QFilter* [28]. We proposed a hybrid image classification model that took advantage of the potential of convolutional networks in Classical computing and replaced the classical filters with variational quantum filters to reduce the computational cost of classical computing. And similarly, this work would be compared with other classification methods and system execution on different servers. It should be remembered that convolutional filters are based on the scalar product and represent a high cost for classical computation. In contrast, this operation (scalar product) is native to quantum computation (Hilbert vector space).

The feasibility of all the algorithms that we have proposed has been made in various environments such as Qiskit, AWS-Braket, D-Wave, Pennylane and Qibo.

## 1.6 Thesis structure

This thesis is organised as follows. In section (2), we review the Optimisation combinatorial problems related to our topic. Chapter (3), we introduce the Systems and useful concepts needed for the methods to solve the optimisation problems. In section (4), we present the Methods to solve combinatorial optimisation problems. The Quantum Mechanic is introduced in chapter (5), reviewing the Quantum mechanics postulates. We introduce the Complexity Class in section (6). After analysing the quantum complexity in chapter (7), we revise the useful Quantum Gates. Quantum Computers and the players are the objects of chapter (8). Chapter (9), we go deeply into the most important technics in Quantum Computing. In the next section (10), we develop the Research Design and analyse our experiment tools, the quantum frameworks, and some experiments over quantum algorithms and technics. Section (11), we present our approaches to solving the Social Workers' Problem. Section (12) we present a quantum machine learning approach to solve the SWP. In section (13) We generalize the SWP as qRobot. Section (14) and (14.2), we show the results and emphases on a discussion over these results. Our last chapter is over the conclusion and furthers works.



## Chapter 2

# Combinatorial Optimization problems

### 2.1 Introduction

The posed problem is combinatorial. For this reason, it is of utmost importance to study the optimization of discrete and combinatorial objects known as Combinatorial Optimization. Which consists of the structure of a complete exploration algorithm based on the exploration of nodes; the node, the evaluation functions, the concept of Heuristics, the heuristics with BackTracking [128], the Meta-heuristic and heuristic procedure.

Combinatorial optimization means searching for an optimal solution in a finite or infinite set of potential solutions. Optimality is defined concerning some criterion function, which must be minimized or maximized is usually called the Cost Function or Objective Function.

The *Social Workers' Problem* can be seen as the sum of a scheduling and routing problem. Because of this, we will explore various scheduling and routing algorithms. The first study is the Traveling Salesman Problem (TSP) algorithm but with the mind of adding some time variables and redefining the variables to be optimized.

### 2.2 Travelling Salesman Problem

The *Travelling Salesman Problem* known as TSP is defined as follows: *Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?*

TSP is an NP-Hard problem within combinatorial optimization. The problem was first formulated in 1930 and is one of the most studied optimization problems. Although the situation is computationally complex, many exact heuristics and methods are known so that some instances from one hundred to thousands of cities can be solved.

In the TSP formulation (2.1) to (2.4), there are  $N!$  Possible routes (for the exact calculation). However, it can be simplified since, given a path, the starting point does not matter, and this reduces

the number of routes to be examined by a factor  $N$ , leaving  $(N - 1)!$ . Since the direction in which the traveller is travelling does not matter, the number of routes to be examined is again reduced by a factor of 2. Therefore, it is necessary to consider  $\frac{(N-1)!}{2}$  possible routes.

The number of routes is multiplied by the  $N$  factor for each new city, and its growth is factorial. So, the TSP belongs to the NP-complete complexity class.

$$\min \sum_{i=0}^n \sum_{i \neq j, j=0}^n c_{ij} x_{ij}, \quad (2.1)$$

$$0 \leq x_{ij} \leq 1. \quad (2.2)$$

Where  $x_{ij}$  is an integer and could take the following value  $i, j = 0, 1, \dots, n$

$$\sum_{i=0, i \neq j}^n x_{ij} = 1, \quad (2.3)$$

$$\sum_{j=0, i \neq j}^n x_{ij}. \quad (2.4)$$

### 2.2.1 Constructive heuristics

Following our analysis, we find that the constructive heuristics are based on the nearest neighbour algorithm (NN)[65] or also called Greedy algorithm[65], which allows the traveller to choose the nearest unvisited city as the next move. This algorithm quickly returns a lower-cost route. For  $N$  cities randomly distributed on a plane, the algorithm on average returns a path 25% longer than the smallest path possible. However, in many cases, they exist where the distribution of the given cities makes the  $NN$  algorithm return the worst path [114]. This occurs for both symmetric and asymmetric TSPs[115]. Rosenkrantz showed that the  $NN$  algorithm has an approximation factor of order  $O(\log|V|)$  for instances that satisfy the triangular inequality. A variation of the  $NN$  algorithm, called *Nearest Fragment operator*, is the one that connects a group of closest unvisited cities and can find the shortest route with successive iterations. The  $NF$  operator can also be applied to obtain initial solutions for the  $NN$  algorithm and, besides, be improved in an elitist model where only the best solutions are accepted.

Another solution approach is the Job Shop Scheduling Problem (JSSP) that we will analyze next.

## 2.3 Job Shop Scheduling Problems

The *Job Shop Scheduling problem*, known as JSSP, consists of planning a set of  $N$  jobs  $\{J_1, \dots, J_N\}$  on a set of  $M$  resources or physical machines  $\{R_1, \dots, R_M\}$ . Each  $J_i$  job consists of a set of visits or operations  $\{V_{i1}, \dots, V_{iM}\}$  that have to be executed sequentially. Each  $V_{il}$  Visit has a runtime of  $pV_{il}$  of time during which it requires the exclusive use of a single resource,  $V_{il}$ , starting from start time  $stV_{il}$  to be determined.

Each job has an earlier start time and is sometimes considered a later end time, forcing visit start times to take values in finite domains.

Regarding the analogy, in our case, the  $N$  jobs  $A_1, \dots, A_N$  will be the assistants and the physical machines  $P_1, \dots, P_M$  will be patients. So from here, the same description and variables are respected.

The *JSSP* has three restrictions: sequential or precedence restrictions, capacity restrictions and non-expulsion restrictions of the machines, in our case, the patients. Sequential constraints are defined

by the sequence of visits to a job and can be described by linear inequalities of the type:  $stV_{il} + pV_{il} \leq stV_{i(l+1)}$  that is, the visit  $V_{il}$  must be carried out before the visit  $V_{i(l+1)}$ . Capacity constraints restrict the use of each resource to a single visit at each instant of time and can be described as a disjunctive constraint of the form:  $stv + pv \leq stw \vee stw + pw \leq stv$  if  $Pv = Pw$ . Mode to restrict that two visits that use the same resource cannot overlap. Lastly, the non-expulsion restrictions express that once the start time has been determined,  $stV_{il}$  for a visit  $V_{il}$ , this visit must be carried out during the time interval  $[stV_{il}, stV_{il} + pV_{il}[$  without interruption.

The objective is to find optimal planning for a specific criterion, the most common being the following three:

- ◇ Makespan is the end time of the last visit, and time is denoted as  $C_{\max}$ . This version of the problem is known, in the literature, as  $J|| C_{\max}$ .
- ◇ Total Flow: it is the sum of the end times of all the works. This version is known as  $J|| \sum C_i$ , with  $C_i$  is the time of completion of the work  $J_i$ .
- ◇ Tardiness: or delay time of the works. This version is known as  $J|| \sum T_i$  with  $T_i$  as the tardiness,  $T_i = \max\{0, C_i - d_i\}$  with  $d_i$  is the due date of the visit  $i$ .

We can consider a *JSSP* as a *TSP* executed  $M$  times or see it another way as the *TSP* is a particularization of the *JSSP* taking into account  $N$  jobs on a single machine. Being the travelling salesman, the machine and the works the cities.

Another approach to solving the problem is to use a generalization of the *TSP*. In this case, the *VRP*.

## 2.4 Vehicle Routing Problem

The *Vehicle Routing Problem* known as *VRP* is defined as follows: *what is an optimal set of routes for a fleet of vehicles to traverse to deliver to a given group of customers?*

In *VRP*, the goal is to find optimal routes for multiple vehicles visiting a set of locations. (When there is only one vehicle, it is reduced to *TSP*).

What do we understand by optimal routes for a *VRP*? The answer is the same as for a *TSP*; routes with the shortest total distance. However, suppose there are no other restrictions for the *Vehicle Routing Problem*. In that case, the optimal solution is to assign a single vehicle to visit all locations and find the least cost route for that vehicle. This is merely solving the *TSP*.

Since the study focuses on an optimization problem, a good definition for optimal routes would be to minimize the most extended single path among all vehicles. This is the correct and best-known definition if the goal is to complete all deliveries as soon as possible.

It is essential to see that the basis of this formulation is *TSP*. So, we can find a general formulation on the *TSP* that includes restrictions such as:

- ◇ Capacity restrictions
- ◇ Visit time.

It is also observed that the *Vehicle Routing Problem* can serve to encode the difficulty of finding the optimal schedules of the  $n$  social assistants (vehicles) who visit the  $m$  patients (locations).

## 2.5 Summary

In this section, we have introduced some classical combinatorial algorithms on which we will base the design of our *SWP*. We have reviewed state of the art, both the definition and the formulations of

the *TSP* with its techniques of constructive heuristics. We also analysed the *JSSP*, which despite being an algorithm designed for factories, can be used to model combinatorial planning problems and finally, the *VRP*.

To summarise, we observed that even with the best restrictions, heuristics or programming techniques, depending on the size of the data considerably high, the best *TSP*, *JSSP* or *VRP* requires a very high computational cost for classical computing and would continue to approach an exponential cost. When the number of the input increases, any polynomial algorithm is more efficient than any exponential [121]. Another positive feature of polynomial algorithms is that, in a sense, they take better advantage of technological advances.

In these cases, another approach would be needed, such as using the power of quantum computing to solve the problem of this magnitude.

Now we have to relate these classical algorithms with the usual concepts that can be of great help in the development of this thesis.

# Chapter 3

## Systems and useful concepts

### 3.1 Introduction

This chapter will work on concepts and systems we need to solve combinatorial optimization systems. We will also analyze some necessary concepts that relate the resolution of the Ising Hamiltonian with graph theory and the basic Hamiltonian path of solving a routing problem such as SWP. But, first, remember that the Hamiltonian trajectory problem is equivalent to that of a TSP since the task goes through all the nodes in a graph exactly once.

A Hamiltonian system is a dynamic system, that is, a system whose state evolves with time, governed by Hamilton's equations. These systems are essential in dynamic optimization techniques [97].

Next, we will analyze the Hamiltonian path, Hamiltonian cycles, the Hamiltonians of TSP, VRP and the Hamiltonian Ising.

### 3.2 Hamiltonian path and cycles

A *Hamiltonian path*, in the mathematical field of graph theory, is a path of a graph (Figure (3.1)), that is, a succession of adjacent edges that visits all the vertices of the graph at one time. When the last vertex visited is adjacent to the first, the path is a Hamiltonian cycle.

A *Hamiltonian cycle* is defined as a *closed-loop on a graph where every node (vertex) is visited exactly once*.

A *loop* is just an edge that joins a node to itself; a *Hamiltonian cycle* is a path travelling from a point back to itself, visiting every node en route.

Let  $G = (V, E)$ , and  $N = |V|$  be the graph and the Rank respectively. Having the following vertices  $\{1, \dots, N\}$ , and the set of vertices  $uv$  to be directed (the order  $uv$  matters), it is trivial to extend to non-directed graphs, just considering a directed graph with  $vu$  added to the set of arcs each time  $uv$  is added to the arc set.

Before describing how to particularize the Hamiltonian cycles for a TSP, VRP and the research design, we will explain the Ising model, a Hamiltonian system, and their relationship with the computer optimization problems.



### 3.3 Ising model and the Hamiltonian system

The *Ising model* is defined as a mathematical model of ferromagnetism in statistical mechanics. The model consists of discrete variables representing magnetic dipole moments of atomic "spins" in one of two states  $+1$  and  $-1$ .

Unfortunately, solving the spin system is costly because the energy landscape is equivalent to finding the global minimum of a function with several local minima. Thus, Ising's model describes the system's energy, and the operator or equation that describes the system is known as the Hamiltonian.

Finding the lowest energy of the spin system is equivalent to finding its best configuration (the global minimum of the associated function).

This problem is somewhat equivalent to a QUBO problem that is NP-Hard. The only difference is that the QUBO works with binary variables and the Ising model with variables between  $-1$  and  $1$ . Thus, we can summarize that the correspondence between a computer problem (QUBO) and a physical system is variable. But, first, let us detail the Hamiltonian system from the spin glasses.

A *Hamiltonian system* is defined as a mathematical formalism developed to describe the evolution equations of a physical system.

Spin glasses are systems of  $N$  particles in which each particle can take two values:  $-1$  and  $+1$ . The set of all possible values of the  $N$  particles is called the configuration space and is designated as  $\sum_N \{+1, -1\}^N$ . A configuration  $\sigma = \sigma_1, \sigma_2, \dots, \sigma_n$  is an element of the summation in  $N$ , and each of the components  $\sigma_i$  of this configuration is a spin. Each configuration has an associated energy  $H_N(\sigma)$  from a function which is called the *Hamiltonian system*. The type of energy function that is chosen determines the model.

$$H_N = \sum_N \rightarrow R. \tag{3.1}$$

The work of many researchers, both mathematical, physical and electronic, has facilitated the conclusion that spin glasses can be used to model quantum states. The contribution of *Manuel de la Rosa Fernández*[204] reaches the same conclusion. Spin glasses are the basis of Hamiltonian systems that go from neural networks to artificial intelligence problems and go through numerous issues of a physical nature, such as; studies on states of matter and statistical mechanics.

Mathematically, the fact that a problem is NP-complete means that a mapping to the decision form of the Ising model can be found with a polynomial number of steps. This assignment can be reinterpreted as a pseudo-boolean optimization problem [53].

There are constructions known as "p-spin lenses", which often lead to interactions of three bodies or more in  $H$ , and which can later be used to reduce the problem to an Ising spin glass, introducing

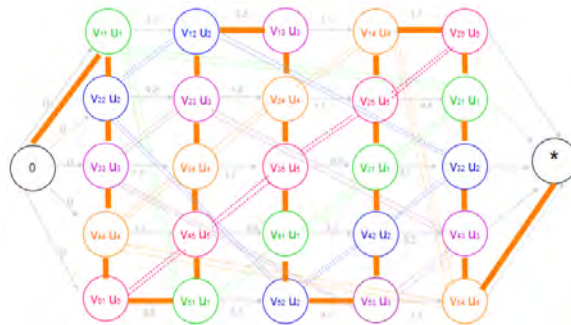


Figure 3.1: Shows a graph of 26 vertices in which a Hamiltonian cycle is highlighted

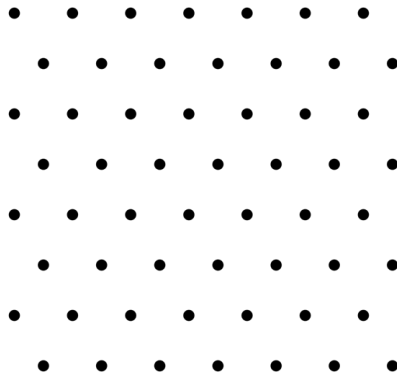


Figure 3.2: A Lattice in the Euclidean plane [240]

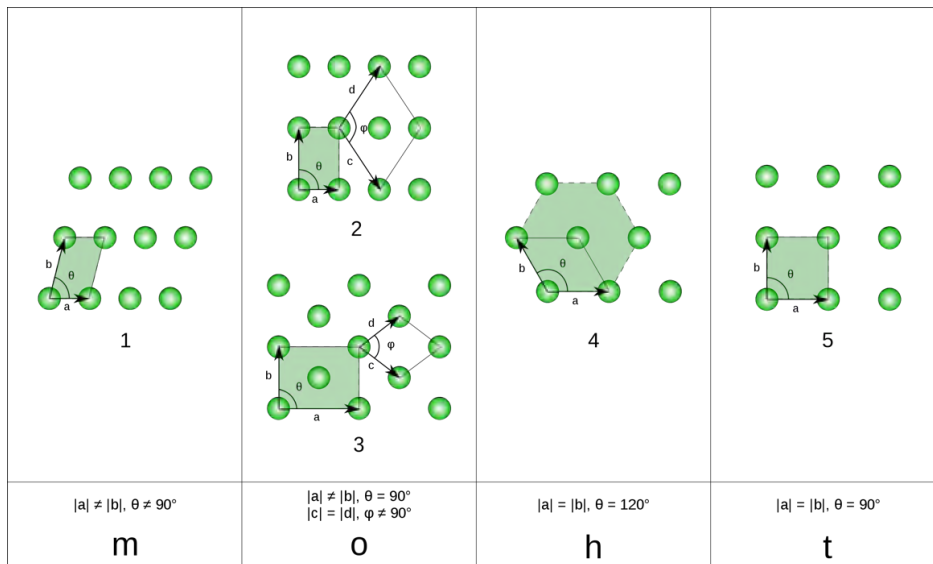


Figure 3.3: Five Lattices in the Euclidean plane [240]

a polynomial number of turns that help reinforce the interaction of three bodies through multiple interactions of two bodies  $s_1 s_2$  [36] [49].

Based on Andrew Lucas's work[155], the author concludes that almost all famous  $NP$ [103] [134] problems can be written as Ising models with a polynomial spin number that does not scale faster than  $N^3$ . This work is essential for writing any quantum algorithm.

### 3.4 The Hamiltonian of a TSP

Let  $G = (V, E)$  be a graph of the TSP, where each arc  $uv$  in the graph has an associated weight  $W_{uv}$  is to find the Hamiltonian cycle so that the sum of the weight of each arc in the cycle is minimized. Usually, the TSP is solved in a complete graph, but with the developed Hamiltonian mapping of Ising, it can be solved in an arbitrary graph.

Let us remember that the decision problem: Is there a path of total weight  $\leq W$ ?

Having the following vertices  $\{1, \dots, N\}$ , and the set of vertices  $uv$  to be directed (the order  $uv$  matters), it is trivial to extend to non-directed graphs, just considering a directed graph with  $vu$  added to the set of arcs each time  $uv$  is added to the arc set.

To solve this problem, we use  $H = H_A + H_B$ , with  $H_A$  the given Hamiltonian for the directed (or undirected) Hamilton cycles problem from the equation (3.2). So, we only need to add (3.3).

In the mathematical solution proposed by Lucas Andrew, he uses  $N^2$  bits  $x_{v,i}$  where  $v$  represents the vertex, and  $i$  represents its order in a prospective cycle. The first two concepts that are required are that each vertex can only appear once in a loop and that there must be a  $j^{\text{th}}$  node in the loop for each  $j$ . Finally, for the nodes in our potential order, if  $x_{u,j}$  and  $x_{u,j+1}$  are both 1, then there should be an energy penalty if  $uv \notin E$ .

These are encoded in the Hamiltonian:

$$H_A = A \sum_{v=1}^n (1 - \sum_{j=1}^N x_{v,j})^2 + A \sum_{j=1}^n (1 - \sum_{v=1}^N x_{v,j})^2 + A \sum_{(u,v) \notin E} \sum_{j=1}^N x_{u,j} x_{v,j+1}. \quad (3.2)$$

$A > 0$  is a Lagrange multiplier and is a constant. The ground state of this system has  $H_A = 0$  only if you have an order of vertices in which each vertex is only visited once, and the adjacent vertices in the cycle have edges in the graph that is one cycle Hamiltonian.

To solve the Hamiltonian path problem, the last sum over  $j$  above one is restricted to  $N - 1$ .  $N^2$  spins are required to solve this problem.

It is easy to slightly reduce the size of the state space for the problem of Hamilton cycles as follows: node one must always be included in a Hamilton cycle and without loss of generality  $x_{1,i} = \delta_{1,i}$ . This only means that the general order of the cycle is chosen so that node 1 is the first. This reduces the number of variables to  $(N - 1)^2$ .

$$H_B = B \sum_{uv \in E} W_{uv} \sum_{j=1}^N x_{u,j} x_{v,j+1}. \quad (3.3)$$

The constant  $B$  must be very small and meet these constraints  $0 < B \max(W_{uv}) < A$ .  $W_{uv} \geq 0$  for each  $(uv) \in E$ .

As previously indicated, if the travelling salesman does not have to return to his initial position, the sum on  $j$  can be restricted from 1 to  $N - 1$ . In the same way, as for Hamilton cycles,  $(N - 1)^2$  spins are required since node one can be adjusted to appear first in the process. From here, the steps defined above can be followed to map the formulation of the Hamiltonian Ising in a quantum computer. What must be taken into account is to ensure that the Hamiltonian of Ising is correctly encoded in the  $Z$  base. For this, a problem of eigenvalue and vector must be solved for a symmetric matrix of dimension  $2^N \times 2^N$ .

### 3.5 The Hamiltonian of a VRP

From a *Vehicle Routing Problem*, a binary polynomial optimization can be constructed with equality constraints considering cases in which  $K = n - 1$ . In these cases, the elimination constraints of the secondary path are not necessary, and the problem is only in the variable. We can write the

Hamiltonian of the vehicle routing problem defined by the sum of (3.4) and (3.5):

$$\begin{aligned}
 H_A = & A \sum_{i=1}^n \left( 1 - \sum_{j \in \delta(i)^+} x_{i,j} \right)^2 + A \sum_{i=1}^n \left( 1 - \sum_{j \in \delta(i)^-} x_{ji} \right)^2 + A \left( k - \sum_{i \in \delta(0)^+} x_{0,i} \right)^2 \\
 & + A \left( k - \sum_{j \in \delta(0)^+} x_{j,0} \right)^2.
 \end{aligned} \tag{3.4}$$

$$H_B = B \sum_{ij \in E} W_{ij} x_{i,j}. \tag{3.5}$$

### 3.6 Summary

In this section, we have reviewed the state of the art of useful concepts to develop this thesis. We have introduced the Hamiltonian path and cycles based on graphs, the Ising model, the Hamiltonian system, the Hamiltonian of a TSP and the VRP one. We have also revised their classical formulation.

All these concepts, theories and analyses are important to solve combinatorial optimization systems, and they will help us fully when it comes to modelling our problem. In our case, we will further develop each constraint subject to the objective function.



## Chapter 4

# Solving combinatorial optimization problems

### 4.1 Introduction

This chapter will review existing methods for solving constrained optimisation problems. For this, we need to analyse the existing programming techniques and improve them in the face of our mission.

### 4.2 Linear Programming

Almost all the approximation algorithms that we know are built based on linear programming (LP).

*Linear Programming* is defined as *the problem of optimising (minimising or maximising) a linear function subject to linear inequality constraints. The function being optimised is called the objective function.*

Linear Programming corresponds to an algorithm through which real situations are solved. It aims to identify and solve difficulties to increase productivity concerning resources (mainly limited and expensive), thus increasing benefits.

How to solve a problem using LP?

The first step in solving a linear programming problem is to identify the essential elements of a mathematical model; these are:

- ◇ Objective Function
- ◇ Variables
- ◇ restrictions

The next step is to determine the following goals, for which we propose to follow the following methodology.

1. Define the objective function criteria
2. Identify and define variables
3. Identify and define constraints
4. Present the Objective Function

### 4.2.1 The Objective function, decision variables and the constraints

The *objective function* is directly related to the general question that one wishes to answer. If different questions arise in a model, the objective function would be associated with the higher-level question, the fundamental question.

The *decision variables* are identified, starting from a series of questions derived from the fundamental question. Decision variables are the controllable factors of the system that one is modelling. They can take various possible values, of which it is necessary to know their optimal weight, contributing to the achievement of the objective of the general function of the problem.

When we talk about *constraints* in a linear programming problem, we are referring to everything that limits the freedom of the values that decision variables can take. The *constraints* of the problem determine the conditions in which the use of resources is feasible, taking into account all the tasks of the problem.

The best way to find them is by thinking about a hypothetical case that was decided to give infinite value to the decision variables. For example, in the case of a TSP, one restriction will be to visit the city only once.

In the case of the allocation algorithm, the decision variables  $x_{ij}$  are incorporated, which are binary variables that take the value  $x_{ij} = 1$  as long as  $b_i$  is assigned to  $a_j$  and that  $x_{ij} = 0$ . If these options were not given, the following restrictions (4.1) and (4.2) would be established:

$$\sum_{i=0, i \neq j}^n x_{ij} \leq 1, \quad (4.1)$$

$$\sum_{j=0, i \neq j}^n a_{ij} x_{ij} \leq b_i. \quad (4.2)$$

The first restriction, equation (4.1), means that the set of visits made by an agent will not exceed the limit of resources that the agent has available. While the second restriction, equation (4.2), means that visits will be made only by an agent. In the case of an exact TSP, the first set of equalities ensures that each exit city  $0, \dots, n$  reaches exactly one city. The second set of equalities provides that from each city  $1, \dots, n$  exit exactly into a town (both restrictions also imply that there is precisely one exit from town 0). Finally, the last limitation (4.3) forces a single road to cover all the cities, and it is not two or more disjoint roads that include all the towns together.

$$u_i - u_j + nx_{ij} \leq n - 1. \quad (4.3)$$

### 4.2.2 Formulation of linear programming of binary integers

Let us introduce a *linear programming of binary integers* as follows: If  $z_1 z_2 z_3 \dots z_n$  are  $N$  binary variables, which are ordered in a vector  $z$ , what is the most substantial value of  $cx$ , for some vector  $c$ ,

given a constraint?

$$Sx = b. \quad (4.4)$$

From (4.4) with  $S$  an  $m \times N$  matrix and  $b$  a vector with  $m$  components, it is known that the resolution of this equation has an NP-hard complexity, with a corresponding NP-complete decision problem. As discussed above, almost most daily challenges are combinatorial optimisation problems, and many of them can be posed as ILP [210]. In our case, a social worker must maximise visits to a home patient, given the regulatory restrictions of her/his contract.

Let  $H = H_A + H_B$  be the Hamiltonian (mathematical and physical model introduced above) defined by (4.5).

$$H_A = A \sum_{j=1}^m \left[ b_j - \sum_{i=1}^N S_{ji} x_i \right]^2. \quad (4.5)$$

With  $A \in R^+$ . The ground states of  $H_A = 0$  enforce the constraint that  $Sx = b$ . Then you get that

$$H_B = -B \sum_{i=1}^N C_i x_i. \quad (4.6)$$

With  $A \gg B$  and  $B$  is a positive constant.

To find restrictions on the required  $A/B$  ratio, we can proceed similarly as before. In many cases, for simplicity, it is assumed that the constraint of the equation  $Sx = b$  can be satisfied with a selection of  $x$ . For such a choice, the most substantial possible value of  $-\Delta H_B$  is, in principle, constraint limits, where:

$$C = \sum_{i=1}^N \max(C_i, 0). \quad (4.7)$$

The smallest possible value of  $\Delta H_A$  is related to the properties of the matrix  $S$  and would occur if only a single constraint were violated, and that constraint was broken by the least amount possible, given by (4.8)

$$S \equiv \min_{\sigma_i \in \{0,1\}} \left( \max \left[ 1, \frac{1}{2} \sum_i (-1)^{\sigma_i} S_{ji} \right] \right). \quad (4.8)$$

This limit could be improved if more specific properties of  $S$  and  $b$  are known.

It can be concluded that if the coefficients  $C_i$  and  $S_{ji}$  are integers  $O(1)$ , we have that  $C \leq N \max(C_i)$  and  $S \geq 1$ , so it is concluded that  $A/B$ .

### 4.3 Quadratic Programming

The *Quadratic programming* can be stated as *the procedure that minimizes a quadratic function of  $n$  variables subject to  $m$  linear constraints of equality or inequality.*

A quadratic program is the simplest nonlinear problem with inequality constraints. Quadratic programming is important because many issues appear naturally as quadratic (least-squares optimisation, with linear constraints). Besides, it is important because it appears as a sub-problem frequently to



solve problems, not more complicated linear. The techniques proposed to solve quadratic issues have a lot in common with linear programming.

Specifically, each inequality must be satisfied as equality. The problem is then reduced to a search for vertices precisely as it was done in linear programming, where  $c$  is a vector of constant coefficients;  $A$  is a matrix ( $m \times n$ ), and it is generally assumed that  $Q$  is a symmetric matrix. Since the constraints are linear and presumably independent, the qualification of the restrictions is always satisfied, so the Karush-Kuhn Tucker conditions are also sufficient conditions to obtain an extreme that will also be a global minimum if  $Q$  is positively defined. On the other hand, if  $Q$  is not defined as positive, the problem could be unbounded or lead to local minima. Quadratic Programming plays a significant role in linear and nonlinear optimisation theory since it is closely related to LP and is an essential intermediate step to effectively solving general Nonlinear Programming problems. In the case of solving the Ising Hamiltonian, the two instances of solving the quadratic minimisation problems will be seen. There are several, but they will not be listed in this thesis.

### 4.3.1 Quadratic Formulation

Let  $f : R^n \rightarrow R$ ,  $f$  be a quadratic function if:

$$f(x_1, x_2, \dots, x_n) = \sum_{i,j=1}^n q_{ij}x_i x_j = (x_1, x_2, \dots, x_n) \begin{pmatrix} q_{1,1} & \cdots & q_{1,n} \\ \vdots & \ddots & \vdots \\ q_{n,1} & \cdots & q_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}. \quad (4.9)$$

Let us focus on one of the properties of the Quadratic function as follows:

$$f(\lambda x_1, \lambda x_2, \dots, \lambda x_n) = \lambda^2 f(x_1, x_2, \dots, x_n). \quad (4.10)$$

On the one hand, Quadratic Minimization Problems subject to equality constraints require minimizing the objective function  $f(x)$  subject to linear equality constraints  $Ax = b$ .

But when these are subject to linear inequality constraints, they require minimising the objective function  $f(x)$  subject to these constraints  $Ax \leq b$ , and may even contain equality constraints.

$$\min f(x) = \frac{1}{2}x^T Qx + c^T x. \quad (4.11)$$

Subject to:

$$Ax \leq b, \quad (4.12)$$

$$x \geq 0. \quad (4.13)$$

Where  $c \in R^n$  is a row vector,  $x \in R^n$  and  $b \in R^n$  are column vectors. Let consider also that  $Q = q_{ij} \in M_{n \times n}$  and  $A = (a_{ij}) \in M_{n \times n}$  are matrices and the superscript  $T$  denotes the transpose. The elements  $q_{ij}$  of  $Q$  are constants given such that  $q_{ij} = q_{ji}$  (that is,  $Q$  is a symmetric matrix,  $Q^T = Q$ ). Thus, the objective function is expressed in terms of these  $q_{ij}$ , the elements  $c_j$  of  $c$  and the

variables  $x_j$ , as follows:

$$f(x) = \frac{1}{2}x^T Qx + c^T x = \sum_{j=1}^n c_j x_j + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j. \quad (4.14)$$

### 4.3.2 Change of variables to simplify the quadratic formulation

In some cases, to simplify the objective function and therefore simplify the programming of the algorithm, it is required to make some variable changes (see Fig. (4.1)). This contribution is analysed below.

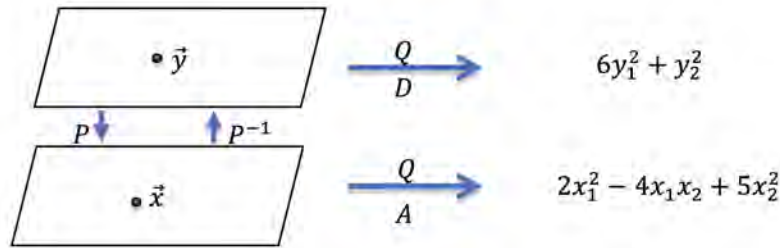


Figure 4.1: Variable change to simplify quadratic programming

Next, the change of variables is studied to simplify quadratic programming. These variable changes and some linear algebra tricks will help with quantum programming.

When a variable change is made for any matrix  $P \in R^{n \times n}$   $\varepsilon$  coordinates to  $B$  coordinates, it is to make the quadratic function easier.

$$\vec{x} = P\vec{y} \leftrightarrow \vec{y}P^{-1}\vec{x} \leftrightarrow [\vec{y}]_B = [\vec{x}]_\varepsilon. \quad (4.15)$$

The idea is:

$$\vec{x}^T A x = (P\vec{y})^T A (P\vec{y}) = \vec{y}^T P^T A P \vec{y} = \vec{y}^T (P^T A P) \vec{y} = \vec{y}^T (D) \vec{y}. \quad (4.16)$$

Given the quadratic function  $Q(x) = 2x_1^2 - 4x_1x_2 + 5x_2^2$  we will perceive what happens with the variable changes. Let us imagine for a moment that the function  $Q(x)$  is the Ising Hamiltonian of the TSP.

Let us find the matrix  $A$ .  $A = \begin{bmatrix} 2 & 5 \\ -2 & 5 \end{bmatrix}$  How is  $A$  found? Following this trick:

$$A = \begin{bmatrix} a & c \\ d & b \end{bmatrix}. \quad (4.17)$$

If we consider that of the function  $Q(x)$ ,  $a$  is the constant that multiplies  $x_1^2$  and  $b$  is the term that multiplies  $x_2^2$ ;  $a = 2$  and  $a = 5$ . So far, relatively easy. How are  $c$  and  $d$  determined?  $c$  and  $d$  are half the coefficient of  $x_1x_2$ . So,  $c = \frac{4}{2} = d$ . From here are the values and eigenvectors to be able to change the variables.

Given the characteristic polynomial  $P(\lambda) = \det(A - \lambda I_n)$  if we want to calculate the eigenvalues, the characteristic polynomial must be zero. This translates to  $\det(A - \lambda I_n) = 0$ , where  $I_n$  is the identity matrix of rank  $n$  and  $\lambda$  the eigenvalue. Mathematically speaking, if we want the characteristic polynomial of matrix  $A$  to be null, the following linear equation must be solved. The characteristic polynomial is of rank  $n$ , and its roots are the eigenvalues of matrix  $A$ . Since in our case matrix  $A$  is of rank 2, it will not be necessary to apply the Laplace expansion formula. Starting from a square matrix of degree  $n$ .

$$\det(A - \lambda I_n) = 0. \quad (4.18)$$

According to Laplace's theorem, the value of its determinant is equal to the sum of the products of the elements of a row or column by their attachments, thus taking any row  $f$ , the determinant is:

$$\det(P) = \sum_{j=1}^n a_{f,j} P_{f,j}. \quad (4.19)$$

And taking a column  $c$ , it will be:

$$\det(P) = \sum_{i=1}^n a_{i,c} P_{i,c}. \quad (4.20)$$

Solving the equation of the characteristic polynomial we get:

$$\det \begin{pmatrix} 2 - \lambda & -2 \\ -2 & 5 - \lambda \end{pmatrix} = 0. \quad (4.21)$$

Since it is a rank two matrix, the calculation of the determinant is direct and is shown through the following quadratic expression:

$$\lambda^2 - 7\lambda + 6 = 0. \quad (4.22)$$

Where the roots of the equation are  $\lambda_1 = 1$  and  $\lambda_2 = 6$ . If the eigenvectors are now calculated to matrix  $A$ , this definition is fulfilled.

$$\exists v \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \neq 0 / \exists \lambda \in C/A \cdot v = \lambda \cdot v. \quad (4.23)$$

Where  $\lambda$  is the eigenvalue and  $v$  is the eigenvector. And, if this definition is applied, it arrives at (4.24)

$$\begin{bmatrix} 2 & 5 \\ -2 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}. \quad (4.24)$$

With this expression  $-2x_1 = x_2$ , we can find the two eigenvectors associated with matrix  $A$ .

$$v_1 \begin{pmatrix} 1 \\ -2 \end{pmatrix} \text{ and } v_2 \begin{pmatrix} 2 \\ 1 \end{pmatrix}. \quad (4.25)$$

Once this point has been reached, if the quadratic form is rewritten, the reader can understand how simple it is and, consequently, its resolution or programming will also be more straightforward.

$$P(y) = \lambda_1 y_1^2 + \lambda_2 y_2^2. \quad (4.26)$$

If we remember that we came from the expression  $Q(x) = 2x_1^2 - 4x_1x_2 + 5x_2^2$ , we can find the matrix  $P$ .

$$P = \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \quad (4.27)$$

If the matrix  $P$  is normalized, the following expression is reached,

$$P = \frac{1}{dist} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \quad (4.28)$$

With  $dist = \sqrt{1^2 + -2^2}$  as the distance.

$$P = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}. \quad (4.29)$$

Retrieving the expression,  $A = PDP^T$  takes us to  $D = P^TAP$ , where  $D$  is the following matrix:

$$D = \begin{bmatrix} 6 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4.30)$$

These variable changes help simplify the objective function by undoing the crossed variables.

## 4.4 Quadratic programming with quadratic constraints

The *quadratically constrained quadratic program* can be defined as *an optimisation problem in which both the objective function and the constraints are quadratic functions*.

When the constraints are of order 1, the quadratic formulation just analyzed works for us. But there are cases when the restrictions must be quadratic; of the type  $x_1(x_1 - 1) \leq 0$  and  $x_1(x_1 - 1) \geq 0$  being equivalent to the constraint  $x_1(x_1 - 1) = 0$ , and, in turn, equal to the restriction  $x_1 \in \{0, 1\}$ . In this case, a generalization of quadratic programming must be used. Almost for all quantum algorithms where we want to improve resolution and computation time, *Quadratically Constrained Quadratic*

*Program* (QCQP) should be used. It is the same studied so far but with quadratic restrictions. For this, the new objective function  $P(x)$  must be minimized

$$\min P(x) = \frac{1}{2}x^T P_0 x + q_0^T x + r_0. \quad (4.31)$$

Subject to:

$$\frac{1}{2}x^T P_i x + q_i^T x + r_i \leq 0 \quad \text{for } i = 1, \dots, m, \quad (4.32)$$

$$Ax = b. \quad (4.33)$$

$P_i \in S_+^n$ , the objective and constraints are quadratic convex, where  $P_0, \dots, P_m$  are matrices  $n \times n$  and  $x \in R^n$  is the optimization variable. If  $P_1, \dots, P_m \in S_{++}^n$ , the feasible region is the intersection of  $m$  ellipsoids and a cognate set.

When trying to model a problem in binary variables, a degree greater than two may appear in the function to minimise (or cost function). As most of the quantum computing technique in this NISQ era is implemented, to handle terms of degree less than or equal to 2, the degree of monomials that do not meet this condition must be reduced.

Let us analyse different ways to transform a cubic monomial of the cost function into a quadratic one. Once we have this technique, applying it repeatedly, we can decrease the degree of any monomial.

A first way is to transform the problem  $\min f(w)$  into the problem  $\min g(k)$  subject to  $t = xy$  con  $f(w) = xyz$  y  $g(k) = tz$ .

Later, we will realise that we cannot have a quadratic constraint, so let us find a way to substitute  $t = xy$  for linear constraints.

Taking the constraints  $t \leq x$ ,  $t \leq y$  and  $t \geq x + y - 1$ , we can transform the problem of minimize  $f(w)$  into the problem:

$$\text{minimize } tz$$

,

$$\text{subject to } \begin{cases} t \leq x \\ t \leq y \\ t \geq x + y - 1. \end{cases}$$

This remodelling involves introducing three inequality constraints. Inequality constraints, while often unavoidable, are somewhat inefficient. This is due to the need to include a set of auxiliary slack variables necessary to transform each inequality into equality. This procedure is similar to the one that occurs when modelling a linear programming problem to be solved using the simplex algorithm. There are more techniques, but we will not dive deeply into them in this thesis. Therefore, let us analyse for other better methods that allow us to reduce the degree of the monomials.

The first technique we will analyze is to exploit the relationship

$$f(w) = \max_{t \in \{0,1\}} \{t(x + y + z - 2)\}.$$

This relationship follows from the fact that, on the one hand, if any of the variables  $x, y, z$  is equal to 0, we have that  $(x + y + z - 2) \leq 0$  and therefore, the maximum on the right will be reached when  $w$  takes the value 0, resulting in 0. If, on the contrary, all the variables are worth 1, we would have the product  $w \cdot 1$ , which takes the maximum value 1 when  $t = 1$ .

Now, if the monomial  $axyz$  from which we want to reduce the degree has coefficient  $a < 0$ , it is true that

$$\min_{x,y,z} axyz = a \max_{x,y,z}(xyz) = a \max_{x,y,z}(\max_t(t(x+y+z-2))) = a \max_{x,y,z,t}(t(x+y+z-2)) = \min_{t,x,y,z} at(x+y+z-2).$$

Therefore, if  $a < 0$  we can simply substitute monomials of the form  $af(w)$  for  $a(t(x + y + z - 2))$ . If  $a > 0$  we cannot use the same technique since, although

$$f(w) = - \min_{t \in \{0,1\}} t(-x - y - z + 2),$$

The above equation with the minus sign does not allow this term to be entered in the cost function.

To solve this problem, let us return to the problem of expressing the equation  $z = xy$  in linear terms. The penalty functions are needed in these cases. We need to construct a penalty function that takes large values when  $z \neq xy$  and the value 0 when  $z = xy$ . Also, it is needed to work carefully on Lagrange multiplier. This task is important for problems where the degree of the restriction can be greater than 2; Problems that go beyond those of this NISQ era.

## 4.5 Summary

In this chapter, we have analysed and studied the methods of solving combinatorial optimisation algorithms. Furthermore, we have studied Linear Programming and its derivative of binary integers, Quadratic Programming and Quadratic Programming with Quadratic Constraints. But here, we have not studied a lead combinatorial optimisation problem-solving algorithm, QUBO. However, the reader can find it in the section (9.1.1) on quantum computing because of its close connection with the Ising models. Moreover, QUBO is a problem central to adiabatic quantum computing, where it is solved by a physical process called quantum annealing.

We can now entirely focus and delve into quantum computing, but to do this, we will introduce some necessary concepts and theories of quantum mechanics.



# Chapter 5

## Introduction Quantum Mechanics

### 5.1 Introduction

*Quantum mechanics* [194] is a theory that describes the physical properties of nature on an atomic scale.

The first revolution[125] of quantum mechanics is the basis of several advances to our modern society. Almost all modern electronics in the last 50 years have been based on the properties of quantum mechanics. Semiconductors such as diodes, transistors, integrated circuits, etc., have led to the considerable growth of electronics today.

Quantum mechanics, in addition to studying the motion of particles, also allows us to understand the properties of materials and their features to manufacture transistors. These are the basis of all modern electronics and the control system of millions of devices that we use today. However, in this thesis work, we focus on the second revolution of quantum mechanics, that is, in using the properties of quantum mechanics to empower computing. For this, we need to know how quantum information science is structured (5.2).

The *Quantum information science* also known as QIS, is the area of information science that depends on the effects of quantum mechanics. Figure (5.2) breaks down the subtasks of quantum information science.

During the last decades, the scientific community has dedicated a lot of time and has provided considerable resources to QIS [251]. As a result of this delivery, improvements have been achieved in the scientific advances that we see reflected in publications, conferences, and concrete solutions such as D-Wave, IBMQ, Xanadu, etc. Quantum computing, in particular, and quantum information science is a hot topic because of its novelty and the promising developments it predicts.

In this thesis work, all the postulates of quantum mechanics on which we rely are referenced by [173].



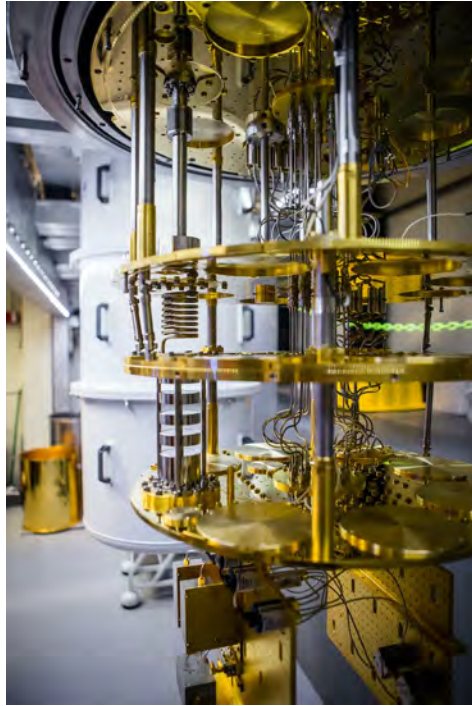


Figure 5.1: TBluefors dilution refrigerators used in the MIT Engineering Quantum Systems group to operate superconducting qubits at near-zero temperatures [235]

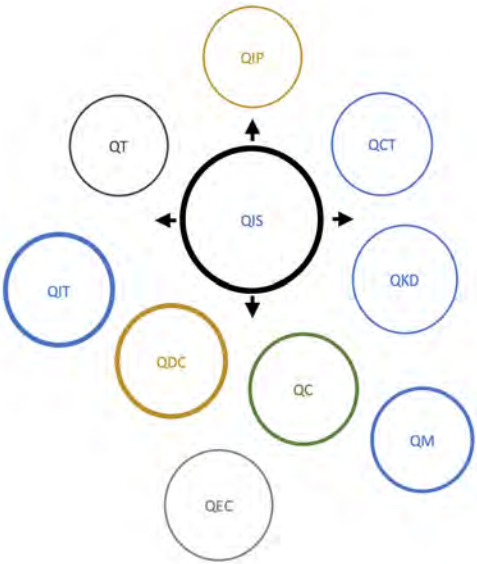


Figure 5.2: The areas of quantum information science (QIS)

## 5.2 Moore Law

Another aspect to consider understanding the importance of the properties of quantum mechanics and thus empower computing is *Moore's law*, which states that the number of transistors that can be integrated into a single chip doubles every 18 months. This process Figure (5.3) leads to a doubling of the memory and a doubling of the calculation speed. Therefore, it is expected that the size of the characteristic of the wafer (thin plate of semiconductor material) will be less than 10nm during this year (2020). At this point, the individual properties of atoms and electrons would begin to dominate, so Moore's law would no longer be valid. Therefore, the demands of the miniaturization of electronics will eventually bring us to the point where quantum effects become important[251].

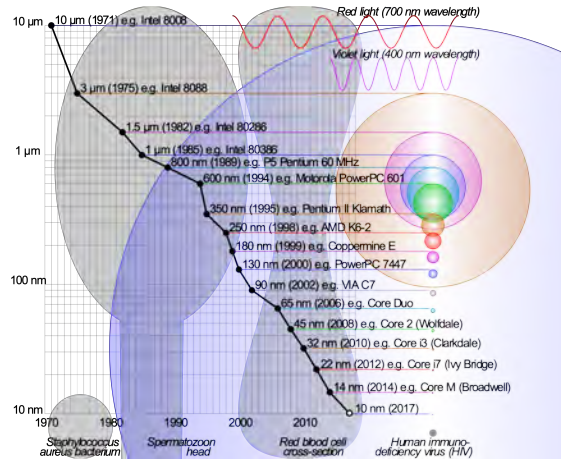


Figure 5.3: Moore law Roadmap: Quantum Computing Stack Exchange [187]

Because Moore's law will stop working today, there are even more reasons for the scientific community to resort to QIP much sooner than it seems [251][123]. On the other hand, multi-core architecture is becoming a practical approach; computational speed improvement can be achieved even without reducing the size of the feature by parallelization.

### 5.2.1 State of the Quantum Technologies

Another point to highlight is the fact that, despite the intensive development of quantum algorithms, the number of available quantum algorithms is still small (Fig. (5.4)) compared to that of classical algorithms basically because the current quantum gates are only several tens of quantum bits (qubits), which is relatively low for any significant quantum computing operation[189].

This delay was partial until very recently, and it was believed that quantum computing would never or seldom come true. However, the latest advances in the different techniques (D-Wave, IBMQ, Xanadu, ...) of quantum gate implementations, as well as the proof of the precision threshold theorem [153], give rise to optimism (Figures (5.5) and (5.6)) that quantum computers (5.1) to large-scale could become a reality quite soon [189]. We can also appreciate (Figures (5.7) and (5.8) ) the metrics and milestones to help monitor the development of quantum computing and the companies and startups involved in the 2018 quantum computing ecosystem.

One of the most significant problems that QIS faces is the physical deployment issues. There are many potential technologies, such as nuclear magnetic resonance (NMR), ion traps, quantum cavity electrodynamics, photonics, quantum dots, and superconducting technologies, to name just a few.

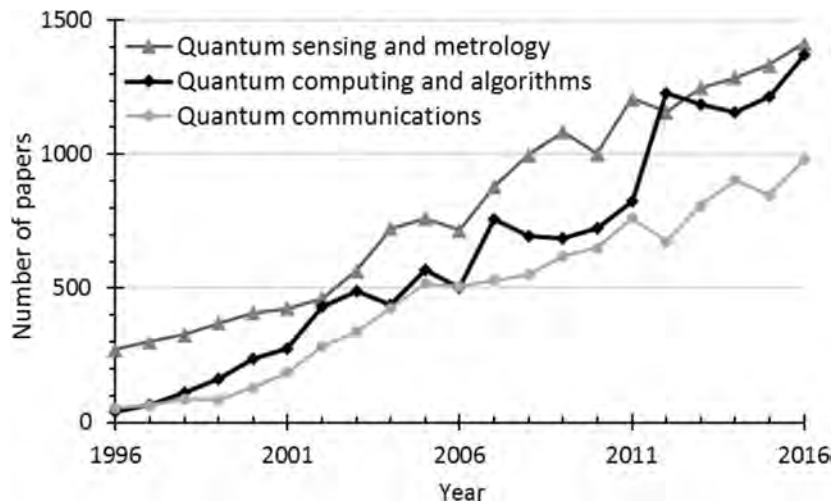


Figure 5.4: The number of research papers published per year in quantum computing and technologies [214]

However, it is not clear which technology will prevail. For example, Xanadu (photonics) technology seems to be most likely for quantum teleportation[127][42].

Right now, there is a big battle going on between the significant manufacturers of quantum computers to achieve the highest number of qubits and 'impose' their scalability plan. In knowing who will be able to define the standard and, above all, get the quantum supremacy [24].

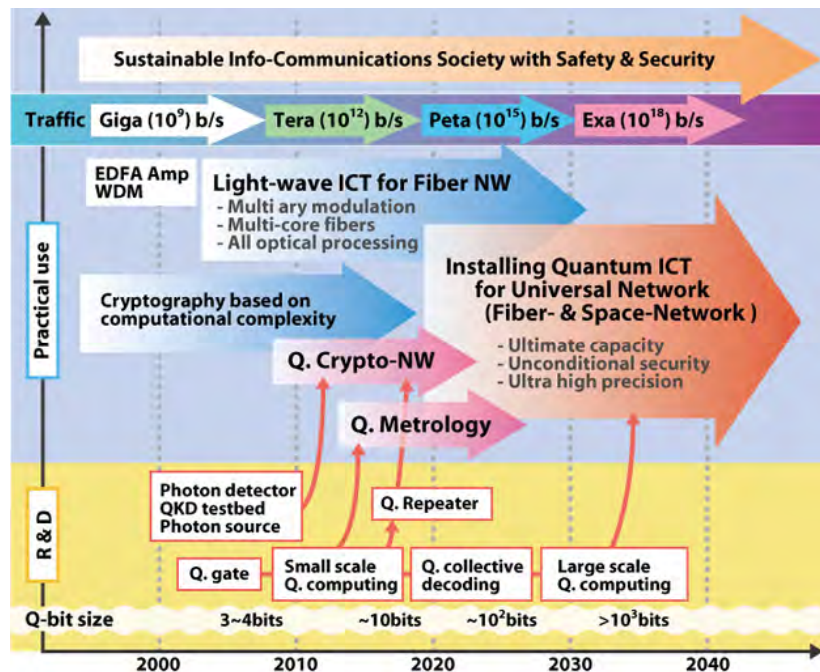


Figure 5.5: Quantum Computing is becoming the talk of the town [4]

### 5.3 Noisy Intermediate-Scale Quantum

The *Noisy Intermediate-Scale Quantum* era known as (NISQ) era [189], is defined as the era leading quantum processors to contain about 50 to a few hundred qubits. Still, it is not advanced enough to reach fault tolerance nor large enough to profit sustainably from quantum supremacy. It is used to describe the current state of the art in the fabrication of quantum processors.

The real problem of this era is related to *decoherence* [60]. *Decoherence* is associated with the interaction of qubits with environments that blur the fragile states of overlap (entanglement). This results in the introduction of random errors due to the environment. However, there are quantum error correction techniques known as the *Quantum Error Correction Concept* (QECC)[194][66]. One of the powerful applications of quantum error correction is based on the protection of quantum information, as it is dynamically subjected to quantum computing. Imperfect quantum gates affect quantum computing by introducing errors into the computed data. Also, imperfect control gates add errors into the processed sequences as incorrect operations can be applied. However, this imperfection gives rise to exciting computing techniques and algorithms based on variational calculations. This opens up a world of possibility to the era of *Quantum Machine Learning* (QML) [47][185][211]. The objective of QECC is to deal with errors introduced by quantum channels and those presented by (imperfect) quantum gates during the encoding and decoding process. Because of this, the reliability of the data processed by quantum computers depends on the probability of error per gate being below a certain threshold known as the precision threshold theorem. This is what NISQ defines as. *'Noisy'* because we don't have enough qubits leftover for error correction, so we'll have to use the imperfect qubits directly on the physical layer and *'Intermediate-Scale'* due to its reduced qubit (but not too small).

To understand quantum mechanics, we have to appreciate its postulates. These postulates provide a connection between the physical world and the mathematical formalism of quantum mechanics. It is important to emphasize that it is unnecessary to understand quantum mechanics in detail to know how to make a program on a quantum computer. But if we want to make intelligent and efficient algorithms, it is more than recommended to understand the fundamentals of quantum mechanics. The

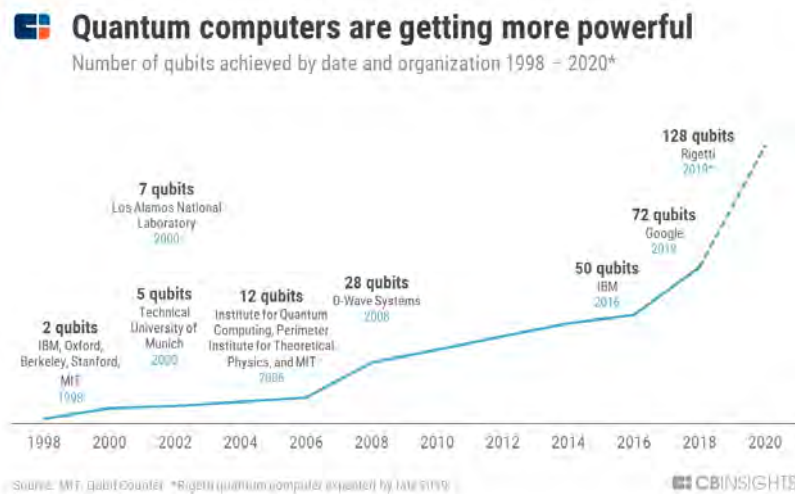


Figure 5.6: The number of qubits achieved by date and organization. Roadmap by MIT [137]

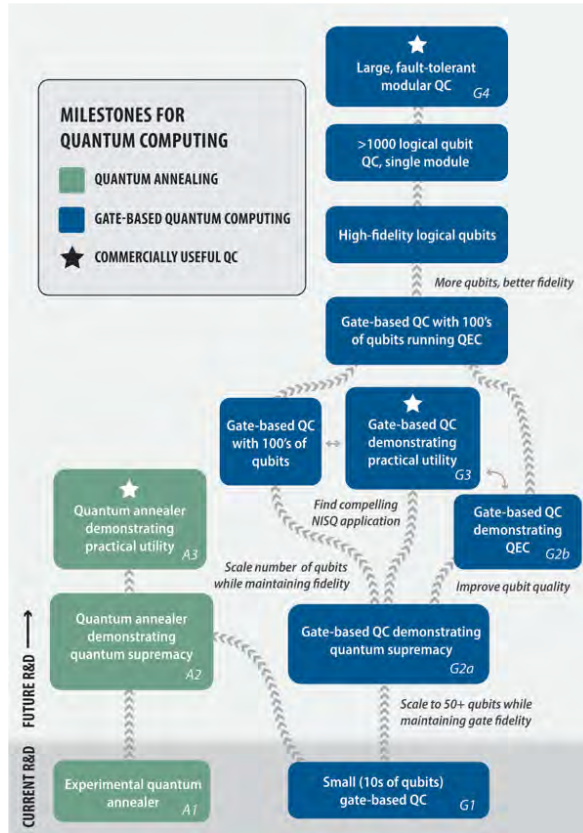


Figure 5.7: Metrics and milestones to help monitor the development of quantum computing. Roadmap by MIT [214]

analogy to understand what we mean is the following: To program we can do it in *PhP*, and *python*, high-level languages, but we can also program in *C* or in *assembler* (machine language). If we want to be experts or make efficient algorithms, it would be useful to know the machine's architecture and to program in the machine language.

## 5.4 Postulate 1

The *postulate 1* is defined as *associated with any isolated physical system is a complex vector space with an internal product (Hilbert space) known as the state space of the system. The system is fully described by its state vector, which is a unit vector in the state space of the system.*

This postulate explains the space of the quantum state and describes the area where quantum mechanics takes place. This area is nothing more than linear algebra in Hilbert vector space [246]. It is essential to know that, given a physical system, quantum mechanics does not tell us what the state space of that system is, nor can it tell us what the system's state vector is. Therefore, it is imperative to have this clear postulate. In other term, what we mean is that it is challenging to know the state of a quantum system at all times.

Let's consider that the simplest quantum mechanical system is the qubit. A qubit has a two-



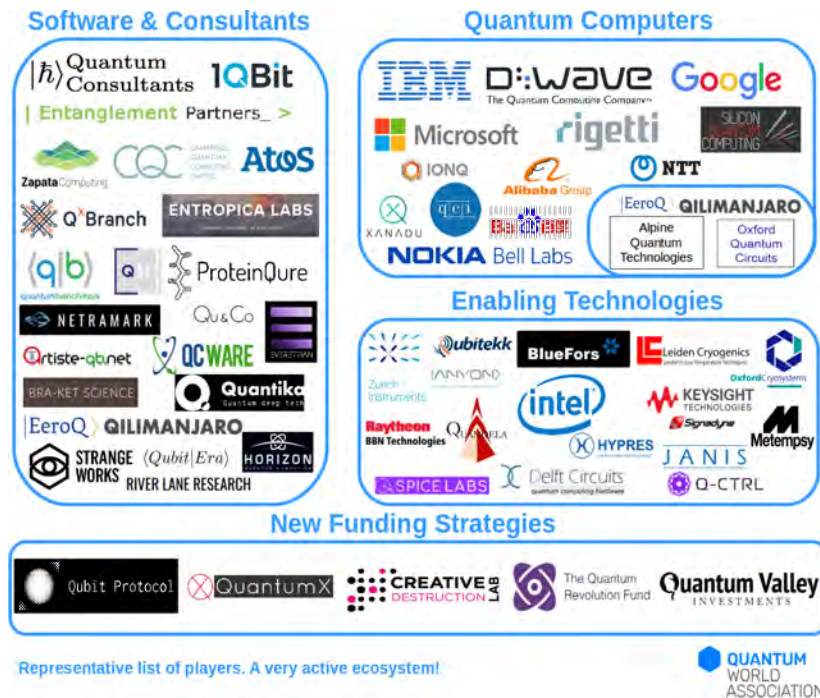


Figure 5.8: Companies and startups involved in the 2018 quantum computing ecosystem. Roadmap by Quantum World Association

dimensional state space  $|0\rangle$  and  $|1\rangle$  that form an orthonormal basis for that state space. Then you can write an arbitrary state vector in the state space.

$$|\psi\rangle = a|0\rangle + b|1\rangle. \quad (5.1)$$

Where  $a$  and  $b$  are complex numbers. The condition that  $|\psi\rangle$  is a unit vector is given by  $\langle\psi|\psi\rangle = 1$ , is therefore equivalent to  $|a|^2 + |b|^2 = 1$ . The term  $\langle\psi|\psi\rangle = 1$  is known as the normalization condition for state vectors.

The way a qubit differs from a classic bit is that there are overlaps of these two states, in the form  $a|0\rangle + b|1\rangle$ , where it is not possible to say that the qubit is definitely in the state  $|0\rangle$ , or definitely in the state  $|1\rangle$ . In other words, what we mean is that the quantum state is a linear combination of the components  $a$  and  $b$ .

If we had a system of more than one qubit, the expression of the quantum state would be of the form  $\sum_i a_i |\psi_i\rangle$ , and we would have a system with the superposition of the states  $|\psi_i\rangle$  and of amplitude  $a_i$  for the state  $|\psi_i\rangle$ .

## 5.5 Postulate 2

The *postulate 2* is defined as *a unitary transformation describes the evolution of a closed quantum system. That is, the state  $|\psi\rangle$  of the system at the time  $t_1$  is related to the state  $|\psi'\rangle$  of the system at the time  $t_2$  by a unit operator  $U$  that depends only on times  $t_1$  and  $t_2$ ,  $|\psi'\rangle = U|\psi\rangle$ . This postulate gives a standard for describing quantum state changes over time.*

From Ref. [174] we know that quantum mechanics does not tell us the state space or quantum state of a particular quantum system; it also does not tell us which unit operators  $U$  describe real-world quantum dynamics. On the other hand, quantum mechanics assures us that the evolution of any closed quantum system can be described in this way.

$$|\psi'\rangle = U|\psi\rangle. \quad (5.2)$$

In a way, we are saying that if we want to describe an  $n$  qubit quantum system subject to time evolution, we must calculate its unit operators  $U$ . The challenge would be to find such unit operators. Quantum gates are somehow the operators that act on quantum states. This postulate is very interesting and requires that the quantum system be closed [194][193]. Moreover, various derivatives emerge from this postulate.

### 5.5.1 Postulate 2'

The *postulate 2'* is defined as *the time evolution of the state of a closed quantum system is described by the Schrödinger equation (5.3).*

$$i\hbar \frac{\delta|\psi\rangle}{\delta t} = H|\psi\rangle. \quad (5.3)$$

Where  $\hbar$  is Planck's constant and  $H$  is a fixed hermitian operator known as the Hamiltonian of the closed system [194].

In some way, we can say that if we know the Hamiltonian of a system, then we understand its dynamics completely. This concept is precious for modelling the system, and we will use it a lot throughout our thesis.

During the 20th century, much of the scientific (physical) community has been dedicated to discovering the Hamiltonian of any quantum system. And the conclusion they reached is that, in general, finding out that the Hamiltonian necessary to describe a particular physical system is challenging.

The Hamiltonian is a Hermitian operator that allows us to make its spectral decomposition with eigenvalues of energy  $E$  and the normalized eigenvectors corresponding to energy  $E$  as eigenstates  $|E\rangle$ .

$$|H\rangle = \sum_E E|E\rangle\langle E|, \quad (5.4)$$

$$|E\rangle = \exp(-i\frac{Et}{\hbar})|E\rangle. \quad (5.5)$$

The lowest energy is known as the ground state energy for the system, and the corresponding energy proper state (or adequate space or steady-state) is known as the ground state.

$$|\psi(t_2)\rangle = \exp[\frac{-iH(t_1 - t_2)}{\hbar}]|\psi(t_1)\rangle = U(t_1 - t_2)|\psi(t_1)\rangle. \quad (5.6)$$

Where we define

$$U(t_1 - t_2) \equiv \exp[\frac{-iH(t_1 - t_2)}{\hbar}]. \quad (5.7)$$

This interpretation is compelling and extremely useful since it is shown that any unitary operator  $U$  can be written in the form (5.8) tanking  $\hbar = 1$  and  $t_2 = 0$  the equation comes out to be (5.6)

$$U = \exp(-iK). \quad (5.8)$$

With  $K$ , a Hermitian operator.

## 5.6 Postulate 3

The *postulate 3* is defined as *quantum measurements are described using a  $M_m$  collection of measurement operators. These are the operators that operate in the state space of the system being measured. The  $m$  index refers to the measurement results in the experiment. If the state of the quantum system is  $|\psi\rangle$  immediately before the measurement, the probability that the result  $m$  will occur is given by the equation (5.9).*

$$p(m) = \langle\psi|M_m^T M_m|\psi\rangle. \quad (5.9)$$

The state of the system after the measurement is (5.10).

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^T M_m|\psi\rangle}}. \quad (5.10)$$

And, the measurement operator satisfies the completeness equation [194] that is given by the following equation (5.11) for all the values of the quantum state  $|\psi\rangle$ .

$$\sum_m M_m^T M_m = I p(m) = \langle\psi|M_m^T M_m|\psi\rangle = 1. \quad (5.11)$$

The value of the measure is the probability described by  $|a|^2 + |b|^2 = 1$ .

If we want to make the different  $M_m$  observations, the operations we would be doing are the following:

$$M_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (5.12)$$

$$M_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5.13)$$

With  $\langle\psi|$  the conjugate of  $|\psi\rangle$ , represented by  $|\psi\rangle = \langle\psi|^T$

$$M_0|0\rangle = |0\rangle\langle 0|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle. \quad (5.14)$$

$$M_0|1\rangle = |0\rangle\langle 0|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0. \quad (5.15)$$



knowing that  $|0\rangle\perp|1\rangle$ ,  $\langle 0|1\rangle$  disappears; It is cancelled. If we project on any state, we will have:

$$M_0|\psi\rangle = |0\rangle\langle 0|\psi\rangle = |0\rangle a|0\rangle + b|1\rangle = a|0\rangle. \quad (5.16)$$

We can generalize (5.14): let  $M_0$ , let  $i$  be any qubit and let  $|\psi\rangle$  be a multi-state of qubits, the measurement or the projection on the state  $|0\rangle$  can be expressed as:

$$Tr \left[ |\psi\rangle\langle\psi|M_0^i \right]. \quad (5.17)$$

If the qubit  $i$  is measured in the state  $|0\rangle$ , then the system will be in the state expressed by (5.18).

$$\frac{M_0^i|\psi\rangle}{\|M_0^i|\psi\rangle\|_2}. \quad (5.18)$$

Another way to get the result of the measurements is to remember that the measurement operator is Hermitian; this translates to  $M_0^2 = M_0$  the same for  $M_1^2 = M_1$ . Also remembering that the relationship of completeness obeys the equation  $I = M_0^T M_0 + M_1^T M_1 = M_0 + M_1$ .

One of the crucial applications of (5.6) is to distinguish quantum states. We recall that since the quantum system must be closed (5.4), getting to find out the quantum states represents a titanic and less intuitive task as in classical computing.

To clarify what we have just advanced, we will demonstrate the absurdity. We consider two non-orthogonal quantum states  $|\psi_1\rangle$  and  $|\psi_2\rangle$  and assume that the measurement is possible. If  $|\psi_1\rangle$  and  $|\psi_2\rangle$  are prepared, the measures (observations) will respond to the completeness (5.11) equations so that we can write:

$$\langle\psi_1|M_1|\psi_1\rangle = 1; \langle\psi_2|M_2|\psi_2\rangle = 1. \quad (5.19)$$

Knowing that  $\sum_i M_i = I$  therefore  $\langle\psi_1|M_1|\psi_1\rangle = 1$  and must  $\langle\psi_1|M_2|\psi_1\rangle = 0$  and  $\sqrt{M_2}|\psi\rangle = 0$ . If we decompose  $|\psi_2\rangle = \alpha|\psi_1\rangle + \beta|\varphi\rangle$  with  $|\psi_1\rangle$  and  $|\varphi\rangle$  they are orthonormal. This leads us to the fact that  $|\alpha|^2 + |\beta|^2 = 1$  and that  $|\beta| < 1$  while  $|\psi_1\rangle$  and  $|\psi_2\rangle$  are not orthonormal. With  $\sqrt{M_2}|\psi\rangle = \beta\sqrt{M_2}|\varphi\rangle$ . What contradicts completeness's equation (5.9).

$$\langle\psi_2|M_2|\psi_2\rangle = |\beta|^2\langle\varphi|M_2|\varphi\rangle \leq |\beta|^2 < 1. \quad (5.20)$$

Continuing with *postulate 3*, projective measurements form a particular case of it, being very useful and straightforward. Due to its simplicity, this measurement is the basis of many algorithms to know the observables.

A projective measurement is described by an observable,  $M$ , a Hermitian operator in the system's state space being observed. The observable has a spectral decomposition.

$$M = \sum_m m P_m. \quad (5.21)$$

where  $P_m$  is known as the projector in the proper space of  $M$  with the right value  $m$ . If we measure on the state  $|\psi\rangle$ , the probability of having the result  $m$  is

$$p(m) = \langle\psi|P_m|\psi\rangle. \quad (5.22)$$

so, the quantum state just after the measurement will be:

$$\frac{P_m|\psi\rangle}{\sqrt{\langle\psi|P_m|\psi\rangle}}. \quad (5.23)$$

Otherwise, what we are saying is that  $\langle M \rangle = \langle \psi | M | \psi \rangle$ . We can generalize this formula with the mean and variance:

$$[\Delta M]^2 = \langle (M - \langle M \rangle)^2 \rangle = \langle M^2 \rangle - \langle M \rangle^2 \Delta M = \sqrt{\langle M^2 \rangle - \langle M \rangle^2}. \quad (62)$$

This formula (62) is the basis of Heisenberg's uncertainty principle [194]. The important thing about the Heisenberg principle is that we cannot simultaneously measure/know the position and velocity of the electron (a particle). Therefore, it is impossible to determine its trajectory. We are somehow telling ourselves that measurement is destructive (but this is not the basis of Heisenberg's principle) because the position and velocity of the particle are in overlap. Measuring one collapses the wave function and destroys the other component/s.

A mathematical explanation can be described as follows. Suppose  $A$  and  $B$  are two Hermitian operators, and  $|\psi\rangle$  is a quantum state. Suppose that  $\langle \psi | AB | \psi \rangle = x + iy$ , where  $x$  and  $y$  are real. Applying the switches and anti-switches to the Hermitian operators (matrices)  $A$  and  $B$ , we arrive at  $\langle \psi | [A, B] | \psi \rangle = 2iy$  and  $\langle \psi | \{A, B\} | \psi \rangle = 2x$ . This implies that:

$$|\langle \psi | [A, B] | \psi \rangle|^2 + \langle \psi | A, B | \psi \rangle^2 = 4|\langle \psi | AB | \psi \rangle|^2. \quad (5.24)$$

For the Cauchy-Schwarz inequality

$$|\langle \psi | AB | \psi \rangle|^2 \leq \langle \psi | A^2 | \psi \rangle \langle \psi | B^2 | \psi \rangle. \quad (5.25)$$

underestimating the negative term, we arrive at:

$$|\langle \psi | [A, B] | \psi \rangle|^2 \leq 4\langle \psi | A^2 | \psi \rangle \langle \psi | B^2 | \psi \rangle. \quad (5.26)$$

if we now consider two observables  $M_1$  and  $M_2$ , and substitute  $A = M_1 - \langle M_1 \rangle$  and  $B = M_2 - \langle M_2 \rangle$  in the last equation, we obtain the Heisenberg uncertainty principle from the equation (5.27).

$$\Delta M_1 \Delta M_2 \geq \frac{\langle \psi | [M_1, M_2] | \psi \rangle}{2}. \quad (5.27)$$

The correct interpretation of the uncertainty principle is that if we prepare a large number of quantum systems in identical states,  $|\psi\rangle$ , and then make measurements of  $M_1$  in some of those systems, and of  $M_2$  in others, then the standard deviation  $\Delta M_1$  of the results of  $M_1$  multiplied by the standard deviation  $\Delta M_2$  of the results of  $M_2$  will satisfy the inequality  $\Delta M_1 \Delta M_2 \geq \frac{\langle \psi | [M_1, M_2] | \psi \rangle}{2}$ .

In the commutation relationship for the observable  $X$  coordinates and moment  $P$  is,  $[X, P] = j\hbar$ . If we introduce it in the equation above, we get to (5.28):

$$\langle \Delta X \rangle^2 \langle \Delta P \rangle^2 = \frac{\hbar^2}{4}. \quad (5.28)$$

We make a parenthesis to explain the behaviour of the phase-in quantum mechanics. We already know that quantum mechanics is defined in the complex vector space (Hilbert space). We also know that a vector representation of a vector (wave) can be described with its argument and angle. Besides, the phase as an operator applied to a quantum state in quantum mechanics does not change the quantum state.

Now, let us consider that the state  $e^{i\theta}|\psi\rangle$ , where  $|\psi\rangle$ , is a state vector and  $\theta$  is a real number. We say that the state  $e^{i\theta}|\psi\rangle$ , is equal to  $|\psi\rangle$  where the factor  $e^{i\theta}$  is known with the global phase of the system [194][123]. This property is fascinating and useful when writing a quantum algorithm. Saving this operation entails a gain in time and computational cost.

Suppose we want to make the measurements of the observables  $M_m$  on the quantum state  $e^{i\theta}|\psi\rangle$ ,

with  $|\psi\rangle$  the state vector. Then, the respective measurements will be given by the following equations:

$$\langle\psi|M_m|\psi\rangle\langle\psi|e^{-i\theta}M_m^TM_me^{i\theta}|\psi\rangle=\langle\psi|M_m^TM_m|\psi\rangle. \quad (5.29)$$

We see that the global phase does not affect measures, but another phase does. This phase is known as the relative phase [123]. We will use an example to explain it. Let's consider two quantum states:

$$|\psi_1\rangle=a|0\rangle+b|1\rangle \quad \text{and} \quad |\psi_2\rangle=a|0\rangle-b|1\rangle. \quad (5.30)$$

We see that the state amplitude  $|1\rangle$  of the first quantum state  $|\psi_1\rangle$  is  $+b$  and  $a$ , the state amplitude  $|0\rangle$  of the second quantum state  $|\psi_2\rangle$  is  $-b$ . We see that the only difference is the sign, not the amplitude. If two quantum states have the same amplitudes and differ exclusively by the phase (sign), this phase is known by the relative phase of the system. And generically, we can write it as follows  $a = \exp(i\theta)b$ . This concept is fascinating and is the basis of quantum gates that will allow quantum computing.

## 5.7 Postulate 4

The *postulate 4* is defined as *the state-space of a composite physical system is the tensor product of the state spaces of the components of physical systems. Furthermore, if we have systems numbered from 1 to n, and the number of system i is prepared in the state  $|\psi_i\rangle$ , then the joint state of the total system is  $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle$ .*

This postulate is the basis for creating the complex system of more than two qubits or quantum states. This postulate also allows us to define one of the most exciting and puzzling ideas associated with composite quantum systems: entanglement. Consider the two-qubit states  $|\psi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ , this state has the remarkable property that there are no unique qubit states  $|a\rangle$  and  $|b\rangle$  such that  $|\psi\rangle = |a\rangle|b\rangle$ . In other words, we say that  $|\psi\rangle \neq |a\rangle|b\rangle$ . This property is one of the reasons for the empowerment of quantum computing.

## 5.8 Postulate 5 and 6

In this thesis work, we are based on the postulates listed in this reference [173] *Quantum Computation and Quantum Information*, where postulate 5 (The temporal evolution of a system) of quantum mechanics [73] is contemplated like our postulate 2' (5.5.1) and the postulate 6 (commutation rules) of quantum mechanics that define the positional and momentum operators that satisfy the following commutation rules, they are treated as operators not like postulate.

## 5.9 Summary

This section's right conclusion highlights that these four postulates of quantum mechanics define how we can conceptualize and face a problem about quantum mechanics and its computation. The first postulate establishes the space for quantum mechanics by specifying how the state of an isolated quantum system should be described. The second postulate illuminates us on the dynamics of the closed quantum systems and its description through the Schrödinger equation and using the unitary evolution. In the third postulate, we are explained how to make the measurement describing the importance of the characteristics (restrictions) when extracting information (measuring) from our

quantum systems. And finally, the fourth postulate reveals how we can create composite systems. One of the most shocking and interesting ideas of quantum mechanics is that we cannot directly observe the state vector and that it is in charge of deciphering the behaviour of any quantum system. Let's imagine that we want to know what position the cat is in (5.9). In a classic system, thinking



Figure 5.9: Observability of a classical system

about computing, the location (state vector) of the cat is given by the variable  $x = 7$



Figure 5.10: Observability of a quantum system

In a quantum system, the position (see figure (5.10)) of the cat is not so quickly known. In quantum, each element in the state vector contains the probability of finding the cat in a specific place. And it is represented by:

$$|x\rangle = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{array}{l} \text{Probability of} \\ \text{cat being at } \cdot \\ \leftarrow \text{position 7} \end{array} \quad (5.31)$$

Classical physics, classical computation, and our intuition tell us that the fundamental properties of an object, such as energy, position, and velocity, are directly accessible to observation. However, these quantities no longer appear essential in quantum mechanics, depend on the state vector, and cannot be directly observed. Furthermore, merely observing a classical system does not necessarily change the system's state; instead, for a quantum system, observation is an invasive procedure that generally changes the state of the system (the state vector).



# Chapter 6

## Complexity Class

### 6.1 Introduction

The *complexity class* is defined as *a set of computational problems of related resource-based complexity, and the two most commonly analyzed resources are time and memory.*

Some problems appear to be "intractable" (complexity is exponential in the number of bits  $n$ ) in classical computing, but they can be solved polynomially in  $n$  by quantum computing. For example, the factorization of prime numbers is an insoluble problem for classical computing, taking into account the size of the numbers to be factored. Still, solvable in quantum by Shor's algorithm [216]. Peter Shor's algorithm is based on the fact that a composite number can be represented in prime numbers in several steps being polynomial in  $n$ . The differential and key value of quantum algorithms are that they can explore all branches of a non-deterministic algorithm in parallel through the concept of quantum parallelism (superposition of basic states).

There are necessarily three properties that can vary in the definition of a complexity class. The resource of interest (time, space, ...), the type of problem being considered (decision problem, optimization problem, ...), and the underlying computational model (deterministic Turing machine, probabilistic Turing machine, quantum computer, ...).

We know that classical algorithms can be classified as useful when the number of steps is a polynomial function of size  $n$ . Therefore, the computational complexity of these algorithms is typically denoted as  $P$ . The class of problems that can be solved in a non-deterministic manner [123] in polynomial time are called  $NP$  [163]. The subclass of these problems, which are the most difficult, is complete  $NP$  problems [163]. For example, the VRP and its derivatives belong to this subclass, and if one of these problems can be solved efficiently, they can all be solved. The kind of problem that can be solved with the amount of the memory polynomial in the input size is called  $PSPACE$  [163]. Also, the kinds of problems can be solved with high probability by using a random generator known as  $BPP$ , which originates from a time-limited error probability polynomial. In addition to all these kinds of complexity, there is a final category that can be solved in polynomial time if the exponential sums of many taxpayers are computable in polynomial time. This is denoted as  $P^{#P}$ . In summary, we can establish this relationship between complexity classes.

$$P \subset BPP, P \subset NP \subset P^{#P} \subset PSPACE. \quad (6.1)$$

Being the quantum analogue of BPP, known as BQP [163], the most interesting in our study of quantum algorithms.

In computational complexity theory, Bounded error Quantum Polynomial-time BQP [163] is the class of decision problems decidable by a quantum computer in polynomial time with an error probability of at least  $1/3$  for all instances. They are problems based on Large Linear Systems, Quantum material simulations and Quantum walk encompassing the SWP.

## 6.2 Quantum complexity

The *quantum complexity theory* is defined as *the sub-field of computational complexity theory that deals with complexity classes defined using quantum computers. This class studies the hardness of computational problems concerning these complexity classes and the relationship between quantum complexity classes and classical ones.*

Alan Turing [194] defined a class of machines, known as Turing machines, that can be used to study the complexity of a computational algorithm. In particular, there are the so-called universal machines, which can be used to simulate any other Turing machine, and the latter can be used to simulate all the operations carried out on a modern computer.

This led to the formulation of the Church-Turing thesis [194]. One of the most powerful arguments in computing remains in force today. The study specifies that the class of functions that a Turing machine can calculate corresponds precisely to the class of functions that one would naturally consider computable by an algorithm. This thesis establishes the equivalence between the mathematical description defined by a Turing machine and an intuitive concept. Some problems are not foreseeable, and there is no known algorithm to solve them. The Church-Turing thesis [194] also applies to quantum algorithms.

We know from the complexity class,[123] that an algorithm is competent if it can be solved in a polynomial number (P) of steps. We also know that Turing machines can describe these practical algorithms. With the fusion of these last two concepts, we can formulate a robust version of the Church-Turing thesis looking for a universality. Any computational model can be simulated in a probabilistic Turing machine with a polynomial number of computational steps.

## 6.3 The deterministic Turing machine

The *Turing deterministic machine* is described by an alphabet  $A$ , a set of control states  $Q$ , and a transition function  $\delta$ .

$$\delta : Q \times A \rightarrow Q \times A \times D. \quad (6.2)$$

The alphabet elements are called letters ( $A$ ), and by concatenating the letters, we get words. Set  $D$  is related to the read-write head, with elements  $D = \{-1, 0, 1\}$   $-1$ ,  $+1$  and  $0$  that indicate the movement of the head to the left, to the right and the foot, respectively. The deterministic Turing machine can be defined as  $(Q, A, \delta, q_0, q_a, q_r)$ , where the state of the machine is specified by  $q \in Q$ . In particular,  $q_0, q_a, q_r \in Q$  denotes the initial state, the acceptance state and the rejection state, respectively.

The configuration of the Turing machine is given by  $c = (q, x, y)$ , where  $x, y \in A'$ , where  $A'$  is the set of all the words obtained by concatenating the letters of  $A$ .

The Turing machine has a tape (memory) specified by  $xy$  with  $x$  being the scan (read). Somehow, we can define computation as a sequence of configurations that start with initial setup  $c_0$  (sometimes

this initial configuration is called the initial state represented by  $\varepsilon_0$ ) until we reach the stop configuration. Computing stops after  $p$  calculation steps (interaction) when one of the configurations does not have a successor or if its state is  $q_a$  or  $q_r$ .

## 6.4 The probabilistic Turing machine.

A *probabilistic Turing machine* is defined as a *non-deterministic Turing machine that chooses between the available transitions at each point according to some probability distribution.*

There are several more computationally dependent Turing machines. The most generic is the probabilistic Turing machine [163]. Since the transition function assigns probabilities to possible operations, the deterministic machine is a specific case of probability. In other words, a probabilistic Turing machine is a non-deterministic Turing machine. That randomly selects possible transitions according to some probability distribution and is defined as follows:

$$\delta : Q \times A \times Q \times A \times D. \quad (6.3)$$

In this case, a stochastic matrix can describe the machine state transitions  $D = [0, 1]$ . In the same way, as in the deterministic Turing machine, a given configuration is a successor configuration with probability  $\delta$ . A terminal structure can be calculated from an initial setup with a probability given by the product of probabilities of intermediate configurations, which lead to it by a particular calculation defined by a sequence of states. For example, a generalized case is the deterministic Turing machine of type  $m$ . Said  $m$  tapes characterize machine, an alphabet  $A$ , a finite state of control states  $Q$  and the following transition function.

$$\delta : (Q \times A^m) \rightarrow Q \times (A \times D)^m. \quad (6.4)$$

The configuration of a machine of type  $m$  is given by  $(q, x_1, y_1, \dots, x_m, y_m)$ , where  $q$  is the current state of the machine,  $(x_i, y_i) \in A' \times A'$ , and  $x_i y_i$  denote the content of the  $i$ -th type. Machines of type  $m$  are suitable for problems related to parallelism [193].

If the computation time required for a machine of a type is  $t$ , then the computation time needed for a device of type  $m$  is  $O(t^{1/2})$ , while in terms of computational complexity, they are comparable.

## 6.5 Quantum Turing Machine

The *Quantum Turing Machine*, also known as the universal quantum computer, is *an abstract machine used to model the effects of a quantum computer.*

The following transition function characterizes the Turing quantum machine:

$$\delta : Q \times A \times Q \times A \times D \rightarrow C. \quad (6.5)$$

In quantum, the transition function moves a given configuration into a range of successor configurations. Each of which has a quantum probability amplitude, which corresponds to a unit transformation of a quantum state that is the set of complex numbers by  $C$ . Scientist and physicist Charles Henry Bennett demonstrated that *m-type Turing machines* could be simulated using reversible Turing machines with a specific reduction in efficiency [193]. Furthermore, Professor of Electrical Engineering



and Automat Theory, Tommaso Toffoli, demonstrated that finite arbitrary mapping could be calculated reversibly by padding strings with zeros, permuting them, and projecting some of the bit strings to other bit strings. In other words, what Toffoli means is that elemental reversible gates can be used to implement bit string permutations. Finally, one of the most powerful contributions to quantum computing based on Postulate 2 came from the physical scientist Paul Benioff, who demonstrated that the unitary evolution of the quantum state (which is reversible) is at least as powerful as a Turing machine [194][123][193].

The classical probabilistic process can be represented by a tree, which grows exponentially with possible results. The key difference in quantum computing is that we assign the quantum probability amplitudes to the branches of a tree, which can interfere with each other.

## 6.6 Study of complexity

In our case, the concept of computational complexity that we use is essentially the same as the complexity of the quantum circuit, that is, the minimum number of quantum gates required to prepare a given unit operator [123]. The computational complexity of  $U(t)$  progresses over time. Both the black hole and the considerations of the quantum circuit suggest the following conjecture summarized in Figure (6.1) [223][95].

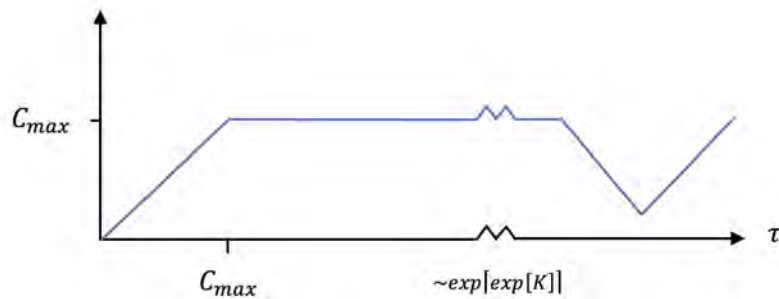


Figure 6.1: The conjectured evolution of the quantum complexity of the operator  $e^{iHt}$  [38]

The complexity  $C(t)$  grows linearly as follow:

$$C(t) = Kt. \tag{6.6}$$

For an exponential time in  $K$  and  $t^K$ , the complexity reaches its maximum possible value  $C_{\max}$  and flattens for a long time. This is the period of complexity. Equilibrium during which the complexity fluctuates over the maximum

$$C_{\max}(t)^K. \tag{6.7}$$

On a much longer order time scale, quantum repeats  $\exp[e^K]$  quasi-periodically return complexity to subexponential values. All this is a conjecture that, at the moment, cannot be proved, but that can be related to other complex assumptions.

The instance-averaging strategy sometimes allows conclusions about generic behaviour that would not be possible for specific cases. A particular example, which has generated recent interest, is the Sachdev-Ye-Kitaev (SYK) approach to coding. By averaging an appropriate set of time-independent

Hamiltonians, it is possible to demonstrate that almost all Hamiltonians saturate the fast coding limit. Potentially, this averaging can also be applied to questions about the evolution of complexity[95].

Another type of randomness is stochastic randomness, over which a statistically fluctuating (noisy) time-dependent Hamiltonian is averaged. The more averaging, the easier it is to conclude, and the stochastic average is easier than averaging over time independent Hamiltonians. In our case, the unit matrix that describes our quantum system is written  $U = e^{iH}$ .

$$H = \sum_{i < j} J_{ij} s_i s_j + \sum_i h_i s_i. \quad (6.8)$$

With  $H$ , the Hamiltonian of our objective function. We express  $H$  as a sum of Pauli's tensor products. This way is even very valid when it comes to finding the unit matrices of various quantum systems (see (5.7)).

There are many theories and strategies to determine the complexity of a quantum circuit. Going through approximation theory, based on entropy, or by the geometry of complexity. It can also be achieved based on the entropy of a classic system as we highlight in the following Lemma [58]:

*Quantum complexity for a  $K$  qubit system behaves similarly to the entropy of a classical approach with  $2^K$  degrees of freedom.*

A practical criterion for calculating the complexity of a unit operator is the concept of the complexity of the quantum circuit. For example, in the case of running a quantum algorithm on IBMQ, the first thing the IBMQ framework does is to compile all the operations of a single qubit on universal gates  $U_1$ ,  $U_2$  and  $U_3$ .

Next, we will discuss the complexity of the circuit consists of all  $K$  qubit circuits composed of  $k$ -local gates allow us to prepare  $U$ . For simplicity, we take the gates to act in series,

$$U = g_N g_{N-1} \dots g_1. \quad (6.9)$$

The complexity of the  $U$  circuit is denoted  $C(U)$ . It is the minimum number of  $k$ -local gates needed to build  $U$ . At last, it depends on the choice of allowable gates and some measures of the distance and concatenation on the gates. Hence, the concept of quantum relative complexity.

Relative complexity can also be defined for a pair of unit operators. This means that if we have a system composed of  $n = 2$  unit matrices  $U_n$ , the relative complexity would be the complexity  $U_1 U_2$  and, in a generic case,  $U_1 U_2 \dots U_n$ . Inspired by Nielsen's ideas, we can build various auxiliary theories [174]. We consider an interesting approach is a theory based on a complexity metric. This theory is based on the idea of the complexity geometry that consists of defining a new parameter in  $SU(2^K)$  different from the standard metric, in which the distance between two elements of  $SU(2^K)$  reflects its complexity relative and where  $SU$  is a space of the unit matrices of  $2^K$  dimensions. To be precise, in our case, we are calculating the Kolmogorov complexity[79] of a time-independent Hamiltonian with a tolerance  $\delta$ . The distance between two spins gives this tolerance  $J_{ij}$  within the lattice model (see figures (3.2) and (3.3)) [88][149] when determining the Hamiltonian of Ising.

In the Kolmogorov complexity calculation, time  $t$  does not appear. In this case, the first part is a fixed overhead that does not scale with  $t$ . The second part specifies the time required  $\log t$  bits. And the third part establishes that the Hamiltonian also requires bits  $\log t$ , because to approximate  $e^{-iHt}$  for a time  $t$  requires precision in  $H$  that is an inverse polynomial in  $t$ .

The simplest model is given when a single gate operates at every moment. So, the Kolmogorov complexity per gate would be ordering one insignificant. Although the choice also encompasses which group of  $k$  qubits performance between the gate at each stage. For example; in the case of  $k = 2$ , there are  $K(K - 1)/2$  possibilities to choose. That means, for each gate, we must add a Kolmogorov  $\sim \log K^2$  complexity. We can quickly realize this by assigning a  $\log K^2$  of complexity to each gate. The total complexity of a unit operator in the stochastic model is  $\log K^2$  times the minimum number of gates required to prepare  $U$ .

## 6.7 Summary

This chapter is significant because it touches on determining the complexity of a quantum algorithm. We have reviewed that the complexity of a unitary operator is from the concept of complexity of the quantum circuit, Nielsen's idea that marries the postulate 4 of quantum mechanics as the tremendous and pragmatic contribution Kolmogorov by the gate. Before all this, we have seen that the Turing machine (probabilistic) is useful for quantum computing.

SWP is a problem that belongs to the BQP complexity class because it behaves like the Quantum Walk algorithm [20].

# Chapter 7

## Quantum Gates

### 7.1 Introduction

Quantum gates are the operators applied to a quantum system to modify its state (state vector).

As discussed above, Ref. (5.2) allows us to define quantum gates with these operators that act on quantum states  $|\psi'\rangle = U|\psi\rangle$ .

### 7.2 General U-gate

One of the most important contributions to postulate two based quantum computing came from the physical scientist Paul Benioff [64] who has shown that the unitary evolution of the quantum state (which is reversible) is at least as powerful as a Turing[123][174]. Both quantum and classical computing can be viewed as associations of universal quantum gates. For simplicity, we take the gates to act in series as is shown in figure (7.1),

$$U = g_N g_{N-1} \dots g_1. \tag{7.1}$$

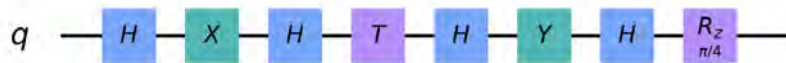


Figure 7.1: One-qubit circuit of quantum gate association to illustrate unit transformation matrix  $U$

Along these lines, there are two fundamental theorems related to the construction of quantum circuits.

**Theorem 1** [87][174]: *All quantum circuits can be constructed using the only unit and NOT controlled gates.*

**Theorem 2 [87][174]** : All quantum circuits can be constructed (in an approximate sense) using only Hadamard gates and Toffoli gates.

In the same way, Emil Post announced that any circuit could be built with only three boolean AND, OR and NOT (fan-out) gates using the Toffoli gates. We won't go into detail in this demo, but the following reference validates it [174].

Next, we will look at some new logic gates and then regain the notion of universality for quantum computing.

**Theorem 3[87][174]**: A map  $U : H_n \rightarrow H_n$  is unitary if, and only if, the matrix it represents in some coordinate representation,  $A$ , satisfies  $A^*A = AA^* = I_n$ , where  $*$  is the transposition of the complex conjugate vector. This property is independent of the chosen coordinate representation.

Let  $X$  be the operator associated with a square matrix, whose elements are  $X_{mn} = \langle a^{(m)} | X | a^{(n)} \rangle$ , which can be written as follows:

$$X_{mn} = \begin{bmatrix} \langle a^{(1)} | X | a^{(1)} \rangle & \langle a^{(1)} | X | a^{(2)} \rangle & \cdots & \langle a^{(1)} | X | a^{(n)} \rangle \\ \langle a^{(2)} | X | a^{(1)} \rangle & \langle a^{(2)} | X | a^{(2)} \rangle & & \langle a^{(2)} | X | a^{(n)} \rangle \\ & \vdots & \ddots & \vdots \\ \langle a^{(m)} | X | a^{(1)} \rangle & \langle a^{(m)} | X | a^{(2)} \rangle & \cdots & \langle a^{(m)} | X | a^{(n)} \rangle \end{bmatrix}. \quad (7.2)$$

### 7.3 Single qubit gates

Let be a qubit with the matrix representation in the computational base  $\{|0\rangle, |1\rangle\}$  we assign to these the representation of natural coordinates:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (7.3)$$

we can write based on the linear combination of the qubit by a matrix. Let's define the fundamental quantum gates: the Hadamard  $H$  gate, the phase shift  $S$  gate, the  $CNOT$  gate, and the Pauli X, Y, Z operators. Recall that the Hadamard H-gate, S-phase shift gate, and the NOT quantum gate have matrix representations in the computational base (CB)  $\{|0\rangle, |1\rangle\}$ .

### 7.4 Gate I

$$I_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 1|. \quad (7.4)$$

Gate  $I$ , figure (7.2), is identity. Many kinds of literature consider her as the first operator of Pauli.

## 7.5 Gate NOT

$$X_{22} = \begin{bmatrix} \langle 0|X|0\rangle & \langle 0|X|1\rangle \\ \langle 1|X|0\rangle & \langle 1|X|1\rangle \end{bmatrix} = \begin{bmatrix} \langle 0|1\rangle & \langle 0|0\rangle \\ \langle 1|1\rangle & \langle 1|0\rangle \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = |0\rangle\langle 1| + |1\rangle\langle 0|. \quad (7.5)$$

We see that the NOT gate (figure (7.3)) maps  $|0\rangle$  to  $|1\rangle$  and  $|1\rangle$  to  $|0\rangle$ . We can also write the following operations.

$$X|0\rangle = |1\rangle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |1\rangle. \quad (7.6)$$

In the same way:

$$X|1\rangle = |0\rangle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |0\rangle. \quad (7.7)$$

The NOT gate is known as the  $X$  gate.

## 7.6 Gate Y

$$Y_{22} = \begin{bmatrix} \langle 0|Y|0\rangle & \langle 0|Y|1\rangle \\ \langle 1|Y|0\rangle & \langle 1|Y|1\rangle \end{bmatrix} = \begin{bmatrix} \langle 0|1\rangle & -i\langle 0|0\rangle \\ i\langle 1|1\rangle & \langle 1|0\rangle \end{bmatrix} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = i|0\rangle\langle 1| - i|1\rangle\langle 0|. \quad (7.8)$$

We see that the gate  $Y$  (figure (7.4)) maps  $|0\rangle$  towards  $|1\rangle$  and  $|1\rangle$  towards  $|0\rangle$  and at the same time, introduces a phase change. We can also write the following operations

$$Y|0\rangle = i|1\rangle i \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = i \begin{bmatrix} 0 \\ 1 \end{bmatrix} = i|1\rangle. \quad (7.9)$$

In the same way:

$$Y|1\rangle = -i|0\rangle i \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -i \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -i|0\rangle. \quad (89)$$



Figure 7.2: Circuit of the Quantum Identity gate  $I$

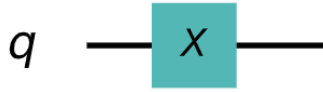


Figure 7.3: The Circuit of the quantum Pauli's first gate if we do not consider gate I

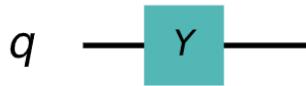


Figure 7.4: Circuit of the quantum gate Y. The second gate of Pauli

## 7.7 Gate Z

$$Z_{22} = \begin{bmatrix} \langle 0|Z|0\rangle & \langle 0|Z|1\rangle \\ \langle 1|Z|0\rangle & \langle 1|Z|1\rangle \end{bmatrix} = \begin{bmatrix} \langle 0|0\rangle & \langle 0|1\rangle \\ \langle 1|0\rangle & -\langle 1|1\rangle \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|. \quad (7.10)$$

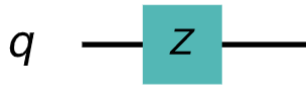


Figure 7.5: Circuit of the quantum gate Z. Third gate of Pauli

We see that the gate  $Z$  (figure (7.5)) maps  $|0\rangle$  towards  $|1\rangle$  and  $|1\rangle$  towards  $|0\rangle$  and at the same time, introduces a phase change. We can also write the following operations.

$$Z|0\rangle = |0\rangle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle. \quad (7.11)$$

In the same way:

$$Z|1\rangle = -|1\rangle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -|1\rangle. \quad (7.12)$$

If we look closely, we will realize that the operator  $Z$  applied to the quantum system does not alter the output. At most, add a 180-degree phase. This phase does not affect the system since it is an absolute phase. This is because states  $|0\rangle$  and  $|1\rangle$  are the two proper states of gate  $Z$ . This is why the base  $Z$  is used as the computational base (the base formed by states  $|0\rangle$  and  $|1\rangle$ ). Many other bases

are used as bases as follows.

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (7.13)$$

and

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (7.14)$$

## 7.8 The Pauli Operators

The Pauli operators  $X, Y, Z$  (also known as  $\sigma_x, \sigma_y, \sigma_z$ ) correspond to the measurement of the turn along the  $x^-$ ,  $y^-$  and  $z^-$  axes respectively. Its actions in the base states are given by where it is clear that the base states are elements of  $Z$ :

$$X|0\rangle = |1\rangle, X|1\rangle = |0\rangle, Y|0\rangle = -j|1\rangle, Y|1\rangle = j|0\rangle, Z|0\rangle = |1\rangle, Z|1\rangle = -|1\rangle. \quad (7.15)$$

These operators fulfil the unit transformation property and the Hermitian property, that is,  $A^*A = AA^* = I_n$ .

$$XX^* = X^2 = YY^* = Y^2 = ZZ^* = Z^2 = I. \quad (7.16)$$

They also fulfil some interesting abelian group permutation properties [174][251][186].

$$XY = jZYX = -jZ. \quad (7.17)$$

## 7.9 Hadamard gate

It is one of the fundamental gates of quantum computing that is described as follows:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (7.18)$$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (7.19)$$

$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \quad (7.20)$$

The Hadamard gate  $H$  (figure (7.6)) is one of the universal gates of quantum computing. Therefore, it meets  $HH^* = H^2 = I$ . What the Hadamard gate does, is convert a  $|0\rangle$  in  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ , Halfway between  $|0\rangle$ , and  $|1\rangle$  and converts  $|1\rangle$  in  $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ , which is also “halfway” between  $|0\rangle$ , and  $|1\rangle$ . This property is known as an overlay and will be very useful when performing the Bell EPR (entanglement). Geometrically, we can see it as a rotation on the Bolch sphere that responds to the transformation of the qubit state between the bases  $X$  and  $Z$ .



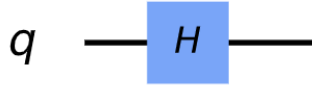


Figure 7.6: Hadamard Gate Circuit. A significant gate to get the superposition

This gate combined with the Pauli operators allows us to establish some exciting operations for quantum computing (See Fig. (7.7)).

$$HZH = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X. \quad (7.21)$$

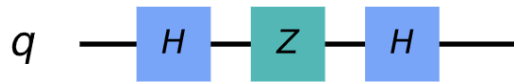


Figure 7.7: A circuit in Qiskit of the operation equivalent to  $HZH = X$

$$XHZH = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X. \quad (7.22)$$

where  $HZH = XHZH$

Generically, the Hadamard gate of  $m$  components can be described by the following equation:

$$H_m |x\rangle_{xy} = \frac{1}{2^{\frac{m}{2}}} \sum_{y \in F_2^m} (-1)^{xy} |y\rangle. \quad (7.23)$$

If we look a little, we can see that the Hadamard gate is a specific case of the Fourier quantum transform in the vector base  $F_2^m$  for  $|x\rangle = |0\rangle$ .

$$H_m |0\rangle_{xy} = \frac{1}{2^{\frac{m}{2}}} \sum_{y \in F_2^m} (-1)^{xy} |y\rangle. \quad (7.24)$$

We realize that we have an identical superposition of all the states of the orthonormal base  $F_2^m$ . This is widely used to initialize quantum systems to have the same probability distribution.

$$U = e^{i\alpha} \begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix}. \quad (7.25)$$

With  $\alpha \in \mathbb{R}$  and with  $u_i \in \mathbb{C}$  so it meets that,  $u_0 \bar{u}_0 + u_1 \bar{u}_1 = 1$ . This implies that there is only one  $\theta \in [0, \pi]$  such that  $|u_0| = \cos \frac{\theta}{2}$  and  $|u_1| = \sin \frac{\theta}{2}$ . Therefore, we can write that  $u_0 = e^{-i\lambda} \cos \frac{\theta}{2}$

and  $u_1 = -e^{i\mu} \sin \frac{\theta}{2}$ . With  $\lambda$  and  $\mu \in R$ . Rewriting our unit transformation matrix, we will have:

$$U = e^{i\alpha} \begin{bmatrix} u_0 & u_1 \\ -\bar{u}_1 & \bar{u}_0 \end{bmatrix} = \begin{bmatrix} e^{-i\lambda} \cos(\frac{\theta}{2}) & -e^{i\mu} \sin(\frac{\theta}{2}) \\ -e^{i\mu} \sin(\frac{\theta}{2}) & e^{-i\lambda} \cos(\frac{\theta}{2}) \end{bmatrix}. \quad (7.26)$$

If we note that  $\lambda = (\beta + \gamma)/2$  y  $\mu = (\beta - \gamma)/2$ ,  $\beta, \gamma \in R$ , we can arrive at the generic expression of the unit matrix as an association of the rotation matrices.

$$\begin{bmatrix} e^{-i\lambda} \cos(\frac{\theta}{2}) & -e^{i\mu} \sin(\frac{\theta}{2}) \\ -e^{i\mu} \sin(\frac{\theta}{2}) & e^{-i\lambda} \cos(\frac{\theta}{2}) \end{bmatrix} = R_z(\beta) R_y(\theta) R_z(\gamma). \quad (7.27)$$

With

$$R_z(\varphi) = \begin{bmatrix} e^{-i\varphi/2} & 0 \\ 0 & e^{i\varphi/2} \end{bmatrix} = \cos\left(\frac{\varphi}{2}\right) I_2 - i \sin\left(\frac{\varphi}{2}\right) Z = e^{-\frac{i\varphi}{2} Z}. \quad (7.28)$$

$$R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} = \cos\left(\frac{\theta}{2}\right) I_2 - i \sin\left(\frac{\theta}{2}\right) Y = e^{-\frac{i\theta}{2} Y}. \quad (7.29)$$

$$R_x(\psi) = \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) & -\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\psi}{2}\right) & \cos\left(\frac{\psi}{2}\right) \end{bmatrix} = \cos\left(\frac{\psi}{2}\right) I_2 - i \sin\left(\frac{\psi}{2}\right) X = e^{-\frac{i\psi}{2} X}. \quad (7.30)$$

To finish writing that  $U(\theta, \beta, \gamma) = R_z(\beta) R_y(\theta) R_z(\gamma)$ . We can also use the Euler decomposition of the unit transformation matrix  $U$  to arrive at the expression[174]:

$$U = e^{i\alpha} A X B X C. \quad (7.31)$$

With  $A$ ,  $B$  and  $C$  elements of the Clifford group[186][174],  $ABC = I_2$ . We can now write  $U = e^{i\alpha} R_z(\beta) R_y(\theta) R_z(\gamma)$  with:

$$A = R_z(\beta) R_y\left(\frac{\theta}{2}\right). \quad (7.32)$$

$$B = R_y\left(-\frac{\theta}{2}\right) R_z\left(-\frac{\beta + \gamma}{2}\right). \quad (7.33)$$

$$C = R_z\left(\frac{\gamma - \beta}{2}\right). \quad (7.34)$$

These operators fulfil the property of the unitary transformation and the Hermitian property, which means,  $U^*U = UU^* = I_n$ .

## 7.10 Gate $R_\phi$

Let the universal gate  $U_3$  be a simplification of the universal matrix described above, we define the rotate gate  $R_\phi$  (Fig. (7.8)) as a specific case of the universal gate  $U$ .

$$U_3(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & e^{i\lambda+i\phi} \cos(\frac{\theta}{2}) \end{bmatrix}. \quad (7.35)$$

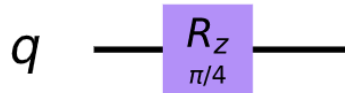


Figure 7.8:  $R_\phi$  Gate of a qubit. This gate is significant for rotational operations

$$R_\phi = U_3(\theta, \phi, \lambda) |_{\theta=\phi=0} = U_3(0, 0, \lambda) = U_1 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}. \quad (7.36)$$

## 7.11 Gate T

The gate  $T$  (Fig. (7.9)) it is equivalent to the gate  $R_\phi$  with  $\lambda = \pi/4$ . So,

$$R_{\phi|\lambda=\pi/4} = U_1 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}. \quad (7.37)$$

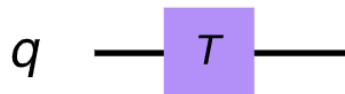


Figure 7.9:  $T$  Gate of one qubit. Remember that the inverse of this gate is not itself

## 7.12 Gate S

The gate  $S$  (Fig. (7.10)) is a particular case of the  $R_\phi$  gate with  $\phi = \pi/2$ . So,

$$R_{\phi|\phi=\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix}. \quad (7.38)$$

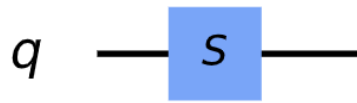


Figure 7.10:  $S$  Gate of one qubit. Remember that the inverse of this gate is not itself

We want to make two relevant comments about this gate to our path to quantum computing. First, gate  $S$  makes a quarter turn around the Bloch sphere, and its inverse is not in itself like the other quantum gates described in this work. We can see that gates  $I$ ,  $Z$ ,  $S$  and  $T$  were individual cases of gate  $R_\phi$ . Similarly, gate  $U_3$  is the most general of all single-qubit quantum gates. Each gate could be specified as  $U_3(\theta, \phi, \lambda)$  and with this, we can conclude that the single-bit quantum universality resides at gate  $U_3(\theta, \phi, \lambda)$ . Qiskit provides gates  $U_2$  and  $U_1$ , which are specific cases of gate  $U_3$  where  $\theta = \frac{\pi}{2}$  and  $\theta = \phi = 0$  respectively

$$U_3 \left( \frac{\pi}{2}, \phi, \lambda \right) = U_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -e^{i\lambda} \\ e^{i\phi} & e^{i\lambda+i\phi} \end{bmatrix} U_3(0, 0, \lambda) = U_1 = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}. \quad (7.39)$$

Before running on real IBMQ quantum hardware, all single-qubit operations are compiled on universal gates  $U_1$ ,  $U_2$ , and  $U_3$ . The universality of this quantum gate is seen from the fact that there is an infinite number of possible gates.

## 7.13 More than one qubit system.

To express a qubit system, the tensor product is used. The tensor product [225] between more than two states is denoted as  $\otimes$ . If we write a three-qubit system (Fig. (7.11)), it would be like the following case  $|011\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle$

$$|011\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} |011\rangle. \quad (7.40)$$

$$|011\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (7.41)$$

For a two-qubit system, it would be  $|01\rangle = |0\rangle \otimes |1\rangle = |0\rangle|1\rangle$ . A binary quantum gate is a unitary

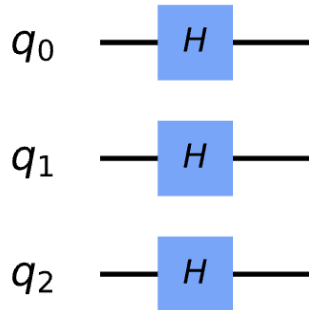


Figure 7.11: A 3-qubit system where we apply a Hadamard gate to each qubit. With this application, we assign each of the input qubits with the same probability

operation in two qubits, that is, a unitary map  $F_{22} \rightarrow F_2 \otimes F_2$  with the base  $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

$$|00\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (7.42)$$

Let  $K$  and  $L$  be qubits. Let  $C_{NOT}$  be a unitary quantum gate acting on  $L$ . The gate  $C_{NOT}$  controlled is the binary gate in  $K \otimes L$  defined by:

$$L_{C_{NOT}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes I_2 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes L. \quad (7.43)$$

In this case,  $L = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$$L_{C_{NOT}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes L = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0L & 0L \\ 0L & 1L \end{bmatrix}. \quad (7.44)$$

$$= \begin{bmatrix} 1 & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ 0 & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0 & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ 0 & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (7.45)$$

Where:

$$L_{C_{NOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (7.46)$$

What  $C_{NOT}$  gate (Fig. (7.12)) does is to map  $|00\rangle \rightarrow |00\rangle, |01\rangle \rightarrow |01\rangle, |10\rangle \rightarrow |11\rangle, |11\rangle \rightarrow |10\rangle$ . Generically,  $C_{NOT}$  is described as:

$$L_{C_{NOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q_{00} & q_{01} \\ 0 & 0 & q_{10} & q_{11} \end{bmatrix} = \begin{bmatrix} I_2 & 0_2 \\ 0_2 & X_2 \end{bmatrix}. \quad (7.47)$$

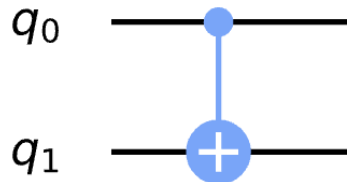


Figure 7.12: The two qubits  $C_{NOT}$  gate

The following list gives all the gates we need to create our algorithm  $U_1, U_2, U_3, C_{NOT}$  and the identity gate  $I$ . Other types of quantum computers have different native gates, such as the two-qubit atomic gate [83].

## 7.14 Toffoli gate

As we have discussed above, the Toffoli gate (Fig. (7.13)) is a fundamental gate for the construction of quantum circuits. The operation performed by the Toffoli gate is described by the relationship  $|q_0\rangle|q_1\rangle|q_2\rangle \rightarrow |q_0\rangle|q_1\rangle|q_0 \oplus q_1\rangle$ . This gate can be built using the Hadamard gate, and phase-controlled rotation gate and CNOT gates. In the figure (7.13), we observe the implementation of this gate in Qiskit.

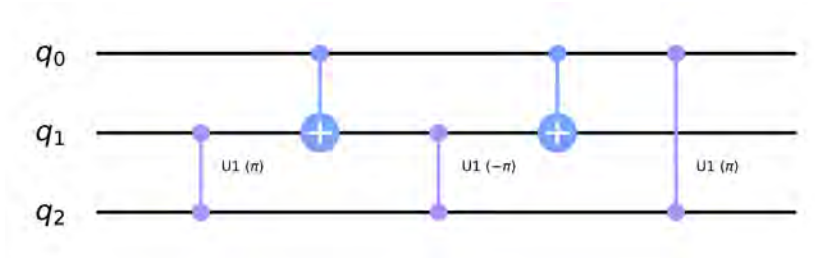


Figure 7.13: The Toffoli gate implemented with 2  $C_{NOT}$  gate, 2  $R_{\phi|\phi=\pi}$  gates and another gate  $R_{\phi|\phi=-\pi}$

## 7.15 Summary

At this point, we are ready to create complex quantum circuits. We just have to remember that if we want to create a quantum system, we can combine two or more than two systems following *postulate 2*. And, based on the works of Toffoli, Barenco and Emil Post, we can create any quantum circuit based on quantum gates fundamentals ( $C_{NOT}$ , Hadamard  $H$ , the Paulis gates  $X$ ,  $Y$  and  $Z$  and the  $R_{\phi}$  rotation gate). But before we get fully into quantum computing, we will review the state of the art of quantum computers in the next chapter.

# Chapter 8

## Quantum Computers

### 8.1 Introduction

From Feynman [162] to today, quantum computers are computers based on quantum mechanics techniques to perform calculations. These computations are based on the probability of an object's state on a complete inner-product space known as the Hilbert space[225]. The states represent logical operations using an electron/photons spin. Spin-up is assigned to logic 1 and spin-down to logic 0. These discrete states allow for digital calculation [174][166]. The quantum system with two states exploiting electron spin is a qubit. The calculations, or quantum mechanical representation, can process exponentially more data compared to classical computers. Because in quantum systems, the discrete states can exist in multiple states simultaneously. Quantum development kits – for coding in quantum assembly languages QCaaS subscription services – for use-case exploration, algorithm development, and simulation Partnerships with startups – to drive rapid innovation in QC initiatives Quantum hardware – to access quantum machines with enormous compute capacities.

One of the quantum mechanics principles used in quantum computing is superposition. The superposition propriety is when a qubit can be both 0 and 1 simultaneously. As observed, the Hilbert space is the space of complex vectors. Therefore, the qubit's superposition can be comprehended like a linear combinatory of each vector of the basis. Fundamentally, when a qubit is in a superposition of states, a logical operation applied to it will operate on both states simultaneously. Another principle of quantum computing that gives quantum computing one of the significant rewards is the entanglement [195]. This propriety is well-defined when the states of individual qubits are dependent on others.

What the companies (see Fig. (8.1)) do is mostly combine these principles (superposition and entanglement) with the foremost objective to create the core power of quantum computing [189], hence quantum parallelism. With this combination, quantum computers perform computation on all possible inputs instantaneously. This enables quantum computers to explore and design algorithms that no classical computer will ever be able to create. This is a way to define the Quantum Supremacy[163] [189][24].



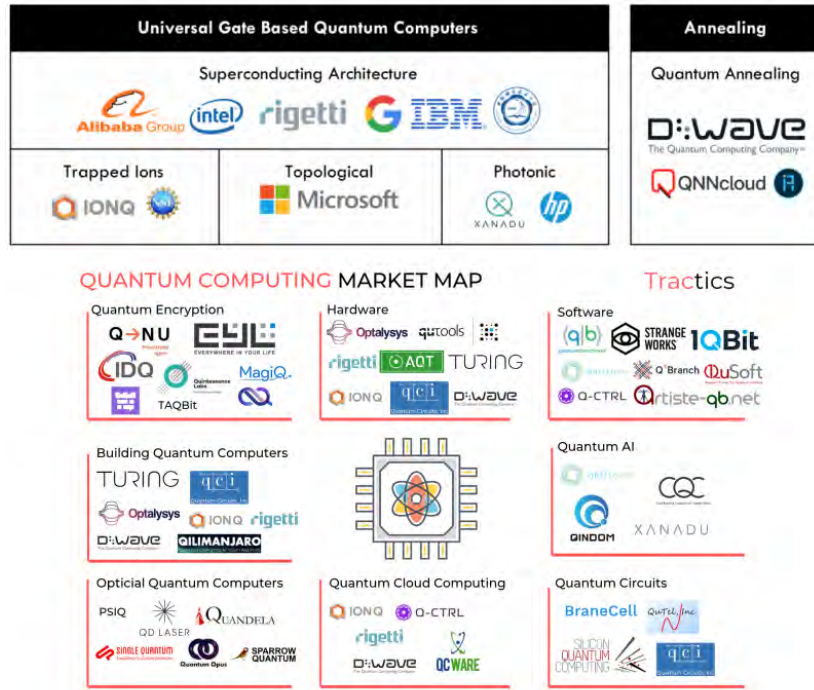


Figure 8.1: The quantum computing ecosystem [76]

## 8.2 Simulated annealing

The *Simulated annealing* (SA) is described as a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimisation in a large search space for an optimisation problem. It is often used when the search space is discrete.

It is one of the most successful and used heuristic techniques. This method is advantageous and appropriate for optimisation problems in large (considerable) search spaces to solve the issues through an exhaustive search and with certain cost functions. Simulated annealing (Fig. (8.2)) can be seen as a random walk through the solution space, where each element (quantum particles) creates a path across the optimisation horizon. A heuristic is a technique to find an approximate solution. As we already know, they are useful in a scenario where the time needed to find an exact solution can be more than considerable. The motion described by the particle but viewed uphill is described as a thermal jump. This is because Simulated annealing is a physics algorithm that mimics the behaviour of materials as they cool slowly. The "walker" is like an atom in a metal driven by temperature to reconfigure itself. These changes are random but are more likely to move to lower energy settings than higher energy configurations. That is why it is said that the "walker" follows a biased random movement.

This technique is used to model, formulate, and solve combinatorial problems.

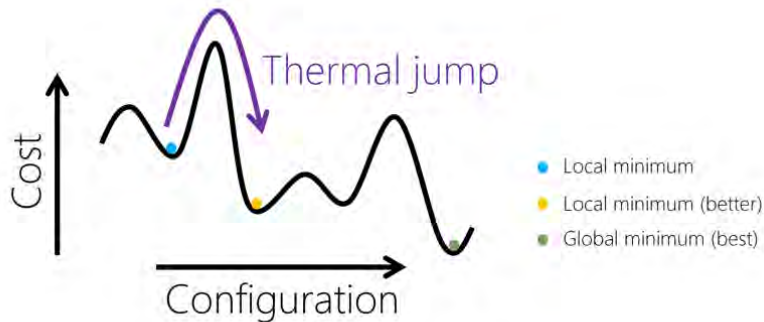


Figure 8.2: Simulated annealing concept. The motion described by the particle but viewed uphill is described as a thermal jump. This is because Simulated annealing is a physics algorithm that mimics the behaviour of materials as they cool slowly. The "walker" is like an atom in a metal driven by temperature to reconfigure itself. These changes are random but are more likely to move to lower energy settings than higher energy configurations. That is why it is said that the "walker" follows a biased random movement.

### 8.3 Quantum annealing

The *Quantum annealing* (QA) is described as *a metaheuristic for finding the global minimum of a given objective function over a given set of candidate solutions that are stated by a process using quantum decoherence.*

Quantum annealing (Fig. (8.3)) copies the simulated annealing philosophy by introducing a few small changes. In simulated annealing, the solution space is explored by doing thermal jumps from one solution to the next. But in quantum annealing, what you do is use the quantum effect called the quantum tunnel [195], which allows us to travel through these energy barriers. It is the same effect that gives life to tunnel diodes.

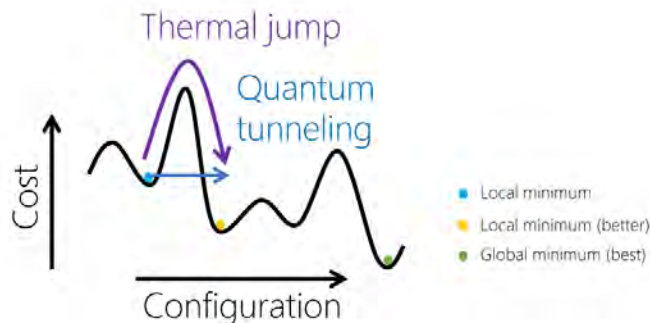


Figure 8.3: Advantage of quantum tunnelling over thermal jump taking advantage of the Simulated Annealing concept

### 8.3.1 Quantum Tunneling

The *Quantum tunnelling* is described as *the quantum mechanical phenomenon where a wavefunction can propagate through a potential barrier*.

To understand this characteristic of Quantum Tunneling, we can analyse the basic model to study the qualitative behaviour of a quantum system. In this model, we have the potential barrier seen in Figure (8.4) with height  $V_0$  And width  $d = x_2 - x_1$ . Let us consider that a particle of mass  $m$  with

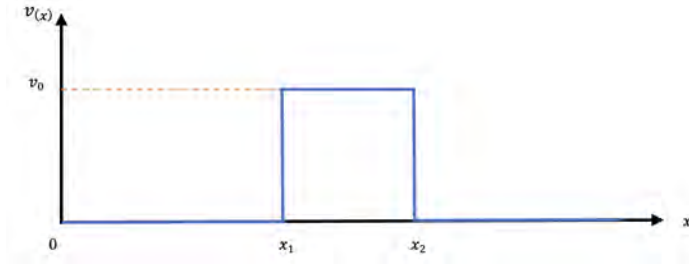


Figure 8.4: one-dimensional potential barrier

energy  $E < V_0$  passes through this potential energy, and we want to know the probability of transition of the particle  $P \equiv \frac{|\psi_b|^2}{|\psi_a|^2}$  where  $\psi_a$  and  $\psi_b$  are the functions waveform on the left and right side of the barrier. Therefore, it is necessary to use the equation of motion of quantum systems, the Schrödinger equation, whose general form is:

$$H\psi = i\hbar \frac{\partial \psi}{\partial t}. \quad (8.1)$$

In the case of the Figure (8.4), the Hamiltonian is equal to

$$H = \frac{\hbar^2}{2m} \frac{d^2 \psi}{dx^2} + V_0 [\theta(x_1 - x) - \theta(x_2 - x)] \psi. \quad (8.2)$$

Now, applying the Schrödinger equation, we arrive at (8.3):

$$H = \frac{\hbar^2}{2m} \frac{d^2 \psi}{dx^2} + V_0 [\theta(x_1 - x) - \theta(x_2 - x)] \psi = i\hbar \frac{\partial \psi}{\partial t}. \quad (8.3)$$

The particle will behave like a flat wave on the left side of the barrier, remaining as  $\psi_a = Ae^{-ikx} + Be^{ikx}$  with  $k = \sqrt{\frac{2mE}{\hbar^2}}$ . On the right side of the barrier, the particle is also a free particle  $\psi_b = Ce^{-ikx} + De^{ikx}$ . Using normalization and continuity conditions throughout the space, the probability of transition would be:

$$P = \frac{1}{1 + \frac{V_0^2 \sin^2 h^2 \Delta}{4E(V_0 - E)}}. \quad (8.4)$$

With

$$\Delta \equiv d \sqrt{\frac{2m(V_0 - E)}{\hbar^2}}. \quad (8.5)$$

If we now apply the limit to  $P$ , taking into account that  $V_0$  constant and we increase  $d$ , the transition probability becomes smaller and, in the limit,  $\lim_{d \rightarrow \infty} P = 0$ . In Quantum annealing, the problem to be solved is modelled by mapping qubits, which are the variables. The energy of an allocation/mapping given to qubits is the value of the cost function that has a close relationship with the Hamiltonian. How does it develop? Initially, the system begins with the quantum state in a full overlay over many possible qubit assignments. Then, the quantum field strength is varied instead of varying the temperature, as we did in simulated annealing. The intensity of the quantum field is a parameter that defines the radius of the neighbouring states to which we can move. As time goes by and we get closer to a solution, this radius gets smaller and smaller. At the end of the annealing process, the system is established in a particularly low energy configuration (ground state) that can then be measured, giving us the optimisation problem solution. The cleanest mathematical formulation of quantum annealing is known for adiabatic quantum optimisation, and this is what almost all quantum optimisation methods attempt to emulate.

Adiabatic Quantum Computing (AQC)[163] uses a concept of quantum physics known as the adiabatic theorem[148]. The process followed by the AQC can be summarised in two very recognisable steps:

1. The first step is to prepare a system and initialise it to its lowest energy state, known as the ground state.
2. The second step allows us to transform/map our problem in the system.

The adiabatic theorem states that as long as this transformation occurs slow enough, the system has time to adapt and will remain in the lowest energy configuration. When we are done with our transformations, we will find solutions.

To use the adiabatic theorem for computation, researchers H. Nishimori and T. Kadowaki demonstrated that it is necessary to use Ising's transversal model to code the problem to be optimised and activate it slowly. The Hamiltonian illustrating this is as follows,

$$H_{Ising} = -A(t) \left( \sum_{i < j} J_{ij} Z_i Z_j + \sum_i h_i Z_i \right) + B(t) \sum_i h_i X_i. \quad (8.6)$$

With  $Z_i$  the Pauli matrix in  $Z$ . In many books, we can see instead of  $Z_i$ ,  $\sigma_z$  and instead of  $X_i$ ,  $\sigma_x$  or simply  $\sigma_i$   $\sigma_x$ ,  $\sigma_y$ ,  $\sigma_z$  generally representing the Pauli matrices or operators in  $X$ ,  $Y$  y  $Z$ .

Taking into account that:

$$A(t)|_{t=0} = 0 \quad \text{and} \quad B(0)|_{t=0} = 1. \quad (8.7)$$

With  $X_i$  the Pauli matrix in  $X$  (generically  $\sigma_x$ ). As mentioned above, if we want the system to be in the ground state, the initial condition must be

$$|\psi(t)\rangle|_{t=0} = |+_1\rangle_x \otimes |+_2\rangle_x \otimes \dots \otimes |_{+n-1}\rangle_x \otimes |+_n\rangle_x. \quad (8.8)$$

With  $|+_i\rangle_x$  is the proper state of  $\sigma_i^x$ . In some books or articles, we can see the following notification  $|+\rangle^{\otimes N}$  with  $N$  is the number of qubits.

Now suppose that quantum annealing ends at  $T \gg 1$ , where  $(t)|_{t=T} = 1$  and  $B(t)|_{t=T} = 0$ . The adiabatic theorem ensures that  $|\psi(t)\rangle|_{t=T}$  will be the ground state of  $H(t)$  such that the energy of the system will be the energy destined for the ground state. This is the basis of adiabatic quantum computing. There are several derivatives of this technique, but it is not in line with this research.

## 8.4 Quantum Computers

The *Quantum computers* can be defined as *computers that use the quantum mechanic's properties to store data and perform computations that can be extremely worthwhile for certain tasks. As a result, they could vastly outperform even our best supercomputers and classical computer.*

There are several techniques for building a quantum computer, and the way it is currently done is by combining multiple various multicore processors [46]. All of this brings to life numerous models of quantum computing. Theoretical models, quantum circuit models, adiabatic quantum computing, measurement-based quantum computing, and topological quantum computing, are equivalent to each other within the reduction of polynomial-time. The most widespread and considerably developed model is the circuit model for gate-based quantum computation. As discussed in the chapter on quantum gates, the conceptual generalisation of Boolean logic gates (AND, OR, NOT, NAND, etc.) is used for classical computing works for quantum computing (Emil Post). With the combination of these basic gates and the appropriate memory structures based on architecture, it gives life to the quantum computer.

Quantum annealing has a somewhat different software stack structure than gate-based model quantum computers. The annealing-based computer must be viewed as a specific case of a quantum accelerator based on quantum gate algorithms. So instead of a quantum circuit, as in the case of IBM and company, the level of abstraction is Ising's classic model. Let's take back everything we've already talked about solving problems with the Ising model. Like gate model superconducting quantum computers, superconducting quantum annealers also suffer from limited connectivity. This means we have to find a smaller embedded graph, combining several physical qubits into one logical qubit. Since this operation is already NP-Hard, it is necessary to use probabilistic heuristics.

Below, from Figure (8.5) to (8.9), we are presented with examples of Full-stack quantum computer architectures. To see the architecture being followed for the construction of the quantum independently of the Quantum Processing Unit (QPU). The QPU, also known as a quantum chip, is a physical chip (manufactured by players based on the techniques discussed above) that contains several interconnected qubits. It is the fundamental component of a full quantum computer, which includes the "motherboard\*" environment for the QPU, the control electronics, and many other components. Currently, there are two dominant configurations for quantum computing. Continuous-Time Quantum Calculus (D-Wave), in which problems are coded in quantum Hamiltonians and the natural dynamics of physical systems, and the Gate Model Quantum Calculus (IBM) [163][174] [225], in which the calculation is made through a series of discrete gate operations.

In continuous-time quantum computation, optimisation is achieved by mapping the Hamiltonian optimisation problem of a controllable quantum system so that the low-energy states correspond to optimal solutions. The quantum superconducting circuit analysers produced by D-Wave Systems Inc are the most mature[163][189].

A D-Wave machine is a physical representation of the Ising model and, as such, has a Hamiltonian "problem" of the form of the equation (6.8).

Returning to our study, the main problem to solve is assigned to the previous Hamiltonian. The system begins with the Hamiltonian:  $H_{initial} = \sum_i h_i Z_i$  and the annealing parameter,  $s$  is used to assign the initial Hamiltonian  $H_{initial}$  to the Hamiltonian problem  $H_{ising}$  using  $H(s) = (1-s)H_{initial} + sH_{ising}$ .

The process is done slow enough based on the annealing theorem to stay close to the system's ground state. At the same time, the Hamiltonian varies according to the problem, using tunnels to remain close to the ground state.

---

\*In this case, to be a purist, it does not make sense to speak of the motherboard as in classic computers, but of housing thinking of a datacenter. Since today's quantum computers live in exceptional houses, another approach would be to think of a future with possible quantum laptops.

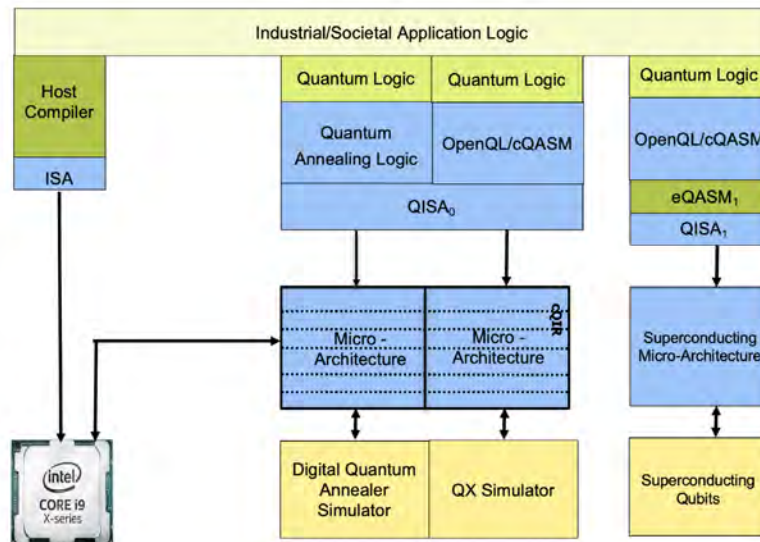


Figure 8.5: Full-Stack execution [46]

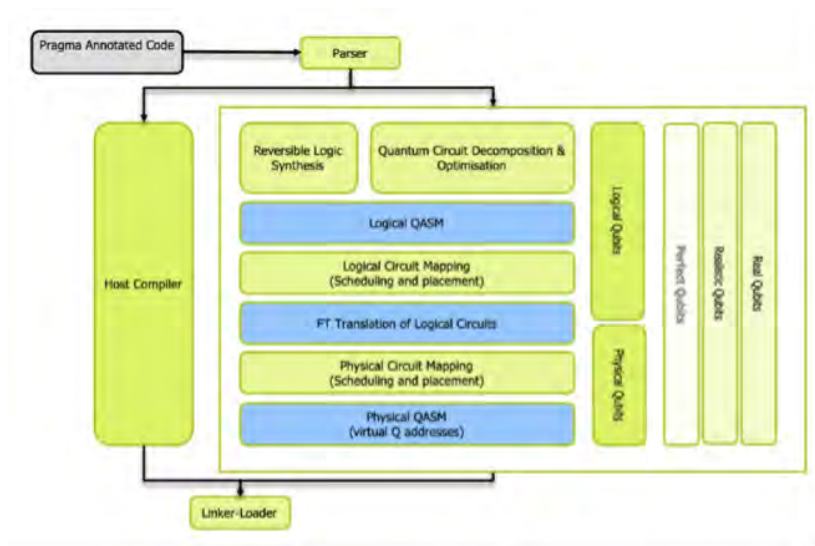


Figure 8.6: compiler infrastructure [46]

For quantum gate-based computers, one of the most promising algorithms for optimisation is the one known by Ansatz Alternative Quantum Operator, also known as the Approximate Quantum Optimization Algorithm (QAOA) [112] [189][183][177][39]. QAOA is exclusively designed to run in polynomial time on NISQ devices [189] and find optimal solutions for optimisation problems. This algorithm is used to solve critical optimisation problems that classically require exponential computational complexity to find the optimal solution exactly. Although, in principle, QAOA could be considered a gate model or a continuous-time configuration [112] [189].

Based on the work of Andrew Lucas, Ising formulations of many NP problems [156], any problem in NP can be assigned to an NP-hard in polynomial time, and integer factorisation is, in fact, an



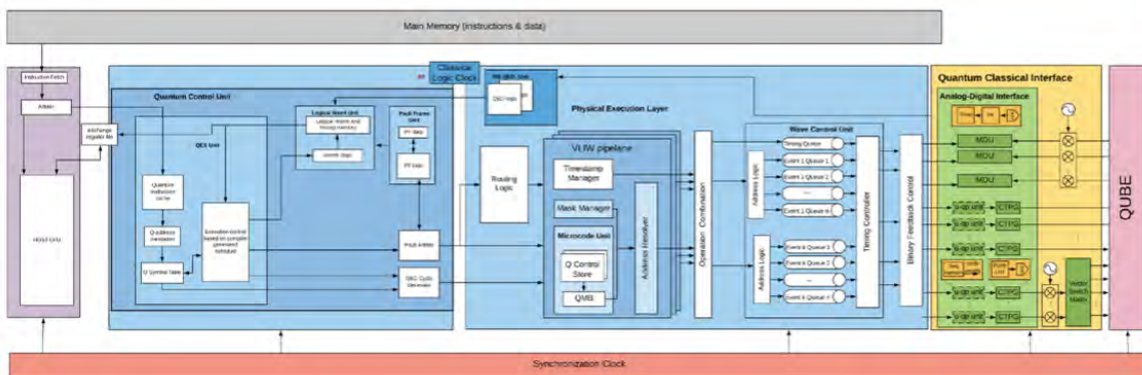


Figure 8.7: An example of general-purpose quantum microarchitecture [46]

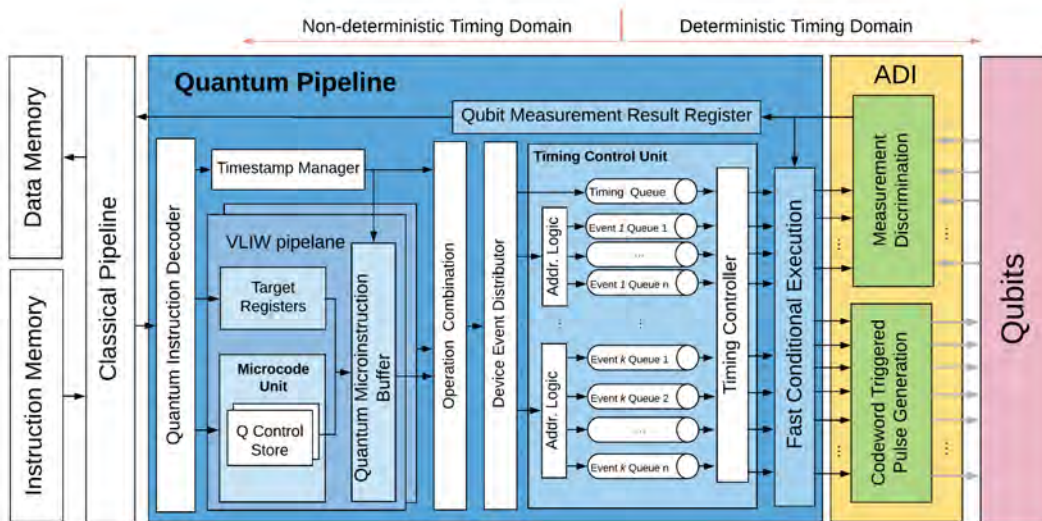


Figure 8.8: Example of experimental implementation of microarchitecture for (real) superconducting qubits [46]

NP problem. Ising's Hamiltonian is a quadratic function that corresponds to a binary optimisation problem without quadratic constraints (QUBO), being in NP-hard. A universal Quantum Computing can solve the Hamiltonian of the same problem in polynomial time. The thing that a Quantum Annealing cannot do, since it cannot simulate a universal Quantum Computing in polynomial time[13].

Gate-based quantum algorithms are designed, so that solution states interfere constructively. In contrast, non-solutions interfere destructively by skewing the final probability distribution to measure solutions. However, the error rates are still around  $10^{-2}$  and  $10^{-3}$  and need to be substantially improved.

Adiabatic quantum computing (AQC) was the first quantum computation model to solve combinatorial optimisation problems. Unlike the gate, based on the quantum computation model, it was based on the adiabatic theorem of quantum mechanics. In this model, to perform any calculation, we need two Hamiltonians called  $H_{mixer}$  and  $H_{cost}$ . Among them, the fundamental state of  $H_{mixer}$  must be easily prepared, like  $|+\rangle^{\otimes N}$  and the ground state of  $H_{cost}$  encode the solution to our problem.

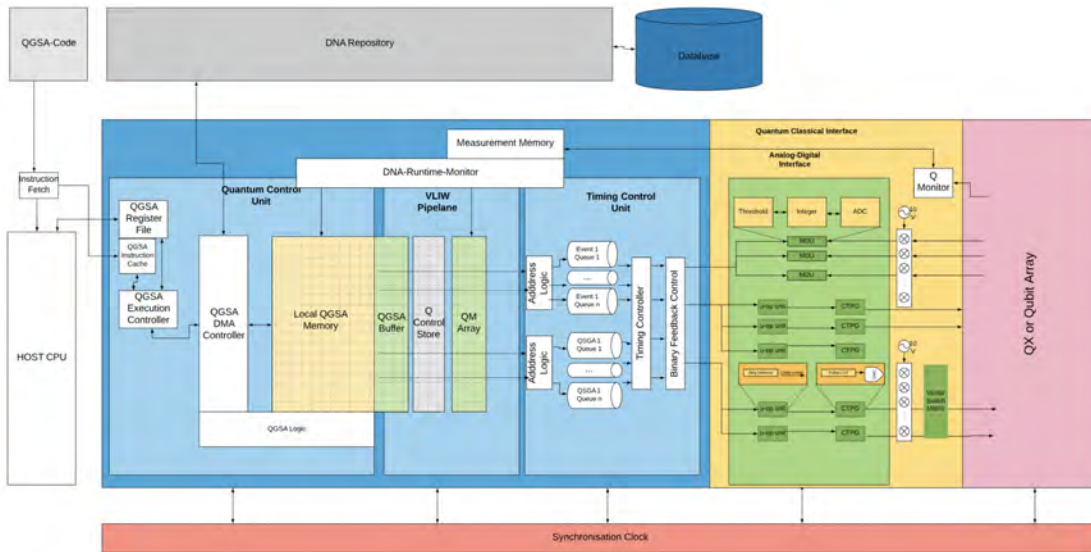


Figure 8.9: Example of the new microarchitecture for the quantum genome sequencing accelerator [46]

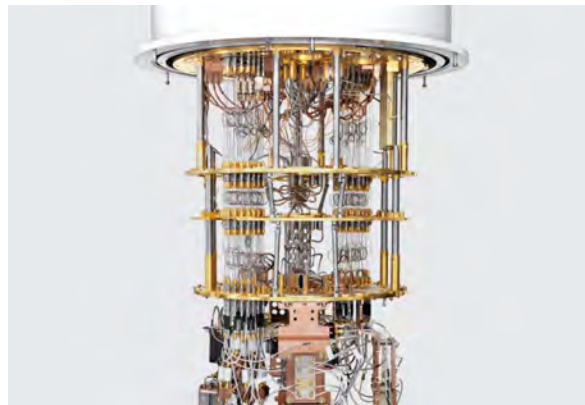


Figure 8.10: Microsoft Quantum Computer [105].

### 8.4.1 Quantum gate-based computers

In this section, we will present the leading companies that follow the career of the construction of the quantum computer based on gates.

IBM (Fig. (8.11)) is one of the most advanced technology companies in quantum computing with cloud computing; IBM Quantum Experience (IBMQ)<sup>†</sup>. To date, there are more than 250,000 users who have tried the IBMQ service<sup>‡</sup>. In addition, more than 100 companies are paying for its service, using IBM hardware, getting advice from the company’s experts, and counting on all kinds of services.

Currently, IBM has 15 New York-based quantum cloud computers. The objective is to ensure that each country can have one, but while waiting for agreements with the different governments, IBM will

<sup>†</sup><https://quantum-computing.ibm.com/>

<sup>‡</sup>Private data from slack and IBMQ profiles





Figure 8.11: IBM Quantum Computer [18].

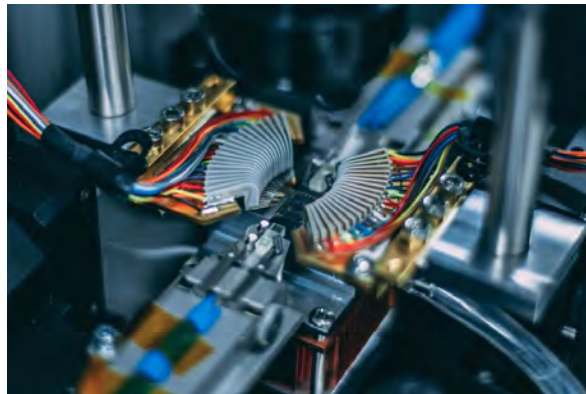


Figure 8.12: Xanadu Quantum Computer [54].



Figure 8.13: Inside D-Wave Quantum Computer [104]

begin its distribution in Germany and Japan.

Recently, IBM offered a quantum processor of up to 53 qubits. On the other hand:

Intel is analysing semiconductor and superconductor qubits, but fundamentally they are more interested in the qubit semiconductor processor. The essence is to focus on qubit production, partly backed by a solid microarchitecture. Microsoft (Fig. (8.10)) has some preference for the Majorana-based approach [138], but they have yet to do the first qubit based on that quasi-particle. Nevertheless, they are very active in software development. Alibaba is betting very heavily in the field of quantum

computing. It has a quantum lab that focuses on a range of activities ranging from developing a quantum processor and classic quantum algorithms to simulating quantum physics[14]. Google is also one of the leaders in superconducting qubits. The team led by John Martinis has developed the quantum computer Sycamore<sup>§</sup> of 54 qubits and a specific algorithm to show quantum supremacy (the algorithm generates sequences of random numbers and has no relevant practical applications) [106]. Rigetti is a startup company focused on the superconducting quantum processor. They are progressing well, but there is still no applicable processor on the market, even though a processor can be used for some testing purposes. Rigetti's Aspen-7-25Q-B Quantum Processing Unit (QPU) has 25 qubits with 24 programmable two-qubit gate interactions. The company released its last 31 qubits Aspen -8 QPU on May 5, 2020 [110][109]. Xanadu (Fig. (8.12)) focuses on continuous-variable quantum computing based on squeezed light photonics[45]. A slightly different model compared to qubits. The essential elements of its photonic system are qumodes, each of which can be represented by superimposing on different numbers of photons [42][127]. Xanadu designs and incorporates photonic quantum silicon chips into existing hardware to create a complete quantum computing experience.

Following the race to get a quantum computer, Intel researchers (qHiPSTER<sup>¶</sup>), MIT and the University of Toronto are addressing the question of how the performance of next-generation quantum devices submodules can be designed and tested using existing quantum computers. Furthermore, the methods drawn with the researchers open a new path to the design of quantum processors, candidates when the demands to calculate the properties of the submodule exceed the capabilities of classical computing resources. All these steps lead us to the well-known Full-stack [169]. With the

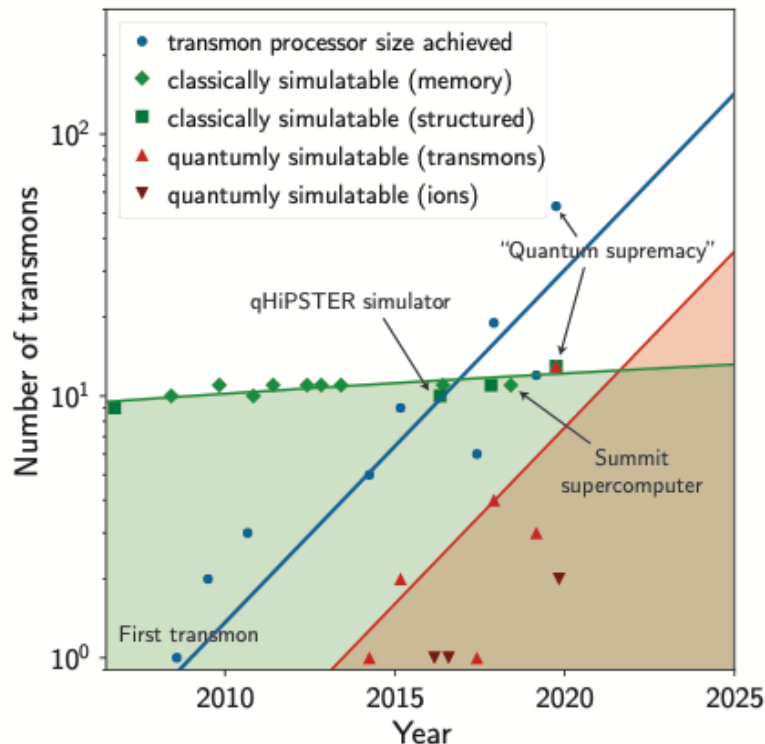


Figure 8.14: The challenge of simulating transmon quantum processors [143]

increasing size of quantum processors, the submodules that make up the processor will be too large to

<sup>§</sup><https://www.nytimes.com/2019/10/30/opinion/google-quantum-computer-sycamore.html>

<sup>¶</sup>Intel Quantum Simulator

simulate a classic computer accurately. Therefore, one will soon have to fabricate and test each new design primitive and parameter choice in time-consuming coordination between design, fabrication, and experimental validation.

### 8.4.2 Quantum annealing-based computers

This section will present the leading companies that follow the career of the construction of the quantum computer based on annealing.

D-Wave (Fig. (8.13)) is the pioneer company in the construction of quantum computers during this last decade. In 2018, its technology reached up to 2000 superconducting qubits. D-Wave Systems' commitment is in quantum hardware. However, it also offers a programming environment Ocean<sup>||</sup>, for basic operations compared with Qiskit or PennyLane. Fujitsu has invested in the development of digital annealing. Therefore, its bet is not entirely a quantum computer, but it is used to solve QUBO problems through simulation[201]. Hitachi also specialises in making a quantum annealing-based quantum accelerator using semiconductor qubits. Its bet<sup>\*\*</sup> is similar to Fujitsu. Seen from

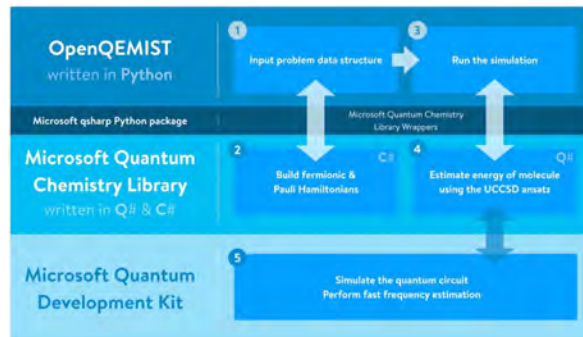


Figure 8.15: Microsoft Quantum Development Kit. Provides an end-to-end scalable quantum development environment and leverages the Q# quantum programming language, which enables users to design, compile, and simulate quantum algorithms [167].

the outside, 1QBit is a software company that identifies insoluble industrial problems and creates the necessary software to take advantage of the best classical and quantum hardware technologies to solve them. 1QBit is instead focused on computational finance, materials science, quantum chemistry, and life sciences. Its algorithms serve both gate quantum computing and quantum annealing. Its framework, 1Qloud, focuses on optimisation issues that map to QUBO for quantum annealing computers and similar devices from D-Wave. Fujitsu, in contrast, its QEMIST<sup>††</sup> platform focuses on advanced materials and quantum chemistry research, with universal quantum computing processors.

We can observe from the figure (8.14) the challenge of simulating transmon quantum processors. We can also analyse Microsoft's framework that offers an end-to-end scalable environment from the figure (8.15).

<sup>||</sup> <https://ocean.dwavesys.com/>

<sup>\*\*</sup> [https://www.hitachi.com/rd/sc/story/cmos\\_annealing2/index.html](https://www.hitachi.com/rd/sc/story/cmos_annealing2/index.html)

<sup>††</sup> <https://1qbit.com/qemist/>

## 8.5 summary

This chapter has analysed and defined the state of the art of quantum computers and the companies creating such computers. We have seen that to date, and there are two potential techniques; Quantum Annealing and Quantum Gate-Based. Furthermore, we have revised several strategies for building a quantum computer and are currently done by combining multiple multicore processors. Finally, we also focus on the challenge of simulating transmon quantum processors.

We can focus on research since we have the necessary ingredients to experiment and implement a quantum algorithm for the combinatorial optimisation problem.



# Chapter 9

## Quantum Computing

### 9.1 Introduction

In this chapter, we will experiment with quantum computing, mainly on Qiskit (Fig. (10.12)). We will explore variational techniques and some of their libraries to do this.

#### 9.1.1 Quadratic Unconstrained Binary Optimization Problems

The *Quadratic Unconstrained Binary Optimization*, commonly called QUBO, is known as *a combinatorial optimization problem with many applications, from finance and economics to machine learning*.

QUBO problems are traditionally used in computer science. True and False variables, states that correspond to 1 and 0 values. A QUBO problem is defined using an upper-diagonal matrix  $Q$ , which is an  $N \times N$  upper-triangular matrix of real weights, and  $x$ , a vector of binary variables, as minimizing the function.

$$H(x)_{\text{QUBO}} = \sum_i \sum_{j < i}^N Q_{ij} x_i x_j + \sum_i^N Q_i x_i. \quad (9.1)$$

Where the diagonal terms  $Q_{ij}$  are the linear coefficients and the nonzero off-diagonal terms are the quadratic coefficients  $Q_{ij}$ . The QUBO is unconstrained in that there are no constraints on the variables other than those expressed in  $Q$ . The diagonal entries of  $Q$  are the linear coefficients that bias the qubits. The nonzero off-diagonal terms are the quadratic coefficients that define the strength of the coupling between variables.

The input may be full or sparse. Both upper- and lower-triangular values can be used;  $(i, j)$  and  $(j, i)$  entries are added together. An exception is raised if a  $Q$  value is given to a coupler not present. Only entries indexed by working couplers may be nonzero. This concept is so powerful that we will use it in our formulation, taking into account that with the docplex, we can't formulate inequation constraints.

The QUBO is usually represented more concisely as equation (9.2)

$$\min_{x \in \{0,1\}^n} x^T Q x. \quad (9.2)$$

We can also find the Hamiltonian of the objective function expressed as an in the scalar notation of the QUBO form as follows:

$$H(a_i, b_{ij}; q_i)_{\text{QUBO}} = \sum_i^N \sum_{j < i} b_{ij} q_i q_j + \sum_i^N a_i q_i. \quad (9.3)$$

This form can be easily found in the D-Wave formulation. The transformation between Ising and QUBO is  $s = 2x - 1$ . Let demonstrate that QUBO and the Ising of the Hamiltonian are similar. This means that we will have the algorithm in Ising form by writing an algorithm for QUBO with a single variable change. That is very useful to have the algorithm for various platforms based on quantum gates or quantum annealing.

$$H(s) = \sum_i^N \sum_{j < i} J_{ij} s_i s_j + \sum_i^N h_i s_i. \quad (9.4)$$

With  $s_i$  and  $s_j \in \{-1, 1\}$ . For the translation in QUBO formulation, let's consider  $x_i$  and  $x_j \in \{0, 1\}$  and using the spin relation  $s = 2x - 1$ , we can remap the Hamiltonian as follow:

$$\begin{aligned} H(s) &= \sum_i^N \sum_{j < i} J_{ij} (2x_i - 1)(2x_j - 1) + \sum_i^N h_i (2x_i - 1) \\ &= \sum_i^N \sum_{j < i} J_{ij} (4x_i x_j - 2x_i - 2x_j + 1) + \sum_i^N (2h_i x_i - h_i). \end{aligned} \quad (9.5)$$

Regrouping all the constants into  $C_0$  we have the following expression:

$$H(s) = \sum_i^N \sum_{j < i} 4J_{ij} x_i x_j - \sum_i^N x_i \sum_{j < i} 2J_{ij} - \sum_i^N \sum_{j < i} 2J_{ij} x_j + \sum_i^N 2h_i x_i - C_0. \quad (9.6)$$

By grouping  $x_i$  terms together, we can write,  $a_i = \sum_{j < i}^N 2J_{ij} + 2h_i$ , so:

$$H(s) = \sum_i^N \sum_{j < i} 4J_{ij} x_i x_j - \sum_i^N a_i x_i - 2 \sum_i^N \sum_{j < i} J_{ij} x_j + C_0. \quad (9.7)$$

Let us develop the term  $\sum_i^N \sum_{j < i} J_{ij} x_j$  in term of  $x_i$ .

$$\begin{aligned} \sum_i^N \sum_{j < i} J_{ij} x_j &= J_{10} x_0 + J_{20} x_0 + J_{21} x_1 + J_{30} x_0 + J_{31} x_1 + J_{32} x_2 + \dots \\ &+ J_{N0} x_0 + J_{N1} x_1 + J_{N2} x_2 + J_{N2} x_3 + J_{N,N-1} x_{N-1}. \end{aligned} \quad (9.8)$$

We can observe that each column has a term:

$$(J_{10} + J_{20} + J_{30} + \dots + J_{N0}) x_0 + (J_{21} + J_{31} + \dots + J_{N1}) x_1 + \dots + (J_{N,N-1}) x_{N-1}. \quad (9.9)$$

Terms				
Problem Expression	Linear Coefficient	Quadratic Coefficient	Variable	States
QUBO (scalar)	$a_i$	$a_{i,j}$	$q_i$	$\{0, 1\}$
QUBO (matrix)	$Q_{i,j}$	$Q_{i,j}$	$x_i$	$\{0, 1\}$
Ising	$h_i$	$J_{i,j}$	$s_i$	$\{-1, 1\}$

Table 9.1: Comparison between Ising and QUBO representation and related terminology.

Let's cast the  $J_{ij}$  as some constants  $b_i$ , so:

$$b_0x_0 + b_1x_1 + b_2x_2 + \dots + b_{N-1}x_{N-1}. \quad (9.10)$$

Since there is no  $x_N$  in the original sum, so  $b_N = 0$ . So we can write that:

$$\sum_i^N \sum_{j<i} J_{ij}x_j = \sum_i^N b_i x_i. \quad (9.11)$$

So, our Hamiltonian can be written as:

$$H(s) = \sum_i^N \sum_{j<i} 4J_{ij}x_i x_j - \sum_i^N a_i x_i - 2 \sum_i^N b_i x_i + C_0. \quad (9.12)$$

Let  $J'_{ij} = 4J_{ij}$  and  $h'_i = a_i - 2b_i$ , therefore,

$$H(s) = \sum_i^N \sum_{j<i} J'_{ij}x_i x_j - \sum_i^N h'_i x_i + C_0 \quad (9.13)$$

Where we can experiment that Ising  $H(s)$  and QUBO  $H(x)$  are similar in form and relations; they are isomorphic.

To convert the coefficients from QUBO to Ising:

$$\begin{aligned} J_{ij} &= \frac{1}{4}Q_{ij} \\ h_i &= \frac{1}{2}Q_{ii} + \frac{1}{4} \sum_{i<j} Q_{ij}. \end{aligned} \quad (9.14)$$

Or from Ising to QUBO:

$$\begin{aligned} Q_{ij} &= 4J_{ij} \\ Q_{ii} &= 2h_i - \frac{1}{2} \sum_{i<j} Q_{ij}. \end{aligned} \quad (9.15)$$

## 9.2 Variational calculation

The *variational method* is defined as *the way to find approximations to the lowest energy eigenstate or ground state and some excited states. It allows calculating approximate wavefunctions such as molecular orbitals. The basis for this method is the variational principle.*



The *variational calculation* [224] is the basis of the variational principle [119]. We can say that the variational calculus consists of looking for maximums and minimums or extensively looking for relative ends of a function of functions (functional) over a space of functions. This calculation can be seen as a generalization of the elementary calculus of a variable's maximum and minimum real functions.

Mathematically speaking, when we talk about optimization, we are talking in some way to find the maximum or minimum of the function that models our scenario; Our objective function. That is, calculate the minimum or maximum of our objective function. Although it seems easy, in many cases, the calculation of the minimum or ceiling is not entirely trivial because of the structure of the data, the size of the data or basically or for the computational cost required to make this calculation makes it non-trivial. The computational cost [192] is one of the limits of all scientific advances. For that same reason, the scientific community is working to equip itself with machines that can give it the most significant computational capacity [82].

Several branches defined and designed alternatives in solving optimization problems by calculating variations. One of the most contemplated approaches is from Richard Bellman [207], who developed dynamic programming [207, 35] with apparent alternatives to the calculation of variations.

The work of Sturm-Liouville [206] and Rayleigh-Ritz method [198], are the basis of the Variational Quantum Eigensolver; the VQE. A dynamic control system that allows us to make the variational calculation of a quantum state  $|\psi(\theta)\rangle$  associated with its expectation value  $H$ .

Let  $|\psi_i\rangle$  be an eigenvector, of a matrix  $A$  which is invariant under transformation by  $A$  up to a scalar multiplicative constant (the eigenvalue  $\lambda_i$ . That is  $A|\psi_i\rangle = \lambda_i|\psi_i\rangle$ . If we define the Hamiltonian  $H$  a matrix that is Hermitian,

$$H = \sum_{i=1}^N \lambda_i |\psi_i\rangle \langle \psi_i|. \quad (9.16)$$

where each  $\lambda_i$  is the eigenvalue corresponding to the eigenvector  $|\psi_i\rangle$ . Furthermore, the expectation value of the observable on  $H$  on an arbitrary quantum state  $|\psi\rangle$  is given by:

$$\langle H \rangle = \langle \psi(\theta) | H | \psi(\theta) \rangle. \quad (9.17)$$

Substituting the value of  $H$  and using the Hermitian property in the equation,

$$\begin{aligned} \langle H \rangle &= \langle \psi(\theta) | \left( \sum_{i=1}^N \lambda_i |\psi_i\rangle \langle \psi_i| \right) | \psi(\theta) \rangle = \left( \sum_{i=1}^N \lambda_i \langle \psi(\theta) | \psi_i \rangle \langle \psi_i | \psi(\theta) \rangle \right) \\ &= \sum_{i=1}^N \lambda_i |\langle \psi_i | \psi(\theta) \rangle|^2 \langle H \rangle = \sum_{i=1}^N \lambda_i |\langle \psi_i | \psi(\theta) \rangle|^2. \end{aligned} \quad (9.18)$$

This  $\langle H \rangle = \sum_{i=1}^N \lambda_i |\langle \psi_i | \psi(\theta) \rangle|^2$  demonstrates that the expectation value of an observable on any state can be expressed as a linear combination using the eigenvalues associated with  $H$  as the weights. Moreover, each of the weights in the linear combination is greater than or equal to 0, as  $|\langle \psi_i | \psi(\theta) \rangle|^2 \geq 0$  and so it is clear that  $\langle H \rangle_{\psi(\vec{\theta})} \geq \lambda_{\min}$ .

This is very powerful; it implies that the expectation value of any wave function will always be at least the minimum eigenvalue associated with a given  $H$ . Moreover, the expectation value of the state  $|\psi_{\min}\rangle$  is provided by  $\langle H \rangle_{\psi_{\min}(\vec{\theta})} = \lambda_{\min} = E_{gs}$ . This is the base of the VQE, where  $E_{gs}$  is the ground state energy of that system related to the Hamiltonian  $H$ .

A fundamental and relevant concept is knowing that a fixed variational form with a polynomial number of parameters can only generate transformations to a polynomially dimensioned subspace of

all states in a Hilbert space of exponential size [174]. This is very important for all algorithms based on variational calculation. It is also imperative to mention that the ability to generate an arbitrary state ensures that during the optimization process, the variational form does not limit the set of achievable states on which the expected value can be taken. Usually, this limitation is given by the classical algorithm. This leads us to see that there are several variational forms. Some, like  $Ry$  and  $RyRz$ , are heuristically designed, regardless of the target domain.

While building a variational form, we must balance two different objectives.

Normally, a variational form  $n$  qubits could generate any possible state  $|\psi\rangle$  where  $|\psi\rangle \in C$  and  $N = 2^n$ . Nevertheless, we would like the variational form to use as few variables/parameters as possible. This is the contradiction. We must have expertise when creating the ideal variational form of our circuit.

In the experimentation chapter, we will work intuitively until we reach our algorithm, fulfilling the two mentioned objectives.

### 9.2.1 Variational Quantum Eigensolver

The *Variational Quantum Eigensolver*, most known as VQE, is a *flagship algorithm for quantum chemistry using near-term quantum computers. It is an application of the Ritz variational principle, where a quantum computer is trained to prepare the ground state of a given molecule.*

Unfortunately, we're still in the Noisy Intermediate-Scale Quantum (NISQ) [189] era because we don't have a perfect quantum computer yet. To compensate for the fact that quantum isn't perfect yet, researchers started developing algorithms that work both quantum and classical to solve problems. The VQE[183][117][89] is based on the variational principle, which is dynamic programming. Using the VQE, we can make smart and adaptive algorithms (Figures (9.1) and (9.2)). This area is known as Quantum Machine Learning (QML) [48] and one of the warmest QML algorithms nowadays is the Variational Quantum Eigensolver. This is because its applications range from finance, biology, scheduling and chemistry. One of the essential characteristics of molecules is their ground state energy. The ground state energy is just the lowest possible energy state that a molecule can be in. The ground state energy of a molecule is vital because it gives us more information about the electron configuration of that molecule.

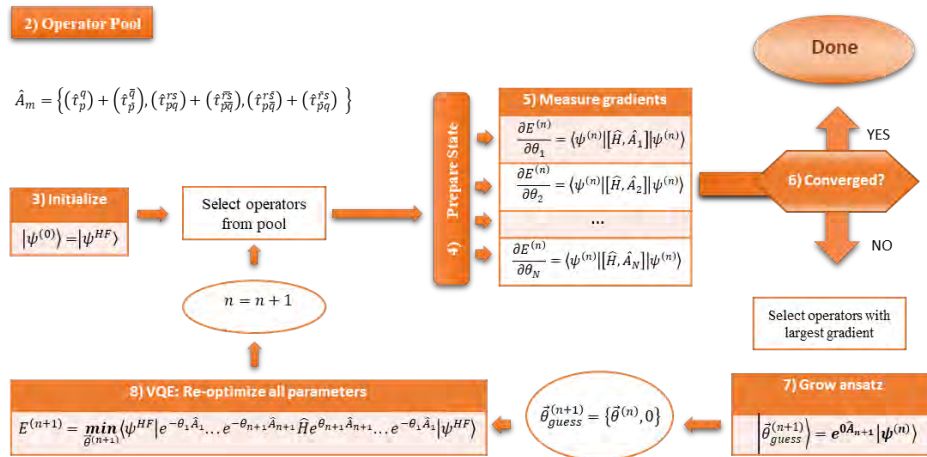


Figure 9.1: Adaptive Variational Quantum Eigensolver

The VQE is a hybrid quantum/classical algorithm originally proposed to approximate the ground

state of a quantum chemical system (the state attaining the minimum energy).

By varying the experimental parameters in the preparation of the state and calculating the Rayleigh-Ritz ratio [238] using the subroutine in a classical minimization, unknown eigenvectors can be prepared. At the end of the algorithm, the reconstruction of the eigenvector stored in the final set of experimental parameters that define the state will be done. The variational method in quantum mechanics is used, which is a way of finding approximations to the energetic state of lower energy or fundamental state, and some excited states. This allows to calculate approximate wave functions, such as molecular orbitals and is the basis of this method. It is the variational principle that will enable us to write the following equation  $\langle H \rangle_{\psi(\vec{\theta})} \geq \lambda_i$ . With  $\lambda_i$  as eigenvector and  $\langle H \rangle_{\psi(\vec{\theta})}$  as the expected value. The problem that the VQE solves is reduced to finding such an optimal choice of parameters  $\vec{\theta}$ , that the expected value is minimized and that a lower eigenvalue is located.  $\langle H \rangle = \langle \psi(\theta) | H | \psi(\theta) \rangle$ .

Architecture of the quantum-variational eigensolver (See Fig. (9.2)). **Algorithm 1:** Quantum states that have been previously prepared are fed into the quantum modules, which compute  $\langle H_i \rangle$ , where  $H_i$  is any given term in the sum defining  $H$ . The results are passed to the CPU, which computes  $\langle H \rangle$ . **Algorithm 2:** The classical minimization algorithm, run on the CPU, takes  $\langle H \rangle$  and determines the new state parameters, which are then fed back to the QPU. First, we prepare the trial

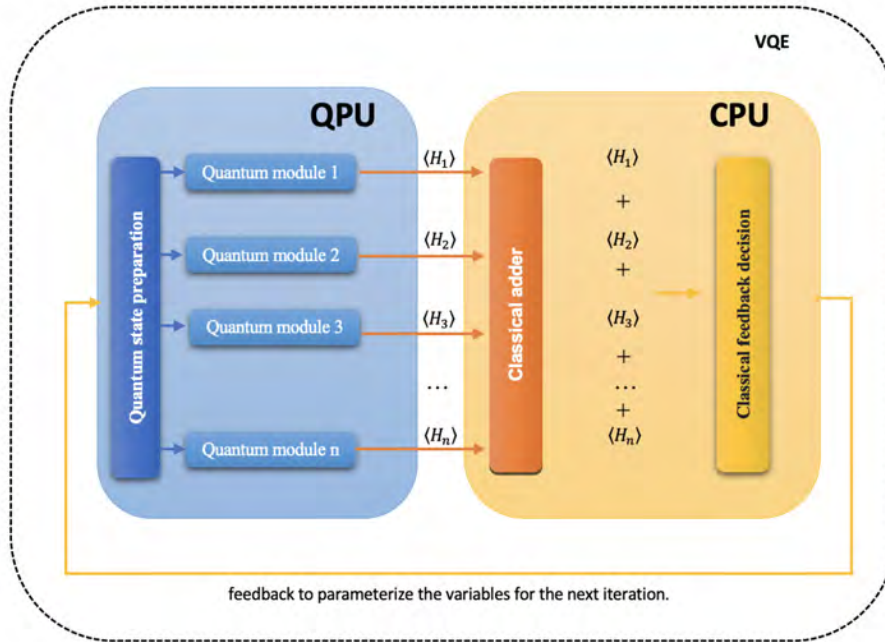


Figure 9.2: Architecture of the Variational Quantum Eigensolver. The architecture of the quantum-variational eigensolver. Algorithm 1: Quantum states that have been previously prepared are fed into the quantum modules, which compute  $\langle H_i \rangle$ , where  $H_i$  is any given term in the sum defining  $H$ . The results are passed to the CPU, which computes  $\langle H \rangle$ . Algorithm 2: The classical minimization algorithm, run on the CPU, takes  $\langle H \rangle$  and determines the new state parameters, which are then fed back to the QPU

wavefunction on a quantum processor. Then, we measure the qubits, resulting in an  $n$ -bit string  $x_0 \dots x_{n-1}$ . Each observed string easily translates into a sample from  $\langle \psi(\theta) | H | \psi(\theta) \rangle$ , because  $H$  is a weighted summation of tensor products of Pauli Z-matrices, and each such term can be computed

with a simple parity check. We denote these samples by  $H_k(\theta)$ ,  $k = 1, \dots, K$ , where  $K$  is a natural number and is the number of samples.

The sample mean (9.19) is an estimator for  $\langle H \rangle = \langle \psi(\theta) | H | \psi(\theta) \rangle$  and is used as the objective function for the classical optimization algorithm.

$$mean = \frac{1}{K} \sum_{k=1}^K H_k(\theta). \quad (9.19)$$

The need for the VQE algorithm is seen when solving an optimization problem related to a Hamiltonian. Once the Hamiltonian is built, we use/apply VQE to determine the ground state, from which an optimal solution to our objective function can be sampled with a probability of 1.

Assuming  $n$  qubits, we start by applying single qubit  $Y$  rotations to every qubit, parametrized by an angle  $\theta_{0,i}$  for qubit  $i$ . We then repeat the following  $p$  times. We apply controlled  $Z$  gates to all qubit pairs  $(i, j)$  satisfying  $i < j$ , where  $i$  denotes the control qubit and  $j$  the target qubit; and we add another layer of single-qubit  $Y$  rotations to every qubit, parametrized by  $\theta_{k,i}$  for qubit  $i$  and repetition  $k \in \{1, \dots, p\}$ .

### 9.2.2 Noisy VQE for the Optimization problem

In Quantum computing, we can classify the algorithms in 3 groups. Gate circuits [197], Annealing [55] and Variational [117][24][224]. Although the logic gates are not perfect and have noise by having

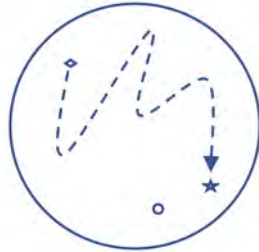


Figure 9.3: Minimization principle used in Variational Quantum Eigensolvers to empower Quantum Machine Learning

the variational principle, we can see in NISQ devices a golden opportunity to have a machine that analyzes Hilbert's vector space.

The term "Noisy Intermediate Scale Quantum" describes the era in which we find ourselves today. Noisy, because the computer still does not offer enough qubits to save for error correction. So, we have imperfect qubits at the physical layer, last but not at least "intermediate scale" because of their small number of qubits.

This leads us to forget about the imperfections of the gates and only think of the variational ansatz [117] and that this ansatz can be analyzed efficiently on a quantum computer. Something we can't do with a classic computer. Fig. (9.3). summarizes the idea. In short, we can say that we have a quantum computer that generates variational states, known as a Variational Quantum Computer [117]. Another way to see it could be, in each of the iterations, we have a quantum circuit close to the solution we would be looking for. This is the basis of Quantum learning [48]. We are doing machine learning (ML) on circuit design.

With this vision, we are developing a perfect machine to solve optimization and classification problems by exploring the entire configuration space of quantum molecules.

As commented, the VQE are useful because they find the lowest possible eigenvalue of a given Hermitian matrix  $H$  (it doesn't matter the size of  $H$ ) using the variational method or the variational principle. It's also known that the expected value must always be equal or greater than the lowest possible eigenvalue. This means that if we just keep minimizing that expectation value, we only get closer and closer to the minimum eigenvalue of the given  $H$  matrix and never below it [117]. With this powerful concept, the great clue is how to map our objective function into a Hamiltonian model of a given molecular system. To do that, first, we map the molecular Hamiltonian into a qubit. This essentially means that we are mapping the electron orbital interactions inside the molecules onto our qubits. Next, we prepare the set. Our set to be shallow has to cover a good enough range for our trial wave functions, so since we don't face our ground state energy. With the information given by a specific Hamiltonian, now, we calculate the energy of that electron configuration.

At this point, the algorithm measures those values and send them through to the classical optimizer. The classical optimizer minimizes our parameters, getting a lower expectation value  $H$ . After that, we feed all these values back into the quantum part and reiterate it with this loop many times until it converges onto the lowest possible energy state for that interatomic distance to follow all the described steps. All this is achieved regardless of the noise or imperfection of the logic gates.

### 9.2.3 Quantum approximate optimization algorithm

The *Quantum Approximate Optimization Algorithm* most known as QAOA, is now as far as the best quantum approximate optimization algorithm developed and introduced by Farhi, Goldstone, and Gutmann, that finds a right solution (not always find the exact solution) to an optimization problem in polynomial time [144, 93].

One reason why the work done by researchers makes QAOA (see Figures (9.4), (9.5) and (9.6)) very interesting and highly useful for the NISQ era is its potential to display quantum supremacy [106].

This algorithm belongs to the NISQ era algorithm class already discussed, which are from the hybrid algorithm class. More precisely, classical-quantum hybrid variational algorithms. We can safely consider it an evolution of adiabatic time trotted in  $p$  steps towards the ground state of a Hamiltonian (characteristics of a quantum circuit/system) that encodes the problem. The algorithm does "discretize" by introducing steps in the time variable (time evolution). Where the steps variable  $p$  defines the accuracy of the solution.

These graphs conceptually show us the differences/evolution between quantum annealing (left) that follows this adiabatic time evolution path. Simulated annealing (middle) follows the same way in discrete steps and QAOA that follows adiabatic time evolution path in  $p$  steps. These define the accuracy of the solution.

This algorithm builds on and improves run time on Simulated Annealing. Therefore, intuitively, we can believe before any demonstration that QAOA is the best candidate to solve combinatorial optimization problems assigned to the minimization of a Hamiltonian Ising in NISQ devices.

The Adiabatic Quantum Computing (AQC) has one serious problem when the computation time to solve any problem rises exponentially as  $\Delta E$  becomes infinitesimally small. This bound AQC's capability to solve a specific instance of hard optimization problems [6].

This means that the difference between the ground state and first excited state energy of the Hamiltonian  $H(t)$  of the problem we will resolve bounds the step size one can take to follow the adiabatic pathway. QAOA resides in this limitation of AQC. So, and according to the Fig. (9.4), we trotterize in  $p$  step relating to some configuration parameters. The Trotterization [6] formula is known by the equation (9.20).

$$e^{-iHt} = \left( e^{-iH_0 \frac{t}{p}} * e^{-iH_1 \frac{t}{p}} * \dots * e^{-iH_{k-1} \frac{t}{p}} \right)^P + f(\gamma). \quad (9.20)$$

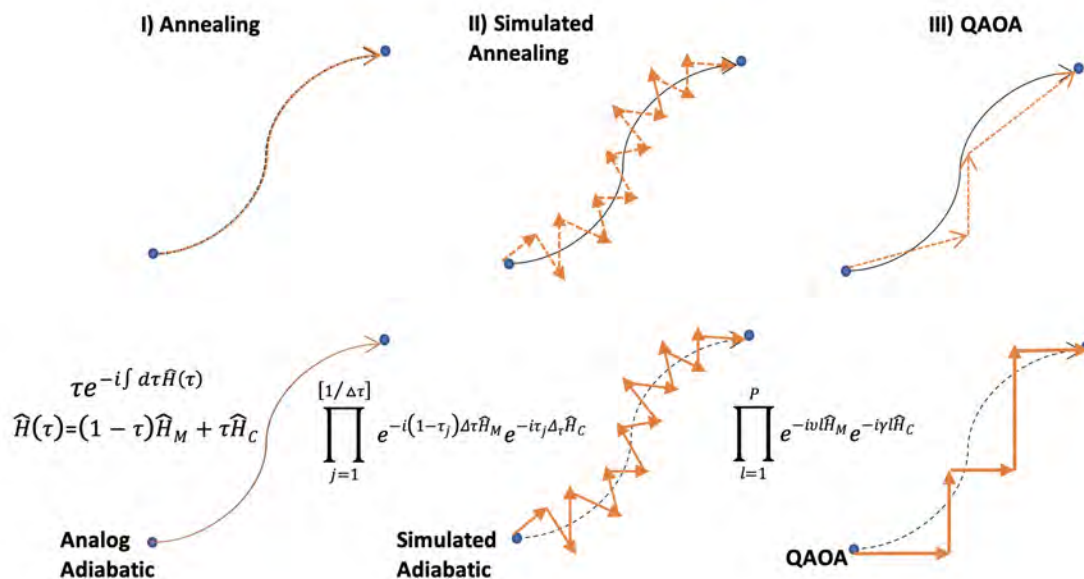


Figure 9.4: From Annealing to QAOA. These graphs conceptually show us the differences/evolution between quantum annealing (left) that follows this adiabatic time evolution path. Simulated annealing (middle) follows the same way in discrete steps and QAOA that follows adiabatic time evolution path in  $p$  steps. These define the accuracy of the solution.

With  $f(\gamma)$  as some polynomial factors equal to zero.

As Hamiltonians are Hermitian operators that are usually a sum of a large number of individual Hamiltonians  $\sum_j H_j$ , we can use Lie product formula [135] as shown by the equation (9.21).

$$e^{-i(\sum_j H_j)t} = \lim_{N \rightarrow \infty} \left( e^{-iH_0 \frac{t}{N}} e^{-iH_1 \frac{t}{N}} * \dots * e^{-iH_{m-1} \frac{t}{N}} \right)^N. \quad (9.21)$$

Since the limit of this series is infinite, when we implement this in quantum computing, we must truncate the function by introducing a quantifiable bounded error  $\epsilon$  refers to  $\|e^{-iHt} - U\| \leq \epsilon$ . This truncation is known as Trotterization, and it's widely used to simulate non-commuting Hamiltonians on quantum computers. We will take advantage of this technique in the development of EVA [15].

Suppose we want to simulate one circuit given by the following Hamiltonian:

$$H = X_0 + Y_1 + Z_2. \quad (9.22)$$

Where  $X$ ,  $Y$  and  $Z$  are, Pauli matrices (In some books, we can find this notation relating to the Pauli matrices  $\sigma_i$  ( $\sigma_x, \sigma_y, \sigma_z$ )) and the subscripts label the qubits that the Hamiltonians apply to. We can't simulate each Hamiltonian separately because they don't commute (the anti-commutation can be achieved by noting that  $XZ = ZX$  and  $-YZ = ZY$ ) [174]. This is the main reason why we use Trotterization, where we evolve the whole Hamiltonian by repeatedly switching between evolving  $X$ ,  $Y$  and  $Z$  each for a small period. The first step from the given system is finding the quantum gates (we are developing on Quantum gate-based computers) that implement each of its terms.

This case is simple since the quantum gates will implement the individual terms correctly.

$$R_x(\theta) = e^{-i\frac{\theta}{2}X}, R_y(\theta) = e^{-i\frac{\theta}{2}Y}, R_z(\theta) = e^{-i\frac{\theta}{2}Z}. \quad (9.23)$$

Where  $\theta$  specifies the angle by which to rotate the state in a specified axis, this notion is pretty powerful because it determines how long (as a time ( $t$ )) to apply the Hamiltonian to the qubit.

If we consider, in this case, that  $p = 2$  and  $t = 1$  we can simulate/compute our circuit (the Hamiltonian) by using the Lie formula as shown:

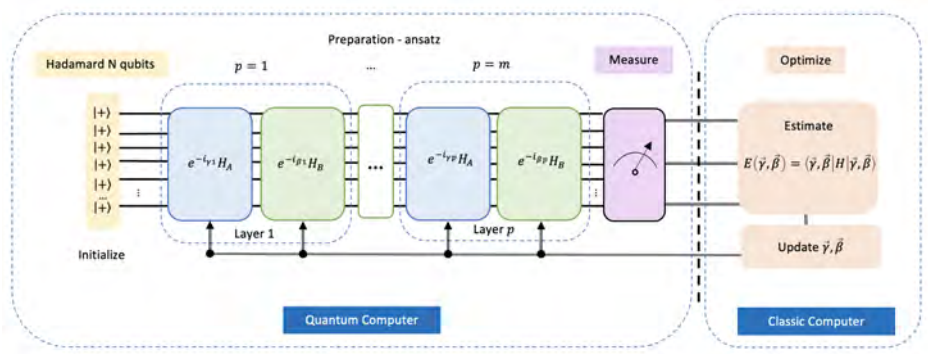


Figure 9.5: QAOA algorithm operation. The system is initialized along the y-direction in the Bloch sphere in the  $|+\rangle^{\otimes N}$  state. The unitary evolution under  $H = X_0 + Y_1 + Z_2$  is implemented for  $\gamma_i \beta_i$  angles for p times. At the end of the algorithm, global measurements in the x and the y basis are performed to compute the average energy  $\langle H \rangle = E(\vec{\gamma}, \vec{\beta})$ , which is compared to the theoretical ground state energy  $E_{groundstate}$

$$e^{X_0+Y_1+Z_2} = \left( e^{-iX_0-iY_1-iZ_2} \right) \left( e^{-i\frac{X_0}{2}-i\frac{Y_1}{2}-i\frac{Z_2}{2}} \right). \quad (9.24)$$

In a generic case, the Hamiltonian considered is from the cross-field. The algorithm divides it into two components  $H = H_A + H_B = H_{Ising}$ . With (9.25) and (9.26).

$$H_A = -A(t) \left( \sum_{i<j} J_{ij} Z_i Z_j + \sum_i h_i Z_i \right). \quad (9.25)$$

$$H_B = B(t) \sum_i h_i X_i. \quad (9.26)$$

And execute the steps mentioned above. The graphical representation of these steps is that of the figure (9.5) and following the processes defined by the figure (9.6). The state obtained after players of the QAOA can be written as:

$$|\vec{\gamma}, \vec{\beta}\rangle = \prod_{k=1}^p e^{-i\beta_k (H_B/J_0)} e^{-i\gamma_k (H_A/J_0)} |\psi_0\rangle. \quad (9.27)$$

Where  $|\psi_0\rangle$  is the initial state, where the evolution times  $\beta_k$  and  $\gamma_k$  are variational parameters used in the k-th QAOA layer to minimize the final energy  $E(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | H | \vec{\gamma}, \vec{\beta} \rangle$  and where  $J_0$  is the average nearest neighbour coupling, and in the case of  $B = 0$ . For each p (interaction), represent the unusual optimal angles by  $\{(\beta^{*(p)}, \gamma^{*(p)})\}$  which we can also think of as a pair of angle curves, as a



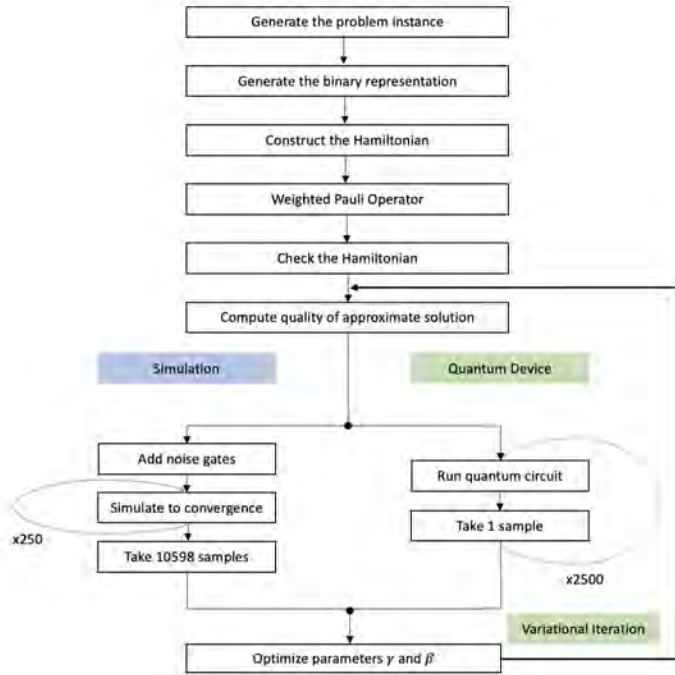


Figure 9.6: QAOA’s diagram process to perform it experimentally with quantum devices. In our study, 1000 samples are used to estimate the value of the cost function  $E(\vec{\alpha}, \vec{\beta}) = \langle \vec{\alpha}, \vec{\beta} | H | \vec{\alpha}, \vec{\beta} \rangle$  at each variational iteration.

function of step-index  $i$ . As  $p$  is varied, we may think of these minima as a set. And one can do several studies on this set, for example, the convergence study of variables  $p$  and  $N$ . This interesting article [39] reviews how to improve even the QAOA algorithm. At this point, we just need to experiment and focus on our problem. The next chapter will experiment and solve the Quantum Social Assistant Workers Scheduling Problem.

### 9.3 Summary

In this chapter, we have experimented with quantum computing in the Qiskit environment. We have experimented with variational techniques, variational algorithms like VQE, QAOA, and QUBO techniques. We tested the VQE as one of the best algorithms of this quantum era, and we have also verified the detailed operation of the QAOA, which is the bet of the gate-based computers that work as an annealing algorithm. As stated in the QAOA, one of the advantages is that we can increase the precision arbitrarily. In contrast, QA will only find the solution with probability one when the time goes to infinity, which is impractical. In addition, if  $T$  is too long, it is possible not to find the outcome as the probability is not monotonic.

Now we have the basis to contribute to the scientific community.





## Part II

# Contributions



# Chapter 10

## Research Design

### 10.1 Introduction

In this section, we define our research scenario, which is illustrated by figure (10.1). But, first, we will define the problem and the necessary tools.

The figure (10.1) summarises the work scenario on the tasks proposed for the thesis. First, we define a problem to help us answer our main hypothesis. Then, we ask whether it is possible to design a quantum algorithm to solve combinatorial optimisation problems with strict restrictions.

The designed exercise is a project called SWP, which combines routing, planning and combinatorial tasks. From this point, we solve the SWP using two different techniques: the first with the top-down approach and the second with the Machine learning technique developing a new qCBR algorithm.

Second, we propose and solve the Batching and Picking Problem with the generalisation of the SWP called qRobot.

To improve the computation time and make a leap in quality in chemistry, we have proposed new algorithms known as EVA and reduced-EVA [15], the latter as a new proposal for calculating the expected value to improve the flagship quantum algorithm VQE.

Therefore, we will continue working to improve the routing and optimisation algorithms for another sector such as banking in this era of very few useful qubits, pending the lecture of this doctoral thesis.

### 10.2 Proposed problem

Let  $n$  be the number of patients (users) and consider a weekly calendar of visits for each of them. Our objective is to find an optimal meeting's timing that minimises the cost of time travel; hence, money and maximises the number of patient visits in a work schedule. And if there are changes from the social workers or they hire or fire new workers or any change in patients, the system must adapt dynamically. Figure (10.2) illustrates the resolution of SWP with VQE. In the end, we assign social workers to the group with the resultant optimal hours.

In our case study, the daily schedule Table (10.1), is set at 8 hours, and the distance between patients is at least 15 minutes.

◇ A set of social workers ( $N_1, N_2, N_3, \dots$ )

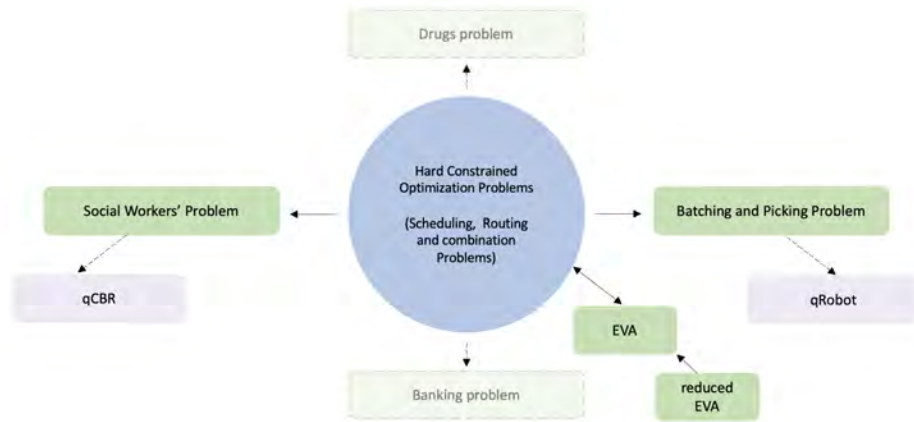


Figure 10.1: My research's path over these 3 years

A sample of the social workers' schedule					
Time	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 – 10:00	$U_1$	$U_1$	$U_1$	$U_1$	$U_1$
9:30 – 10:30	$U_4$				
10:15 – 11:15		$U_4$			
11:30 – 12:30					$U_4U_5$
11:45 – 12:45	$U_5$				
12:00 – 13:00	$U_2$				
14:45 – 15:45		$U_2$			
15:00 – 16:00			$U_3$		
15:15 – 16:15	$U_3$				
15:45 – 16:45			$U_3$		
16:00 – 17:00			$U_2$		
16:30 – 17:30		$U_3$			
17:00 – 18:00					$U_3$

Table 10.1: Schedule of patient visits without any association with social workers. Where  $U_1$  to  $U_2$  are the patients (users) and equal to the variable  $i$  or  $j$  of the mathematical formulation

- ◇ A set of patients ( $P_1, P_2, P_3, \dots$ )
- ◇ A set of visits ( $U_1, U_2, U_3, \dots$ )
  - each visit is linked to a patient: a patient can have multiple appointments on a day
  - for each visit, we know the start time and duration
- ◇ Social workers can work at most 8 hours per day (that means, social workers cannot do a visit the first visit at 9:00 and end the last tour at 18:00) (removed to reduce the number of variables, so this can be solved using Qiskit)
- ◇ We know the cost of travelling between each pair of patients. The cost can be seen as a function of travel time and distance.

The objective is the following:

- ◊ Find a schedule where each visit is assigned to a social worker
- ◊ We minimise the travel cost while also respecting that a social worker does not work more than 8 hours per day.

Taking into account, the advances on the speedup of the *Quantum Approximate Optimization Algorithm* (QAOA)[112] [189][183][177][39] in the NISQ era, we can affirm that the experiment’s scenario and its design are representatives. Because the only difficulty that can be added here, for this combinatorial optimisation problem, is the number of patients and some restrictions. These difficulties have to do directly with the computational cost (not with the formulation/algorithm) where quantum computing is called to be more efficient [189]. Hoping to access a more powerful computer, we limited our test to a 20 qubits computer, the most potent public quantum computer.

In this work, we have chosen  $n = 5$  (20 qubits) because the experiments from the problem of this class are among the most difficult in our test bench. We will review it in the discussion chapter. We

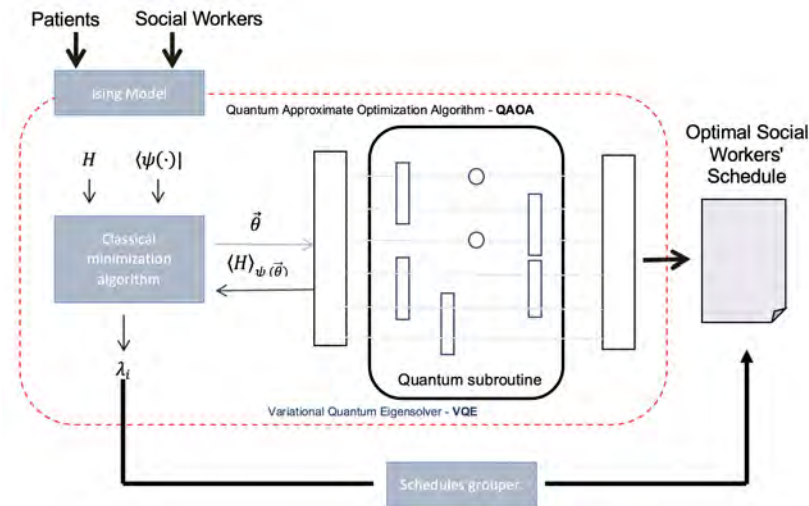


Figure 10.2: Using the Variational-Quantum-Eigensolver (VQE) as Quantum Machine Learning to creates an Intelligent social workers schedule problem solver

need to analyse and choose a programming environment to solve our problem. Next, we will focus on the tools to carry out our experiments.

## 10.3 Experimentation tools

### 10.3.1 Programming environments

The race to leadership in quantum computing involves determining standards. Everything currently points to the NISQ era being quite stable than public opinion imagine. Perhaps, it is not necessary to think about quantum desktop computers but to focus on generating applications control parameters for the devices of this era. Based on the work done by Mark Fingerhuth, Toma’s Babej and Peter Wittek, Open source software in quantum computing[185][211][96], which we had to update, we see that several players are working hard on high-level programming environments. A change in philosophy was also seen in mid-2019 with companies specialising in compiler writing and different collaborations between technology giants such as IBM, Google, Microsoft, AWS-Braket with high-tech startups such

as Xanadu. All this leads us to think that we will see many more applications in quantum computing and, above all, quantum datacenters in a few months.

Next, we will analyze the frameworks (Figure (10.3), (10.4) and (10.11) ) and decide on an environment for our experimentation.

Name	Tagline	Programming language	Licence	Supported OS
Cirq	Framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits.	Python	Apache-2.0	Windows, Mac, Linux
Cliffords.jl	Efficient calculation of Clifford circuits in Julia.	Julia	MIT	Windows, Mac, Linux
dimod	Shared API for Ising/quadratic unconstrained binary optimization samplers.	Python	Apache-2.0	Windows, Linux, Mac
dwave-system	Basic API for easily incorporating the D-Wave system as a sampler in the D-Wave Ocean software stack.	Python	Apache-2.0	Linux, Mac
FermiLib	Open source software for analyzing fermionic quantum simulation algorithms.	Python	Apache-2.0	Windows, Mac, Linux
Forest (pyQuil & Grove)	Simple yet powerful toolkit for writing hybrid quantum-classical programs.	Python	Apache-2.0	Windows, Mac, Linux
OpenFermion	The electronic structure package for quantum computers.	Python	Apache-2.0	Windows, Mac, Linux
ProjectQ	An open source software framework for quantum computing.	Python, C++	Apache-2.0	Windows, Mac, Linux
PyZX	Python library for quantum circuit rewriting and optimisation using the ZX-calculus.	Python	GPL-3.0	Windows, Mac, Linux
QGL.jl	A performance orientated QGL compiler.	Julia	Apache-2.0	Windows, Mac, Linux
Qbsolv	Decomposing solver that finds a minimum value of a large quadratic unconstrained binary optimization problem by splitting it into pieces.	C	Apache-2.0	Windows, Linux, Mac
Qiskit Terra & Aqua	Quantum Information Science Kit for writing experiments, programs, and applications.	Python, C++	Apache-2.0	Windows, Mac, Linux
Qiskit Tutorials	A collection of Jupyter notebooks using Qiskit.	Python	Apache-2.0	Windows, Mac, Linux
Qiskit.js	Quantum Information Science Kit for JavaScript.	JavaScript	Apache-2.0	Windows, Mac, Linux
Qrack	Comprehensive, GPU accelerated framework for developing universal virtual quantum processors.	C++	GPL-3.0	Linux, Mac
QuantumEog	Python tools for analyzing both classical and quantum Bayesian networks.	Python	BSD-3-Clause	Windows, Mac, Linux
Quantum++	A modern C++11 quantum computing library.	C++, Python	MIT	Windows, Mac, Linux
Qubiter	Python tools for reading, writing, compiling, simulating quantum computer circuits.	Python, C++	BSD-3-Clause	Windows, Mac, Linux
Quirk	Drag-and-drop quantum circuit simulator for your browser to explore and understand small quantum circuits.	JavaScript	Apache-2.0	Windows, Mac, Linux
reference-qvm	A reference implementation for a Quantum Virtual Machine in Python.	Python	Apache-2.0	Windows, Mac, Linux
ScaffCC	Compilation, analysis and optimization framework for the Scaffold quantum programming language.	C++, Objective C, LLVM	BSD-2-Clause	Linux, Mac
Strawberry Fields	Full-stack library for designing, simulating, and optimizing continuous variable quantum optical circuits.	Python	Apache-2.0	Windows, Mac, Linux
XACC	eXtreme-scale Accelerator programming framework.	C++	Eclipse PL-1.0	Windows, Mac, Linux
XACC VQE	Variational quantum eigensolver built on XACC for distributed, and shared memory systems.	C++	BSD-3-Clause	Windows, Mac, Linux

<https://doi.org/10.1371/journal.pone.0208561.t001>

Figure 10.3: Overview of all published quantum computing programming environments [96]

Name	Quantum computing paradigm	Quantum algorithms	Quantum circuits	Quantum compiler	Quantum computer simulator	QPU backend	Full-stack
Cirq	Discrete gate model	✓	✓	✓	✓	✗	✓
Cliffords.jl	Discrete gate model	✗	✓	✗	✓	✗	✗
FermiLib	Discrete gate model	✓	✗	✗	✗	✗	✗
Forest (pyQuil & Grove)	Discrete gate model	✓	✓	✓	✓	✓	✓
OpenFermion	Discrete gate model	✓	✓	✗	✗	✗	✗
ProjectQ	Discrete gate model	✓	✓	✓	✓	✓	✓
PyZX	Discrete gate model	✗	✗	✓	✗	✗	✗
QGL.jl	Discrete gate model	✗	✗	✓	✗	✗	✗
Qiskit Terra & Aqua	Discrete gate model	✓	✓	✓	✓	✓	✓
Qiskit Tutorials	Discrete gate model	✓	✗	✗	✗	✗	✗
Qiskit.js	Discrete gate model	✓	✓	✓	✓	✓	✓
Qrack	Discrete gate model	✗	✓	✓	✓	✗	✗
Quantum Fog	Discrete gate model	✓	✓	✗	✗	✗	✗
Quantum++	Discrete gate model	✗	✓	✗	✓	✗	✗
Qubitier	Discrete gate model	✓	✓	✓	✓	✓	✓
Quirk	Discrete gate model	✓	✓	✗	✓	✗	✗
reference-qvm	Discrete gate model	✗	✓	✗	✓	✗	✗
ScaffCC	Discrete gate model	✗	✗	✓	✗	✗	✗
Strawberry Fields	Continuous gate model	✓	✓	✓	✓	✗	✓
XACC	Discrete gate model	✓	✓	✓	✓	✓	✓
XACC VQE	Discrete gate model	✓	✗	✗	✗	✗	✗
Name	Hardware platform	Hamiltonian generation	Minor embedding	Post-processing	Classical solver	QPU backend	Full-stack
dimod	Quantum annealing	✗	✓	✓	✓	✓	✗
dwave-system	Quantum annealing	✗	✓	✓	✓	✓	✗
Qbsolv	Quantum annealing	✗	✗	✗	✓	✓	✗

<https://doi.org/10.1371/journal.pone.0208561.t002>

Figure 10.4: Detailed characteristics of all published quantum computing programming environments. Data, August 2018 [96]

### 10.3.1.1 PennyLane:

*PennyLane*\* [45] (See the Fig. (10.5)) is a *Python 3*-based programming environment for optimization and machine learning in quantum and hybrid (quantum-classical) computing. The library provides a unified architecture for quantum computing devices of the era we are in and supports qubit and continuous variable paradigms. The main feature of *PennyLane* is the capability to calculate gradients of variational quantum circuits that are compatible with classical techniques such as backpropagation.

*PennyLane* extends the standard machine differentiation algorithms in optimisation and machine learning to include quantum and hybrid calculations. A plugin system makes the framework compatible with any simulator or gate-based quantum hardware. For example, Xanadu, through its framework, provides plugins for Strawberry Fields[45], Rigetti Forest<sup>†</sup>, Qiskit, Cirq<sup>‡</sup> y ProjectQ, allowing *PennyLane* optimisations to run on publicly accessible quantum devices provided by Rigetti and IBMQ. Furthermore, *PennyLane* can be used as a classic environment thanks to its interaction with accelerated machine learning libraries such as TensorFlow, PyTorch and autograd. Also, *PennyLane* can be used for optimisation problem solving using VQE, QAOA, QML or any quantum formulation. The following is a list of the plugins already developed to make *PennyLane* universal. *PennyLane-SF*: Supports integration with Strawberry Fields, a full-stack Python library to simulate photon quantum

\*<https://github.com/XanaduAI/pennylane/>

<sup>†</sup><http://docs.rigetti.com/en/stable/>

<sup>‡</sup><https://cirq.readthedocs.io/en/stable/>



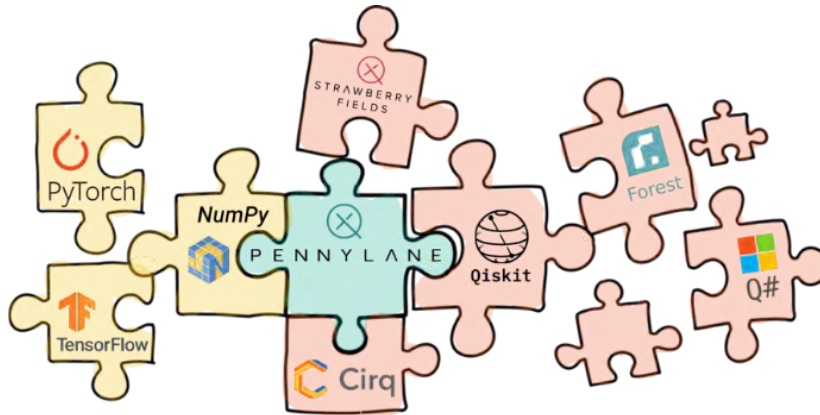


Figure 10.5: Xanadu bets with its PennyLane platform [44].

computing. *PennyLane-qiskit* : Supports integration with Qiskit, an open-source quantum computing environment from IBM that provides support for Qiskit Aer quantum simulator devices and IBMQ hardware devices. *PennyLane-cirq*: Supports integration with Cirq, an open-source Google quantum computing environment. *PennyLane-Forest*: Supports integration with PyQuil, Rigetti Forest SDK and Rigetti QCS, an open-source quantum computing environment from Rigetti that provides device support with *Quantum Virtual Machine* (QVM) and *Quantum Processing Units* (QPU) hardware. *PennyLane-Qsharp*: Supports integration with Microsoft Quantum Development Kit, a quantum computing framework that uses the Q# quantum programming language.

### 10.3.1.2 Cirq

*Cirq* is Google’s commitment (Fig. (10.6)) as a *framework for the programming and simulation of quantum computers*. *Cirq is a software library for writing, manipulating and optimising quantum circuits and then running them on and simulating quantum computers*.

Cirq takes into account the NISQ era, which is the one we are in, to get the most out of the hardware, analysing all the details of it before executing the quantum circuit or not. The current



Figure 10.6: Google bets with its Cirq platform.

version of Cirq is alpha, ready to test<sup>§</sup>.

<sup>§</sup><https://cirq.readthedocs.io/en/stable/>

### 10.3.1.3 Qsharp

*Qsharp<sup>¶</sup> is Microsoft's push for the bet on its quantum hardware (Fig. (10.7) and (10.8)). Q# is a new high-level programming language focused on the quantum.*

Its features rich integration with Visual Studio, Visual Studio Code and interoperability with the Python programming language. Enterprise-grade development tools provide the fastest path to quantum programming on Windows, macOS, or Linux.



Figure 10.7: Bet on Microsoft with its Qsharp platform [168].

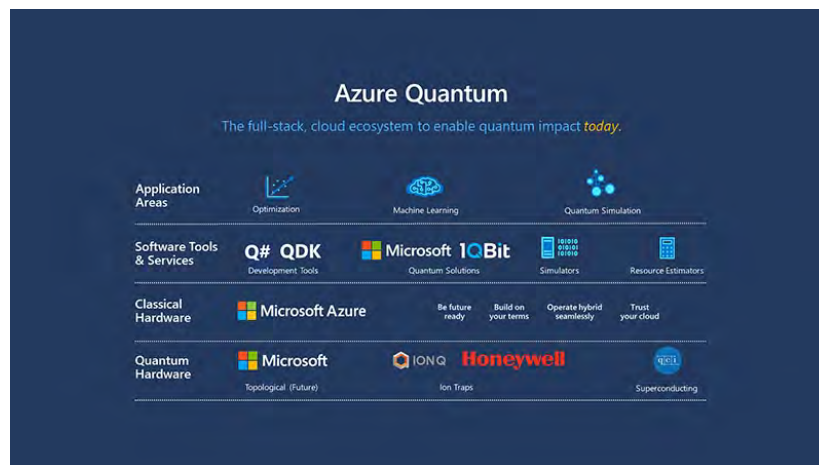


Figure 10.8: Azure Quantum Computing [167]

Microsoft as IBM bets for the cloud solution in the short term. In the case of Microsoft, its bet is Quantum Azure.

*Azure Quantum is a cloud platform for quantum computing where developers use the QDK to write Q# programs and run on quantum hardware or formulate problems to run on quantum-inspired solvers.*

The heatmap, figure (10.9), displays the evaluation results from source code documentation, README files, changelogs, user documentation, and tutorials on a scale of 1 (bad) to 5 (good). The data was obtained in August 2018 [96].

*Qiskit is textitthe IBM programming environment consists of Terra<sup>¶</sup> (central compiler and libraries for quantum programming), Aer<sup>\*\*</sup> (noise modelling and simulators without noise), Ignis (error char-*

<sup>¶</sup><https://docs.microsoft.com/es-es/quantum/overview/what-is-qsharp-and-qdk>

<sup>¶</sup><https://qiskit.org/terra>

<sup>\*\*</sup><https://qiskit.org/aer>

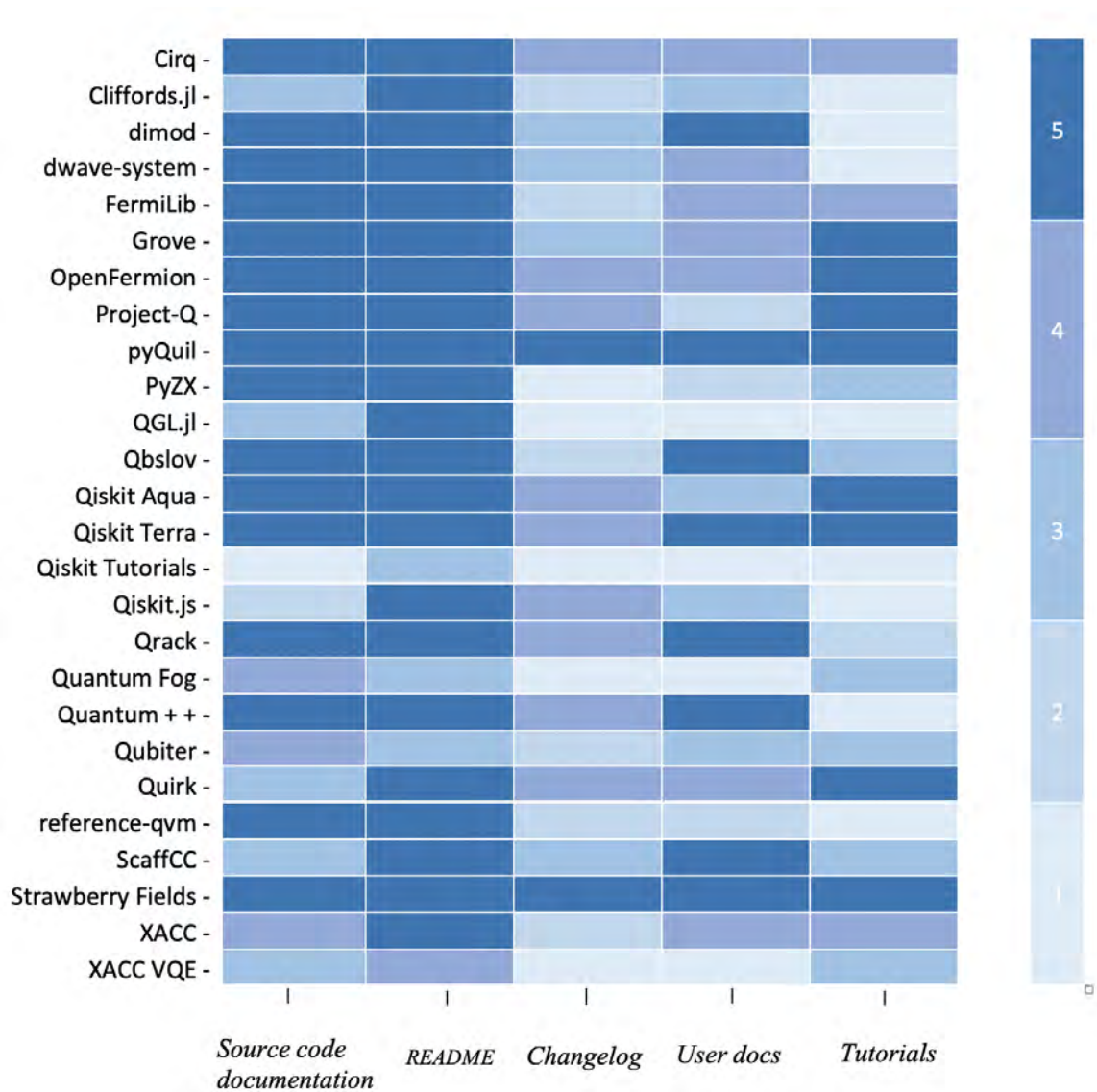


Figure 10.9: The heat map on the results of analysis and documentation

acterization and QEC) and Aqua<sup>††</sup> (Applications).

Next, I will go into more detail with each of the components to understand it better, as is shown in figure (10.12). *Terra* provides the fundamental roots for our software stack. Within Terra, there is a set of tools for composing quantum programs at the circuit and pulse levels, to optimize the constraints of a particular quantum physical processor, and to manage batch execution of experiments on remote access backends.

*Aqua*, the "water" element, is the element of life. This library will solve everything related to Chemistry, Optimisation or AI. Aqua is accessible to experts in chemistry, optimisation, or artificial intelligence domains who want to explore the benefits of using quantum computers as accelerators for

<sup>††</sup><https://qiskit.org/aqua>

Project	Roadmap	Releases	Contributors	User-discussion channels	Developer-discussion channels	Public review process	Community profile
Cirq	X	✓	28	Stack Exchange	-	E+I	4/7
Cliffords.jl	X	✓	7	-	-	E	3/7
dimod	X	✓	11	Forum	-	E+I	5/7
dwave-system	X	✓	6	Forum	-	E+I	4/7
FermiLib	X	✓	10	-	-	E+I	3/7
Forest - Grove	X	✓	24	Slack	Slack	E+I	3/7
Forest - pyQuil	X	✓	46	Slack	Slack	E+I	3/7
OpenFermion	X	✓	26	-	-	E+I	3/7
ProjectQ	X	✓	10	-	-	E+I	3/7
PyZX	X	X	3	-	-	-	3/7
QGL.jl	X	X	3	-	-	E+I	3/7
Qbsolv	X	✓	18	Forum	-	E+I	5/7
Qiskit Aqua	X	✓	14	Forum	-	E+I	7/7
Qiskit Terra	✓	✓	67	Forum, Slack	Slack	E+I	7/7
Qiskit Tutorials	X	X	37	-	-	E+I	3/7
Qiskit.js	X	✓	4	Forum	-	E	7/7
Qrack	X	✓	2	-	-	E+I	3/7
Quantum Fog	X	X	2	-	-	E	3/7
Quantum++	X	✓	3	Gitter	-	E	5/7
Qubiter	X	X	2	-	-	E	3/7
Quirk	X	✓	3	-	-	E	4/7
reference-qvm	X	✓	8	-	-	E+I	3/7
ScaffCC	X	✓	7	-	-	E	3/7
Strawberry Fields	X	✓	5	Slack	Slack	E+I	7/7
XACC	X	X	6	-	-	E	4/7
XACC VQE	X	X	2	-	-	E	3/7

<https://doi.org/10.1371/journal.pone.0208561.t003>

Figure 10.10: Results of the evaluation for the community analysis. For each project, we indicate whether there is a public development roadmap and whether the software is published in the form of releases. Besides, we report the GitHub community profile score, the total number of contributors, the type of discussion channel focused on the user and the developer, and the type of public code review process, specifically if it applies to internal (I) external (E) taxpayers. Data were obtained in August 2018 [96]

specific computational tasks without worrying about how to translate the problem into the language of quantum machines.

*Ignis*, the "fire" element, is dedicated to fighting noise and mistakes to forge a new path. While *Aer*, the item "air", permeates all the aspects of Qiskit. Simulators, Emulators, and Debuggers.

#### 10.3.1.4 IBM Quantum Experience

*IBM Quantum Experience*<sup>‡‡</sup> (IBMQ) (Figures (10.13) to (10.19)) is an online platform that provides users of the general public with access to a prototype suite of IBM quantum processors via the cloud.

An online Internet forum to discuss relevant quantum computing issues, a set of tutorials on how to program IBMQ devices and other educational materials on quantum computing.

It is of utmost importance to perceive and understand the strategy of IBM when he describes the Qiskit roadmap as a platform and business bet. This strategy determines how to propose and design the algorithm for the end-user. In just two years of research that I have been following the firm,

<sup>‡‡</sup><https://quantum-computing.ibm.com/>

Name	Version control system	Issue tracking system	Issues/ PRs	Attention rate	Average response time (days)	Test suite	Code coverage	Complexity
Cirq	Git	GitHub	448/686	0.54	2.6	✓	94%	2.99
Cliffords.jl	Git	GitHub	6/12	0.33	<1	✓	-	-
dimod	Git	GitHub	110/201	0.30	5.3	✓	94%	2.96
dwave-system	Git	GitHub	54/72	0.24	8.2	✓	87%	3.47
FermiLib	Git	GitHub	24/134	0.31	<1	✓	99%	2.43
Forest - Grove	Git	GitHub	53/130	0.51	17.7	✓	72%	3.25
Forest - pyQuil	Git	GitHub	293/385	0.41	10.6	✓	88%	2.65
OpenFermion	Git	GitHub	137/345	0.61	1.3	✓	100%	2.46
ProjectQ	Git	GitHub	84/198	0.75	4.0	✓	100%	4.02
PyZX	Git	GitHub	6/2	0.80	<1	✓	51%	4.42
QGL.jl	Git	GitHub	17/13	0.75	130.6	✓	-	-
Qbsolv	Git	GitHub	50/85	0.17	22.2	✓	95%	-
Qiskit Aqua	Git	GitHub	43/141	0.20	1.8	✓	67%	3.04
Qiskit Terra	Git	GitHub	526/713	0.11	16.0	✓	76%	2.56
Qiskit Tutorials	Git	GitHub	94/274	0.40	8.6	✗	-	-
Qiskit.js	Git	GitHub	19/8	0.33	4.4	✓	66%	-
Qrack	Git	GitHub	7/78	0.07	8.7	✓	87%	-
Quantum Fog	Git	GitHub	17/1	1.00	<1	✗	0%	3.32
Quantum++	Git	GitHub	8/45	0.88	<1	✓	72%	-
Qubiter	Git	GitHub	14/3	0.75	<1	✗	0%	-
Quirk	Git	GitHub	286/131	0.96	<1	✓	-	-
reference-qvm	Git	GitHub	6/14	0.44	75.6	✓	80%	3.99
ScaffCC	Git	GitHub	15/11	0.18	10.1	✓	-	-
Strawberry Fields	Git	GitHub	16/20	0.73	1.2	✓	97%	2.70
XACC	Git	GitHub	65/14	0.65	<1	✓	-	-
XACC VQE	Git	GitHub	22/4	0.33	8.8	✓	-	-

<https://doi.org/10.1371/journal.pone.0208561.t004>

Figure 10.11: Evaluation results for the static analysis of each project and its source code [96]



Figure 10.12: Components of Qiskit's framework [241]

IBM is already beginning to encapsulate many libraries as a black box to allow users to work more at a high level. This has enormous benefits for the scientific community since quantum computing developers will increase exponentially. Because the complexity of understanding quantum computing fundamentals will be reduced, notably, the part of the quantum mechanics and, thus, would not be necessary. But this will also limit understanding some quantum phenomena better because it will be



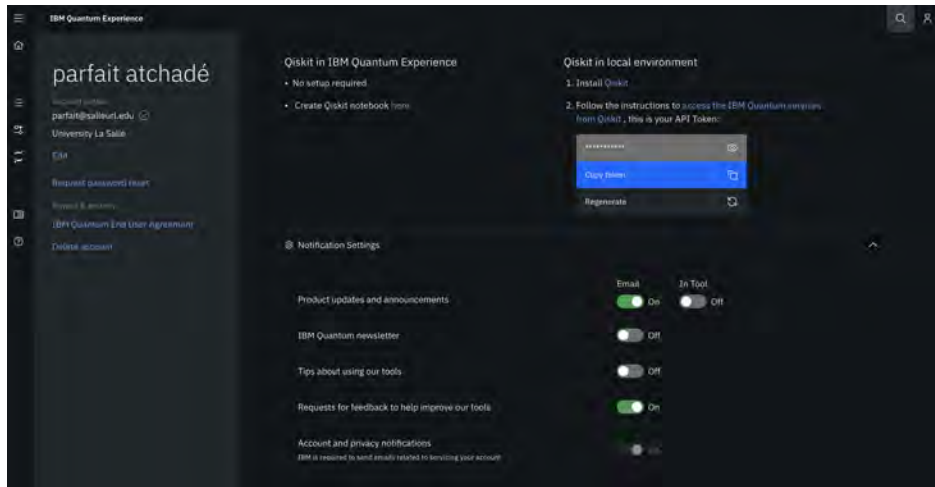


Figure 10.13: IBMQ User Profile

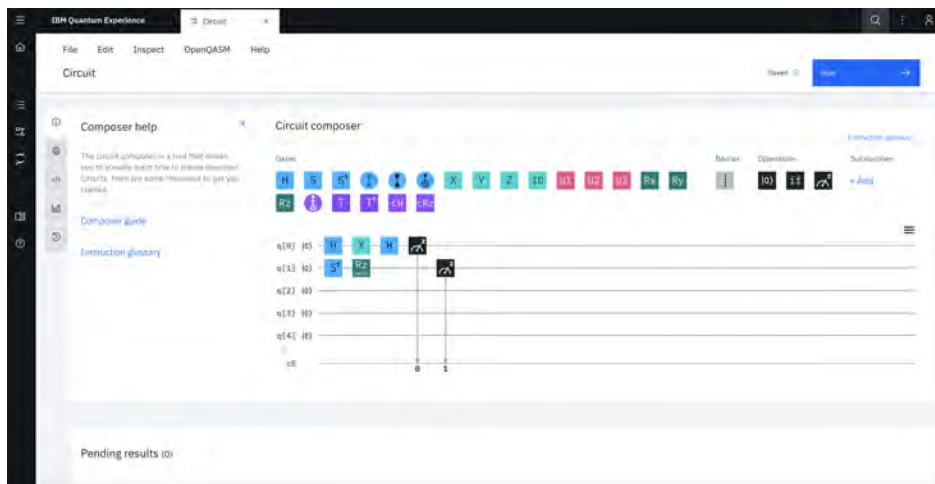


Figure 10.14: IBMQ Circuit Composer

increasingly difficult to reach the low level. However, the low level will be reserved at the level of the circuit composer.

### 10.3.2 Experimentation algorithms

For its simplicity, the optimisation based on the decent gradient, where each parameter is updated in the direction that produces the most significant local change in energy (in our case), is used intuitively. We also know that the number of evaluations carried out is proportional to selected optimisation parameters. This will quickly help the algorithm find a local optimum in the proposed search space. However, this optimisation method is often stuck at a local optimum in many cases. This and the relative cost in the number of circuit evaluations performed make the descent gradient method not recommended for use in VQE.

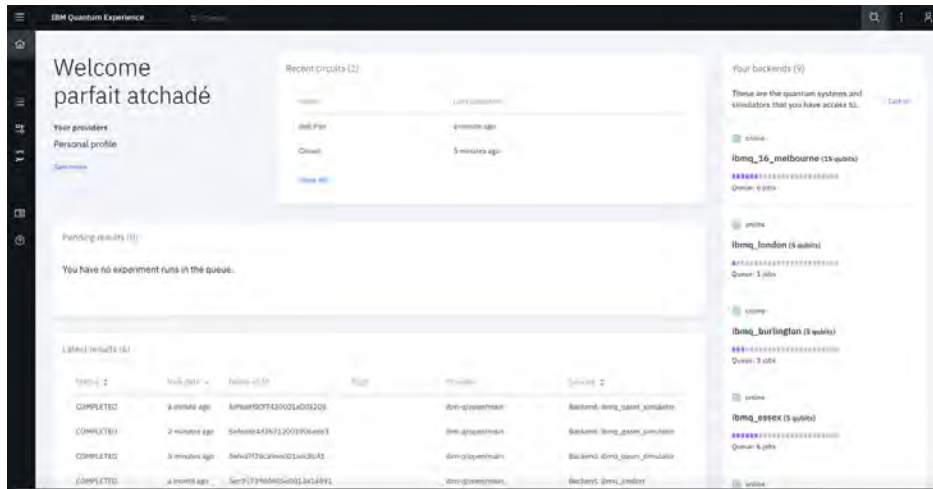


Figure 10.15: IBMQ Laboratory Profile

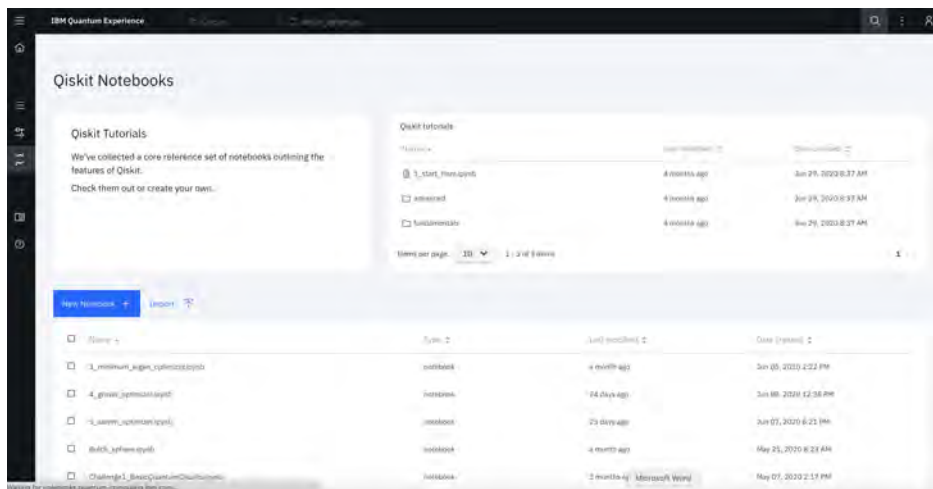


Figure 10.16: IBMQ Jupyter Notebook Files view

### 10.3.2.1 SPSA

Introducing the *Simultaneous Perturbation Stochastic Approximation Optimizer* known as *SPSA*[100] approximates the gradient of the objective function with only two measurements. It does this by simultaneously disturbing all parameters randomly, in contrast to the decent slope where each parameter is independently disturbed. When we need to add a circuit in real condition, that is, with noise, it is strongly recommended to use SPSA as the classic optimiser. The above description makes the SPSA the appropriate optimiser for optimising a noisy objective function.

### 10.3.2.2 SLSQP

The *Sequential Least-Squares Programming* also known as *SLSQP* is a sequential least squares programming algorithm that uses the Han – Powell quasi-Newton method with a BFGS update of the B – matrix and an L1 – test function in the step-length algorithm.

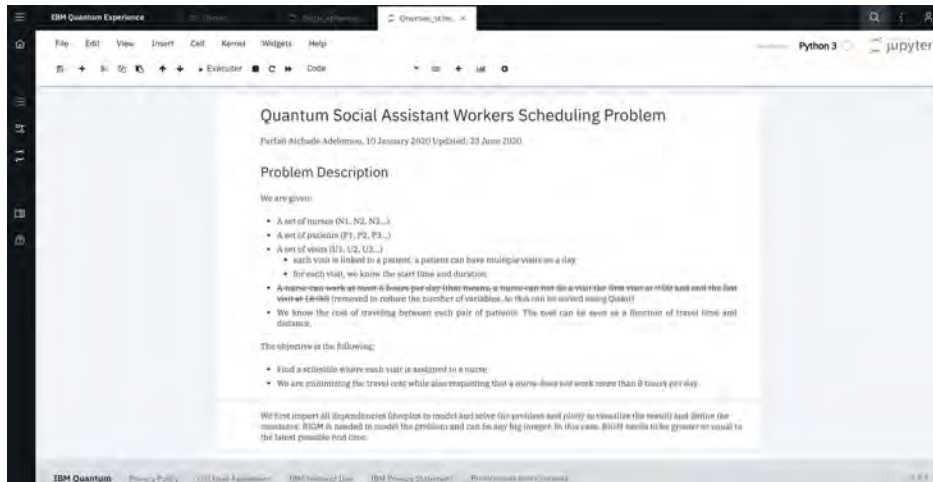


Figure 10.17: IBMQ Jupyter Notebook view

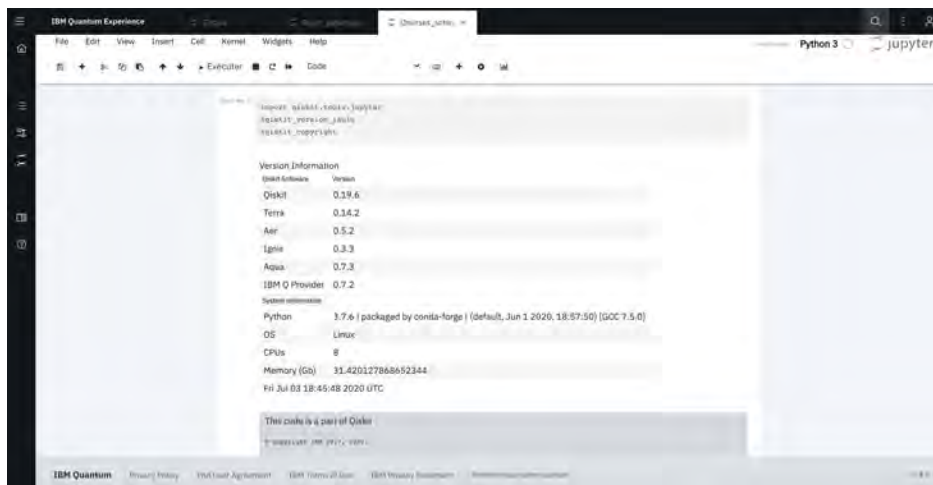


Figure 10.18: From IBMQ’s Jupyter Notebook, the Qiskit version

### 10.3.2.3 COBYLA

The textitclassic optimizer Constrained Optimization BY Linear Approximation commonly called *COBYLA* only evaluates the objective function per optimization iteration. The number of evaluations is independent of the cardinality of the set of parameters. Therefore, it is recommended to use *COBYLA* if the objective function does not produce noise, and it is convenient to minimise the number of evaluations carried out.

Qiskit, and more specifically in the Aqua library, provides us with several classical optimisers.

### 10.3.3 VQE Experimentation

As discussed above, it is not possible for a polynomial parameterised variational form to generate a transformation to any state. Variational forms can be grouped into two categories, depending on how they address this limitation. The first category of variational forms uses domain or application-



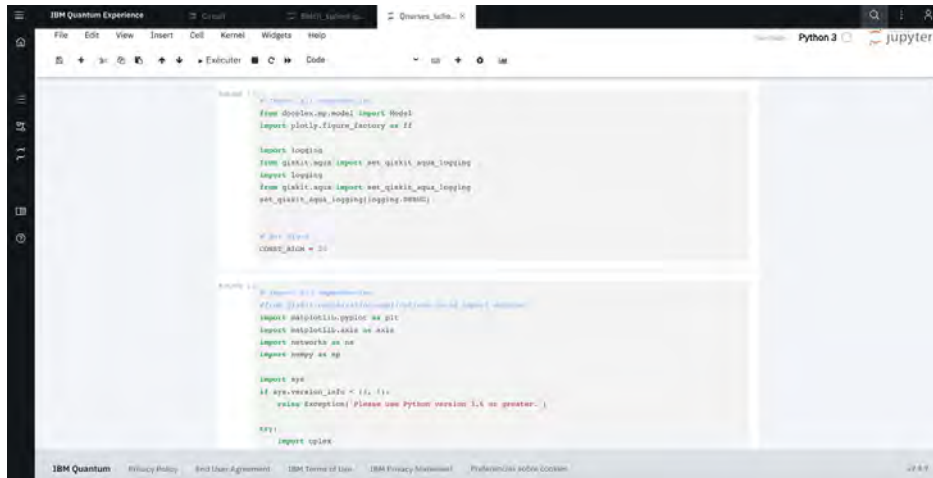


Figure 10.19: From IBMQ’s a code view from Jupyter Notebook

Name	Status	Status
Assembler	Stable	Completed in version 0.9
Circuit	Unstable	-
Copiler	Stable	Completed in version 0.9
Converters	Unstable	-
Dagcircuit	Remove	Will be part of circuits
Extensions	Remove	Will be part of circuits
Ignis.characterization	-	-
Ignis.mitigation	-	-
Ignis.characterization	-	-
Providers	Stable	Completed in version 0.7
Pulse	Unstable	-
Gasm	Remove	Passer location to be determined
Gobj	Remove	Moved into the provider
Quantum_info	Unstable	-
Result	Remove	Moved into the provider
Schemas	Stable	Completed in version 0.7
Tests	Unstable	-
Tools	Unstable	Various elements to be removed
Transpiler	Stable	Completed in version 0.9
Validation	Stable	Completed in version 0.7
Visualization	Stable	Completed in version 0.9

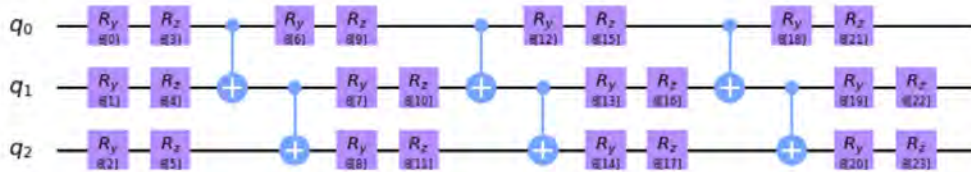
Table 10.2: Status and release of the modules towards Qiskit 1.0

specific knowledge to limit the set of possible output states. The second approach uses a heuristic circuit without prior mastery or application-specific expertise.

Therefore, variational shapes have been built for specific quantum computing architectures. Furthermore, the circuits are tuned to take full advantage of the connectivity and gates natively available from a given quantum device. This leads us to think about the importance of compilers if one wants to make multiplatform algorithms (thinking of using them on several different quantum computers.).

In the second approach, the gates are layered to obtain a good approximation in a wide range

Experiment of VQE with 3 qubits on Linear Entanglement



Experiment of VQE with 3 qubits on Full Entanglement

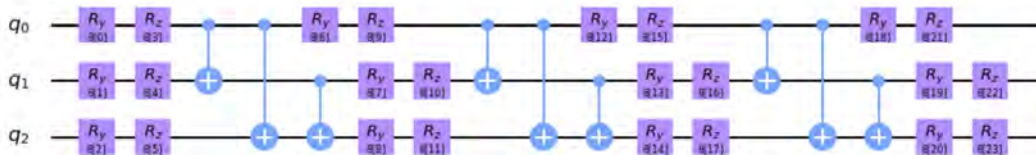


Figure 10.20: VQE experiment with three qubits on linear and full entanglement

of states. Qiskit Aqua supports three such variational forms (see Fig. (10.20)): *RyRz*, *Ry* and *SwapRz* (we will only discuss the first two). All of these variational forms accept multiple user-specified configurations. Three necessary arrangements are the number of qubits in the system, the depth setting, and the entanglement setting. A single layer of a variational form specifies a specific pattern of single-qubit rotations and *CX* gates. The depth setting says how frequently the variational form should repeat this pattern. By increasing the depth setting, at the cost of increasing the number of parameters that must be optimised, the set of states in the variational form can generate increases. Finally, the entanglement setting selects the configuration, and implicitly the number, of *CX* gates. For example, when the entanglement setting is linear, *CX* gates are applied to adjacent qubit pairs in order (and thus  $(n - 1)CX$  gates are added per layer). When the entanglement setting is full, a *CX* gate is applied to each qubit pair in each layer. The circuits for *RyRz* corresponding to entanglement = "full" and entanglement = "linear" can be seen by executing the following code snippet:

In the second approach, the quantum gates are layered to obtain good approximation in a wide range of states. Qiskit Aqua supports three variational forms: *RyRz*, *Ry* and *SwapRz*. All of these variational forms accept multiple user-specified settings. Three essential settings/configurations are:

- ◇ the number of qubits in the system,
- ◇ the depth setting,
- ◇ and the interlace setting.

A single layer of a variational shape specifies a specified pattern of unique qubit rotations and *CX* gates. The depth setting indicates how frequently the variational form should repeat this pattern by increasing the depth setting. The set of states that the variational form can generate increases at the cost of increasing the number of parameters to be optimised. Finally, the entanglement pattern selects the configuration, and implicitly the number, of *CX* gates. For example, when the entanglement

configuration is linear,  $CX$  gates are applied to adjacent qubit pairs in order and thus  $(n-1)CX$  gates per layer are added. When the entanglement configuration is full, a  $CX$  gate is applied to each pair of qubits in each layer. The circuits for  $RyRz$  corresponding to entanglement = "full" and entanglement = "linear" can be seen in Fig. (10.20). Let  $d$  be the depth setting and  $n$  the number of qubits, the  $RyRz$  parameters have  $n \times (d + 1) \times 2$ ,  $Ry$  with linear entanglement has  $2n \times d + \frac{1}{2}$  parameters, and  $Ry$  with full entanglement has  $d \times n \times \frac{(n+1)}{2} + n$  parameters.

### 10.3.4 CPLEX

From IBM literature, linear programming (LP) was revolutionised when the  $CPLEX^{\text{®}}$  software was developed. CPLEX (Fig. (10.21)) was the first commercial linear optimiser on the market to be written in the  $C$  programming language. CPLEX gave operations researchers unprecedented flexibility, reliability, and performance, enabling them to create new optimisation algorithms, models, and applications. The term CPLEX itself is based on the concept of a Simplex algorithm[124], invented

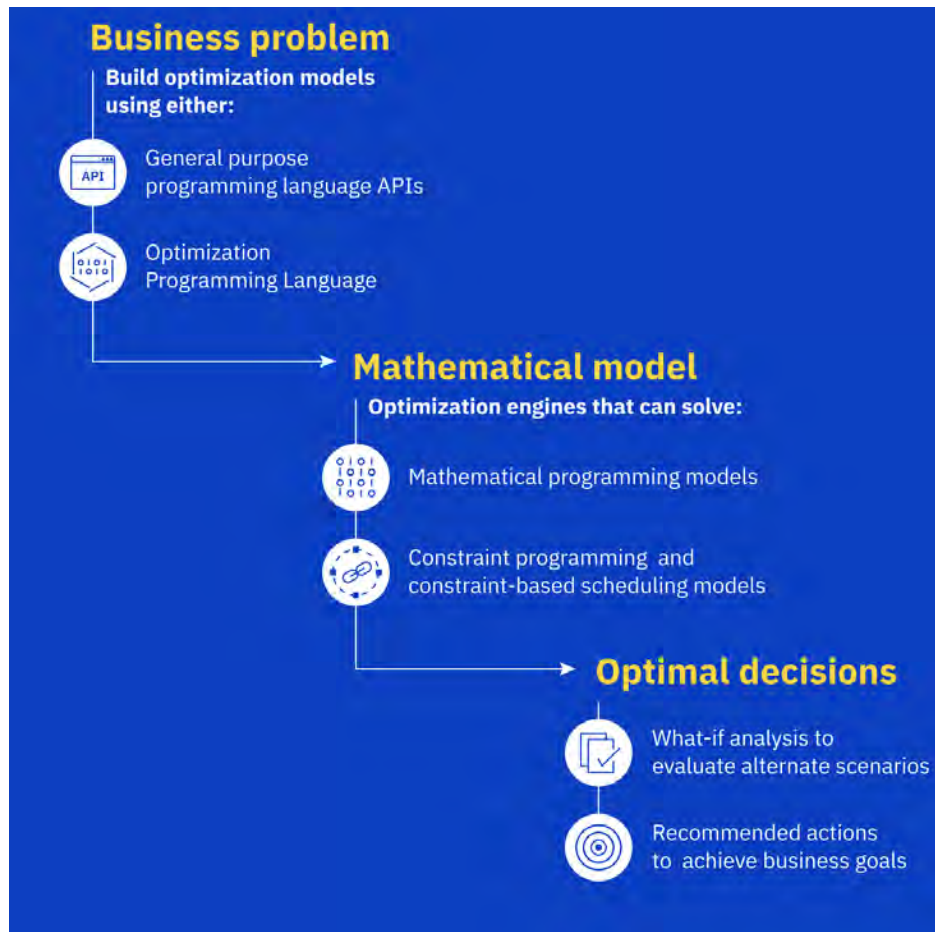


Figure 10.21: IBM ILOG CPLEX Optimization Studio uses decision optimisation technology to create optimal plans, and schedules [160].

by George Dantzig in 1947, that is written in C: C-Simplex resulted in CPLEX. The Simplex algorithm became the groundwork of the entire field of mathematical optimisation and provided the

first practical method of solving a linear programming problem. CPLEX developed over time to adapt and become a leader in the secondary categories of linear programming, such as integer programming (IP), *Mixed Integer Programming* (MIP), and the latest quadratic programming (QP). The integer programs are much more challenging to solve than linear programs, but they have a lot of essential business applications. This was IBM's bet 20 years ago, and they provided CPLEX with sophisticated mathematical techniques to solve these very hard integer programs. These techniques involve the systematic search for possible combinations of the discrete decision variables, using linear or quadratic programming relaxations to calculate the limits of the value of the optimal solution. MIP include formulation of problems such as Vehicle routing, Facility location, Personnel scheduling, . . . These techniques are interesting and necessary when one does not want to program from scratch. We will propose five different ways of solving our problem in our case. IBM ILOG CPLEX offers C, C++, Java, .NET, and Python libraries that solve linear programming (LP) and related problems. Correctly, it solves linearly or quadratically constrained optimisation problems where the objective to be optimised can be expressed as a linear function or a convex quadratic function. In addition, the variables in the model may be declared as continuous or further constrained to take only integer values. CPLEX includes the following technics for high mathematical programming solvers:

- ◇ *Problem modelling*: IBM® ILOG® CPLEX® Optimizer provides a framework to model business issues mathematically.
- ◇ *Improved profits*: IBM ILOG CPLEX Optimizer's mathematical programming provides technology to help improve efficiency, reduce costs, and increase profitability.
- ◇ *Fundamental algorithms*: IBM ILOG CPLEX Optimizer provides flexible, high-performance mathematical programming solvers for linear programming, mixed-integer programming, quadratic programming, and quadratically constrained programming problems. These include a distributed parallel algorithm for mixed integer programming to leverage multiple computers to solve difficult problems.
- ◇ *Robust algorithms for large problems*: IBM ILOG CPLEX Optimizer has solved issues with millions of constraints and variables.
- ◇ *Industry-leading support*: IBM has an impressive rate of product improvement and ample support resources to serve you.
- ◇ *High performance*: IBM ILOG CPLEX Optimizer delivers the power needed to solve large, real-world optimisation problems, as well as the speed required for today's interactive decision optimisation applications.
- ◇ *Robust and reliable*: A large installed base helps us make IBM ILOG CPLEX Optimizer better with each release. Every new feature is tested on the world's biggest, most diverse model library.

### 10.3.5 DOCPLEX

The *Decision Optimization CPLEX Modeling Python*, commonly known as docplex from IBM, is a tool with which data scientists can use their preferred scientific tools (Python, with NumPy/pandas/scipy stack) to develop optimisation-based projects. The Docplex library is composed of 2 modules:

- ◇ IBM® Decision Optimization CPLEX Optimizer Modeling for Python - with namespace docplex.mp. This library is for Mathematical Programming Modeling for Python.
- ◇ IBM® Decision Optimization CP Optimizer Modeling for Python - with namespace docplex.cp. This library id for Constraint Programming Modeling for Python.

Just add that on May 1, 2020, IBM managed to integrate into its docplex the quadratic programming for Ising model for MIP. However, until the mentioned date, This tool developed by IBM had some limitations and didn't allow inequality constraints. Last July 9, IBM unveiled a new module that will boost research, development, and benchmarking of quantum optimisation algorithms for this NISQ era[189]. To map a classical model to quantum computing, we need to find the Hamiltonian of the Ising model. Nevertheless, the Hamiltonians of the Ising model are highly complicated and have no intuitive[163][84][174]. So, mapping a combinatorial optimisation problem to the Hamiltonian of the Ising model can be very challenging, complicated, and may require specialised knowledge as the vectorisation of the matrices with the Kronecker product [85] to express matrix multiplication as a linear transformation in matrices[244]. With the translator, all kinds of users can write a quantum optimisation algorithm model using docplex. With this tool, many things become much easier than writing the Hamiltonian of the Ising model manually basically because the model is short and intuitive.

## 10.4 Summary

In this chapter, we have experimented on the different quantum programming environments, and we are ready to choose the background on which we will focus our efforts based on figure (10.10) and (10.2). Of course, this does not mean that our algorithms are only for that environment. Still, we need to make a strategic decision to move forward and, above all, make the most of the community and the libraries that offer us to do some or other tests. Let's take a look at Qiskit for its community, the facilities that IBM provides, the scientific community and the network of talent it has. As mentioned above, Qiskit already has a community of more than 25,000 developers and an available and affordable team for any questions. It has at any time for all the libraries, especially for its scalability plan. IBMQ is our bet. However, we experiment and will continue to research and monitor the other frameworks such as Dwave, AWS-Braket, Cirq, PennyLane, etc.

# Chapter 11

## Approaches to solve the SWP

### 11.1 Introduction

We design this formulation keeping in mind that our device does not admit inequality restrictions. We will encode the time information that will represent the limits of the constraints in the said formula.

This section will work on the various formulation proposals to adequately solve the proposed problem, always considering the era (NISQ) we are in and the chosen framework (Qiskit). But one of the algorithm proposals will assess the generality of the platforms. This algorithm could be executed (with very few splices) either in a Dwave Ocean, PennyLanne or a Qiskit.

### 11.2 Social Workers' Problem based on new VRP formulation

In this proposal, we take advantage of the formulation of the VRP to establish our proposal. We pursue to avoid using the inequality constraints and use the least number of qubits. We have very few real qubits with the real environments we are using. But to do some comparative studies, we would rather use simulators with a greater number of qubits than the quantum computer. To be able to test our algorithm. Therefore, we will base our algorithm on techniques (VRP, TSP) already consolidated to achieve efficiency in many qubits. We use VRP's universal formulation to model the main part of the proposed algorithm.

Let  $G = (V, E, c)$  be a complete graph directed with  $V = \{0, 1, 2, \dots, n\}$ , as the set of nodes and  $E = \{(i, j) : i, j \in V, i \neq j\}$  as the set of arcs, where node 0 represents the central, for a team of  $k$  social workers with the same capacity  $\rho$  and  $n$  remaining nodes that represent geographically dispersed patients.

Each patient  $i \in V - \{0\}$  has a specific demand for visits  $d_i \leq \rho$ . The non-negative travel cost  $W_{ij}$  is associated with each arc  $(i, j) \in E$ . To simplify, we consider that the distances are symmetrical. Where  $x_{ij}$  are the decision variables of the paths between two patients. The minimum number of social workers needed to care for all patients is  $\frac{\sum_{i=1}^n d_i}{Q}$ .

Minimize:

$$\sum_{k=1}^m \sum_{i=1}^l \sum_{j=1, i \neq j}^n W_{ij} x_{ijk}, \quad (11.1)$$

Subject to:

$$\sum_{k=1}^m \sum_{j=1}^n x_{ijk} = 1 \quad \forall i \in 1, \dots, n, \quad (11.2)$$

$$\sum_{k=1}^m \sum_{i=1}^n x_{jik} = 1 \quad \forall j \in 1, \dots, n, \quad (11.3)$$

$$\sum_{i=1}^l di \sum_{j=1}^n x_{ijk} \leq q \quad \forall k \in 1, \dots, m, \quad (11.4)$$

$$\sum_{j=1}^n x_{0jk} = K \quad \forall k \in 1, \dots, m, \quad (11.5)$$

$$\sum_{j=1}^n x_{j0k} = K, \quad \forall k \in 1, \dots, m, \quad (11.6)$$

$$\sum_{i=1}^n x_{ihk} - \sum_j x_{hjk} = 0 \quad \forall h \in 1, \dots, n \quad \text{and} \quad \forall k \in 1, \dots, m, \quad (11.7)$$

$$x_{ijk} \in 0, 1, \quad \forall i, j \in 0, \dots, n, i \neq j \quad \forall k \in 1, \dots, m. \quad (11.8)$$

In this formulation, the objective function equation (11.1) minimizes total travel savings taking into account the new cost function with the time window. The restrictions equations (11.2) and (11.3) impose that exactly the arcs  $k$  leave the plant; (11.5) and (11.6) are the restrictions on the degree of entry and exit of the social worker from the central office. The constraints (11.7) are the route of continuity and the elimination of sub-courses, which ensure that the solution does not contain a sub-route disconnected from the exchange. Restrictions (11.8) are mandatory. The restriction (11.4) is the demand of the social workers.

Up to this point, the mathematical formulation of equations (11.1) to (11.8) represents a conventional CVRP. To solve a scheduling problem, we need a time variable. The introduction of time (schedule) in the original QAOA and QUBO formulations of the CVRP is a significant obstacle to formulating several important VRP restrictions associated with the VRPTW time window.

Our CVRP QAOA formulation proposal incorporates the schedule (calendar) of table (10.1). This new formulation describes the temporal evolution of each social worker equivalent to the VRPTW. Our proposal is expressed by equations (11.9) and (11.10).

$$W_{ij} = d_{ij} + f(t_{ij}). \quad (11.9)$$

$$f(t_{ij}) = \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}}. \quad (11.10)$$

Where  $W_{ij}$  is our cost/weight and time window function,  $d_{ij}$  is the distance between the patient

$i$  and  $j$  and  $f(t_{ij})$  is our time window's function.  $f(t_{ij})$  is a growing function, and we model it by a quadratic function to weigh short distances concerning large ones. We are taking into account that the first weight function  $W_{ij} = d_{ij}$  is a distance function, we want to make  $f(t_{ij})$  behave like  $d_{ij}$ , and thus, be able to take full advantage of the behaviour of the primary objective function.  $\gamma > 0$  is a weighted degree parameter of our time window function;  $\tau_i$  is the start time of a time slot for patient  $i$  and  $\tau_j$  for the patient  $j$ . where  $d_{\max}$  represents the maximum distance between all patients and,  $d_{\min}$  is the minimum distance between the gaps of all patients. The term  $T_{ij} = (\tau_i - \tau_j) > 0$  is the time window.

The simplified Hamiltonian resulting from the schedule optimization problem is as follows:

$$H = \sum_{ij \in E} W_{ij} x_{i,j} + A \sum_{i=1}^n \left(1 - \sum_{j \in \delta(i)^+} x_{i,j}\right)^2 + A \sum_{i=1}^n \left(1 - \sum_{j \in \delta(i)^-} x_{ji}\right)^2 + A \left(k - \sum_{i \in \delta(0)^+} x_{0,i}\right)^2 + A \left(k - \sum_{j \in \delta(0)^+} x_{j,0}\right)^2. \quad (11.11)$$

### 11.2.1 Social Workers' Problem based on a QUBO approach

Within the pedagogical nature of this work, we will detail the resolution of the problem step by step to meet one of the objectives of the thesis.

Let us solve our formulation in QUBO form by considering  $n = 4$  patients. Where  $Q$  is a  $2^N \times 2^N$  matrix with  $N = n(n-1)$  as the number of qubits. So, in this case,  $N = 12$  qubits. Let's remember that we are using binary variables so,  $(x_{u,v}^i)^2 = x_{u,v}^i$ .

$$\sum_{ij \in E} W_{ij} x_{i,j} = \sum_i^3 \sum_{\substack{j \\ i \neq j}}^3 W_{ij} x_{i,j} = \begin{pmatrix} W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + \\ W_{1,0}x_{1,0} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,0}x_{2,0} + W_{2,1}x_{2,1} + \\ W_{2,3}x_{2,3} + W_{3,0}x_{3,0} + W_{3,1}x_{3,1} + W_{3,2}x_{3,2} \end{pmatrix}. \quad (11.12)$$

$$\begin{aligned} & A \sum_{i=1}^3 \left(1 - \sum_{j \in \delta(i)^-} x_{ji}\right)^2 \\ &= A(1 - x_{0,1} - x_{2,1} - x_{3,1} + 1 - x_{0,2} - x_{1,2} - x_{3,3} + 1 - x_{0,3} - x_{1,3} - x_{2,3})^2 \\ &= A(3 - x_{0,1} - x_{2,1} - x_{3,1} - x_{0,2} - x_{1,2} - x_{3,2} - x_{0,3} - x_{1,3} - x_{2,3})^2 \end{aligned}$$



$$\begin{aligned}
&= A \left( \begin{aligned}
&3^2 + (x_{0,1})^2 + (x_{2,1})^2 + (x_{3,1})^2 + (x_{0,2})^2 + (x_{1,2})^2 + (x_{3,2})^2 + (x_{0,3})^2 + (x_{1,3})^2 + (x_{2,3})^2 \\
&\quad - 6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} - 6x_{1,3} - 6x_{2,3} \\
&+ 2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} \\
&\quad + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} \\
&\quad + 2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} \\
&\quad + 2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} \\
&\quad + 2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\
&\quad + 2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} \\
&\quad + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} \\
&\quad + 2x_{1,3}x_{2,3}
\end{aligned} \right),
\end{aligned}$$

$$A \sum_{i=1}^3 \left( 1 - \sum_{j \in \delta(i)^+} x_{i,j} \right)^2$$

$$= A (1 - x_{1,0} - x_{1,2} - x_{1,3} + 1 - x_{2,0} - x_{2,1} - x_{2,3} + 1 - x_{3,0} - x_{3,1} - x_{3,2})^2$$

$$= A (3 + -x_{1,0} - x_{1,2} - x_{1,3} - x_{2,0} - x_{2,1} - x_{2,3} - x_{3,0} - x_{3,1} - x_{3,2})^2$$

$$\begin{aligned}
&= A \left( \begin{aligned}
&3^2 + (x_{1,0})^2 + (x_{1,2})^2 + (x_{1,3})^2 + (x_{2,0})^2 + (x_{2,1})^2 + (x_{2,3})^2 + (x_{3,0})^2 + (x_{3,1})^2 + (x_{3,2})^2 \\
&\quad - 6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} - 6x_{3,1} - 6x_{3,2} \\
&+ 2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} \\
&\quad + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} \\
&\quad + 2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} \\
&\quad + 2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} \\
&\quad + 2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\
&\quad + 2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} \\
&\quad + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} \\
&\quad + 2x_{3,1}x_{3,2}
\end{aligned} \right)
\end{aligned}$$

$$\begin{aligned}
& A \left( k - \sum_{i \in \delta(0)^+}^3 x_{0,i} \right)^2 = A (k - x_{0,1} - x_{0,2} - x_{0,3})^2 \\
& = A \left( \begin{array}{c} k^2 + (x_{0,1})^2 + (x_{0,2})^2 + (x_{0,3})^2 - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} \\ + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3} \end{array} \right) \\
& = A \left( k^2 + (x_{0,1})^2 + (x_{0,2})^2 + (x_{0,3})^2 - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3} \right),
\end{aligned}$$

$$\begin{aligned}
& A \left( k - \sum_{j \in \delta(0)^+}^3 x_{j,0} \right)^2 = A (k - x_{1,0} - x_{2,0} - x_{3,0})^2 \\
& = A \left( \begin{array}{c} k^2 + (x_{1,0})^2 + (x_{2,0})^2 + (x_{3,0})^2 - 2kx_{1,0} - 2kx_{2,0} - 2kx_{3,0} \\ + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0} \end{array} \right) \\
& = A \left( k^2 + (x_{1,0})^2 + (x_{2,0})^2 + (x_{3,0})^2 - 2kx_{1,0} - 2kx_{2,0} - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0} \right),
\end{aligned}$$

Grouping the terms (11.12) to (11.16)) we reach out to the following expression:

$$\begin{aligned}
& W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3} + \\
& A \left( \begin{array}{c} +3^2 + x_{0,1}^2 + x_{2,1}^2 + x_{3,1}^2 + x_{0,2}^2 + x_{1,2}^2 + x_{3,2}^2 + x_{0,3}^2 + x_{1,3}^2 + x_{2,3}^2 \\ -6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} - 6x_{1,3} - 6x_{2,3} \\ +2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} \\ +2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} \\ +2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} \\ +2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} \\ +2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\ +2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} \\ +2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} \\ +2x_{1,3}x_{2,3} \end{array} \right),
\end{aligned}$$

$$\begin{aligned}
& +A \left( \begin{aligned}
& 3^2 + (x_{1,0})^2 + (x_{1,2})^2 + (x_{1,3})^2 + (x_{2,0})^2 + (x_{2,1})^2 + (x_{2,3})^2 + (x_{3,0})^2 + (x_{3,1})^2 + (x_{3,2})^2 \\
& -6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} - 6x_{3,1} - 6x_{3,2} \\
& +2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} \\
& +2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} \\
& +2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} \\
& +2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} \\
& +2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\
& +2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} \\
& +2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} \\
& +2x_{3,1}x_{3,2}
\end{aligned} \right) + \\
& A \left( k^2 + (x_{0,1})^2 + (x_{0,2})^2 + (x_{0,3})^2 - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3} \right) \\
& +A \left( k^2 + (x_{1,0})^2 + (x_{2,0})^2 + (x_{3,0})^2 - 2kx_{1,0} - 2kx_{2,0} - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0} \right), \quad (11.17)
\end{aligned}$$

Now let's apply the binary variable property  $x_{i,j}^2 = x_{i,j}$  so,

$$W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3}$$

$$\begin{aligned}
& +A \left( \begin{aligned}
& 3^2 + x_{0,1} + x_{2,1} + x_{3,1} + x_{0,2} + x_{1,2} + x_{3,2} + x_{0,3} + x_{1,3} + x_{2,3} \\
& -6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} - 6x_{1,3} - 6x_{2,3} \\
& +2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} \\
& +2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} \\
& +2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} \\
& +2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} \\
& +2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\
& +2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} \\
& +2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} \\
& +2x_{1,3}x_{2,3}
\end{aligned} \right) +
\end{aligned}$$

$$\begin{aligned}
& A \left( \begin{aligned}
& 3^2 + x_{1,0} + x_{1,2} + x_{1,3} + x_{2,0} + x_{2,1} + x_{2,3} + x_{3,0} + x_{3,1} + x_{3,2} \\
& -6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} - 6x_{3,1} - 6x_{3,2} \\
& +2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} \\
& +2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} \\
& +2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} \\
& +2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} \\
& +2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\
& +2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} \\
& +2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} \\
& +2x_{3,1}x_{3,2}
\end{aligned} \right) + \\
& A (k^2 + x_{0,1} + x_{0,2} + x_{0,3} - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) \\
& + A (k^2 + x_{1,0} + x_{2,0} + x_{3,0} - 2kx_{1,0} - 2kx_{2,0} - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0}) \quad (11.18)
\end{aligned}$$

Grouping similar terms:

$$W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3}$$

$$+A \left( \begin{array}{c} 3^2 \\ -5x_{0,1} - 5x_{2,1} - 5x_{3,1} - 5x_{0,2} - 5x_{1,2} - 5x_{3,2} - 5x_{0,3} - 5x_{1,3} - 5x_{2,3} \\ +2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} \\ +2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} \\ +2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} \\ +2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} \\ +2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\ +2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} \\ +2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} \\ +2x_{1,3}x_{2,3} \end{array} \right)$$

$$+A \left( \begin{array}{c} 3^2 \\ -5x_{1,0} - 5x_{1,2} - 5x_{1,3} - 5x_{2,0} - 5x_{2,1} - 5x_{2,3} - 5x_{3,0} - 5x_{3,1} - 5x_{3,2} \\ +2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} \\ +2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} \\ +2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} \\ +2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} \\ +2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\ +2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} \\ +2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} \\ +2x_{3,1}x_{3,2} \end{array} \right) +$$

$A(k^2 + x_{0,1} + x_{0,2} + x_{0,3} - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) + A(k^2 + x_{1,0} + x_{2,0} + x_{3,0} - 2kx_{1,0} - 2kx_{2,0} - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0})$  (11.19) Now let's group the

linear terms as following:

$$\begin{aligned}
& 2A3^2 + 2Ak^2 + (W_{0,1} + A(1 - 5 - 2k))x_{0,1} + (W_{0,2} + A(1 - 5 - 2k))x_{0,2} + \\
& (W_{0,3} + A(1 - 5 - 2k))x_{0,3} + A(W_{1,0} - A(1 - 5 - 2k))x_{1,0} + A(W_{2,0} - A(1 - 5 - 2k))x_{2,0} \\
& + A(W_{3,0} - A(1 - 5 - 2k))x_{3,0} + (W_{1,2} - 10A)x_{1,2} + (W_{1,3} - 10A)x_{1,3} + (W_{2,1} - 10A)x_{2,1} \\
& + (W_{2,3} - 10A)x_{2,3} - (W_{3,1} - 10A)x_{3,1} - (W_{3,2} - 10A)x_{3,2}
\end{aligned} \tag{11.20}$$

Now let's group Quadratic terms.

$$\begin{aligned}
& + 4Ax_{0,1}x_{0,2} + 4Ax_{0,1}x_{0,3} + 4Ax_{0,2}x_{0,3} \\
& + 2Ax_{0,1}x_{1,2} + 2Ax_{0,1}x_{1,3} + 2Ax_{0,1}x_{2,1} + 2Ax_{0,1}x_{2,3} + 2Ax_{0,1}x_{3,1} + 2Ax_{0,1}x_{3,2} + 4Ax_{1,0}x_{2,0} + 4x_{1,0}x_{3,0} \\
& + 4Ax_{2,0}x_{3,0} + 2Ax_{1,0}x_{1,2} + 2Ax_{1,0}x_{1,3} + 2Ax_{1,0}x_{2,1} + 2Ax_{1,0}x_{2,3} + 2Ax_{1,0}x_{3,1} + 2Ax_{1,0}x_{3,2} \\
& + 2Ax_{2,1}x_{1,2} + 2Ax_{2,1}x_{1,3} + 2Ax_{2,1}x_{3,0} + 4Ax_{2,1}x_{3,1} + 4Ax_{2,1}x_{3,2} + 2Ax_{2,1}x_{0,2} + 2Ax_{2,1}x_{0,3} + 4Ax_{2,1}x_{2,3} \\
& + 2Ax_{1,2}x_{0,3} + 2Ax_{1,2}x_{2,0} + 2Ax_{1,2}x_{2,1} + 2Ax_{1,2}x_{3,0} + 2Ax_{1,2}x_{3,1} + 4Ax_{1,2}x_{1,3} + 4Ax_{1,2}x_{3,2} + 4Ax_{1,2}x_{2,3} \\
& + 2Ax_{3,1}x_{0,2} + 2Ax_{3,1}x_{1,2} + 4Ax_{3,1}x_{3,2} + 2Ax_{3,1}x_{0,3} + 2Ax_{3,1}x_{1,3} + 2Ax_{3,1}x_{2,3} \\
& + 2Ax_{1,3}x_{2,0} + 2Ax_{1,3}x_{2,1} + 4Ax_{1,3}x_{2,3} + 2Ax_{1,3}x_{3,0} + 2Ax_{1,3}x_{3,1} + 2Ax_{1,3}x_{3,2} \\
& + 2Ax_{0,2}x_{1,2} + 2Ax_{0,2}x_{3,2} + 2Ax_{0,2}x_{1,3} + 2Ax_{0,2}x_{2,3} \\
& + 2Ax_{2,0}x_{2,1} + 2Ax_{2,0}x_{2,3} + 2Ax_{2,0}x_{3,1} + 2Ax_{2,0}x_{3,2} \\
& + 2Ax_{3,2}x_{0,3} + 2Ax_{3,2}x_{1,3} + 2Ax_{3,2}x_{2,3} \\
& + 2Ax_{2,3}x_{3,0} + 2Ax_{2,3}x_{3,1} + 2Ax_{2,3}x_{3,2} \\
& + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} \\
& + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2}
\end{aligned} \tag{11.21}$$

Recalling the quadratic form  $x^T Qx + g^T x + C$  with the following terms.

	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$	$x_{1,0}$	$x_{1,2}$	$x_{1,3}$	$x_{2,0}$	$x_{2,1}$	$x_{2,3}$	$x_{3,0}$	$x_{3,1}$	$x_{3,2}$
$x_{0,1}$	0	4	4	0	2	2	0	2	2	0	2	2
$x_{0,2}$	0	0	4	0	2	2	0	2	2	0	2	2
$x_{0,3}$	0	0	0	0	2	2	0	2	2	0	2	2
$x_{1,0}$	0	0	0	0	2	2	4	2	2	4	2	2
$x_{1,2}$	0	0	0	0	0	4	2	2	4	2	2	4
$x_{1,3}$	0	0	0	0	0	0	2	2	4	2	2	2
$x_{2,0}$	0	0	0	0	0	0	0	2	2	4	2	2
$x_{2,1}$	0	0	0	0	0	0	0	0	4	2	4	4
$x_{2,3}$	0	0	0	0	0	0	0	0	0	2	2	2
$x_{3,0}$	0	0	0	0	0	0	0	0	0	0	2	2
$x_{3,1}$	0	0	0	0	0	0	0	0	0	0	0	4
$x_{3,2}$	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.1: QUBO  $Q$  Matrix for the Social Workers' Problem

$$X = \begin{bmatrix} x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \end{bmatrix} \quad Q = \begin{bmatrix} 0 & 4 & 4 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 4 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 & 4 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 2 & 2 & 4 & 2 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad g = \begin{bmatrix} W_{0,1} - A(4 + 2k) \\ W_{0,2} - A(4 + 2k) \\ W_{0,3} - A(4 + 2k) \\ W_{1,0} - A(4 + 2k) \\ W_{1,2} - 10A \\ W_{1,3} - 10A \\ W_{2,0} - A(4 + 2k) \\ W_{2,1} - 10A \\ W_{2,3} - 10A \\ W_{3,0} - A(4 + 2k) \\ W_{3,1} - 10A \\ W_{3,2} - 10A \end{bmatrix}, \quad (11.22)$$

and

$$C = 2Ak^2 + 2A(n - 1)^2, \quad (11.23)$$

We put equations(11.9) and (11.10) together, and we arrive at the following expression:

$$W_{ij} = d_{ij} + \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}}, \quad (11.24)$$

There are several strategies to implement the objective function with inequality constraints, but these strategies require additional variables that work as a stack. The approach most frequently used in the scientific community is to use auxiliary binary variables to convert inequality to equality and

	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$	$J_6$	$J_7$	$J_8$	$J_9$	$J_{10}$	$J_{11}$	$J_{12}$
$J_1$	0	4	4	0	2	2	0	2	2	0	2	2
$J_2$	0	0	4	0	2	2	0	2	2	0	2	2
$J_3$	0	0	0	0	2	2	0	2	2	0	2	2
$J_4$	0	0	0	0	2	2	4	2	2	4	2	2
$J_5$	0	0	0	0	0	4	2	2	4	2	2	4
$J_6$	0	0	0	0	0	0	2	2	4	2	2	2
$J_7$	0	0	0	0	0	0	0	2	2	4	2	2
$J_8$	0	0	0	0	0	0	0	0	4	2	4	4
$J_9$	0	0	0	0	0	0	0	0	0	2	2	2
$J_{10}$	0	0	0	0	0	0	0	0	0	0	2	2
$J_{11}$	0	0	0	0	0	0	0	0	0	0	0	4
$J_{12}$	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.2:  $J_{ij}$  Interaction forces between grid neighbours. We assume that  $J_{ij} = 0$  for  $i$  and  $j$  are not adjacent.

then proceed, as usual, by squaring the equality constraint following penalty theory. These additional variables are translated into qubits; today's extra qubits are scarce for some experiments. To do this, we define a strategy that allows us to solve combinatorial optimisation problems with inequality constraints without increasing the number of qubits required. Our strategy is based on coding the variables of time inequality, the time window following the formulation (11.24).

However, IBM's significant contribution\* opens up promising horizons in quadratic programming with inequality constraints.

Now we only have to code the information in table (10.1), the data (distance, costs and correction) of each patient and generate our weight matrix, which in turn will serve to calculate our variables linear  $g$ . With all this, we already have all the components for one, write our objective function in the form QUBO and second solve it.

As we already have our objective function as a QUBO in the form  $\langle x^T | Q | x \rangle$  and as we have shown in chapter (9), based on equations (9.14) and (9.15), and summarized in table (9.1), going from the QUBO to Ising formulation leads to calculating the values of  $J_{ij}$  and  $h_i$ .

Next, we calculate these variables. We start with  $J_{ij}$  as is summarized in table (11.2).

$$J_{ij} = \begin{cases} \frac{q_{i,j} + q_{i,i}}{4} & i < j \\ 0 & otherwise \end{cases} \quad (11.25)$$

Let us calculate the external forces  $h_i$ :

$$h_i = \frac{1}{4} \sum_{k=1}^N (q_{i,k} + q_{k,i}) h_i = \frac{1}{4} \begin{bmatrix} (q_{i,1} + q_{1,i}) + (q_{i,2} + q_{3,i}) + (q_{i,3} + q_{3,i}) \\ + (q_{i,4} + q_{4,i}) + (q_{i,5} + q_{5,i}) + (q_{i,6} + q_{6,i}) \\ + (q_{i,7} + q_{7,i}) + (q_{i,8} + q_{8,i}) + (q_{i,9} + q_{9,i}) \\ + (q_{i,10} + q_{10,i}) + (q_{i,11} + q_{11,i}) + (q_{i,12} + q_{12,i}) \end{bmatrix}. \quad (11.26)$$

\*<https://medium.com/qiskit/towards-quantum-advantage-for-optimization-with-qiskit-9a564339ef26>



Now let calculate  $i = 1$ .

$$\begin{aligned}
 h_1 &= \frac{1}{4} \left[ \begin{array}{l} (q_{1,1} + q_{1,1}) + (q_{1,2} + q_{2,1}) + (q_{1,3} + q_{3,1}) \\ + (q_{1,4} + q_{4,1}) + (q_{1,5} + q_{5,1}) + (q_{1,6} + q_{6,1}) \\ + (q_{1,7} + q_{7,1}) + (q_{1,8} + q_{8,1}) + (q_{1,9} + q_{9,1}) \\ + (q_{1,10} + q_{10,1}) + (q_{1,11} + q_{11,1}) + (q_{1,12} + q_{12,1}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 0) + (4 + 0) + (4 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = 5.
 \end{aligned} \tag{11.27}$$

Where  $h_1 = 5$ . Now let calculate  $i = 2$ .

$$\begin{aligned}
 h_2 &= \frac{1}{4} \left[ \begin{array}{l} (q_{2,1} + q_{1,2}) + (q_{2,2} + q_{2,2}) + (q_{2,3} + q_{3,2}) \\ + (q_{2,4} + q_{4,2}) + (q_{2,5} + q_{5,2}) + (q_{2,6} + q_{6,2}) \\ + (q_{2,7} + q_{7,2}) + (q_{2,8} + q_{8,2}) + (q_{2,9} + q_{9,2}) \\ + (q_{2,10} + q_{10,2}) + (q_{2,11} + q_{11,2}) + (q_{2,12} + q_{12,2}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 4) + (0 + 0) + (4 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = 5.
 \end{aligned} \tag{11.28}$$

Where  $h_2 = 5$

Now let calculate  $i = 3$ .

$$\begin{aligned}
 h_3 &= \frac{1}{4} \left[ \begin{array}{l} (q_{3,1} + q_{1,3}) + (q_{3,2} + q_{2,3}) + (q_{3,3} + q_{3,3}) \\ + (q_{3,4} + q_{4,3}) + (q_{3,5} + q_{5,3}) + (q_{3,6} + q_{6,3}) \\ + (q_{3,7} + q_{7,3}) + (q_{3,8} + q_{8,3}) + (q_{3,9} + q_{9,3}) \\ + (q_{3,10} + q_{10,3}) + (q_{3,11} + q_{11,3}) + (q_{3,12} + q_{12,3}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 4) + (0 + 4) + (0 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = 5.
 \end{aligned} \tag{11.29}$$

Where  $h_3 = 5$  Now let calculate  $i = 4$ .

$$\begin{aligned}
 h_4 &= \frac{1}{4} \left[ \begin{array}{l} (q_{4,1} + q_{1,4}) + (q_{4,2} + q_{2,4}) + (q_{4,3} + q_{3,4}) \\ + (q_{4,4} + q_{4,4}) + (q_{4,5} + q_{5,4}) + (q_{4,6} + q_{6,4}) \\ + (q_{4,7} + q_{7,4}) + (q_{4,8} + q_{8,4}) + (q_{4,9} + q_{9,4}) \\ + (q_{4,10} + q_{10,4}) + (q_{4,11} + q_{11,4}) + (q_{4,12} + q_{12,4}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 0) + (0 + 0) + (0 + 0) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (4 + 0) + (2 + 0) + (2 + 0) \\ + (4 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = 5.
 \end{aligned} \tag{11.30}$$

Where  $h_4 = 5$  Now let calculate  $i = 5$ .

$$\begin{aligned}
 h_5 &= \frac{1}{4} \left[ \begin{array}{l} (q_{5,1} + q_{1,5}) + (q_{5,2} + q_{2,5}) + (q_{5,3} + q_{3,5}) \\ + (q_{5,4} + q_{4,5}) + (q_{5,5} + q_{5,5}) + (q_{5,6} + q_{6,5}) \\ + (q_{5,7} + q_{7,5}) + (q_{5,8} + q_{8,5}) + (q_{5,9} + q_{9,5}) \\ + (q_{5,10} + q_{10,5}) + (q_{5,11} + q_{11,5}) + (q_{5,12} + q_{12,5}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 0) + (4 + 0) \\ + (0 + 2) + (2 + 0) + (4 + 0) \\ + (2 + 0) + (2 + 0) + (4 + 0) \end{array} \right] = 7.
 \end{aligned} \tag{11.31}$$

Where  $h_5 = 7$  Now let calculate  $i = 6$ .

$$\begin{aligned}
 h_6 &= \frac{1}{4} \left[ \begin{array}{l} (q_{6,1} + q_{1,6}) + (q_{6,2} + q_{2,6}) + (q_{6,3} + q_{3,6}) \\ + (q_{6,4} + q_{4,6}) + (q_{6,5} + q_{5,6}) + (q_{6,6} + q_{6,6}) \\ + (q_{6,7} + q_{7,6}) + (q_{6,8} + q_{8,6}) + (q_{6,9} + q_{9,6}) \\ + (q_{6,10} + q_{10,6}) + (q_{6,11} + q_{11,6}) + (q_{6,12} + q_{12,6}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 4) + (0 + 0) \\ + (2 + 0) + (2 + 0) + (4 + 0) \\ + (2 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = \frac{13}{2}.
 \end{aligned} \tag{11.32}$$

Where  $h_6 = 6, 5$

Now let calculate  $i = 7$ .

$$\begin{aligned}
 h_7 &= \frac{1}{4} \left[ \begin{array}{l} (q_{7,1} + q_{1,7}) + (q_{7,2} + q_{2,7}) + (q_{7,3} + q_{3,7}) \\ + (q_{7,4} + q_{4,7}) + (q_{7,5} + q_{5,7}) + (q_{7,6} + q_{6,7}) \\ + (q_{7,7} + q_{7,7}) + (q_{7,8} + q_{8,7}) + (q_{7,9} + q_{9,7}) \\ + (q_{7,10} + q_{10,7}) + (q_{7,11} + q_{11,7}) + (q_{7,12} + q_{12,7}) \end{array} \right] = \\
 \frac{1}{4} \left[ \begin{array}{l} (0 + 0) + (0 + 0) + (0 + 0) \\ + (0 + 4) + (0 + 2) + (0 + 2) \\ + (0 + 0) + (2 + 0) + (2 + 0) \\ + (4 + 0) + (2 + 0) + (2 + 0) \end{array} \right] &= 5.
 \end{aligned} \tag{11.33}$$

Where  $h_7 = 5$

Now let calculate  $i = 8$ .

$$\begin{aligned}
 h_8 &= \frac{1}{4} \left[ \begin{array}{l} (q_{8,1} + q_{1,8}) + (q_{8,2} + q_{2,8}) + (q_{8,3} + q_{3,8}) \\ + (q_{8,4} + q_{4,8}) + (q_{8,5} + q_{5,8}) + (q_{8,6} + q_{6,8}) \\ + (q_{8,7} + q_{7,8}) + (q_{8,8} + q_{8,8}) + (q_{8,9} + q_{9,8}) \\ + (q_{8,10} + q_{10,8}) + (q_{8,11} + q_{11,8}) + (q_{8,12} + q_{12,8}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 0) + (4 + 0) \\ + (2 + 0) + (4 + 0) + (4 + 0) \end{array} \right] = 7.
 \end{aligned} \tag{11.34}$$

Where  $h_8 = 7$

Now let calculate  $i = 9$ .

$$\begin{aligned}
 h_9 &= \frac{1}{4} \left[ \begin{array}{l} (q_{9,1} + q_{1,9}) + (q_{9,2} + q_{2,9}) + (q_{9,3} + q_{3,9}) \\ + (q_{9,4} + q_{4,9}) + (q_{9,5} + q_{5,9}) + (q_{9,6} + q_{6,9}) \\ + (q_{9,7} + q_{7,9}) + (q_{9,8} + q_{8,9}) + (q_{9,9} + q_{9,9}) \\ + (q_{9,10} + q_{10,9}) + (q_{9,11} + q_{11,9}) + (q_{9,12} + q_{12,9}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 4) + (0 + 4) \\ + (0 + 2) + (0 + 4) + (0 + 0) \\ + (2 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = 7.
 \end{aligned} \tag{11.35}$$

Where  $h_9 = 7$

Now let calculate  $i = 10$ .

$$\begin{aligned}
 h_{10} &= \frac{1}{4} \left[ \begin{array}{l} (q_{10,1} + q_{1,10}) + (q_{10,2} + q_{2,10}) + (q_{10,3} + q_{3,10}) \\ + (q_{10,4} + q_{4,10}) + (q_{10,5} + q_{5,10}) + (q_{10,6} + q_{6,10}) \\ + (q_{10,7} + q_{7,10}) + (q_{10,8} + q_{8,10}) + (q_{10,9} + q_{9,10}) \\ + (q_{10,10} + q_{10,10}) + (q_{10,11} + q_{11,10}) + (q_{10,12} + q_{12,10}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 0) + (0 + 0) + (0 + 0) \\ + (0 + 4) + (0 + 2) + (0 + 2) \\ + (0 + 4) + (0 + 2) + (0 + 2) \\ + (0 + 0) + (2 + 0) + (2 + 0) \end{array} \right] = \frac{9}{2}.
 \end{aligned} \tag{11.36}$$

Where  $h_{10} = 4, 5$

Now let calculate  $i = 11$ .

$$\begin{aligned}
 h_{11} &= \frac{1}{4} \left[ \begin{array}{l} (q_{11,1} + q_{1,11}) + (q_{11,2} + q_{2,11}) + (q_{11,3} + q_{3,11}) \\ + (q_{11,4} + q_{4,11}) + (q_{11,5} + q_{5,11}) + (q_{11,6} + q_{6,11}) \\ + (q_{11,7} + q_{7,11}) + (q_{11,8} + q_{8,11}) + (q_{11,9} + q_{9,11}) \\ + (q_{11,10} + q_{10,11}) + (q_{11,11} + q_{11,11}) + (q_{11,12} + q_{12,11}) \end{array} \right] \\
 &= \frac{1}{4} \left[ \begin{array}{l} (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 2) + (0 + 2) \\ + (0 + 2) + (0 + 4) + (0 + 2) \\ + (0 + 2) + (0 + 0) + (4 + 0) \end{array} \right] = \frac{13}{2}.
 \end{aligned} \tag{11.37}$$

Where  $h_{11} = 6, 5$

$h_1$	5
$h_2$	5
$h_3$	5
$h_4$	5
$h_5$	7
$h_6$	6.5
$h_7$	5
$h_8$	7
$h_9$	7
$h_{10}$	4.5
$h_{11}$	6.5
$h_{12}$	7

Table 11.3: Calculated values of the external force  $h_i$

Now let calculate  $i = 12$ .

$$\begin{aligned}
h_{12} &= \frac{1}{4} \left[ \begin{aligned} &(q_{12,1} + q_{1,12}) + (q_{12,2} + q_{2,12}) + (q_{12,3} + q_{3,12}) \\ &+ (q_{12,4} + q_{4,12}) + (q_{12,5} + q_{5,12}) + (q_{12,6} + q_{6,12}) \\ &+ (q_{12,7} + q_{7,12}) + (q_{12,8} + q_{8,12}) + (q_{12,9} + q_{9,12}) \\ &+ (q_{12,10} + q_{10,12}) + (q_{12,11} + q_{11,12}) + (q_{12,12} + q_{12,12}) \end{aligned} \right] \\
&= \frac{1}{4} \left[ \begin{aligned} &(0 + 2) + (0 + 2) + (0 + 2) \\ &+ (0 + 2) + (0 + 4) + (0 + 2) \\ &+ (0 + 2) + (0 + 4) + (0 + 2) \\ &+ (0 + 2) + (0 + 4) + (0 + 0) \end{aligned} \right] = 7.
\end{aligned} \tag{11.38}$$

Where  $h_{12} = 7$

With the calculated  $J_{ij}$  (table (11.2)) and  $h_i$  (table (11.3)), we can now solve our Social Workers' Problem with VQE  $\langle \psi(\theta) | H | \psi(\theta) \rangle$  or QAOA  $\langle \vec{\gamma}, \vec{\beta} | H | \vec{\gamma}, \vec{\beta} \rangle$ , or with any mentioned variational method. We are considering that the superposition state  $N$  qubits of  $Q = q_1 \cdots q_N$  is described by  $|\psi\rangle = |\psi_1 \cdots \psi_N\rangle$ .

$$H_P |\psi\rangle = \sum_i h_i \sigma_i^z + \sum_{i < j} J_{ij} \sigma_i^z \sigma_j^z |\psi\rangle. \tag{11.39}$$

Where the energy function is as follow and where the notation  $\sigma_i^z$  means that the Pauli-Z operator is applied to the single-qubit  $= |\cdots \psi_i \cdots\rangle$ .

$$\epsilon_I = \sum_{i \in V} h_i (-1)^{\psi_i} + \sum_{(i,j) \in V} J_{ij} (-1)^{\psi_i} (-1)^{\psi_j}. \tag{11.40}$$

Another view of the equation (11.40) is given by (11.41) and (11.42).

$$\sigma_i^z \longrightarrow \underbrace{I}_{1^{st} \text{ position}} \otimes \cdots \otimes \underbrace{\overset{z}{\sigma}}_{i^{th} \text{ position}} \otimes \cdots \otimes \underbrace{I}_{N^{th} \text{ position}}, \quad (11.41)$$

and

$$\sigma_i^z \sigma_j^z \xrightarrow{i < j} \underbrace{I}_{1^{st} \text{ position}} \otimes \cdots \otimes \underbrace{\overset{z}{\sigma}}_{i^{th} \text{ position}} \otimes \cdots \otimes \underbrace{\overset{z}{\sigma}}_{j^{th} \text{ position}} \otimes \cdots \otimes \underbrace{I}_{N^{th} \text{ position}}. \quad (11.42)$$

## 11.2.2 The algebraic approach of the Hamiltonian Formulation

We are looking for a more compact Hamiltonian way to make it easier to codify in quantum. To do this, we vectorise the decision variables  $x_{ij}$ . We know that vectorization [176][222] of a matrix is a linear transformation that converts the matrix into a column vector. So, let  $vec(X)$ , the column vector  $mn \times 1$  obtained by stacking the columns of the matrix  $X$  one on top of the other:

$$vec(X) = [x_{1,1}, \dots, x_{1,m}, \dots, x_{1,n}, \dots, x_{m,n}]^T. \quad (11.43)$$

We also know that vectorisation is frequently used with the Kronecker product to express matrix multiplication as a linear transformation in matrices. In particular.

$$\begin{aligned} vec(ABC) &= (C^T \otimes A)vec(B)vec(ABC) = (I^T \otimes AB)vec(C) = \\ &= (C^T B^T \otimes I)vec(A)vec(AB) = (I^T \otimes A)vec(B) = (B^T \otimes I)vec(A). \end{aligned} \quad (11.44)$$

If we apply vectorization as a linear sum, the matrix vectorization operation can be written in terms of a linear sum. Let  $e_i$  be the  $n$ th canonical base vector for  $n$  dimensional space, that is:

$$e_i = [0, \dots, 0, \dots, 0, \dots, 1, \dots, 0, \dots, 0]^T. \quad (11.45)$$

Let  $B_i$  a block matrix  $mn \times m$  defined as follows:

$$B_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e_i \otimes I_m. \quad (11.46)$$

$B_i$  consists of  $n$  block matrices of size  $m \times m$ , stacked in columns, and all these matrices are all zero except the  $i$ th, which is an identity matrix  $m \times m I_m$ .

Then the vectorized version of  $X$  can be expressed as follows:

$$vec(X) = \sum_{i=1}^n B_i X e_i. \quad (11.47)$$

The multiplication of  $X$  by  $e_i$  extracts the  $i$ th column, while the multiplication by  $B_i$  places it in the desired position in the final vector. Alternatively, the linear sum can be expressed using the Kronecker product:

$$vec(X) = \sum_{i=1}^n (e_i \otimes I_m) X e_i = \sum_{i=1}^n (e_i \otimes I_m) X^T. \quad (11.48)$$

With  $X e_i = X^T$ . Where  $^T$  is the transpose.

We can simplify our objective function to be implemented in quantum.

### 11.2.3 Vectorization form of our formulation

The formulation of quantum annealing has been described in countless articles [163] [236] [230] as one of the combinatorial optimisation algorithms solving approaches. Moreover, this formulation is the basis of the new techniques called to be the flagship of quantum computation. Our QAOA/VQE formulation takes after reference [233], showing the detailed development for our case. Let the vector  $Z$  of the decision variables  $x_{ij}$  with  $Z \in 0, 1^N$  and  $N = n(n+1)$ :

$$Z = [x_{01}, x_{02}, x_{03}, \dots, x_{10}, x_{12}, x_{13}, \dots, x_{n(n-1)}]^T. \quad (11.49)$$

And applying equation (11.49) and (11.48) taking into account that the decision variable is now  $Z$ , we arrive at equation (11.50).

$$H = W^T Z + A \sum_{i=1}^n (1 - (e_{in}^T Z))^2 + A \sum_{i=1}^n (1 - v_i^T Z)^2 + A (k - (e_{0n})^T Z)^2 + A (k - v_0^T Z)^2. \quad (11.50)$$

With:

$$W_{ij} = d_{ij} + \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}}. \quad (11.51)$$

Now, let's develop the resultant equation.

$$\begin{aligned}
H &= W^T Z + A \sum_{i=1}^n \left( (e_{in}^T Z)^2 - 2(e_{in}^T z + 1) \right) + A \sum_{i=1}^n \left( (v_i^T Z)^2 - 2v_i^T Z + 1 \right) \\
&+ A \left[ \left( (e_{0n})^T Z \right)^2 - 2k(e_{0n})^T Z + k^2 \right] + A \left[ (v_0^T Z)^2 - 2k(v_0^T Z) + k^2 \right],
\end{aligned} \tag{11.52}$$

$$\begin{aligned}
H &= W^T Z + A \sum_{i=1}^n \left[ (e_{in}^T Z)^2 + [v_i^T Z]^2 - 2A[(e_{in}^T) + v_i^T] Z + 2A + A[(e_{0n}^T) + v_0^T]^2 Z \right. \\
&\left. - 2AK e_{0n}^T + v_0^T Z + 2Ak^2 \right].
\end{aligned}$$

Regrouping the terms in the quadratic formulation  $Z^T Q Z + g^T Z + C$ :

$$\begin{aligned}
H &= A \sum_{i=1}^n \left[ (e_i \otimes I_n)^2 Z^2 + [v_i^T]^2 Z^2 \right] + w - 2A \sum_{i=1}^n [(e_i \otimes I_n^T) + v_i^T] \\
&- 2Ak \left[ (e_0 \otimes 1_n)^T + v_0^T \right] + 2An + 2Ak^2.
\end{aligned} \tag{11.53}$$

With the variables  $Q$ ,  $g$  and  $C$ :

$$Q = \sum_{i=1}^n \left[ (e_i \otimes I_n)^2 + [v_i^T]^2 \right] = \sum_{i=1}^n [e_i \otimes I_n e_i \otimes I_n^T + [v_i v_i^T]]. \tag{11.54}$$

$$g = w - 2A \sum_{i=1}^n [(e_i \otimes I_n^T) + v_i^T] - 2Ak \left[ (e_0 \otimes 1_n)^T + v_0^T \right]. \tag{11.55}$$

$$C = +2An + 2Ak^2. \tag{11.56}$$

#### 11.2.4 Mapping the SWP's formulation into a list of Pauli operators

There is a very close relationship between Clifford's algebras [182] with Quantum computing. Clifford's algebras are abstract structures that have been widely used in various theories of the physical and mathematical types. In particular, they favoured a comprehensive treatment of operators involved in Quantum Computing. For example, Clifford's group defines the computational subspace operations on the Hilbert vector space.

$$\sigma_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \text{ and } \sigma_3 = Z = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}. \tag{11.57}$$

Nevertheless, we use the Pauli group for quantum computing. The Pauli group on one qubit is the 16-element matrix group consisting of the  $2 \times 2$  identity matrix  $I$  and all of the Pauli matrices together with the products of these matrices with the factors  $\pm 1$  and  $\pm i$ :



$$G_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ, \} \equiv \langle X, Y, Z \rangle. \quad (11.58)$$

In the case of  $n$  qubits, the Pauli group is defined as  $G_n$  in the tensor product of the Hilbert space  $(C^2)^{\otimes n}$ .

$$G_n = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ, \}^{\otimes n}. \quad (11.59)$$

With  $2^n \times 2^n$  matrix acting on an  $n$  qubits Hilbert space. Using the Pauli group properties, we can avoid doing costly matrix operations.

The Pauli group is also used for the correction of the quantum error.

#### 11.2.4.1 Mapping of quadratic variables of the Hamiltonian in the Z-basis.

Having the QUBO form of the SWP (equation (11.53)) from the Kronecker product and vectorization, that can now map these  $Q$ (11.54),  $g$ (11.55) and  $C$ (11.56) into the computational the Z-basis and Pauli's operators.

$$Q_z = \left( \frac{Q}{4} \right). \quad (11.60)$$

$$g_z = \left( -\frac{g}{2} - \left( I_v \otimes \frac{Q}{4} \right) - \left( \frac{Q}{4} \otimes I_v \right) \right). \quad (11.61)$$

$$c_z = \left( c + \left( \frac{g}{2} \otimes I_v \right) + \left( I_v \otimes \left( \frac{Q}{4} \otimes I_v \right) \right) \right). \quad (213)$$

$$c_z = c_z + tr(Q_z). \quad (11.62)$$

$$Q_z = Q_z - \text{diag}(\text{textdiag}(Q_z)). \quad (11.63)$$

#### 11.2.4.2 Coding the Hamiltonian into the Pauli Operators

We can now implement our algorithm with the Aqua library of the Qiskit framework. Using the group theory that forms the Pauli operators to reduce computational costs by defining operation over the Pauli components. To do this, we can decompose the Hamiltonian of Ising into a list of the Pauli operators that make up the Clifford group (a set of mathematical transformations which affect permutations of the Pauli operators.)  $P_n = I, (X, Y, Z)$ .

#### 11.2.4.3 Quantum solution from the ground up and closer to the Quantum Native form.

For quantum solver, we used Qiskit. This way to solve the problem is close to the natural way. Now, we embedded some, IBMQ functions as *WeightedPauliOperator*, *NumPyMinimumEigensolver* and *TwoLocal*. For the generic, this can be rewritten quickly.

First, we derived the solution from the ground up, using a class *QuantumOptimizer* that encodes the quantum approach to solve the problem. Then we instantiated it and solved it. We defined the following methods inside the class:

1. *binary\_representation*: encodes the proposed problem into the Ising Hamiltonian QP.

2. `construct_hamiltonian`: constructs the Ising Hamiltonian in terms of the  $Z$  basis.
3. `check_hamiltonian`: makes sure that the Ising Hamiltonian is correctly encoded in the  $Z$  basis: to do this, the function solved an eigenvalue-eigenvector problem for a symmetric matrix of dimension  $2^N \times 2^N$ . For the complexity limit  $N = n(n - 1)$ .
4. `vqe_solution`: solves the proposed problem via VQE by using the SPSA solver with default parameters.
5. `_q_solution`: internal routine to represent the solution in a usable format.
6. `qaoa_solution`: solves the proposed problem via QAOA by using the SPSA solver with default parameters.

All these six steps, according to our objective function defined by the equations (11.53). These steps are necessary for quantum programming.

With these steps, we solve our problem through the exact result, VQE and QAOA. Next, we will discuss the other techniques used to check the code in quantum. But for this, we need the `docplex` and the library of quadratic programming.

### 11.2.5 Resolution of our formulation with `docplex`

Formulating the problem is equivalent to expressing a VRP with time and contract constraints. So, we apply the `docplex` model on the question proposed in the equations (11.11), (11.2), (11.3), (11.5), (11.6) and (11.8). In [25, 27, 26, 16] all the code can be found.

Now, we can solve the model (the problem) by all classical and quantum techniques.

### 11.2.6 Cplex solver

We used the `cplex` solver to solve our algorithm. We use the `cplex` solver to solve our algorithm.

Now, from May 2020, Qiskit supports Quadratically Constrained Quadratic Programs under Quadratic Program (under `docplex`) with binary, integer, and continuous variables, as well as equality and inequality constraints. But, this requires more ancillary qubits.

In Chapter (4), we have seen how to solve a quadratic function. Qiskit has developed a library to solve quadratic functions. We will analyse it below.

### 11.2.7 Solving our Quantum SWP with `MinimumEigenOptimizer`

For that, we needed to use the Quadratic Program from IBM, so let's Introduce the `MinimumEigenSolver` and `MinimumEigenOptimizer` [191]. For this translation, the binary variables taking values in  $\{0, 1\}$  are replaced by spin variables taking matters in  $\{-1, +1\}$ , which allows replacing the resulting spin variables by Pauli  $Z$  matrices, and therefore, a Hamiltonian. Qiskit provides automatic conversion from a suitable Quadratic Program to an Ising Hamiltonian, which then permits to leverage all the Minimum Eigen Solver available in Qiskit Aqua, such as VQE, QAOA, or *Numpy Minimum Eigensolver* (classical exact method solver) (Fig.(11.1)).

Qiskit wraps the translation to an Ising Hamiltonian (in Qiskit Aqua, also called Operator), the call to a *Minimum Eigensolver* as well as the conversion of the results back to Optimization Result in the *Minimum Eigen Optimizer*.

In the following solver, we first illustrate the conversion from a *Quadratic Program* to an Operator and then show how to use the *Minimum Eigen Optimizer* with different *Minimum Eigen solver* to solve a given Quadratic Program. The algorithms in Qiskit automatically try to convert a given

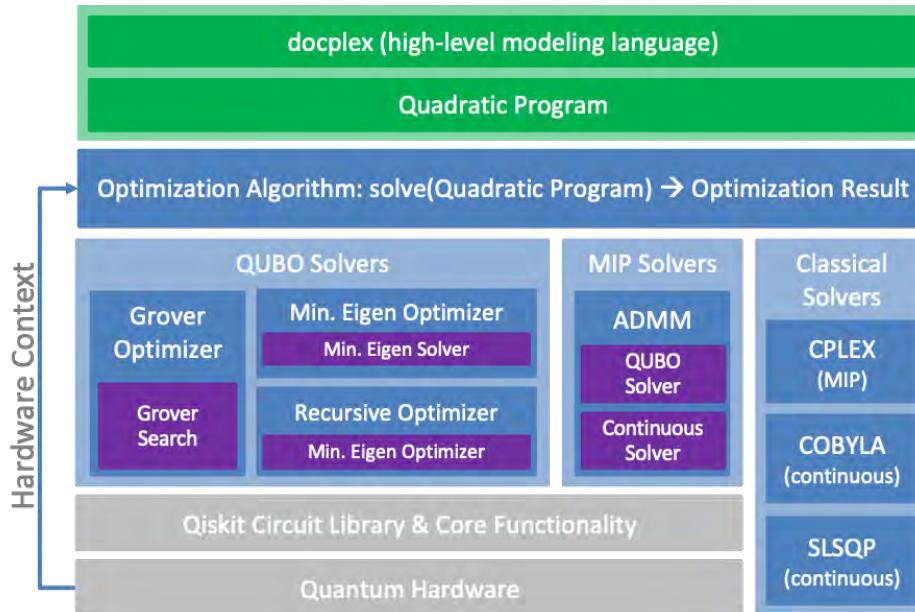


Figure 11.1: Quantum Optimization in Qiskit. A high-level modelling language, a modular set of state-of-the-art quantum optimisation algorithms for different problem classes, leveraging Qiskit’s fundamental quantum algorithms as well as core circuit functionality; combined with a uniform and flexible interface for easy testing, benchmarking, and validation of models and algorithms [191].

problem to the supported problem class if possible; for instance, the Minimum Eigen Optimizer will automatically translate integer variables to binary variables or add a linear equality constraint as a quadratic penalty term to the objective.

The circuit depth of QAOA potentially has to be increased with the problem size, which might be prohibitive for NISQ devices. A possible alternative is Recursive QAOA. Qiskit generalises this concept to the Recursive Minimum Eigen Optimizer.

### 11.2.8 Solving our quantum SWP by using the ADMM Optimizer

Another technique we used to solve our proposed problem is by using the ADMM Optimizer[219]. The ADMM optimiser can solve classes of mixed-binary constrained optimisation problems (MBCO) as a QUBO subproblem on the quantum device via variational algorithms (VQE, QAOA) and continuous convex constrained subproblem, which can be efficiently solved with classical optimisation solvers.

To solve our formulation with the ADMM module, we follow these steps:

We first initialise all the algorithms we plan to use. We initialise this case by setting the phase’s parameters and the QUBO and convex solvers. We used the default parameters for ADMM module  $\rho = 1001$ ,  $\beta = 1000$ . With the penalization factor  $c = 900$  of equality constraints  $Gx = b$ . The tolerance for primal residual convergence is set to  $e^{-6}$ . The 3-block implementation of the ADMM is guaranteed to converge for Theorem number 4 of the ref: [219], because we active the inequality constraint of the continuous variable as recommended by [219]. More details can be found at [191].

As we introduced above, to solve the Quadratic Programming from qiskit as QUBO problems can be chosen between all the following packages. We are talking about Minimum Eigen Optimizer using different Minimum Eigensolver, such as VQE, QAOA or Numpy Minimum Eigensolver (classical), GroverOptimizer, CplexOptimizer (the user needs to install CPLEX is installed. Remember that it is

for classical).

For the classical optimizer (Cobyla, SLSQP, . . . ) as shown in the Fig. (11.1) and analyze in chapter 9. We used the Cobyla Optimizer and the Minimum Eigen Optimizer using the Numpy Minimum Eigensolver for QUBOs as classical alternatives to CPLEX for testing, validation, and benchmarking. In the next step (second), we followed for each ADMM iterations ( $k = 1, 2, \dots$ ) until termination; we solved our Proposed QUBO problem with the quantum solver and updated the dual variables. And the last step is to return optimisers and cost variables. All the experiments code can be found at [25, 27, 26, 16], the reader could see the files of the resolution of our algorithms. They can also be found here [25, 9].

### 11.2.9 Backtracking

One of the classic techniques that we experimented with to solve our proposed problem is Backtracking [209, 71]. We also did this study to make comparisons of the methods that we will review in the chapter reserved for discussions. However, the Social Workers' Problems that are discussed can be solved classically using other known algorithms like the brute force; Dynamic Programming [41, 207] or Greedy Algorithms. In logarithmic, linear-logarithmic time complexity in input data size, and therefore, in some cases (without regard to the brute force), outshine the backtracking algorithm in every respect.

Backtracking is not the most efficient algorithm to solve this problem, but it is suitable for the type of CSP problem (Constraint Satisfaction Problems). Backtracking depends on a user-given scenario that defines the problem to be solved, the nature of the partial solutions, and how they are scaled into complete solutions. It is, therefore, a metaheuristic rather than a specific algorithm; it is guaranteed to find all outcomes to a finite (limited) problem in a bounded amount of time since backtracking algorithms are generally exponential in both time and space.

In our case, we use the Backtracking algorithm to consider all possible issues within the constraints.

### 11.2.10 First approach of the SWP's generalization

In this section, we will propose a more generic formulation that does not consider the limitations of the era in which we are. This formulation can be programmed on any quantum computer based on Quantum Annealing or Based on quantum gates.

Let  $H$  be our new Hamiltonian as follow,

$$H = H_m + H_c + H_t, \quad (11.64)$$

with

$$H_m = A \sum_{i=1}^{n+1} \sum_{u=0}^n \sum_{\substack{v=0 \\ v \neq u}}^n x_{u,v}^i W_{u,v}. \quad (11.65)$$

$$\begin{aligned}
H_c = & B(1 - \sum_{v=1}^n x_{0,v}^1)^2 + B \sum_{i=2}^n \left( 1 - \sum_{u=1}^n \sum_{\substack{v=1 \\ v \neq u}}^n x_{u,v}^i \right)^2 + B \left( 1 - \sum_{u=1}^n x_{v,0}^{n+1} \right)^2 \\
& + B \sum_{u=1}^n \left( 1 - \left( x_{u,0}^{n+1} + \sum_{i=2}^n \sum_{\substack{v=1 \\ v \neq u}}^n x_{u,v}^i \right) \right)^2 + B \sum_{v=1}^n \left( 1 - \left( x_{0,v}^1 + \sum_{i=2}^n \sum_{\substack{u=1 \\ u \neq v}}^n x_{u,v}^i \right) \right)^2. \tag{11.66}
\end{aligned}$$

$$H_t = \tau_1 + \tau \sum_{i=2}^n \tau_i W_i + \tau \sum_{i=1}^n \Lambda_i, \tag{11.67}$$

where

$$\Lambda_i = \left( 1 - \sum_{k=1}^k t_{k,i} \right)^2 \quad (1 \leq i \leq n). \tag{11.68}$$

Where  $v$  is with a time window  $[e_v, l_v]$

$$T_1 = \left( \sum_{v=1}^n x_{0,v}^1 w_{0,v} + \sum_{k=1}^k k t_{k,1} - \sum_{v=1}^n x_{0,v}^1 l_v \right)^2. \tag{11.69}$$

$$T_i = \left( \sum_{d=1}^i \sum_{u=1}^n \sum_{\substack{v=1 \\ v \neq 1}}^n x_{u,v}^d w_{u,v} + \sum_{k=1}^k k t_{k,i} - \sum_{u=1}^n \sum_{\substack{v=1 \\ v \neq u}}^n x_{u,v}^i l_v \right)^2 \quad (2 \leq i \leq n). \tag{11.70}$$

$$t_{k,i} = \begin{cases} 1, & \text{the time margin in the tour } k \\ 0, & \text{otherwise.} \end{cases} \tag{11.71}$$

The binary variable  $i$  represent the position of the tour. The constants  $B$ ,  $A$  and  $\tau$  are positive constants, which must be chosen according to our requirements. Therefore,  $A < B$ , to ensure that the

constraints of  $H_c$  are fulfilled. At last,  $\tau < B$  ranks the time windows constraints over the objective to minimise the path (tour).

## 11.3 Research Results

### 11.3.1 Experimentation

We test our algorithm mapped on VQE, QAOA, Numpy Minimum Eigensolver (classical), GroverOptimizer, CplexOptimizer, Numpy Minimum Eigensolver for QUBOs, Backtracking and CP-SAT Solver from google on the *ibmq-16-melbourne v1.0.0*, *ibmq-qasm-simulator (up to 32 qubits)* with COBYLA and SLSQP as the classical optimizer. We can consider QAOA as a particular case of VQE. This is one of the reasons we can apply the Hamiltonian of the Ising model of our proposed formulation directly to QAOA almost without modify. The Figures (11.2) to (11.13) show the results of our algorithm once we executed our algorithm under the IBMQ. The optimal visit considers the hours of visits to form the optimal schedule. We have done several experiments with the QML defining different scenes using shot configuration. With our quantum machine, we can configure the number of repetitions of each circuit for sampling we need. With that, we will be doing machine learning on circuit design for each shot. And when the loop ends, we will get to the ground state energy. Consequently, we solved our problem by creating one quantum circuit for each shot, and the best circuit will be the one that optimizes our Social Workers' Problem. Next, we will see the results (figures and tables) obtained in each case studied and presented above.

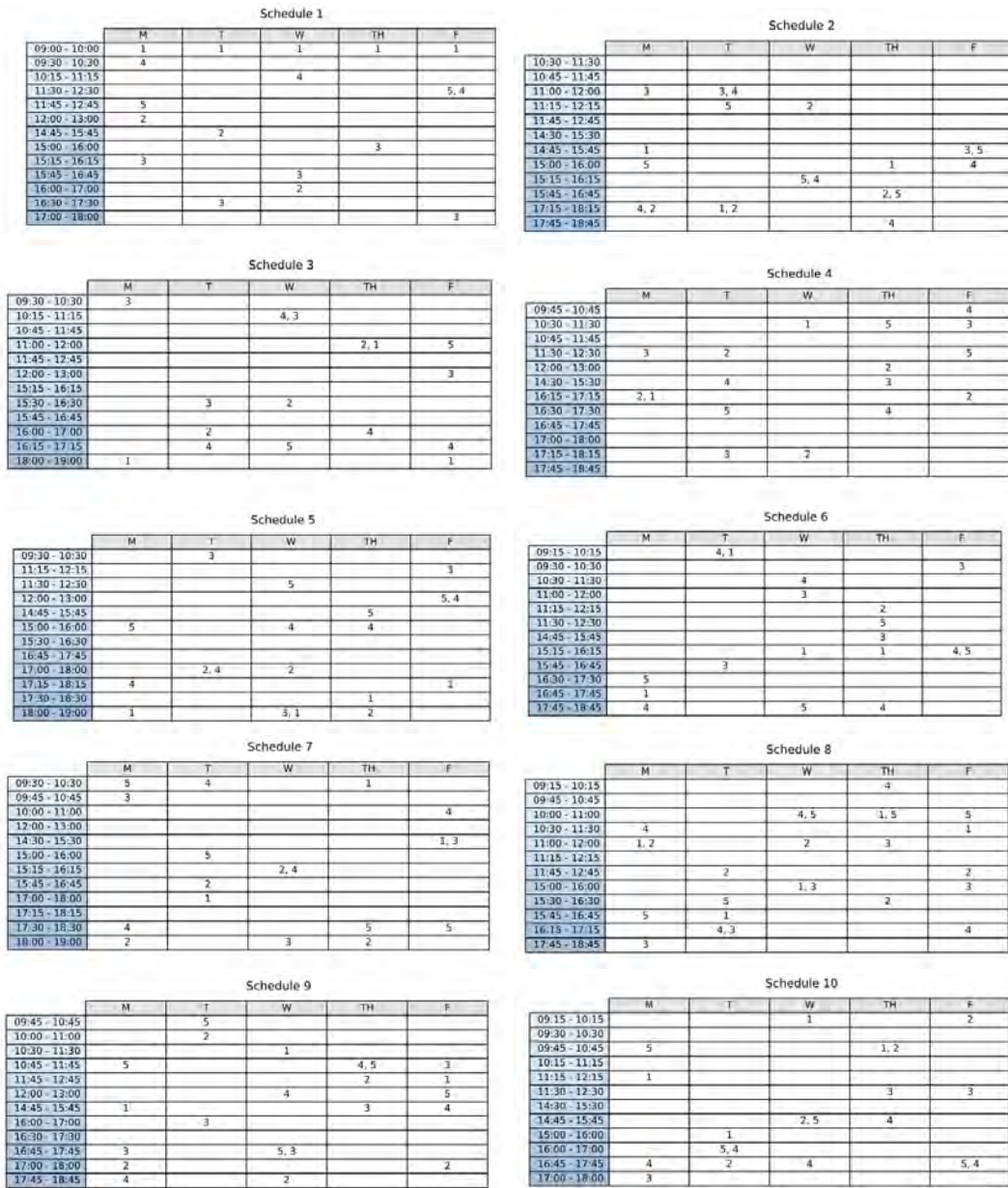


Figure 11.2: Generated 10 Social Workers' schedules

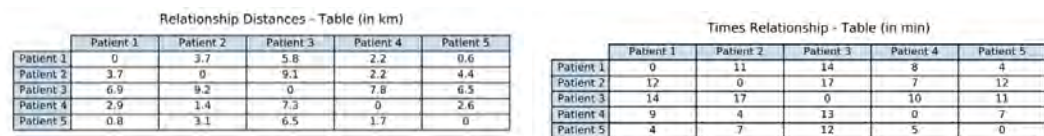


Figure 11.3: Generated 10 Social Workers' Distance Matrix



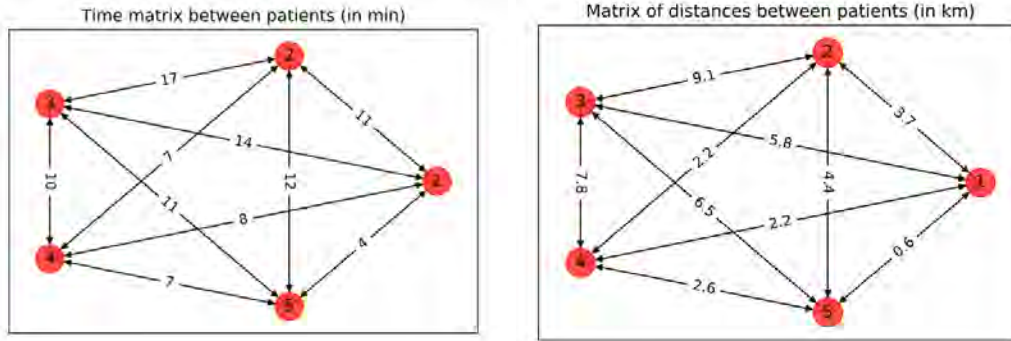


Figure 11.4: Generated 10 Social Workers' Distance Nodes

## 11.4 Discussions

The VQE works very well and empowers the QML era. From what we expected, the VQE has solved it by far. With the help of the IBMQ Aqua<sup>†</sup> program environment (qiskit<sup>‡</sup>), we were able to test our algorithm. Too bad that we do not have access to a quantum computer with more qubits. Since in our case for  $n = 6$ , the number of qubits necessary is  $n(n - 1) = 30qubits$  for the QASM\_Simulator and  $n = 4$  for the real quantum computer (15 qubits). We have not been able to do many more tests for values of  $n$  greater than six since the computer on which we test our algorithm takes too long.

The VQE and the docplex can help the domain of programming in quantum computing, where we require a high computing capacity. But also can be useful to empower intelligent solutions as the objective of this article: Using the Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers schedule problem solver.

The evaluation of the algorithm on an *ibmq\_16\_melbourne v1.0.0* from IBM was fulfilled. With any change in the input, variables are mapped proportionally to our cost variable with a time window.

With this work, we are trying to offer to cities an instrument [243] which could optimize the costs allied to the management of social workers and improve social gains. Moreover, this work could also be a starting point for many countries in Africa that are seizing the opportunity of the mobile technology revolution [220] to develop and increase their industrious and e-health system [152].

We would like to add that the suggested formulation (11.50) and (11.51) is not only specific to the proposed problem. It can be used to solve any family planning, scheduling and routing problem related to a list of tasks, restrictions, allocation of resources on location and time. The test performed and showed in Fig. (11.27) allows us to see the behaviour of our formulation with the variation of the correction factor  $\varepsilon$ . We understand how our time window  $T_{ij} = (\tau_i - \tau_j)$  adapts perfectly at the extremes to the cost variable in the distance. This achievement is due to the chosen quadratic function (11.51). We want it to be adapted in this way so that our resultant function weights together the short distances and time and the long distances and late times. Other functions can be studied to have a test bench to compare the final results. QAOA, like VQE, takes a qubit operator from the Hamiltonian of the Ising model. The only mapping that gets done is when QAOA builds a variational form based on the qubit operator and what we figured was that does not change the original qubit operator input. Figure (11.25) and (11.26) reveal the comparison work between VQE and QAOA algorithms for the same configuration parameters. After several tests, we confirmed that our algorithm takes less execution time with the QAOA instead of the VQE and requires fewer samples for the optimal solution. But in many cases, we have to increase the seed value to get a reasonably stable result.

<sup>†</sup>[www.qiskit.org/aqua](http://www.qiskit.org/aqua)

<sup>‡</sup>[www.qiskit.org](http://www.qiskit.org)

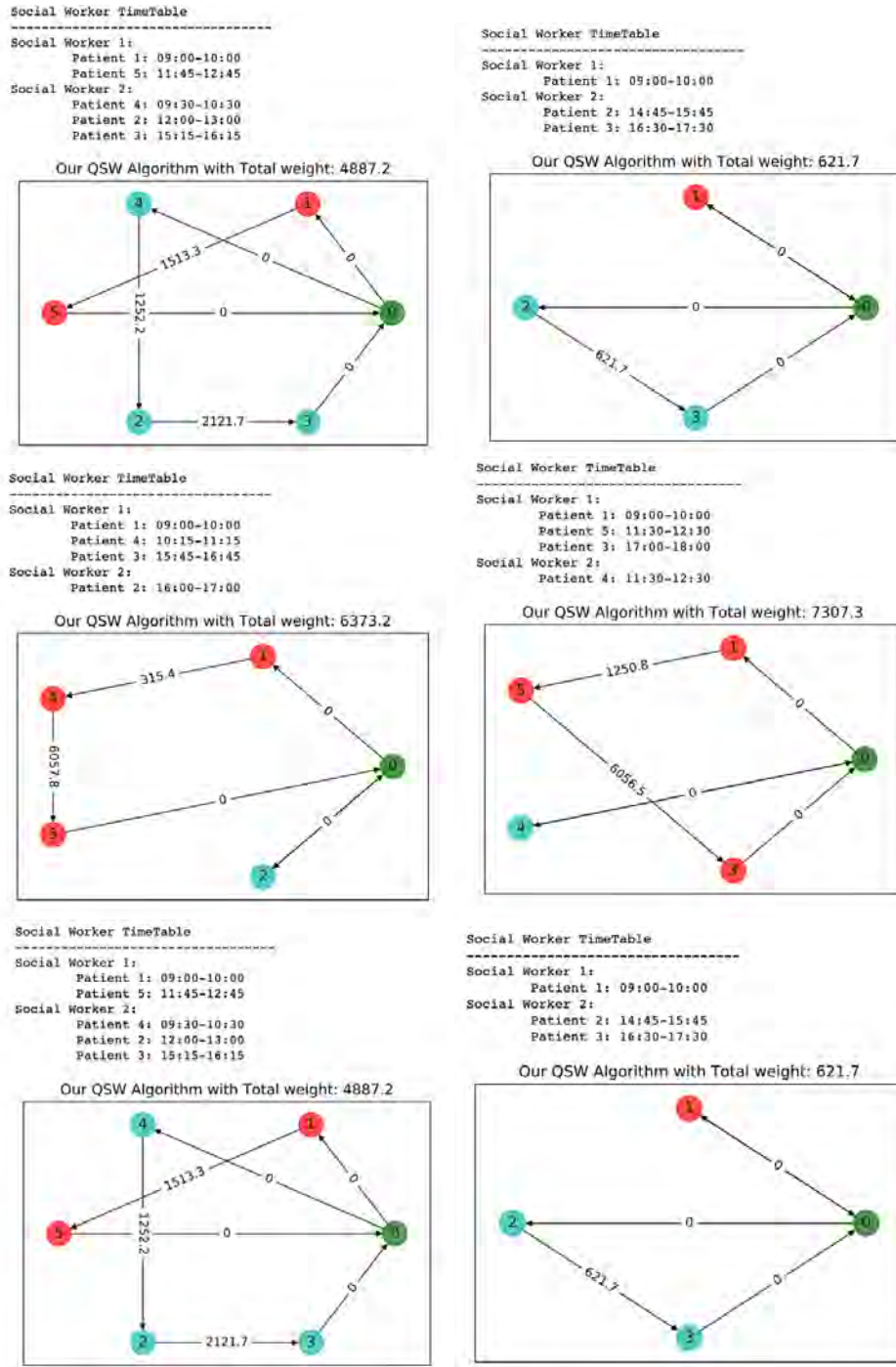


Figure 11.5: Outcomes from the SWP by the Backtracking algorithm. From Monday to Wednesday

## 11.5 Benchmarks of SWP

In this section, we are going to make a comparison of all the techniques we used to solve the SWP.

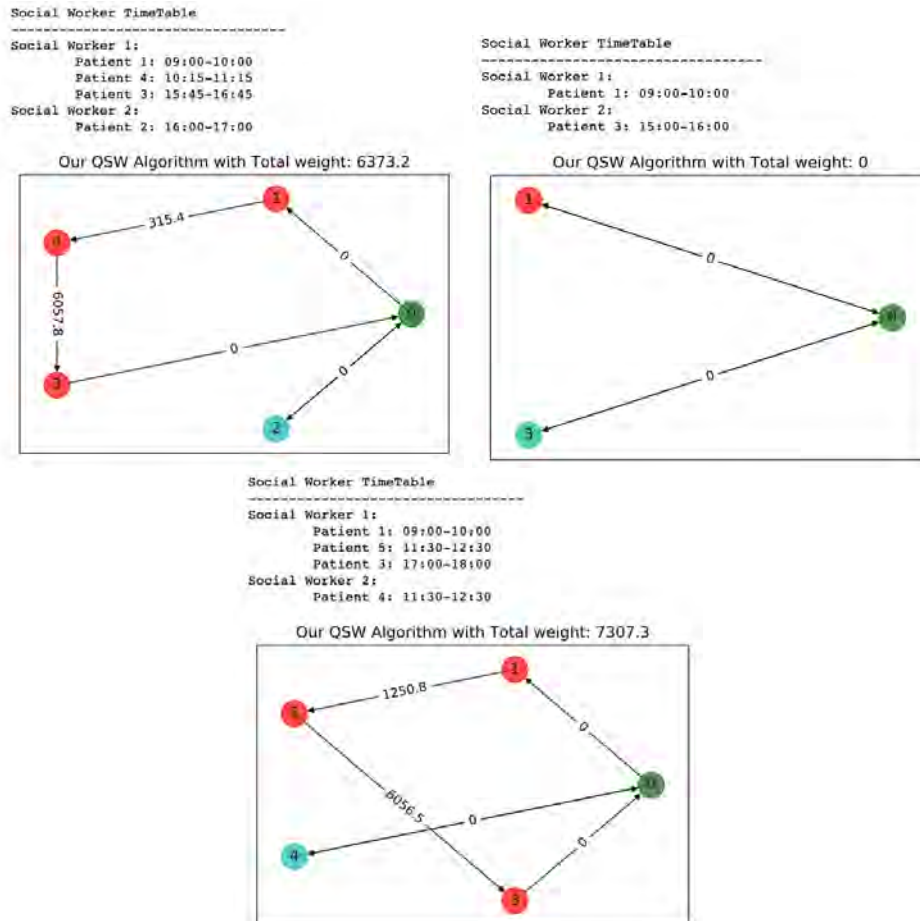


Figure 11.6: Outcomes from the SWP by the Backtracking algorithm. From Thursday to Friday

### 11.5.1 Comparing VQE solution with classical solver

The results obtained with the VQE, compared with the classical and idyllic quantum computing results, demonstrate that the results obtained are correct. The only significant difference is the time it takes for this era’s quantum computers to compute our algorithm right now. It is worth noting that no such time has anything to do with the complexity class. On the contrary, nowadays, what is gained with quantum computers is to observe how well our algorithms work well (See Fig. (11.5) to (11.22)).

### 11.5.2 Backtracking

Backtracking is not the most efficient algorithm to solve this problem, but it is suitable for the type of CSP problem (Constraint Satisfaction Problems). Backtracking depends on a user-given scenario that defines the problem to be solved, the nature of the partial solutions, and how they are scaled into complete solutions. It is, therefore, a meta-heuristic rather than a specific algorithm; it is guaranteed to find all outcomes to a finite (limited) problem in a limited amount of time since backtracking algorithms are generally exponential in both time and space. In our case, we use the Backtracking algorithm to consider all possible issues within the constraints. We can observe that,

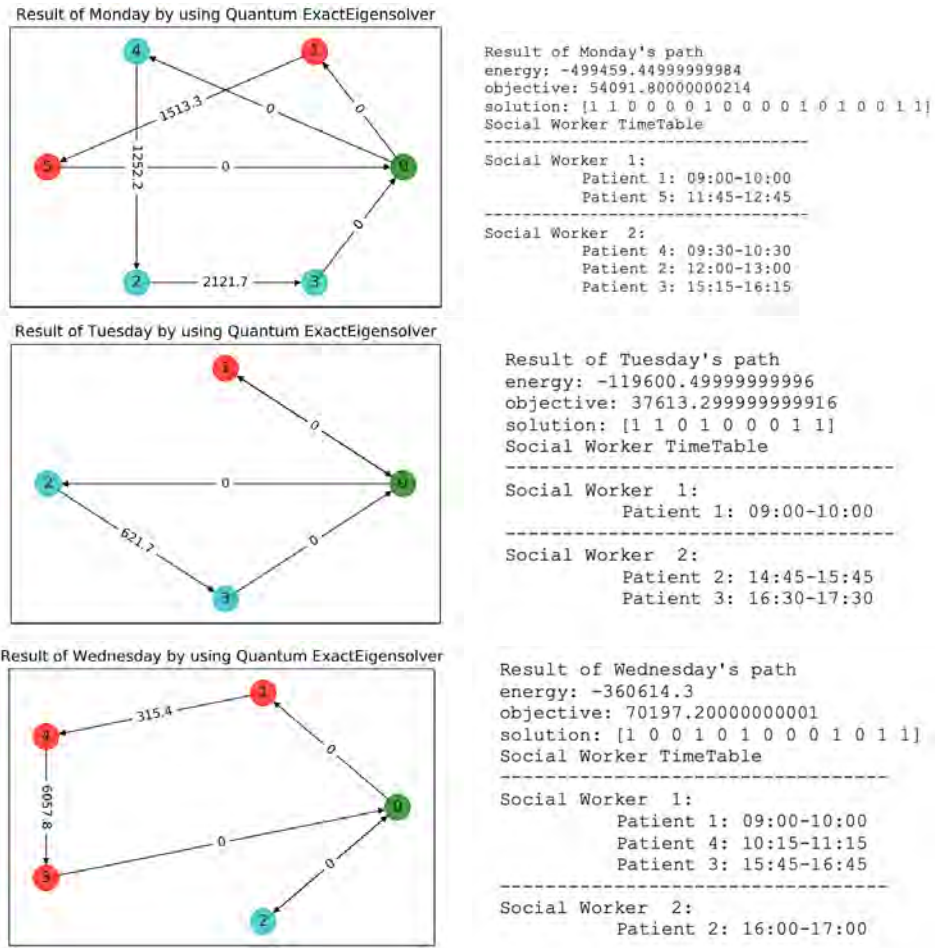


Figure 11.7: Outcomes from the SWP by the Quantum Exact solver algorithm. From Monday to Wednesday

while Backtracking presents an exponential behaviour as the number of patients (which would be the nodes of the graph) increases, the VQE algorithm, without considering the cost of evaluation and calibration of the algorithm, has a growth linear. This, as the number of patients grows, will offer more considerable advantages than a classic algorithm, such as Backtracking, since its temporary cost will be much lower for more complex problems. Finally, we compare the time cost of executing the classic algorithm used with the VQE algorithm on the *ibmq\_16\_melbourne*. To do this, we changed the back-end that allows us to perform the algorithm on a real quantum computer with the highest number of qubits available (15 qubits).

The *ibmq\_16\_melbourne* returns in the result array a parameter that indicates the total time required to run the algorithm, including the calibrations necessary before running the algorithm in each trial. It is essential to keep in mind since the results' time will always be very high, but we are interested in the asymptotic trend that the algorithm follows as we increase the total number of patients to visit. Besides, it is necessary to consider that this kind of algorithm shows its potential in the face of highly complex problems. So, the problem we are trying to solve with such a small number of patients does not allow the VQE algorithm to offer an advantage over a classic algorithm. Still, it allows us to observe its behaviour and validate that it can solve the problem posed. The results are

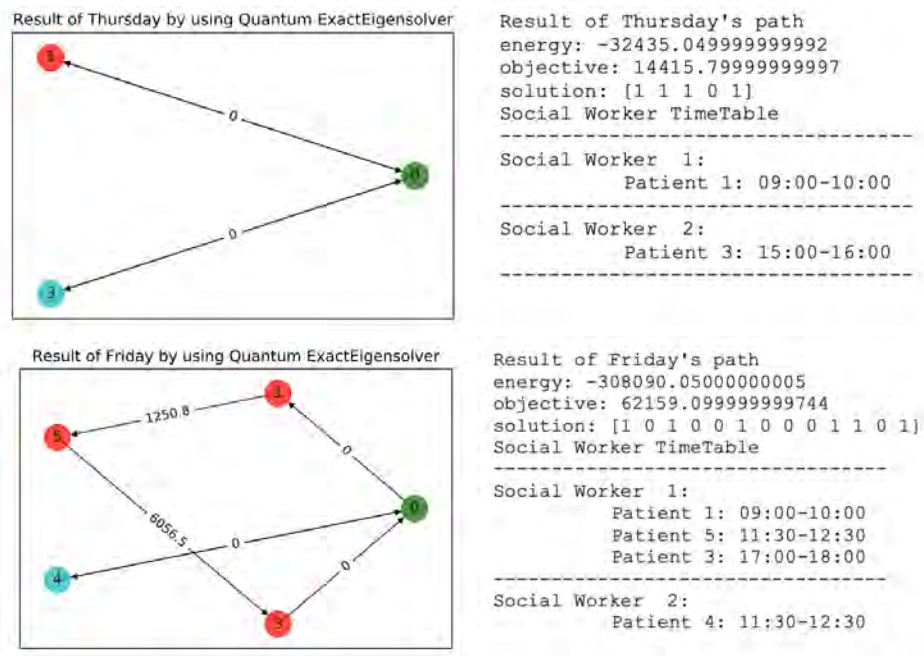


Figure 11.8: Outcomes from the SWP by the Quantum Exact solver algorithm. From Thursday to Friday

shown in table (11.5) and figures (11.5) and (11.6).

All the tests are done with the calibration from table (11.5.2). Nevertheless, this ref. [141] could help to dive deeply. Figure (11.20) expresses the structure of the *ibmq\_melbourne* quantum device from IBM. This computer has fifteen superconducting qubits.

### 11.5.3 Comparison of the algorithms' complexity

We can observe that, while Backtracking (Fig. (11.21)) and the CPLEX present an exponential behaviour as the number of patients (which would be the nodes of the graph) increases, the VQE, QAOA algorithm ((11.22)), without taking into account the cost of evaluation and calibration of the algorithm, have a logarithmic growth. This, as the number of patients grows, will offer more considerable advantages than a classic algorithm, such as Backtracking, since its temporary cost will be much lower for more complex problems (11.21) and (11.22).

### 11.5.4 OR Tool SAT

To validate the correct implementation and solution of problems, we use the Google OR-tools API, which is geared towards solving VRP (Google Developers OR Tools, 2019). To use this API, it is required to assign, at least, a distance between nodes. This distance, for this problem, will be the calculation of weights, conditioned by the time limitations between schedules. This means that if a path is not valid, we assign it a weight corresponding to the system maximum (*sys.maxsize*). Thanks to this, the API can recognize that paths are not valid since they can be added to the maximum distance that a vehicle can move in a journey. Therefore, if we also assign this maximum to *sys.maxsize*, we



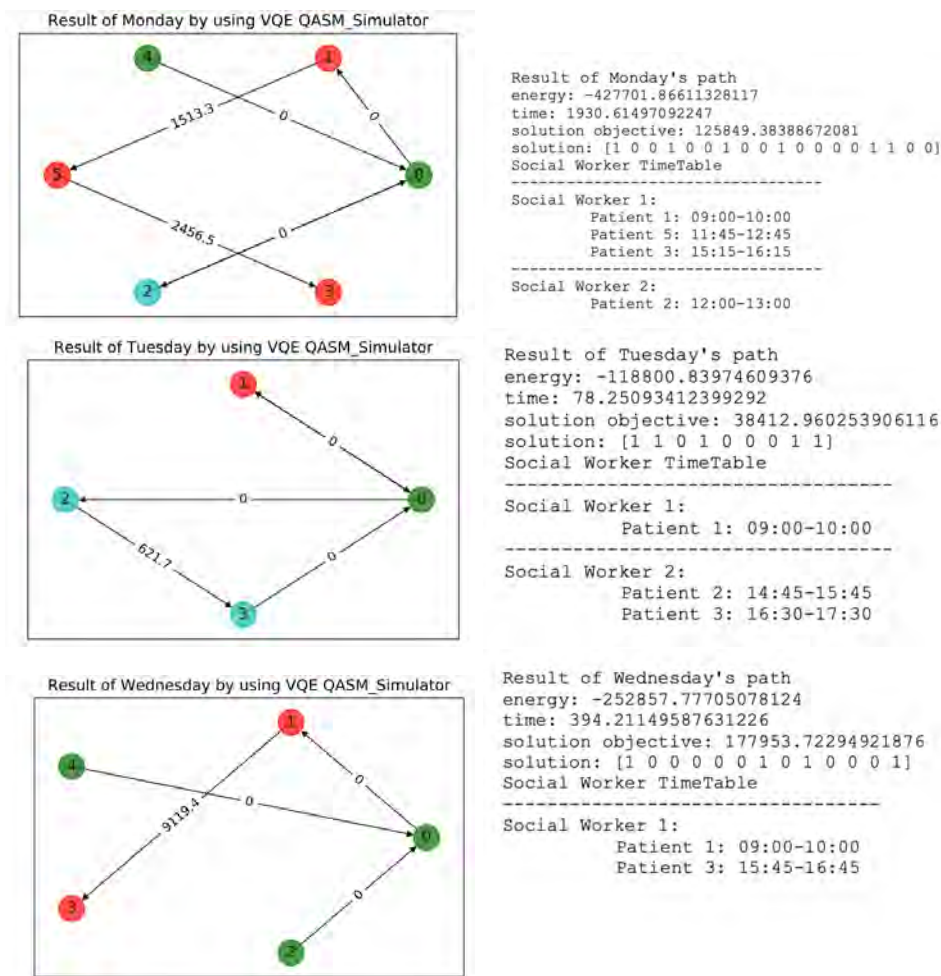


Figure 11.9: Outcomes from the SWP by the Quantum ASM Simulator from IBMQ. From Monday to Wednesday

cannot move the paths with the same length, preventing us from considering these paths valid. The result of the algorithm is then as shown in figure (11.23).

### 11.5.5 Comparing 10 scenarios of SWP with VQE

This section will compare the SWP' schedules in ten scenarios by fixing the patient's location and solving it with the VQE (See Fig. (11.24)).

#### 11.5.5.1 Comparing 5 scenarios of SWP with VQE and QAOA

This section will compare the SWP in five scenarios the algorithm's scalability by varying the patient's number and solving it with the VQE and QAOA (See Fig. (11.25)).

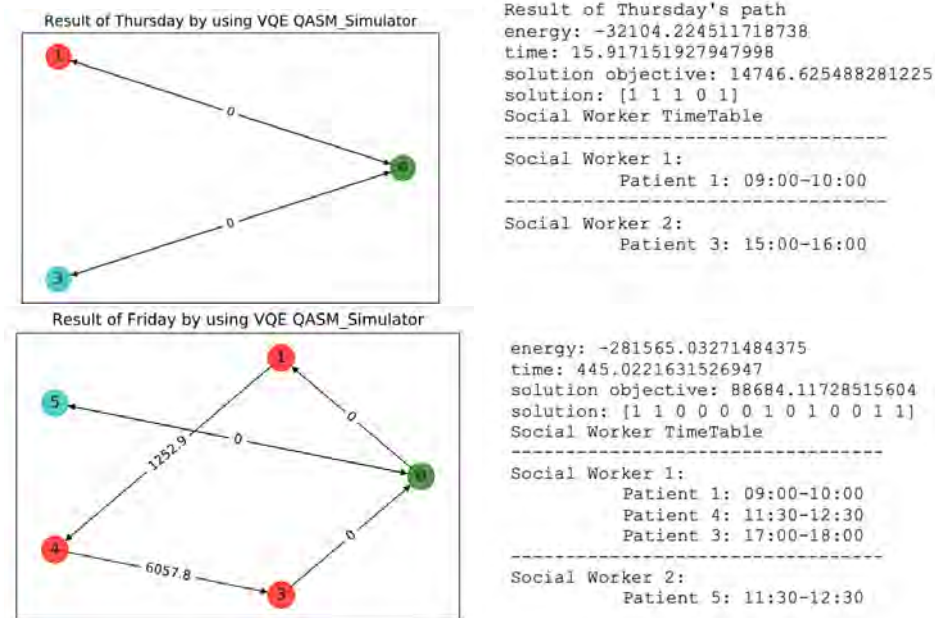


Figure 11.10: Outcomes from the SWP by the Quantum ASM Simulator from Thursday to Friday

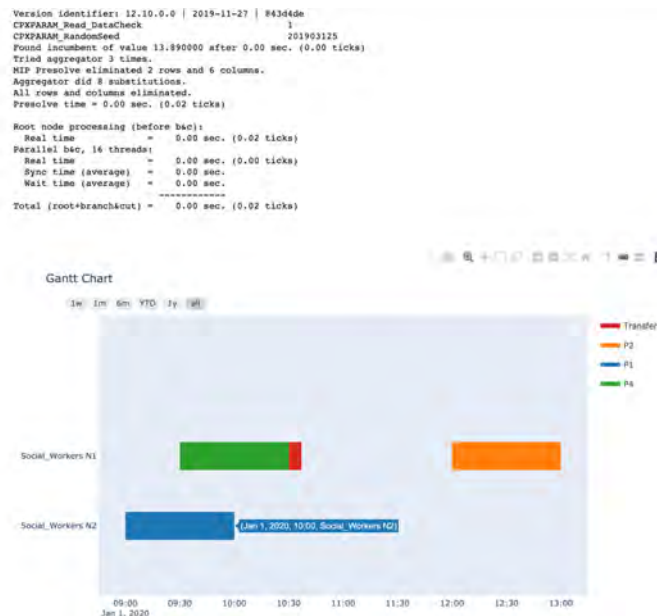


Figure 11.11: Outcomes from the SWP by ibmq\_16\_melbourne v2.1.0 from IBMQ.

### 11.5.6 Comparing 10 scenarios of SWP on VQE and QAOA

This section will compare the SWP in ten scenarios the algorithm's scalability by varying the patient's number and solving it with the VQE and QAOA (See Fig. (11.26)).

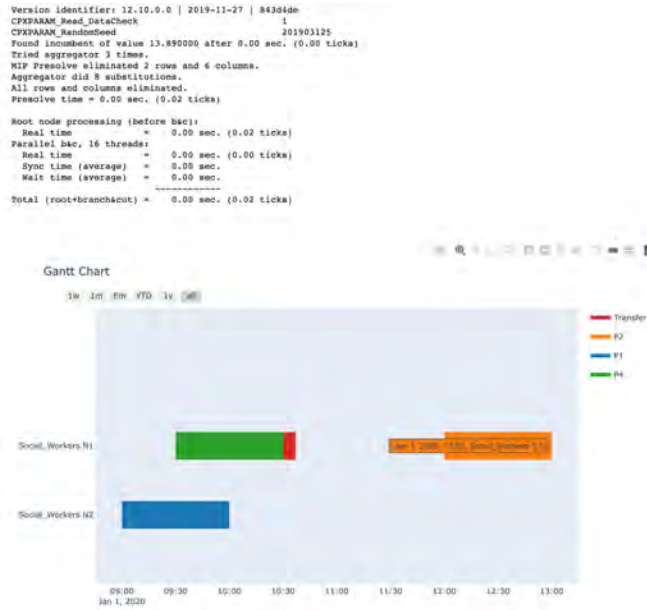


Figure 11.12: Outcomes from the SWP by ADMM optimizer from IBMQ.



Figure 11.13: Outcomes from the SWP by Minimum Eigen Optimizer from IBMQ.

### 11.5.7 Comparing 10 banks of the Social Workers' schedules on QAOA

This section will compare the SWP in ten banks schedules by fixing the patient's location and solving it with the QAOA (See Fig. (11.26) and (11.28)).

### 11.5.8 Comparing the SWP on all quantum algorithms

Since we wanted to make a decent comparison, we mean like many qubit numbers, we had to change the back-end from the *statevector\_simulator* to *qasm\_simulator* and the real quantum computer. We made this change because the superposition calculations of the quantum states or complex amplitudes that the simulator provides to keep track of the algorithm overthrow the computer. It must be said that this has nothing to do with the efficiency of the quantum computer but is a tool that facilitates



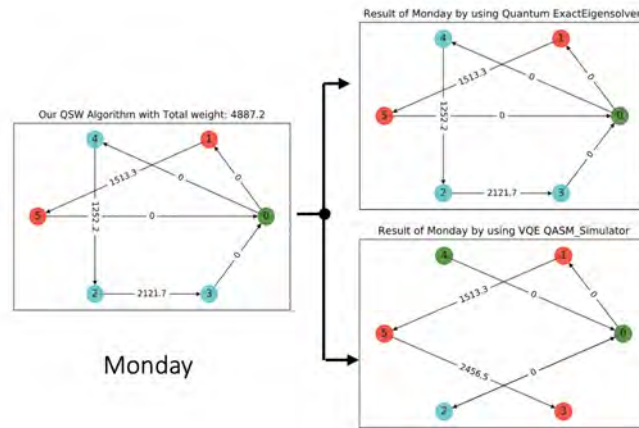


Figure 11.14: Here we present the results by Monday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

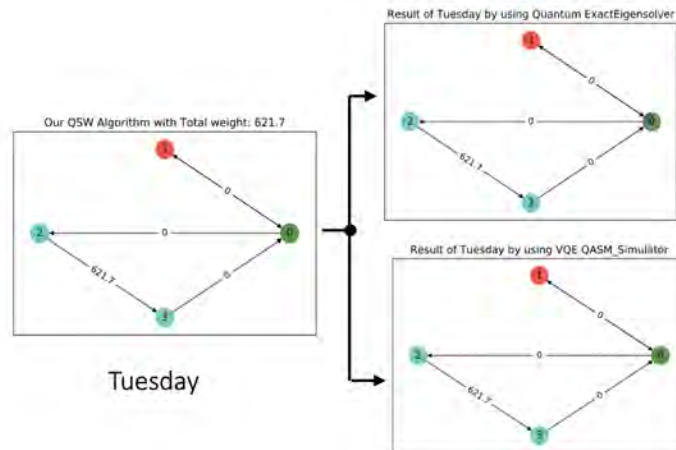


Figure 11.15: Here we present the results by Tuesday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

Qiskit to learn the steps that the algorithm follows.

The results obtained with the *qasm\_simulator* are very similar to the results of the real quantum computer, as seen in the figures (11.12) and (11.13).

We can observe (Fig. (11.19)) in this case that the Quantum exact solver gives the same result as the backtracking. So instead, the VQE provides a solution with but with a patient change. This can be a problem if the patients will not be able to change social workers daily. Another limitation can be the learning process of the top-down approach. Another improvement is to reduce the computational cost of the top-down algorithm.

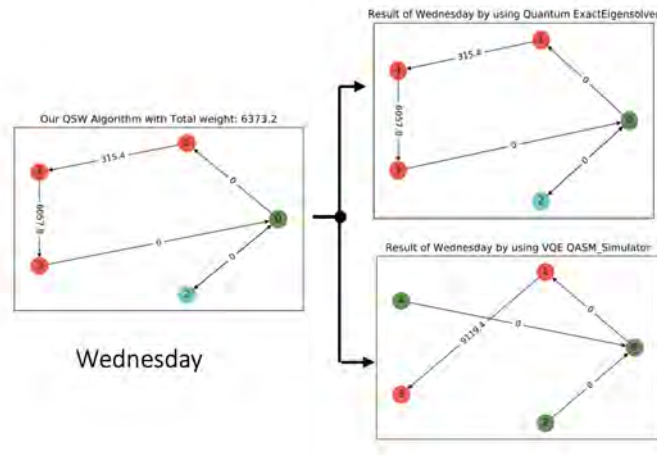


Figure 11.16: Here we present the results by Wednesday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

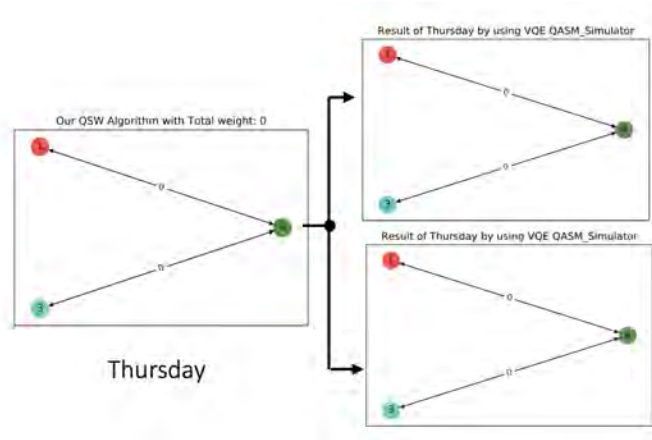


Figure 11.17: Here we present the results by Thursday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

## 11.6 Conclusion and Further Work

This thesis work has studied, implemented, and proved the feasibility of solving the Constraint Search Problem more efficiently with quantum computing. We have researched and implemented, in classic and quantum, several combinatorial optimisation algorithms to solve the problem of social workers visiting patients at their homes. We have also tested the concepts and techniques of combinatorial optimisation in programming environments, Cirq, Qiskit and PennyLane. Considering this, we analysed the state of maturity of the hardware, the framework, of the scientific community and even the response time of the technical support of each leading company in this quantum computing career. We have opted for IBMQ for all the facilities (framework, hardware, support and libraries) they have given us to be within their community. After the implementation, we have run the algorithms on both quantum simulators (*32q\_ibmq\_qasm\_simulator*) and quantum computers (*ibmq\_16\_melbourne v2.3.0*) from IBM. The results and the discussions are developed in detail in this thesis work.

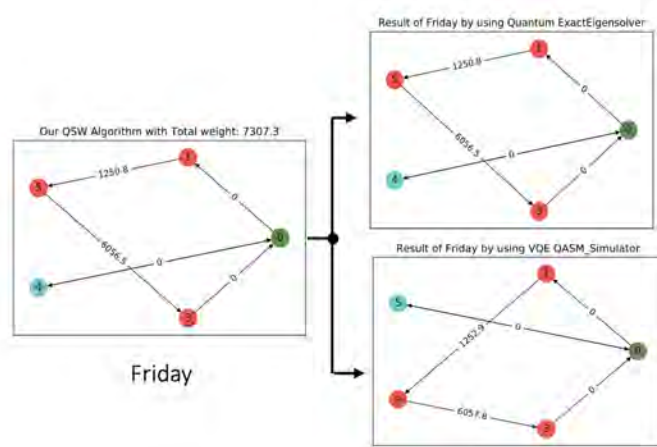


Figure 11.18: Here we present the results by Friday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

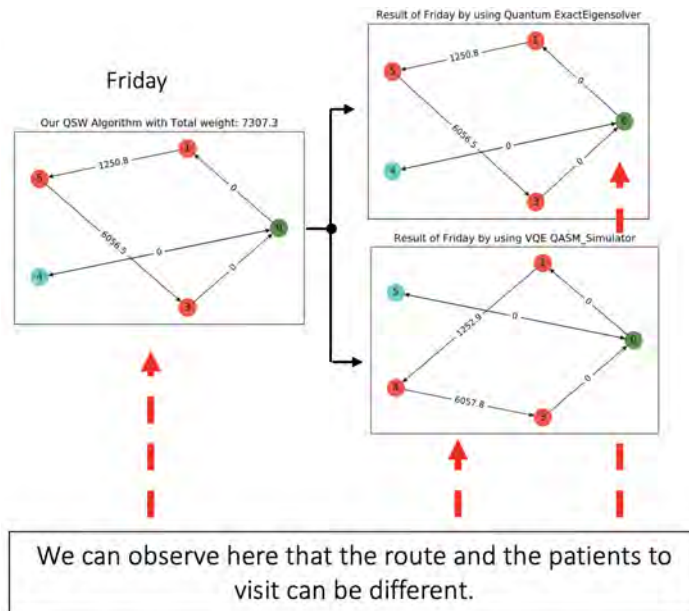


Figure 11.19: Here we present the results by analysing Friday of our algorithm. We are comparing the classical solver and the exact quantum solver, and the Variational Quantum Eigensolver.

### 11.6.1 Conclusions

We have developed several mathematical formulations and proposed one to solve our problem. Our proposed formulation allows us to design a strategy to take advantage of the era we are in (NISQ) (Few useful qubits, limitations in quadratic techniques with inequality restrictions, just to mention these). Moreover, although it seems specific, said formulation is later generalised to encompass more generic applications. And we did a comparative analysis of 5 different implementations made and discussed during this thesis work.

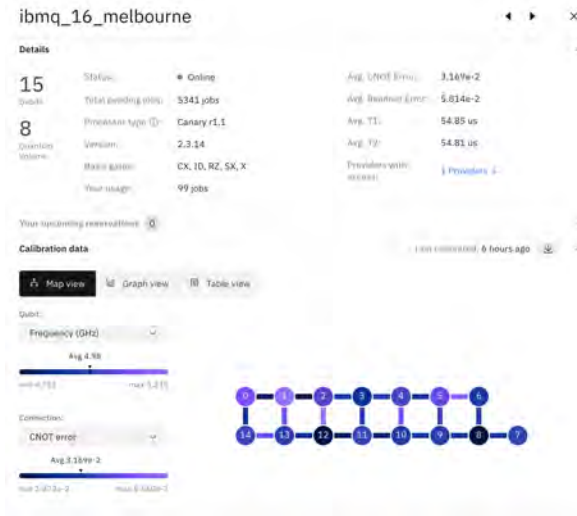


Figure 11.20: The structure of the *ibmq\_melbourne* quantum device from IBM. This computer has fifteen superconducting qubits, which are connected by the  $C_{NOT}$  gate. The bidirectionality of arrows reflects that each of the qubits can be both a control and a target.

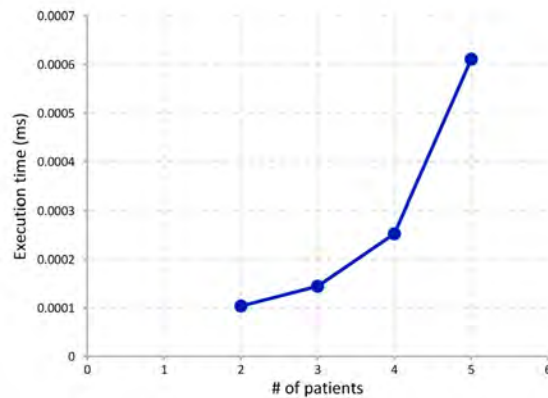


Figure 11.21: Backtracking’s behaviour depending on the execution time and the number of patients.

We have analysed the Ising and the QUBO models in-depth to solve quadratic problems. We have also studied the Cplex optimisation tools such as IBM’s docplex under its framework Qiskit.

Qiskit’s optimisation tool included the Generic Quadratic Programs that help to model any optimisation problem. It is also involved many converters to map the problem to solve to the correct input format. Converters like, *Inequality to Equality* to map inequality constraints into equality constraints with additional slack variables. The converter *Integer To Binary* is useful in the case of the need to convert integer variables into binary variables and corresponding coefficients. *Linear Equality To Penalty* helps to convert equality constraints into additional terms of the object function. And last but not least, the *Quadratic Program To QUBO* converter is used as a wrapper for all the mentioned converters.

We have analysed the Minimum Eigen Optimizer and the optimiser ADMM alternative to VQE and QAOA.

We have also developed some guidelines to make it easier for future PhD or graduate students to

Table 11.4: This is ibmq\_16\_melbourne\_calibrations

Qubit	Frequency (GHz)	T1 (µs)	T2 (µs)	$\sqrt{f}(\text{sr})$ error	ibmq_16_melbourne_calibrations	Single-qubit Pauli-X error	Readout assignment error	CNOT error
1	5.114621711511489	71.32106750982616	102.41446927678529	4.183978644302012e-4	4.183978644302012e-4	1.83978644302012e-4	2.649999999999999e-2	cr0,4 : 2.492e-2, cr0,1 : 1.813e-2
2	5.23503871516284	50.19408723625214	47.728364075207956	1.0042524163122974e-3	1.0042524163122974e-3	1.0042524163122974e-3	3.570000000000000e-2	cr1,3 : 4.305e-2, cr1,2 : 1.473e-2, cr1,0 : 1.843e-2
3	5.0383636392729712	53.16105211069983	62.779117846203924	6.693469486494128e-4	6.693469486494128e-4	6.693469486494128e-4	4.110000000000000e-2	cr2,2 : 5.506e-2, cr2,3 : 2.907e-2, cr2,1 : 1.473e-2
4	4.8944506831891079	58.021400036077925	17.48180445538179	1.0612091500236409e-3	1.0612091500236409e-3	1.0612091500236409e-3	6.059999999999999e-2	cr3,1 : 3.419e-2, cr3,4 : 3.149e-2, cr3,3 : 2.907e-2
5	5.022087248371323	68.25270765048837	73.03990190979944	8.648278220413689e-4	8.648278220413689e-4	8.648278220413689e-4	4.339999999999999e-2	cr4,0 : 4.481e-2, cr4,3 : 2.751e-2, cr4,5 : 3.149e-2
6	4.929465322205785	19.4193930885019857	32.372911443244136	2.7881350400977855e-3	2.7881350400977855e-3	2.7881350400977855e-3	5.780000000000000e-2	cr5,0 : 3.543e-2, cr5,8 : 5.156e-2, cr5,4 : 2.751e-2
7	4.983244873960245	67.881366339618807	73.3076496426577	1.6592402096110034e-3	1.6592402096110034e-3	1.6592402096110034e-3	1.866000000000000e-1	cr5,9 : 3.543e-2, cr5,6 : 5.156e-2
8	4.754431321984846	42.189182464521444	27.512101284092842	1.6103212138957879e-3	1.6103212138957879e-3	1.6103212138957879e-3	6.559999999999999e-2	cr6,8 : 3.061e-2, cr6,5 : 5.156e-2
9	4.9735318170784609	105.48815158260177	78.14201824409058	7.577620344854018e-4	7.577620344854018e-4	7.577620344854018e-4	3.000000000000000e-2	cr7,8 : 2.799e-2
10	4.944608530540347	36.0837808891272	55.028770007179574	3.210266597002919e-3	3.210266597002919e-3	3.210266597002919e-3	4.760000000000000e-2	cr8,8 : 3.061e-2, cr8,5 : 2.650e-2, cr8,7 : 2.799e-2
11	4.997403578451156	50.1126570810803	85.15292984056289	1.1260692077635155e-3	1.1260692077635155e-3	1.1260692077635155e-3	4.100000000000000e-2	cr9,5 : 3.543e-2, cr9,0 : 2.907e-2, cr9,8 : 2.650e-2
12	4.763630144177076	73.55960879293964	54.90241515337368	6.135524392248129e-4	6.135524392248129e-4	6.135524392248129e-4	4.010000000000000e-2	cr10,4 : 4.481e-2, cr10,1 : 2.106e-2, cr10,9 : 2.907e-2
13	4.973575050961678	25.128388160502205	27.790785042098847	8.832373710164472e-4	8.832373710164472e-4	8.832373710164472e-4	5.369999999999997e-2	cr11,3 : 3.419e-2, cr11,2 : 1.786e-2, cr11,0 : 2.106e-2
14	5.007402435608253	37.71333982028009	45.30433211703277	2.7800696362570737e-3	2.7800696362570737e-3	2.7800696362570737e-3	5.940000000000000e-2	cr12,3 : 5.506e-2, cr12,3 : 2.609e-2, cr12,1 : 1.786e-2
				1.5790876435366457e-3	1.5790876435366457e-3	1.5790876435366457e-3	8.299999999999999e-2	cr13,4 : 4.387e-2, cr13,1 : 4.303e-2, cr13,2 : 2.609e-2

take their first firm steps in quantum computing.

### 11.6.2 Future directions

As a future line, we will repeat the comparative studies of our formulation with the generalised formulation. But for this comparison, we need to be able to count a quantum computer of at least 49

Number of patients	Time ( $\mu s$ )
2	0.1039505
3	0.1451969
4	0.2527237
5	0.6108284

Table 11.5: Experiment scenario using the Backtracking technique with 2, 3, 4 and 5 patients

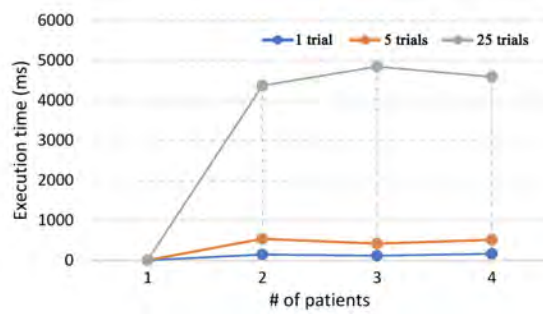


Figure 11.22: VQE’s behaviour depending on the execution time and the number of patients and number of trials

Temporal cost (ms) regarding the number of patients and trials used			
Patients/Trials	1	5	25
2	147.253512	115.247231	164.301820
3	536.902598	421.142229	512.342655
4	4371.009830	4845.496973	4588.167234

Table 11.6: VQE evolution of trials and number of patients

qubits =  $n(n - 1)$ . Where  $n$  is the number of the patients. Another exciting line that we contemplate is the design of an adaptive and specific algorithm that considers any modification or configuration in real-time of the patient or the social worker. Another future line is to design the specific VQE for the proposed problem to gain efficiency. During our experimentation, we realised that the initial configurations are not adequate in some cases; these quantum variational algorithms (VQE, QAOA, ADMM) fail. However, they are today one of the most promising applications of quantum computers in this era (NISQ). We define this future direction to focus on a unique family of quantum circuits called the Hamiltonian Variational Ansatz (HVA), which the QAOA and adiabatic quantum computing inspire. By studying its entanglement spectrum and energy gradient statistics, scientists are experimenting with how HVA exhibits favourable structural properties and optimisation facilities compared to well-studied hardware ansatz. A line of future that interests us a lot is Quantum Machine Learning. However, it is worth mentioning that we are already halfway to QML since all the algorithms based on the variational principle lead us towards this goal. Within this objective, one of the algorithms of the Qiskit framework is the Recursive Minimum Eigen Optimizer that takes a Minimum Eigen Optimizer as input and applies the recursive optimisation structure and strategy to reduce the size of the problem one variable at a time.

```

In [49]: createPathsSocialworkersOR(schedule1, edges, newMax, newMin)

Route for vehicle 0:
0 -> 1 -> 5 -> 0
Distance of the route: 1513m

Route for vehicle 1:
0 -> 4 -> 2 -> 3 -> 0
Distance of the route: 3373m

Maximum of the route distances: 3373m
Route for vehicle 0:
0 -> 1 -> 0
Distance of the route: 0m

Route for vehicle 1:
0 -> 2 -> 3 -> 0
Distance of the route: 621m

Maximum of the route distances: 621m
Route for vehicle 0:
0 -> 1 -> 4 -> 3 -> 0
Distance of the route: 6372m

Route for vehicle 1:
0 -> 2 -> 0
Distance of the route: 0m

Maximum of the route distances: 6372m
Route for vehicle 0:
0 -> 1 -> 0
Distance of the route: 0m

Route for vehicle 1:
0 -> 3 -> 0
Distance of the route: 0m

Maximum of the route distances: 0m
Route for vehicle 0:
0 -> 1 -> 4 -> 0
Distance of the route: 1252m

Route for vehicle 1:
0 -> 5 -> 3 -> 0
Distance of the route: 6056m

Maximum of the route distances: 6056m

Out[49]: []

```

Figure 11.23: SWP solved by OR-Tool-SAT from google

## 11.7 Summary

In this section, we have presented our contribution to the scientific community as its implementation. We have proposed, designed, and implemented the algorithm to formulate the problem of social workers who visit patients at their homes. We have tested brute force techniques, Backtracking, Dijkstra, ADMM, Minimum Eigen Optimizer using different Minimum Eigensolver, such as VQE, QAOA or Numpy Minimum Eigensolver (classical), GroverOptimizer, CplexOptimizer and for the classical optimiser CobyLa, SLSQP, SPSA, ... ).

In the Generalisation of the solution section, we generalised the algorithm for generic scheduling and routing problems to be programmed under Quantum Annealing or Qiskit, etc.

In the case of Dwave, the formulation already works for QUBO. The same goes for Qiskit. In addition, however, we can use the new module from IBMQ (Fig. (11.1)) that boosts research, development, and benchmarking of quantum optimisation algorithms for the NISQ era and beyond.

This last formulation improves our proposed formulation, in the margin of the social worker waiting time, but requires more binary variables than the first one and therefore more qubits (that right now are lacking and very valuable).

Using the Qiskit library, Quadratic Programming, we will model inequality constraints and much



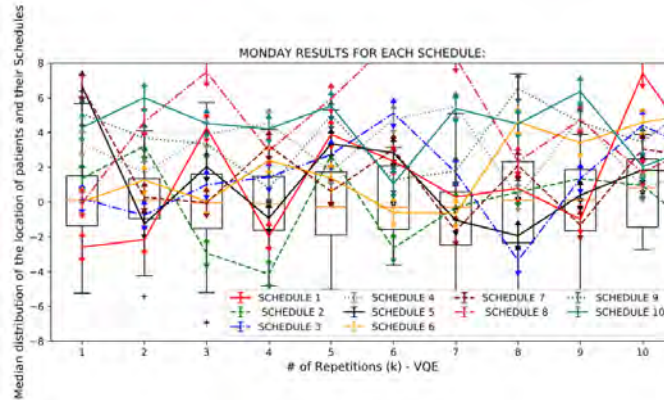


Figure 11.24: Results of the experimentation by comparing ten scenarios of the combinations of Social Workers' schedules for a fixed patient's location (for Monday). We plot the error bar of the median distribution of the location of each patient and their schedules concerning the number of repetitions of the quantum's instance.

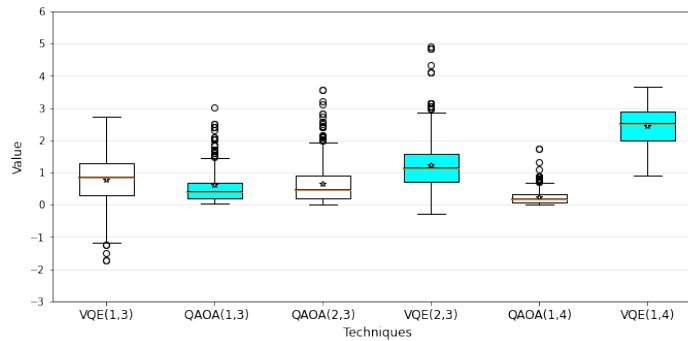


Figure 11.25: Results of the experimentation by comparing six scenarios of the combinations of Social Workers' Problem on VQE and QAOA. By changing the number of patients, the social worker shot configuration intending to analyze the quantum cost to meet the optimal solution. We realize that the QAOA with the same settings as VQE finds the optimal solution with the little sample. We must indeed consider the high values of the seed parameter.

more.

Qiskit, in addition to having converters of inequality to equality constraints, allows modelling from scratch with inequality constraints. So, we have reformulated our system with the library to have inequality restrictions. But as the techniques (slack) of using Qiskit to model, the system applies additional variables that limit us when it comes to having useful qubits. So, both the simulations and the actual tests on the quantum computer are limited since the number of qubits required is higher than the characteristics of the hardware we use.

The number of qubits for integers it is  $\log_2(ub - lb)$  for binaries it is 1:1. Inequality constraints depend on the constraint, i.e. the range of the slack variable to be introduced.



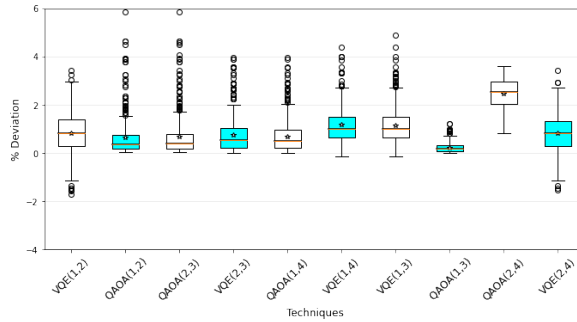


Figure 11.26: Results of the experimentation by hanging the number of patients, social worker shot configuration intending to analyze the quantum cost to meet the optimal solution. We realize that the QAOA with the same settings as VQE finds the optimal solution with the little sample.

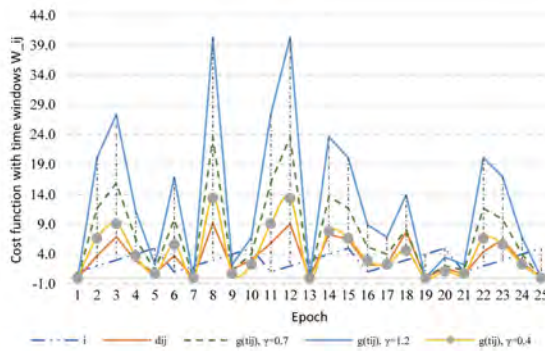


Figure 11.27: The Standard Deviation expected total anneal time for 98% per cent success for each mapping value, with the best  $\epsilon$  for each shoot is shown. Our optimal case is for  $\epsilon = 0, 7$ . Our most representative cases are for  $\epsilon = 0, 4$ ,  $\epsilon = 0, 7$  and  $\epsilon = 1, 1$ .

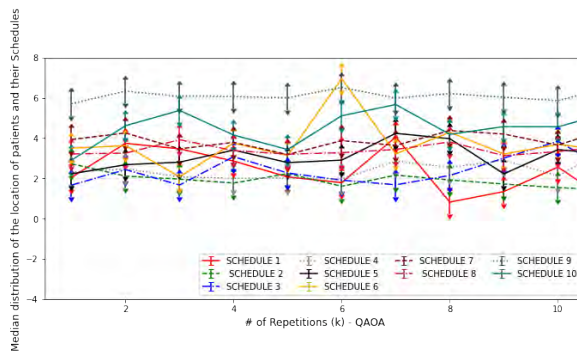


Figure 11.28: Results of the experimentation by comparing ten scenarios of the combinations of Social Workers' schedules for a fixed patient's location. We plot the error bar of the median distribution of the location of each patient and their schedules concerning the number of repetitions of the quantum's instance.

# Chapter 12

## The quantum CBR (qCBR)

### 12.1 Introduction

From our previous works till now, and following the journey of this thesis (10.1), we observe that many problems can be modelled as the social workers' problem (SWP)[31], where their resolution may have two straightforward approaches, Top-Down solution and how to solve them. Top-Down resolution is a resolution that can be set as standard. Given a combinatorial optimisation problem, its objective function can be found/written. And use the existing linear or quadratic programming methods of solving as Simplex [170], Cplex [51] or numerical to solve it. Depending on the problem's complexity class, the algorithm can be improved by introducing some heuristics or restrictions in the objective function to reduce its computational cost for an approximated solution.

On how to solve them, the approach is based on Case-based reasoning[3]. Due to its incremental and sustained learning approach based on the human learning mode, it is based on how human beings solve problems—a way of resolution based on past experiences.

In both cases, when the problem's complexity cost grows, the computational cost begins to be almost intractable for the current computation paradigm. In addition to the above, resolving these problems becomes more challenging when the input data present some overlapping issues or when outstanding accuracy is required, where quantum computing can be the choice.

The problem being solved is the NP complexity class problem, overlapping its input data and demanding considerable accuracy. For this reason, this work presents the qCBR that tries to address both limitations plus the overlapping problems by proposing a Case-Based Quantum Reasoning (qCBR) method.

The proposal seeks to answer the following questions based on experimentation results.

1. Can a quantum CBR be efficiently implemented in this quantum age?
2. Could it be said quantum CBR have a tolerance to an overlapping problem?
3. Could the proposed quantum CBR be scalable while maintaining a mean accuracy?
4. . Can this CBR be used in other combinatorial optimisation problems?

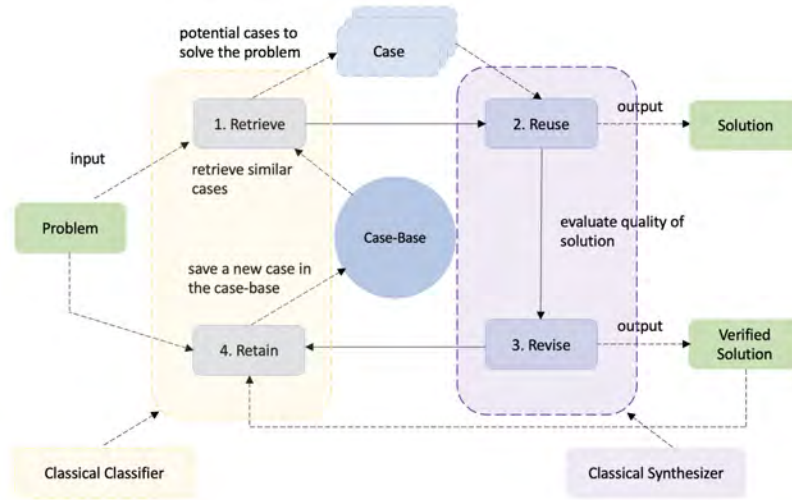


Figure 12.1: Case-Based Reasoning block diagram. For the standard CBR, two essential blocks are distinguished in this work. The classifier and the synthesiser. The classifier is made up of the retrieve and retain blocks and the re-use and revise blocks to make up the synthesis system.

Methods	Brute force	k-d tree [132]	Ball tree [172]
Training time complexity	$O(1)$	$O(dN \log(N))$	$O(dN \log(N))$
Training space complexity	$O(1)$	$O(dN)$	$O(dN)$
Prediction time complexity	$O(KNd)$	$O(K \log(N))$	$O(K \log(N))$
Prediction space complexity	$O(1)$	$O(1)$	$O(1)$

Table 12.1: Table of the NN brute force’s, k-d tree’s, and Ball tree’s complexity method. Where  $d$ , is the data dimensionality,  $N$  is the number of points in the training dataset and  $K$  is the algorithm’s neighbours’ number

### 12.1.1 Related Work

This section summarises previous works in CBR, Quantum CBR and Quantum Machine Learning (QML) algorithms.

CBR technique has been ingrained in society for several years in almost all sectors. For example in a medical reasoning program that improves with experience [158], in the individual prognosis of diabetes long-term risks[23], Case-based sequential ordering of songs for playlist recommendation [37] or ranking order in financial distress prediction [151], for monitoring the elderly at home [158], in software control[3], in the medical field[3], sequencing problems [190] etc. What we have done till now is the Top-Down method. So let us analyse the benefits of using the CBR instead of the Top-Down approach.

In the last two years, algorithms based on QML have increased considerably since the subject’s introduction previous 2014 [212]. Advances in decoherence control [205, 221] and correction system [203] are beginning to be seen to create efficient and intelligent quantum algorithms. Several quantum server providers are also available in the cloud. And with the algorithms based on the variational principle, they have opened the doors to quantum Machine Learning. The most famous is the Variational Quantum EigenSolver (VQE) [184] and the Quantum Approximate Optimization Algorithm (QAOA) [92, 113, 250]. The most promising could be the Quantum Neural Network (QNN) [17, 213,

40] Quantum Support Vector Machine (QSVM)[199, 116, 242], data loading system [181, 161] etc.

In this era of hybrid computing, which defines the potential of QML (combining quantum and classical computing), algorithms such as [249], where a cognitive engine that uses CBR-QGA to adjust and optimise the radio parameters, is presented. An initial quantum bit made up of the matching case parameters is used to avoid blindness of the initial population search and speed up optimisation of the quantum genetic algorithm. Papers [2, 19] were found that propose a new framework that can be adopted in many applications that require Computational Intelligence (CI) solutions. The framework is built under the concepts of Soft Computing (SC), where Fuzzy Logic (FL), Artificial Neural Network (ANN) and Genetic Algorithm (GA) are exploited to perform reasoning tasks based on soft cases. Also, Studies [165] focused on some vital blocks of the CBR were reviewed. It has focused on the quantum version of the k-NN algorithm that allows us to understand the fundamentals when transcribing classic machine learning algorithms into their quantum versions.

However, no quantum case-based reasoning was found to satisfy the abovementioned requirements.

## 12.2 Case-Based Reasoning

CBR [3] is a machine learning technique based on solving new problems using its experience, as humans do. The 'experience' is a case memory containing previously solved cases. The CBR cycle can be summarised in four steps: (1) It retrieves the most similar cases, (2) It adapts them to propose a new solution to the new environment, (3) It revises whether the solution is valid, and finally, (4) It stores it following a learning policy. In the present work, the proposed qCBR modifies these phases as follows (see Fig.(12.1) and (12.3)).

The CBR could be summarised in two large blocks with various functionalities: a classifier and a synthesiser. One of the classical CBR advantages can be seen by its classifier's simplicity. A k-nearest neighbours algorithm (K-NN)[99, 101] classifier is used in many. This apparent advantage can lead to collateral problems [154] at the memory level, at the level of slowness when the volume of data grows considerably and at data synthesis. The synthesis block is in charge of adapting the experience and saving to the new problem. Such adaptation and classification can be costly (Table 12.1) for considerably high data volumes [5]. Where the need for a different approach to further empower this technique.

The strategy to follow is summarised as making a CBR with a quantum classifier [218] instead of a classical neural network, KNN[99, 101] or a Super Vector Machine (SVM) [175] that offers outstanding accuracy and tolerates overlapping problems. And change the classical synthesis technique for the Variational Quantum Eigensolver [183, 117, 111] with *Initial\_point*[241] illustrated in (12.2).

The Variational Quantum Circuit (VQC) [174, 59], consists of a quantum circuit that defines the base structure similar to neural network architecture (Ansatz) while, the variational procedure can optimise the types of gates (one or two-qubit parametric gates) and their free parameters. All this is summarised in a few very identifiable steps. First, the Ansatz must be designed using a set of one- and two-qubit parametric gates. Then, the Ansatz of this circuit can follow a particular path by exploiting the problem's characteristics. A critical block is measuring the quantum state resulting from the given Ansatz. Since the VQC is a feedback system, these measurements evaluate a cost function  $CF(\theta)$ (see in the section 12.5.1) that encodes the problem. The classical optimiser has the role of optimising the cost function to find the value of the parameters that minimise it.

## 12.3 Implementation

The work proposed is the implementation of a quantum case-based reasoning (qCBR) based on figure (12.1). The following strategy replaces the classical classification: an Artificial Neural Network (ANN)

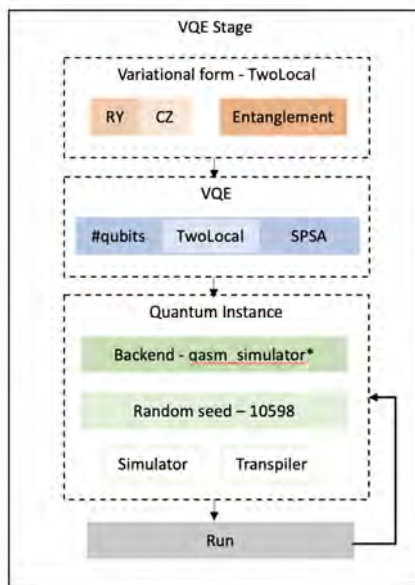


Figure 12.2: VQE working principle based on the quantum variational circuit in Qiskit

or a Super Vector Machine (SVM) or the KNN with a quantum variational classifier that guarantees the required accuracy. And for the quantum synthesis system, use the VQE with and without *Initial\_point* together with a probabilistic decision tree. Figure (12.3) shows the changes that will be introduced to obtain the qCBR, and figure (12.15) shows the detail of the functional blocks implemented with the specific problem of social workers. Finally, the two VQE blocks and the Variational Quantum Classifier are presented before detailing them.

## 12.4 Quantum classifiers

The variational quantum classifier belongs to the variational algorithms like VQE, where classically tunable parameters of a unit circuit are used to minimise the expected value of an observable. The great novelty resides in loading the data in the variational system.

As a universal quantum classifier of  $n$  qubits is pursued, a sub-base in the Hilbert vector space of equitably dividing the hyperplane  $Z$  is described as follows. Let  $B = (\{i, j, k, l, m, n, o, p\})$  be a sub-base within the Hilbert vector space, for the space of the classes  $C^{2^q}$ , the coordinates of the target classes are defined by expression (12.1) with  $q$  as the number of the qubits.

$$\begin{aligned}
 & \{i(1,0,0,0,0,0,0,0); j(0,1,0,0,0,0,0,0); k(0,0,1,0,0,0,0,0); \\
 & l(0,0,0,1,0,0,0,0); m(0,0,0,0,1,0,0,0); n(0,0,0,0,0,1,0,0); \\
 & o(0,0,0,0,0,0,1,0); p(0,0,0,0,0,0,0,1)\}.
 \end{aligned} \tag{12.1}$$

The Ansatz design and data loading (variables  $x_i$  similar to neural networks)[181] are given by

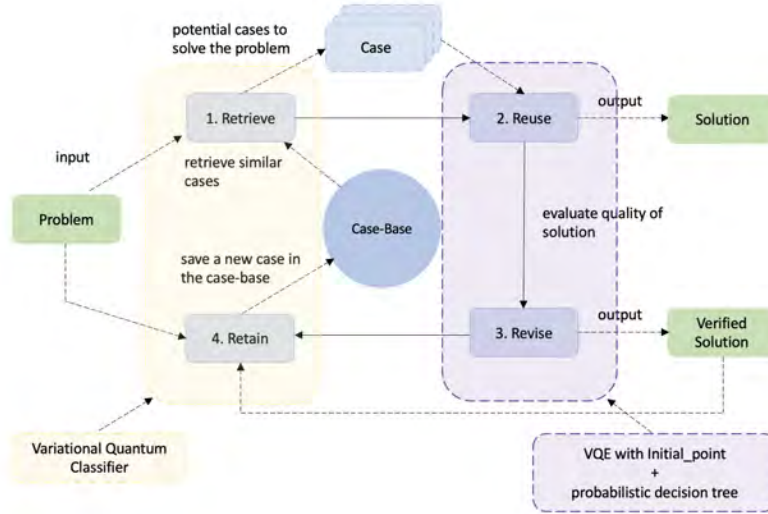


Figure 12.3: Quantum case-based system block diagram. In this scheme, to convert the classic CBR into quantum one, it is proposed to change the retrieve and retain blocks for a quantum variational classifier and the re-use and revise blocks for a synthesis system based on VQE with initial point.

equation (12.2), and its analysis is detailed in 12.5.1.

$$U = (\theta, x) = R_x(\theta_1 x + \theta_2) R_z(\theta_3). \quad (12.2)$$

## 12.5 Variational Quantum Classifier

To date, two dominant categories allow designing quantum classifiers. Although almost all are inspired by the classical classifiers (kernel or neural networks) [5], there is a new category of classifiers that respond to the current era of quantum computing (NISQ); hybrid and variational classifiers.

### 12.5.1 The Ansatz

The basic idea that one pursues is to have an ansatz (Parameterised quantum circuits (PQC)) that, formed by basic gates for quantum computing, is the most representative in the Hilbert vector space. In other words, with the control or parameterisation of these parameters, the Ansatz, in particular, maps the maximum number of points within the Bloch sphere (Fig. (12.4)). Another way to understand the objective of the Ansatz is to find the state vector that best approximates all the points of the Hilbert vector space. We must remember that the Hilbert vector space is the computational space of quantum mechanics, therefore, quantum computing. The Ansatz design inherited from previous works [29, 31] [218]. The way to load the data into the Ansatz is inspired by [181] where the data (variable  $x$ ) is entered using the weights and biases scheme. In this case, the single-qubit gate that serves as the building block for all Ansatz is given by equation (12.3) similar to neural networks.

$$U = (\theta, x) = R_x(\theta_1 x + \theta_2) R_z(\theta_3). \quad (12.3)$$

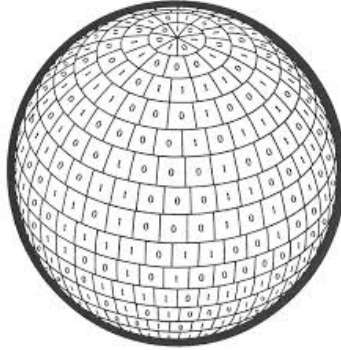


Figure 12.4: Bloch sphere, where the infinite Hilbert vector space resides. It is worth remembering that Hilbert space is the immense place where the states that describe a quantum system live. Let  $N$  be the number of qubits and let  $2^N$  be the space dimension. Given the 65-qubit IBM processor, it will be  $3.6893510^{19}$  dimensions. Many works of literature lead us to use or, at least, to think about using the quantum computer in Machine Learning (ML). By definition, the Hilbert vector space has defined these operations (internal and external products and mapping inputs in a large space) that a quantum computer performs natively and very easily [130].

Being  $\theta$  the vector of the parameters and  $R_x$  and  $R_y$  the unit gates of qubits used to create the Ansatz. To complement the experimentation scenario, it would be necessary to add the CNOT gate and the CRZ, which are the gates that help to achieve entanglement as seen in figures (12.5)(12.6) and (12.7).

The variational quantum classifier structure, figure (12.8) and (12.9), is based on layers of trainable circuit blocks  $L(i) = \prod_{i,j} U(i,j)$  and data coding, as shown in the equation (12.2) for 8 dimensional or in (12.3) for 2 dimensional data size. Additionally, the entanglement can be achieved using the CRZ or CNOT gates.

The number of parameters to optimize the classifier is given by the equation (12.4).

$$NumParam = (nL * 2d * L). \tag{12.4}$$

In this case, with  $n$  the number of qubits,  $n = 3$ ,  $L$  the number of layers (blocks), in the experiment, it is a variable data and  $d$  which is the dimension of the problem. In other words,  $d$  varies with the choice of Ansatz and whether or not entanglement is applied. In the case of the entanglements in figure (12.6), the  $d$  would be summed 1 (CRZ gate has one parameter), which equates to equation (12.5).

$$NumParam = (nL * 2(d + 1) * L). \tag{12.5}$$

### 12.5.1.1 Fidelity cost function

The similarity function follows the same strategy as the re-uploading and path; the Ansatz is different. It uses the definition of quantum fidelity with several qubits and maximizes said average fidelity between the test state and the final state corresponding to its class. Equation (12.6) [180] defines the

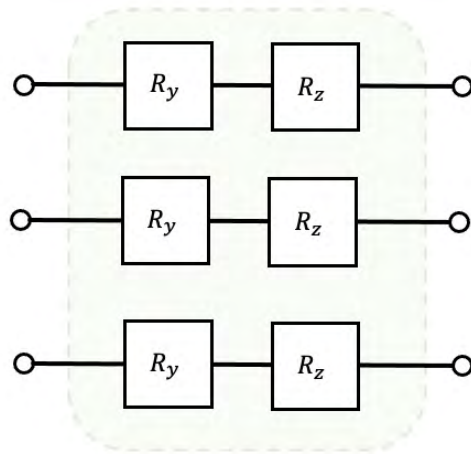


Figure 12.5:  $R_y$  and  $R_z$  Ansatzes without entanglement used in qCBR experimentation.

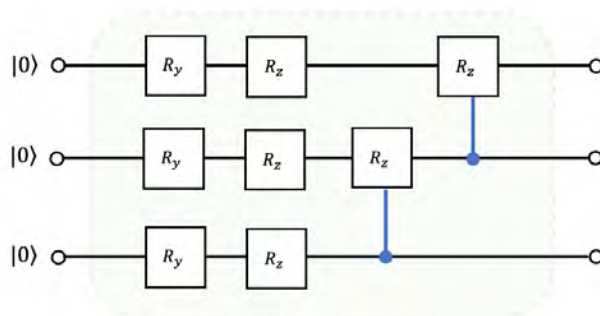


Figure 12.6:  $R_y$  and  $R_z$  Ansatzes with  $CRZ$  entanglement used in qCBR experimentation.

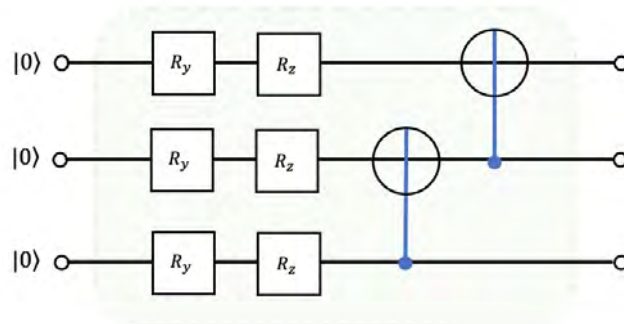


Figure 12.7:  $R_y$  and  $R_z$  Ansatzes with  $CNOT$  entanglement used in qCBR.



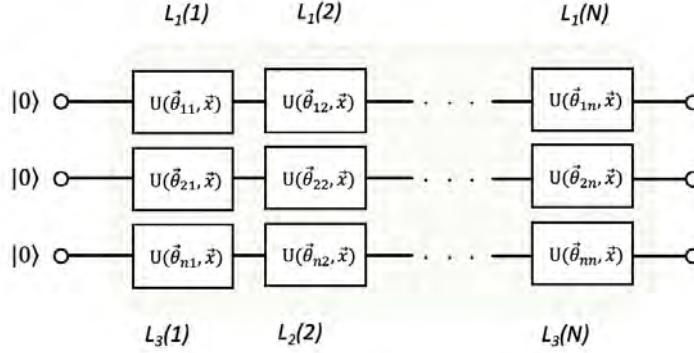


Figure 12.8: Three-qubit quantum classifier circuit without entanglement.

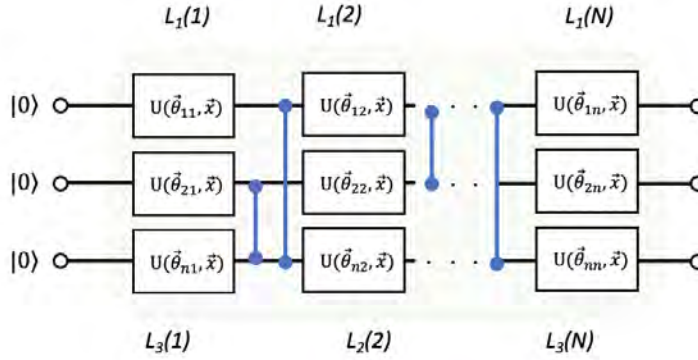


Figure 12.9: Three-qubit quantum classifier circuit with entanglement by using  $CZ$  or  $CNOT$  gates

cost function used.

$$\begin{aligned}
 CF(\vec{\alpha}, \vec{\theta}, \vec{\omega}) &= \\
 &= \frac{1}{2} \sum_{\mu=1}^M \sum_{c=1}^C \left( \sum_{q=1}^Q (\alpha_{c,q} F_{c,q}(\vec{\theta}, \vec{\omega}, \vec{x}_{\mu}) - Y_c(\vec{x}_{\mu}))^2 \right), \tag{12.6}
 \end{aligned}$$

with

$$F_{c,q}(\vec{\theta}, \vec{\omega}, \vec{x}_{\mu}) = \langle \psi_c | \rho_q(\vec{\theta}, \vec{\omega}, \vec{x}) | \psi_c \rangle. \tag{12.7}$$

Where  $\rho_q$  is the reduced density matrix of the qubit to be measured,  $M$  is the total number of training point,  $C$  is the total number of the classes,  $\vec{x}_{\mu}$  are the training points and  $\vec{\alpha} = (\alpha_1, \dots, \alpha_C)$  are introduced as class weights to be optimized together with  $\vec{\theta}$ ,  $\vec{\omega}$ , are the parameters and  $Q$  the numbers of the qubits. Counting on  $Y_c(\vec{x}_{\mu})$  as the fidelity vector for a perfect classification. This cost function (12.6) is weighted and averaged over all the qubit that form this classifier. To complete the hybrid system, it is used for the classical part, the following minimisation methods above cited: L-BFGS-B [62], COBYLA[227] and SPSA[63].

## 12.6 Details of qCBR

The operation of the **retrieve** (prediction) block is given by a new case (schedule). In this experimentation, the schedule that best adapts to the latest issue to be solved is recovered with the predict method, which is executed at a time  $O(\log(MN))$ . It is worth saying that, due to the SWP descriptions, a possible schedule change, a stage of understanding or interpretation is necessary since an adequate resolution of the new schedule cannot be carried out if it is not understood with some completeness. This stage of understanding is a simple decision algorithm with minimal intelligence.

Once having the predicted solution, the synthesis block creates a new solution (proposed solution) by combining recovered solutions. To do this, the algorithm is divided into two main lines (figure(12.15)). A line that determines an acceptable degree of error (after a probabilistic study) that the predicted solution can be considered the proposed solution. The second branch is in charge of improving the expected solution towards a better-proposed solution. To do this, the *Initial\_point* associated with the retrieved schedule is retrieved from the case memory, and the Variational Quantum Eigensolver is executed with very few shots ( $k$  shots). The idea here is to refine the new schedule's similarity with the recovered one. Operating the VQE with *Initial\_point* provides the algorithm with parameter values through the initial point as a starting point for searching for the minimum eigenvalue (similarity between the two times) when the new time's solution point is believed to be close to a matter of the recovered schedule. This is how the **Re-use** block works. These operations have a complexity of  $O(k\log(N) + \log(NM))$ . Where  $N$  is the number of social workers,  $M$  is the number of patients, and  $k$  is the number of shots.

The algorithm's processes to review the proposed solution are seen below the **Re-use** block in figure (12.15). It is essential to classify the best possible solution for the proposed prototype. The best possible solution is calculated with the VQE with the maximum resolution and depth (for the variational part). Once the solution is obtained, it is compared with the proposed solution and said solution with its characteristics is added to the new schedule before storing it (see figures (12.12) and (12.13)). The computational complexity of the Revise is determined by  $O(\log(N) + ICA)$ . In this work, access to data (states) is selected by  $O(\log(MN))$  due to the characteristics of the inner products and superpositions.

One of the most critical blocks in this work is to **Retain**. This block is the heart of the CBR because it is the classifier and because it is the block that allows us to conclude that it has been learned from the previous cases. Unfortunately, not all instances (schedules) are saved in this job, leading to the excessively slow classifier. Therefore, the best issues (timetables) that summarise all the essential information are retained in this part of the algorithm.

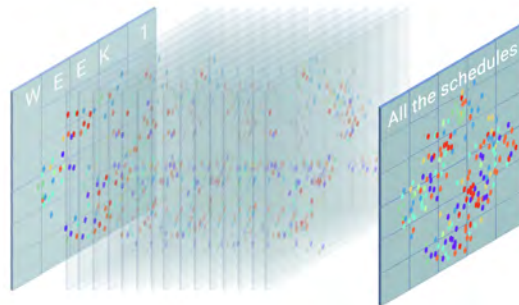


Figure 12.10: Representation of the generation of the  $n$  weekly schedules of the SWP. The last plane, the one to the right of everything, represents all the SWP classes. The overlapping effect generated by the social workers' problem's characteristics and experimentation scenarios can be observed. This experimentation leads us to use the ICA technique to have a resulting dataset regardless of the schedules.

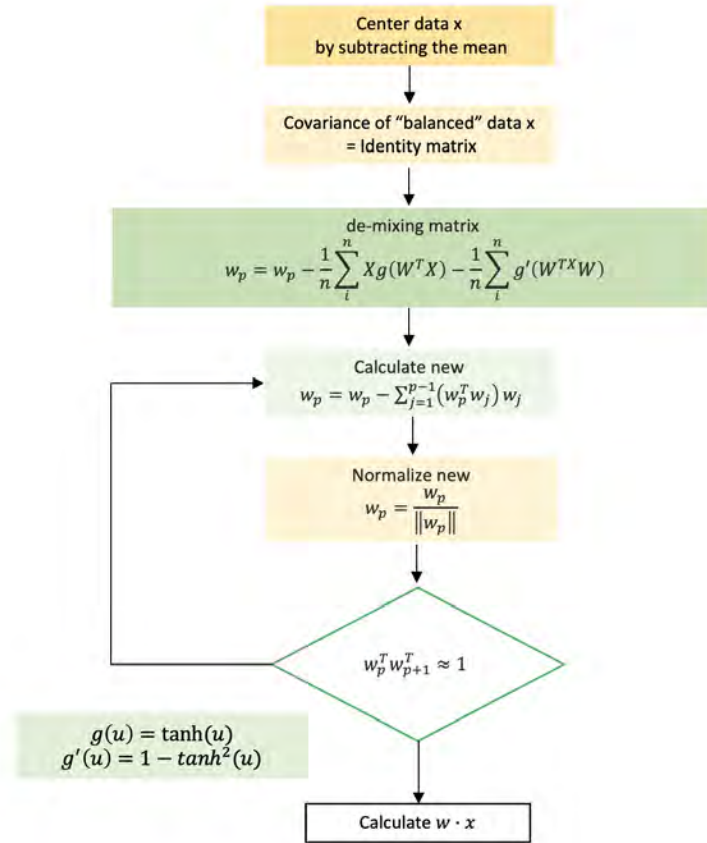


Figure 12.11: Fundamental processes to apply ICA to SWP dataset. The first thing that is done is to centre the data  $x$  by subtracting the mean, balancing the data  $x$  removing its variance, and calculating the unmixing matrix of  $W$ . Then, the new value of  $w$  is calculated. Then  $w$  is normalised before checking if, with the said value, the algorithm converges or not. If it does not converge, a new  $w$  must be recalculated, and if it converges, calculate the scalar product of  $\langle x, y \rangle$  to obtain the independent weekly schedules.

The **Retain** process begins with the treatment of schedules, searching for the algorithm's best efficiency, which is a challenge to solve in this block. In the case of SWP, the patient visits hours have a margin range of 30mn. Therefore, if one schedule starts at 9:00, the next could begin at 9:30, leading to a dataset with overlap between plans if many programs have similar time ranges spread over different days of the week. Suppose we add the non-linearity of the data to this issue. In that case, an almost perfect classifier is needed with a mean accuracy greater than 80% or the treatment and intelligence system classifier can be helped. Then, you can separate the overlapping components of the cases.

In this work, we contemplate both scenarios. First, get an excellent classifier and apply data processing techniques to help a poor classifier. Using the standard classifier, ICA [188] is applied to the original data to reduce the effects of the degree of overlap (figure (12.10)) without losing the fundamental characteristics of the data. Figure (12.11) summarises the processes and operations applied to reduce the overlapping effect observed in the generation of SWP schedules. The complexity of this operation is noted as  $O(ICA)$ . The PCA is then used to reduce the data dimension from 8 to

2 and apply it to the designed variational classifier with the complexity equal to  $O(PCA)$ . Once the best time is determined, we retain the knowledge acquired at the case's resolution.

### 12.6.1 Memory Structure

Next, some test benches based on the memory structure described in figure (12.12) and (12.13) are defined to train the parameterised quantum circuit, and its performance is analysed in terms of the circuit architecture. Finally, the results and discussions session will emphasise the classifier with or without entanglement and a comparative study with different ansatzes.

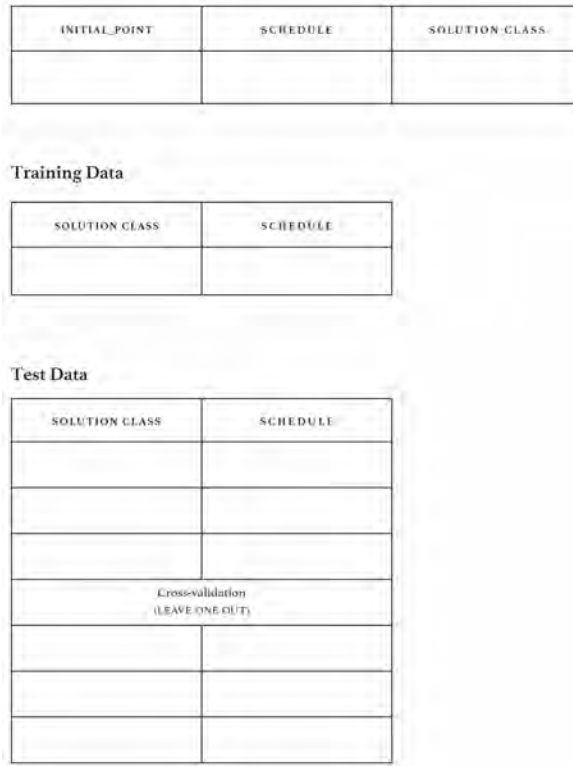


Figure 12.12: qCBR's Memory structure. The use the cross-validation technique helps to improve the quality of the classifier training[59]

The memory structure of the qCBR's retention system is given in figure (12.12). In addition, the solution class (target) related to the different initial points associated with each of the samples that make up the training dataset can be seen. This solution class is also associated with the result of the VQE.

To fill out the data structure created in the classifier to train and test its predictions, the parameters of the *Initial\_point* obtained by VQE are abstracted from the result. And it is composed of each class's coordinates with the following parameters: start time  $sT$  and end time  $eT$  of patient 1 to  $n$ , where  $n$  is the maximum number of patients in the app. Figure (12.14) summarises the data's representation and description that make up the training dataset.

Each coordinate's class corresponds to the VQE solution following the memory structure in figure (12.12) where the class number of the classifier is given by equation (12.8) taking into account the

SOLUTION CLASS	INITIAL_POINT_1	INITIAL_POINT_2	...	INITIAL_POINT_M
A				

SOLUTION CLASS	RESULT
A	{0,0,...,0}

Figure 12.13: Memory structure used for the Training System. A label identifies each class. And in this case, A is one of the labels. For more detail, refer to the code [27].

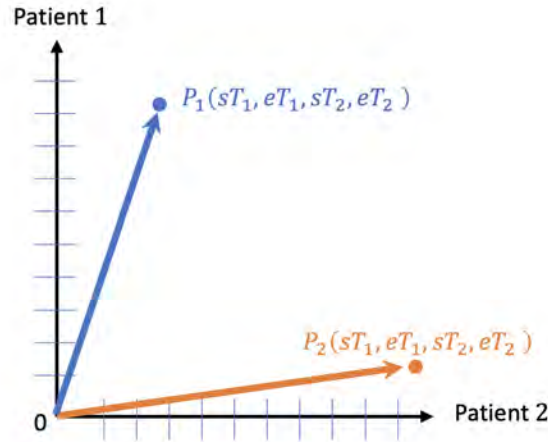


Figure 12.14: The SWP is represented in vector form to take advantage of the Hilbert vector space’s characteristics within quantum computing. It is seen that each patient represents a dimension, and the points that make up the dataset are a dimension of  $2n$  coordinates,  $n$  being the total number of patients. In this figure, to simplify the understanding, we use two patients, therefore, two dimensions

conditions that every worker has a patient and that the workers are indistinguishable (that is, it doesn’t matter whether the social worker  $m_1$  takes care of the patient  $n_1$  and  $m_2$  takes care of  $n_2$  or vice versa).

$$N_{SOL_{SWP}} = \frac{1}{m!} \sum_{k=0}^{m-1} (-1)^k \binom{m}{m-k} (m-k)^n. \quad (12.8)$$

Let  $n$  be the number of patients and  $m$  the number of social workers and knowing that the appearance orders patients in the schedule (from earliest to latest, let  $n_1$  be the patient with the earliest plan and  $n_k$  be the patient with the latest program).

In this work all the tests done are for  $n = 4$  with the data structure equal to  $(sT_1, eT_1, sT_2, eT_2, sT_3, eT_3, sT_4, eT_4)$ ; an 8-dimensional vector for each social worker visit the patient. In this case, the number of qubits will be defined by  $q = \log_2(N_{SOL_{SWP}}) = 3$ . These qubits are used to instance the quantum classifier, and it is worth to mention that the classifier must have  $N_{SOL_{SWP}}$  classes.

The detail of the qCBR’s implementation and analysis is in section 12.6.

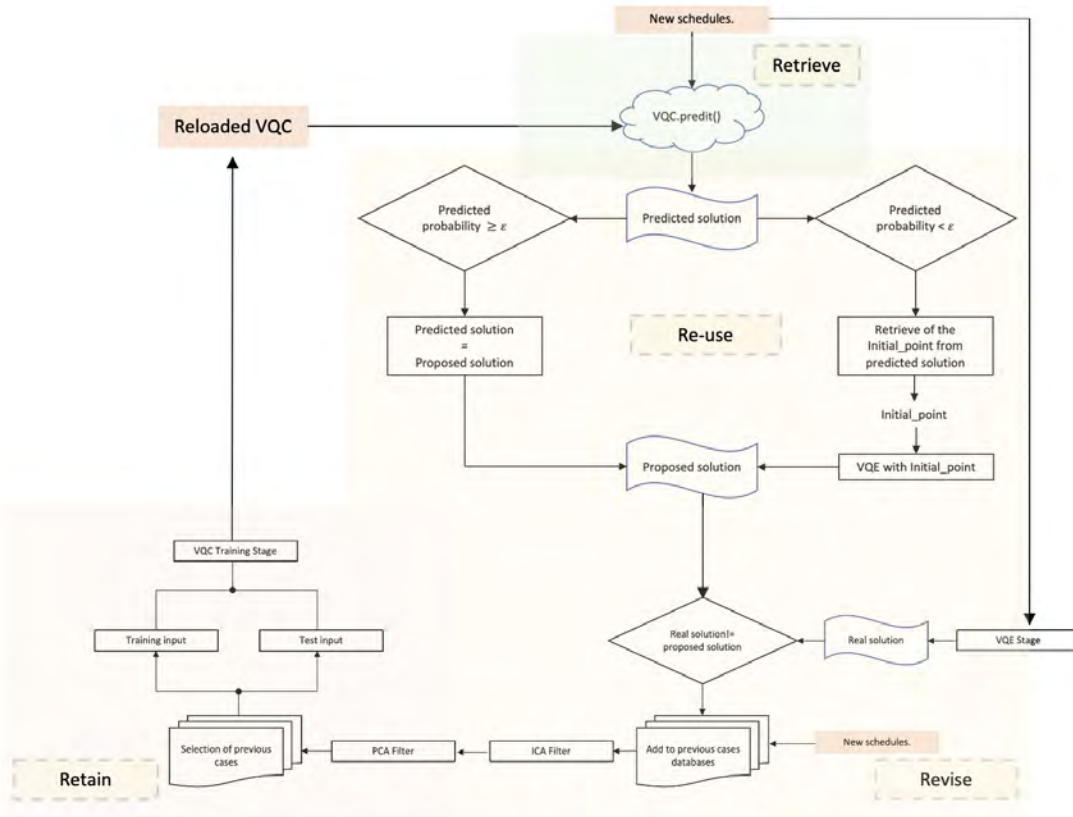


Figure 12.15: Block diagram proposed for the resolution of qCBR, considering a real dataset with an overlap problem between the data components. This block diagram contemplates the treatment of the input data and the use of ICA[188] and PCA[248] before training and classifying the data. In this version, a classifier based on the re-uploader has been designed to handle the classification tasks. And for the synthesis tasks, a decision tree passed in the classifier predictions have been used together with the VQE plus the *Initial\_point*

## 12.7 Results

Figures (12.16) to (12.17) show the implementation outcomes performed in *Qibo*[90] and *Qiskit*[164, 241] to identify the best model architecture and represent functions similar to qCBR.

Tables (12.2) to (12.4) show the global results of qCBR solving the SWP. With table (12.2), the outcome of the different tested scenarios can be observed. Varying the number of patients, social workers, and the quantum circuit's depth to see the global hit number of the qCBR. In table (12.3), we can observe the resolution of the SWP, taking into account five patients, four social workers and setting the depth of the quantum circuit to eight. With this scenario, it can be observed how the behaviour of the qCBR is a function of the number of cases. It is seen how the system, after a threshold of 240 cases already resolved, begins to give very satisfactory results. Table (12.4) repeats the steps of table (12.3) with the only change of the input data; the number of patients and social workers. Tables (12.5) to (12.8) show the result of the implementation of the classical CBR leveraged on ANN and KNN to solve the SWP.

Tables (12.3) and (12.4) show the best performance at the average accuracy level of qCBR for the

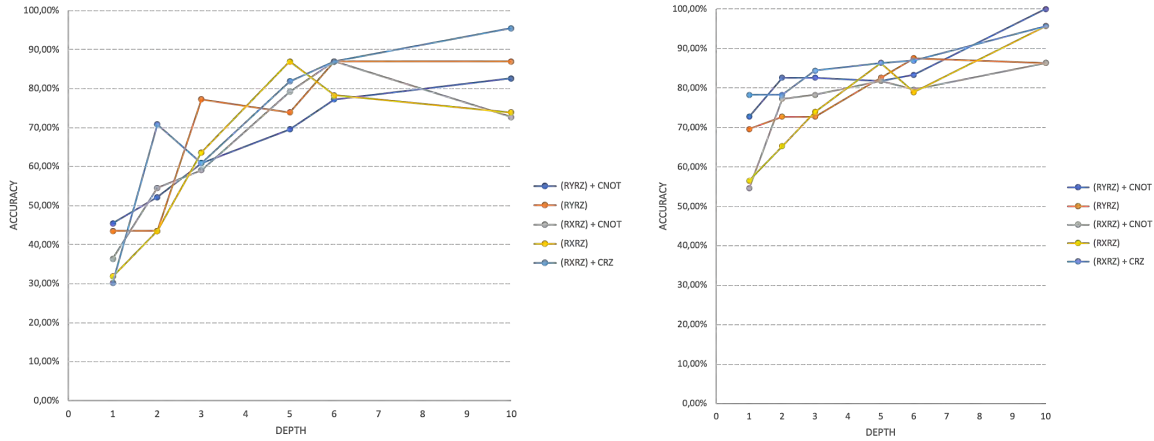


Figure 12.16: Comparative graphs between different ansatzes, taking into account the classifier’s accuracy as a function of depth. Represents the evolution of the ansatzes of dimension two and eight

qCBR solving the Social Workers’ Problem					
#Patients	#SW	#Qubits	#Layers	#Cases	Accuracy
3	2	6	2	580	82.5
4	3	12	3	580	82
5	2	20	4	580	82.5
5	3	20	5	580	87
5	4	20	8	580	92.8
5	4	20	10	580	100

Table 12.2: The result of the qCBR with a variational classifier and using the VQE and the *Initial\_point* with some decision trees as a synthesizer (12.15). This table shows the different studies made as a function of the quantum circuit’s depth (layer). number of the patients and the social workers. The accuracy of the classifier is the maximum with the number of layers equal to 10. SW denotes Social Workers.

classic (Table (12.5) and (12.6)). Table (12.8), (12.7) and (12.2) show the degree of scalability of the qCBR as a function of the variation in the number of patients and social workers. It has also been seen that qCBR is much better shared with overlapping as we wanted to demonstrate.

The experiment was also made of skipping the Principal Component Analysis (PCA) module [248][226], Independent Component Analysis (ICA)[188] and creating a classifier of the same dimension as the data (8 sizes). And the results obtained have been very satisfactory at the level of Ansatz’s accuracy and depth. Still, the need to change the *BFGS* optimiser to the SPSA has become visible due to its slow convergence for the number of data and high parameters. Figure (12.16) describes the behaviour and compare the two scenarios.

Later the Re-use module was analysed using VQE with *Initial\_point* to synthesise the predicted results. In the graph shown in figure (12.17), it is observed how the algorithm, without initial parameters, tends to use a high energy constant of variation to reach an approximation of the fundamental state quickly. Which makes it have to progressively reduce said constant after several iterations,  $n$ , to find the local minimum. On the other hand, when using an *Initial\_point*, the algorithm does not need to start with a high variation to reach energy bands close to the ground state since it is much closer to said energy, reducing the number of iterations necessary reach to the local minimum. We can then see how qCBR can afford to run VQE with *Initial\_point* to refine the accuracy of its results

qCBR solving the Social Workers' Problem For 5 patients and 4 social workers		
Layers	#Cases	Accuracy
8	20	-
	50	12.5
	100	72.5
	240	92.1
	340	95.5
	480	97.2
	500	98.7
580	99.1	

Table 12.3: The result of the qCBR for many patients and social workers fixed at 5 and 4, respectively. The better behaviour of qCBR can be observed for some cases greater than 240. To have a good functioning of the qCBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 92%. The "accuracy" value is the percentage of the number of correct solutions found by the qCBR. A hyphen (-) denotes no solution found within the 20 cases. These tests were done for the quantum circuit depth (layers) equal to 8.

qCBR solving the Social Workers' Problem For 4 patients and 3 social workers		
Layers	#Cases	Accuracy
8	20	-
	50	11.5
	100	73.1
	240	91.1
	340	91.9
	480	96.6
	500	98.1
580	99.0	

Table 12.4: The result of the qCBR for many patients and social workers fixed at 4 and 3, respectively. The better behaviour of qCBR can be observed for some cases greater than 240. To have a good functioning of the qCBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 91%. The "accuracy" value is the percentage of the number of correct solutions found by the qCBR. A hyphen (-) denotes no solution found within the 20 cases. These tests were done for the quantum circuit depth (layers) equal to 8.

since it requires fewer iterations to find the solution closest to the minimum, not assuming such a high computational cost as it would be running VQE without initial parameters.

The qCBR complexity (Table (12.9)) is provided below, where it can be observed that the *Retain* is the highest cost operation and has an exponential improvement compared to a *Retain* of a classic CBR that is usually of the order of  $O(M^2(M + N))$  [199]

## 12.8 Discussions

The proposed qCBR works very well and meets the objectives set using quantum computing to create efficient quantum case-based reasoning. One of the issues to comment on is the improvement observed in (figure(12.16)) both two and height-dimensional classifiers. Due to the small number of depths,



CBR with KNN solving the Social Workers' Problem For 5 patients and 4 socials workers		
Layers	#Cases	Accuracy
1	20	-
	50	42.9
	100	46.5
	240	52.6
	340	55.3
	480	56.8
	500	60.7
	580	63.1

Table 12.5: The classical CBR result on KNN classifier for many patients and social workers fixed at 5 and 4, respectively. The better behaviour of this CBR can be observed for some cases greater than 240. To have a good functioning of the CBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 52.6%. The "accuracy" value is the percentage of the number of correct solutions found by the CBR leveraged on KNN, applying a 10-KFold cross-validation process. A hyphen (-) denotes no solution found within the 20 cases. All these tests were done for the layer equal to 1.

CBR with KNN solving the Social Workers' Problem For 4 patients and 3 socials workers		
Layers	#Cases	Accuracy
1	20	-
	50	55.1
	100	58.5
	240	70.3
	340	71.1
	480	73.6
	500	74.8
	580	76,8

Table 12.6: The classical CBR result on KNN classifier for many patients and social workers fixed at 4 and 3, respectively. The better behaviour of this CBR can be observed for some cases greater than 100. To have a good functioning of the CBR, it must be iterated with the social workers' dataset 239 times. And at the case number of 240, we will have an accuracy of 70.3%. The "accuracy" value is the percentage of the number of correct solutions found by the CBR leveraged on KNN, applying a 10-KFold cross-validation process. A hyphen (-) denotes no solution found within the 20 cases. All these tests were done for the layer equal to 1.

CBR leveraged by CNN solving the Social Workers' Problem				
#Patients	#SW	#Layers	#Cases	Accuracy
3	2	2	580	65.4
4	3	2	580	43.3
5	2	2	580	37.3
5	3	2	580	26.3
5	4	2	580	45.2

Table 12.7: CBR with a neural network classifier and a backtracking algorithm as a synthesizer. SW denotes Social Workers.

#Patients	#SW	#Layers	#Cases	Accuracy
3	2	1	580	95.6
4	3	1	580	77.8
5	2	1	580	47.8
5	3	1	580	44.7
5	4	1	580	63.1

Table 12.8: CBR with a KNN classifier and a backtracking algorithm as a synthesizer. SW denotes Social Workers.

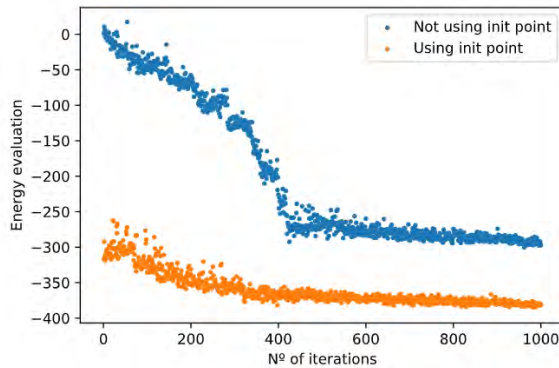


Figure 12.17: Energy comparison between VQE algorithm without using or not the initial point

Methods	Complexity
Retrieve	$O(\log NM)$
Re-use	$O(K \log(N) + \log NM)$
Revise	$O(\log(N)) + ICA$
Retain	$O(\log NM) + PCA$

Table 12.9: Table of the complexity of qCBR counting the PCA cases and the ICA complexity. In this case, K, the number of shots of the VQE, is fixed to 50

but with many more parameters, the height-dimensional classifiers have an average of about 25% of improvements over the two-dimensional ones. With this result, in the case of not wanting an accuracy of around 95%, shallow depth could be used, and computation time saved, depending on the problems. Despite all these improvements, it is essential to highlight some improvement aspects. For example, in the intelligence system that allows deciding the proposed solution, now, the average of the *Initial\_point* of each solution class samples' *Initial\_point* is used. However, it could still be seen based on the predicted solution, which *Initial\_point* is the most suitable for the answer to propose. Thus, the cases to be re-used could be better classified.

Also, one of the improvements is to train the classifier with noisy data further so that the qCBR can adapt to real past situations that adjust to the new position. Because, in practice, there is usually no past case strictly the same as a new one.

The last improvement is generalising the qCBR to serve various problems (betting problem, financial, software maintenance, human reasoning, etc.). For this, It would be a success to focus on the design in memory of cases so that different data sizes can be indexed and, above all, train the classifier with several other data models.

## 12.9 Conclusions and further work

We observed the outstanding performance of qCBR compared to its classical counterpart on the average accuracy, scalability and tolerance to an overlapping dataset. Some of the standard and classical CBR problems have been mitigated in this work. With the design that has been proposed in this work, it has been possible to measure situations of difficult similarity between cases. Despite the non-linear and overlapping attributes, the classifier has been endowed with characteristics that arrive at two similar topics that may seem quite different by having different values in features, but not very important. For example, in the VQE with *Initial\_point*, we can have different *Initial\_point* associated with each training class sample with the same class. With the technique of the average of the "Initial\_point", it is possible to solve this problem by providing the qCBR to distinguish the similarity between cases. Another issue that qCBR mostly solves is the time required to classify a new topic.

With the results of the two implementations (classical and quantum CBR), it is observed that the classical CBR designed with the KNN behaves better for some determined cases (table (12.8)). It is seen that the system has not finished learning thoroughly (table (12.5) and (12.6)) contrary to the qCBR (table(12.3) and (12.4)). This is due to its classifier's accuracy, without forgetting the significant contribution of its synthesis system.

Another improvement that qCBR introduces is when retaining cases, implementing a retention system that maintains model cases and that, together, synthesise the real and most important information. One of the improvements to consider is the implementation of quantum ICA. In this way, the classical ICA analysis's complexity cost will be significantly reduced. Also, counting that the PCA is saved since we have an 8-dimensional classifier, the complexity of the qCBR would be that of the classifier plus some setup constants.

The other exciting line of the future is to design the memory of cases using the quantum technique of random-access memory (qRAM) [50] to improve the memory of stored instances.

Now we will generalise the SWP into Batching and Picking problem by defining qRobot.

We will work on qRobot as a quantum computing approach in mobile robot order picking and batching problem solver optimisation in the next session. We will change the social workers by robots and the patient by item to pick and batch.

## 12.10 Summary

In this section, we have seen how to solve the SWP with a machine learning approach. We have seen how a universal quantum classifier has been created from scratch, developed a formula for calculating the number of solutions, and solved the issue of overfitting and overlapping that the SWP dataset has. In the next chapter, we will see the generalisation of the SWP problem through qRobot.

# Chapter 13

## The qRobot: generalizing the SWP

Following the journey of this thesis work, here, we propose and solve the Batching and Picking Problem with the generalisation of the SWP called qRobot.

### 13.1 Introduction

From DHL, Gartner and others [22, 142, 208], we know that the first wave of automation using smart robotics has reached the logistics industry. Driven by rapid technological advancements and increased affordability, robotic solutions (software and hardware) are forcibly entering labour logistics, supporting flawless processes and boosting productivity. Robots, especially mobile, will adopt more roles in the supply chain, helping workers with storage, transportation and little by little, they will expand their service. In fact, in some countries, there are already robotic delivery services [239].

We are already living an exponential increment of mail-order shopping, online shopping and supply chain systems, requiring large-scale logistic centres. Almost everyone can order products remotely, and the logistic centre increases its functionalities, including keeping and shipping products. While there was a tendency to increase the adoption of automated systems based on robots powered by AI to increase efficiency [234, 217, 11], COVID-19 introduced the concept of touch-less online shopping that reduces the risk of infections. Smart Warehouses are the epicentre of the cost-efficiency of any e-commerce company [228].

The emerging field of hybrid (quantum-classical) algorithms joins CPU and QPU [133] to speed up specific calculations within a classical algorithm. This allows for shorter quantum runs that are less susceptible to the cumulative effects of noise and work well in current devices.

Recently the scientific community are researching the real implementation of quantum computing algorithms in mobile platforms because performances are not here yet [74].

In this work, we demonstrate that we can implement this system in well-known, widely used robotics fields, computer systems like raspberry-pi exploring the performance of a quantum picking and batching model. A hybrid system is proposed to effectively replace the current ones and open the doors to quantum computing in robotics. In addition, we are analysing the results obtained with different public access simulators on the market: IBMQ, Amazon Braket (Dwave), and Pennylane. As far as the authors know, this is the first time this type of implementation has been done.

## 13.2 Work Context

According to [69, 61, 139], supply chains, warehouses and distribution centres occupy a very important position when storing and serving customer demand. Today, to be competitive within this sector, Logistics 4.0 has been created, known as the set of artificial intelligence technologies and techniques that seek the efficiency of the movements of materials and products in a factory or warehouse. In addition, better time management helps logistics companies find and locate a material, reduce fatigue and possible workplace accidents, and spend less time documenting items.

Many works of literature highlight these factors as the main ones where the loss of time and resources in a process require an urgent solution, and precisely, it is technologies such as Artificial Intelligence and the Internet of Things (IoT), which today allow us to optimise them[12, 68, 34].

Researchers have addressed the multiple order picking planning problems in the last decade. The study of the efficiency of a Warehouse can be addressed based on numerous parameters. According to [107], there are three key considerations: 1) Performance Measure (time, cost, productivity, and service), 2) How we model the warehouse (Analytical model, Mathematical Model, or Simulation), and the combination of factors (storage location assignment, routing, order batching, or other order picking planning problems).

Based on data from [107], we can see the percentage of the relevance of the considered order picking planning problems based on the percentage of papers that are related to such challenges:

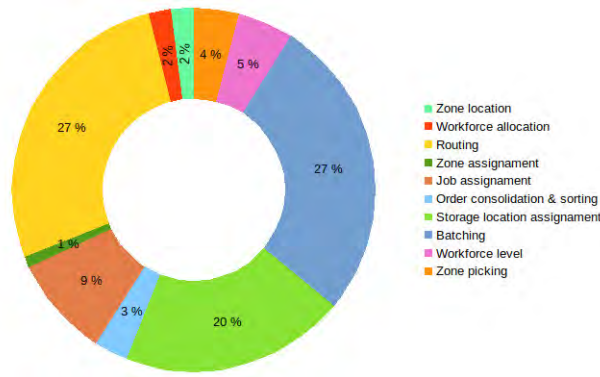


Figure 13.1: Distribution of considered order picking problems based on the percentage of publications.

As we can see in Fig.13.1 Picking and Batching are the top priorities based on the research contributions.

Order preparation (picking) is one of the most frequent and costly operations in labour [69, 61] since it is responsible for recovering the items required by the orders of customer orders (could also be supplied, but in this article, we focus exclusively on sales orders), and to create the batches, grouping several orders of orders in a picking list to collect all the batch demands in a single warehouse tour. In this last part of order preparation, our quantum algorithm comes into action to optimise the routes travelled to achieve efficient picking.

There are many techniques and strategies for solving the picking problem. The most striking are “The selected techniques for evaluation include A \*” [86], “Potential Fields (PF)”, “Rapidly-Exploring Random Trees \* (RRT \*)” [147, 146, 70] and “Variations of the Fast-Marching Method (FMM)” [196]. Other strategies have been explored using the TSP and the VRP as algorithms to solve the picking problem. In this case, if the number of order orders per lot is greater than two [102], picking becomes an NP-Hard problem in which the number of possible lots and binary variables increases exponentially with the number of purchase orders [102]. From there, several heuristic techniques, methods and algorithms (for example, genetic) were created to relax these difficulties [75, 34, 69, 118, 139, 231].

However, and as mentioned above, depending on the volume of data, the computational cost of the algorithm becomes intractable for classical computing.

The latter leads us to explore new approaches to the large-scale picking problem, and one of the approaches to consider to solve this task is quantum computing [6]. Quantum computing could help us change the degree of complexity of the problem, enhanced by its high computing power. Among the great fields where quantum computing is called to stand out is constraint satisfaction problems (CSP) [232]. One of the useful algorithms in this field is Quadratic Unconstrained Binary Optimization (QUBO) problems [140].

Amazon Braket[56] is a cloud-based (Fig.(13.2) and Fig.(13.3)), fully managed quantum computing service that helps researchers and developers get started into quantum world technology to accelerate research and discovery. Amazon Braket provides a development environment to explore, create, test and run quantum algorithms, quantum circuit simulators, and different quantum hardware technologies.

We will use all these related works to define an appropriate strategy for our proposal in this NISQ era.

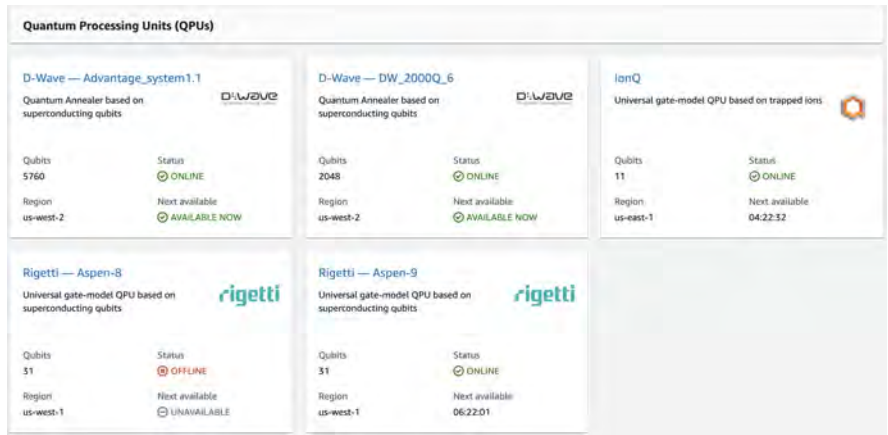


Figure 13.2: Technical Specifications of the Quantum Hardware Technologies (Gate-based superconducting qubits, Gate-based ion traps and Quantum annealing) available in Amazon Braket.

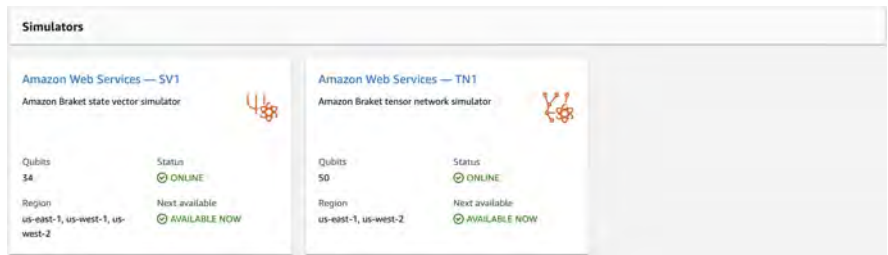


Figure 13.3: Technical Specifications of Quantum Simulators systems where we can see it state vector simulator (34 qubits) and tensor network simulator (50 qubits).

Studying and comparing different optimisation methods of warehouse’s challenge, like picking and batching, [107] proposes three options: analytical models, simulation experiments, and mathematical programming. In our approach, we consider the latter. We use a set of mathematical expressions that describe the problem, represented by an objective mathematical function and constraints within the classical context and translate it to the quantum domain.

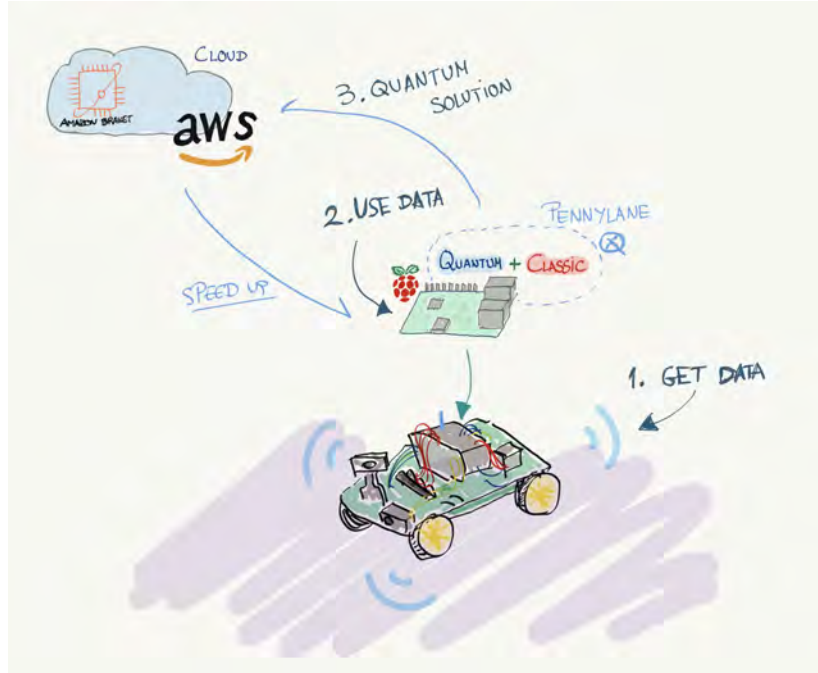


Figure 13.4: We propose a robot that prepares batches and increases the efficiency of picking in a warehouse, taking advantage of the classic Machine Learning experience and leveraging hybrid computing (classical + quantum) in the cloud and distributed. This robot uses the Optimal routing strategy to calculate the shortest route, regardless of the layout or location of the items.

While reviewing the state of the art research, this reference [245] was found. The integrated order routing and the batch problem is modelled in such systems as an extended multi-tank vehicle routing problem with network flow formulations of three indices and two commodities. Such a variable neighbourhood search algorithm provides close to optimal solutions within a computational time acceptable for classical but not quantum computing.

This article intends to bring quantum computing to robotics by proposing an approach that combines the experience of classical robotics computing with the computation of complex and high-cost processes by quantum computing. We suggest preparing an environment to execute the quantum algorithms in the mobile and autonomous robot remotely and locally and designing a quantum algorithm that helps the efficiency of the warehouse management.

From the variational principle in section(9.2), the following equation  $\langle H \rangle_{\psi}(\vec{\theta}) \geq \lambda_i$  can be reached out. With  $\lambda_i$  as eigenvector and  $\langle H \rangle_{\psi}(\vec{\theta})$  as the expected value. By this way, the VQE finds (13.1) such an optimal choice of parameters  $\vec{\theta}$ , that the expected value is minimized and that a lower eigenvalue is located.

$$\langle H \rangle = \langle \psi(\theta) | H | \psi(\theta) \rangle. \quad (13.1)$$

We will use the VQE (Fig. (13.5)) to find the minima of our objective function translated to the Ising model.

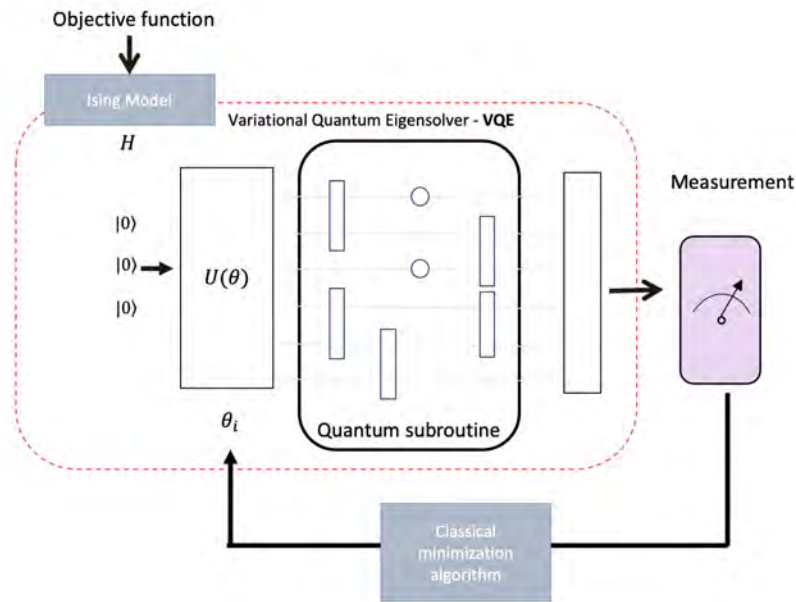


Figure 13.5: VQE working principle based on the quantum variational circuit.

## 13.3 Implementation

To carry out the implementation of our proof of concept (Fig.(13.4) and Fig.(14.2)), we must first prepare the programming environment. Considering that the core of our robot will be the Raspberry Pi 4[98], the first thing to do is prepare it so that it can execute quantum algorithms with the guarantees required for the proposed application and especially for future operations on gradients. It is necessary to install an ARM64 operating system[126, 129] with all the needed packages to run all the required environments to carry out this project. We took advantage of the work for Raspberry Pi Os Desktop (32-bit) on which the author describes how to install and run Qiskit - IBM's open-source quantum computing software framework[241]— on a Raspberry Pi to turn it into a quantum computing simulator and use it to access real IBM quantum computers. In our case, we do need ARM64 because we need to execute at least TensorFlow's version 3.2.1. The tasks to convert the Raspberry Pi 4 in our "quantum computer" are described in the following reference [26].

After setting up the environment, we will focus on designing and experimenting with the announced proof of concept.

### 13.3.1 The problem's formulation

In this formulation, we will seek to optimise the collection of the products, and, later, we will make the batches.

To carry it out, we will consider the following assumptions:

1. The strategy we will follow is the picking routing problem to retrieve each lot which the total distance travelled to retrieve all the items in a lot will be calculated.
2. The warehouse configuration is given in figure(13.8).
3. For the orders of the storage positions, more than one picking robot can be used.



4. Movements in height are not considered.
5. Each product is stored in a single storage position and only one product is stored in each storage position.
6. Each picking route begins and ends at the Depot.
7. The load capacity for each order will not exceed the load capacity of the picking robot.
8. At the moment, the division of order orders is not contemplated. That is, only the batches of closed orders can be prepared.
9. The concept testing will be done on all AWS-Braket, Pennylane, D-Wave and Qiskit environments. And we will stick with the scenario that best benefits our proof of concept.
10. We will use the docplex[120] to model our formulation.

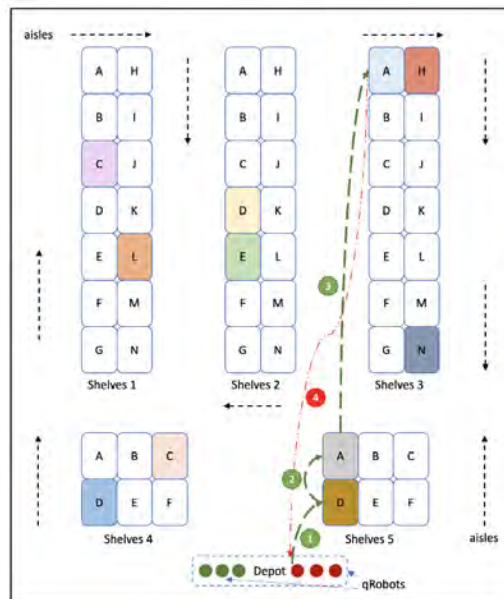


Figure 13.6: Scenario 1, Independent lots. The robot receives the orders and calculates which order is the most optimal according to the coordinates in which each product is found. In this example, lot 4 is the most optimal.

### 13.3.2 Picking and Batching formulation

The formulation is represented as follows. In this scenario, the travel load is described according to the number of robots we have. Let's imagine that we have several robots and that each of them makes a single trip. It would be the same as saying that we have a single robot that makes  $n$  trips.

Let  $N_0$  be the origin node, let  $N_1 \dots N_n$  be the nodes of the products, let  $W_1 \dots W_n$  be the weights in kg associated with for each product, let  $d_{i,j}$  be the distance from node  $i$  to  $j$ , let  $M$  be the maximum load of the qRobots, let  $K$  be the number of qRobots available, let  $t$  be the instant,  $i$  the node (product),  $p$  the robot and let  $x_{t,i,p}$  be our binary variable (for example, for  $x_{2,3,2} = 1$ . It means that at time 2, the qRobot 2 is at node 3). In our formulation, time really tells us the order, that is

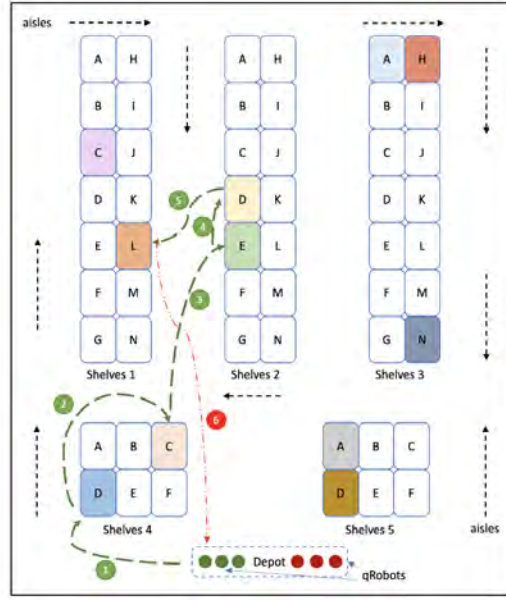


Figure 13.7: Scenario 2, Collecting products in the same route from different batches. The robot will calculate a path that includes all the products to optimize their collection in a single journey. For example, if the product from Lot 2 is next to one from Lot 1, the robot will pick it up and store it in the basket from Lot 2.

to say  $t = 0$  will be the origin  $t = 1$  the moment in which it goes for the first batch. At  $t = 2$  it will be the moment of the second so on.

$$\min_x \sum_{p=1}^K \sum_{t=1}^{n+1} \sum_{i=0}^n \sum_{j=0}^n x_{t-1,i,p} x_{t,j,p} d_{i,j}, \quad (13.2)$$

s.t.

$$\sum_{p=1}^K x_{0,0,p} = K, \quad (13.3)$$

$$\sum_{p=1}^K x_{n+1,0,p} = K, \quad (13.4)$$

$$\sum_{t=1}^{n+1} \sum_{i=1}^n x_{t,i,p} W_i \leq M \quad \forall p \in \{1, \dots, K\}, \quad (13.5)$$

$$\sum_{i=1}^n x_{t,i,p} = 1 \quad \forall t \in \{1, \dots, n+1\} \quad (13.6)$$

$$\forall p \in \{1, \dots, K\},$$

$$\sum_{t=1}^{n+1} \sum_{p=1}^K x_{t,i,p} = 1 \quad \forall i \in \{1, \dots, n+1\}, \quad (13.7)$$

$$x_{t,i,p} \in \{0, 1\} \quad \forall t \in \{0, \dots, n+1\}$$

$$\forall i \in \{0, \dots, n\} \quad (13.8)$$

$$\forall p \in \{1, \dots, K\}.$$

The equation (13.2) is our new objective function. Here we minimise the total distance. We add the distance of all the robots travelling simultaneously, and we will check the nodes' distances. Restriction (13.3) establishes that all the qRobots start from Depot. The restriction (13.4) establishes that all the qRobots end at Depot. The constraint (13.5) establishes any robot  $p$  can carry more load than allowed. The constraint (13.6) declares that each robot can only be one node at any time. (13.7) establish that throughout the entire route, the robots together pass each node only once and the restriction (13.8) describes that  $x_{t,i,p}$  are binary variables.

The number of the qubits to perform this algorithm is equal to  $K(n+1)(n+2) + K\lceil \log_2 M \rceil$ . At this point, we can only map our objective function in quantum and then solve it with a VQE.

### 13.3.3 Mapping the classical to quantum optimization

A common method for mapping classic optimization problems to quantum hardware is by coding it into the Hamiltonian[91] of an Ising model [157].

$$H_{Ising} = \sum_{i < j} J_{i,j} \sigma_i \sigma_j + \sum_i h_i \sigma_i. \quad (13.9)$$

Where  $\sigma_i$  is the product of  $n$  identity matrices  $I$  except a gate  $Z$  in the  $i$ -th position and  $\sigma_i \sigma_j$  product of identities minus gates  $Z$  in positions  $i$  and  $j$ .

As we already can build our objective function as a QUBO in the form  $\langle x^T | Q | x \rangle$ , now we can map our QUBO to Ising Hamiltonian formulation leads to calculating the values of  $J_{ij}$  and  $h_i$ .

The transformation between QUBO and Ising Hamiltonian is  $z_i = 2x_i - 1$ , where  $z_i$  is a new variable that can take the values  $-1$  or  $1$ . This means we will have the algorithm in Ising form by writing an algorithm for QUBO with this single variable change. That is very useful to have the algorithm for various platforms based on quantum gates (IBM Q and PennyLane) or quantum annealing (meanly D-Wave) in case of going from the Hamiltonian form. we can now solve our Picking and Batching Problem with VQE  $\langle \psi(\theta) | H | \psi(\theta) \rangle$ .

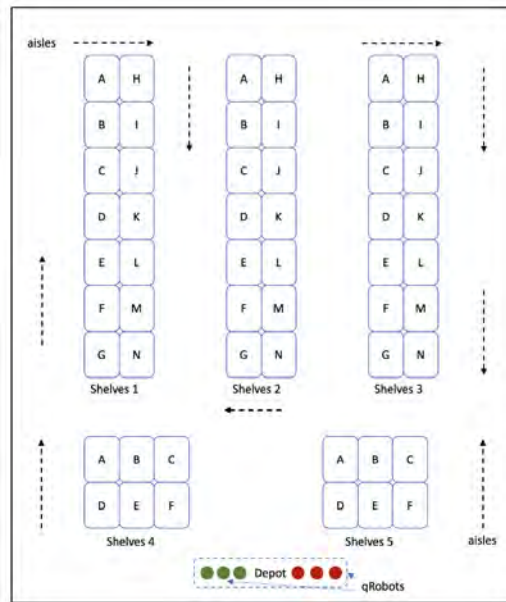


Figure 13.8: Structure of our warehouse with pick locations. The warehouse has a rectangular layout with no unused space. We use all the parallel corridors. This proof of concept contemplates a single warehouse used to take the order and deliver it, and it is also divided into blocks, which contain slots for storing products, and transverse aisles separate them. The cross aisles do not have any products but allow the collector to navigate the warehouse. We base our picking strategy on minimising the route and optimising batch preparation. We do not contemplate shelving of different levels.

## 13.4 qRobot Results

Before analysing all the results of our proof of concept in detail, it is of the utmost importance that we validate its operation globally and affirm that qRobot meets our expectations and works as expected. Let's split the results of this proof of concept in two. 1, the configuration and conversion results of the Raspberry Pi 4 in a quantum computing environment (Fig.(13.12) to Fig.(13.15)) and 2, the picking and batching algorithm results represented by tables (13.1) to (13.3) on one side and Fig.(13.10) and (13.11) on the other.

The block diagram (Fig.(13.9)) summarises the result of the implementation of the qRobot. The first thing we did was determine the mathematical model of our problem. We then used the Docplex to model our objective function and its constraints. For our proof of concept, we used the Docplex library packages to move from Docplex to QUBO. We had two possible operations according to our objectives from this point on. First, we modelled the problem for computers based on quantum gate technology like IBMQ, and second, for annealing computers like D-Wave. Our experiments used the Exact solver and VQE to test the Qiskit framework as samples based on quantum gates. But before using the VQE, we needed to map the QUBO model to the Ising model. Then, when we used the D-Wave computer, we only needed to reform the QUBO output list from Docplex to the QUBO format of the D-Wave computer.

Table (13.1) shows the experimentation results by setting the number of qRobots as their capacities (maximum load) at 1 and 45, respectively. We compared the execution time of our algorithm with different public access simulators on the market during this experimentation, solving the problem of picking and batching. We observed that, for issues of this nature, especially due to the number of qubits required in each scenario, the behaviour of the D-Wave is the desired one at the temporal level,

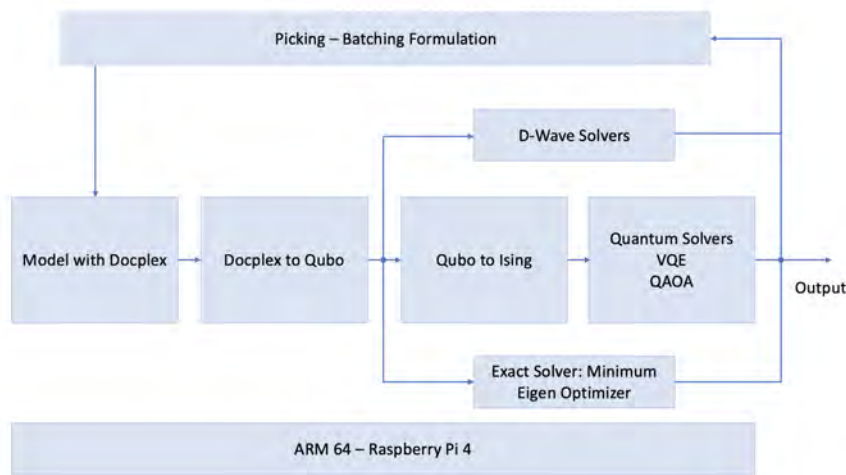


Figure 13.9: qRobot operation diagram. This diagram shows all the necessary blocks and processes that allow modelling the picking-batching problem and its proper functioning on the Raspberry Pi 4.

comparing it with Gate based Quantum Computing. However, it should be taken into account that, for experiments with numbers of qubits less than 20, the behaviour of these simulators is equated with the D-Wave. This experimentation helps to have a clear vision about the feasibility of this proof of concept.

Continuing with the analysis of the results, table (13.2) shows us the computational results of our picking and batching algorithm considering 1 qRobot through AWS-Braket and on the real quantum computer D-Wave Advantage\_system1[247]. The time value is an average and does not count latency time, job creation, and job return time.

We also analyse the latency time when running the algorithm from the qRobot to the quantum computer. The quantum computer was in Oregon (US) and our qRobot in Barcelona (Spain) and Segovia (Spain) in the tests we had done. Out of all the tests we have run, we had an average latency time of around 2 seconds plus all job management processes rising to roughly 8 seconds. For the number of qubits greater than 30, it is very convenient to use AWS-Braket (Advantage\_system1.1) instead of Qiskit or PennyLane for the number of qubits and the execution time; it is differentially better. This scenario makes the use of quantum in robots very viable. For tests with a value of  $M$  less than those in the table, the number of qubits is relaxed, and the execution time is improved. This leads us to normalise the weights of the batches. Since the number of qubits follows the formula  $K(n + 1)(n + 2) + K \lceil \log_2 M \rceil$ , where the  $K \lceil \log_2 M \rceil$  qubits are needed as ancillaries qubits.

We also analyse the quantum real-time execution deeply through table (13.3). We have measured the execution time without counting the latency time, creating jobs, and returning the work.

Fig.(13.10) offers us the algorithm results in different scenarios where we analyse some important cases, which helped us determine viable strategies within our proof of concept. In addition, it is important to note that our algorithm minimises the distance travelled and optimises the number of qRobots. Finally, the Fig.(13.11) repeats almost the same scenario but now considering 7 items with the same number of qRobots.

The benchmark of the qRobot's algorithm in different quantum simulators.				
# items	# qubits	DWave - Time(s)	IBMQ - Time(s)	PennyLane - Time(s)
2	18	1.92	1.89	1.94
3	26	3.2	737.46	656.93
4	36	4.88	-	-
5	48	7.60	-	-
6	62	11.16	-	-
7	78	15.89	-	-
8	96	21.72	-	-
9	116	30.18	-	-
10	138	43.29	-	-
11	162	53.28	-	-
12	188	63.45	-	-

Table 13.1: In this experimentation, both the number of qRobots and their capacities (maximum load) are fixed and are worth 1 and 45 respectively. We compare the execution time of our algorithm in the different public access simulators in the market, solving the picking and batching problem. We see that for issues of this nature, and especially for the number of qubits required in each scenario, the behaviour of the D-Wave is the desired one at the temporal level, comparing it with technologies based on quantum gates. However, it should be noted that for the experiments on numbers of qubits less than 20, the behaviour of these simulators is equated with the D-Wave. This experimentation helps to have a clear vision about the feasibility of this proof of concept.

## 13.5 Discussions

We have achieved that, given a warehouse with a single robot, a list of several products with their respective loads and a list of batches, our system minimises the distance to collect all the products and prepare the batches. Furthermore, this formulation solves the order in which the robot could manage all the products and make the packs pass through the Depot. Another important achievement this approach offers is that each robot makes a single trip. However, it is possible to band the code so that if we find ourselves in a situation where there are many batches to create and only a few robots to do the picking, these robots can be made to make the necessary trips if we have  $k$  qRobots that make at most one trip (we will never need more with  $n$  batches). In this way, we will obtain all the packages for trips we are interested in doing. A more understandable way of explaining it would be that when the first qRobot has finished its journey, it should only be ordered to do the one that would have made the qRobot  $k + 1$ , which does not exist and so on with all the qRobots  $k + 2$ ,  $k + 3$ ,  $k + 4$  etc. until all scheduled batches are finished.

In addition to the processor, quantum computing simulation is closely related to memory. It takes up memory to simulate a quantum computer, but the quantum computer does not need that memory, so it is assumed that it will be better. In this proof of concept, we got the following results using 8GB of RAM on the Raspberry Pi 4. The algorithm of collection and generation of packages take between 2 and 450 seconds to generate the batches and picking. If you want the qRobot to do all these tasks, we must calculate the path before forming the packs. That said, we must remember that if we want to recalculate new routes when the robot has already left, we must consider a lower latency time but close to said interval. A possible solution would be to choose a Raspberry with more RAM capacity. For example, if we had a 64GB Raspberry Pi, this time would be cut to 2/8, and it would take approximately 56.25 seconds (less than a minute) to create the batches. However, in this era of quantum computing, it is not representative to compare times. The computational differences will be noticed when the problems begin to grow, not on the small scales we are currently dealing with.

Effective viability for today's warehouses would consist of splitting the tasks of the robots and

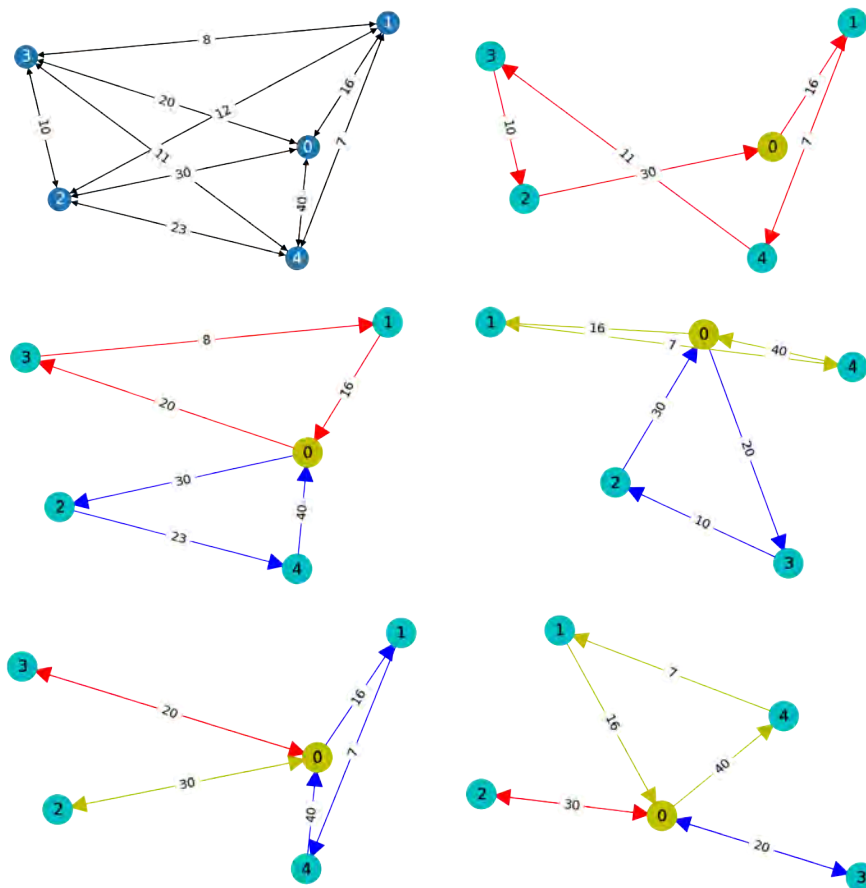


Figure 13.10: In these graphs, we can observe the results of the algorithm in different scenarios. A different colour represents each qRobots; qRobot number 1 is red, next is blue, and the third is yellow, so on. While the depot is the 0 node in yellow color and the rest of the nodes are represented in blue. The weights of each item (not normalized) in kg are  $w_0 = 0, w_1 = 8, w_2 = 8, w_3 = 3$  and  $w_4 = 3$ . The maximum capacity of each qRobots is 45. In this case, we have 4 items and the possibility of using up to 3 qRobots. Reading the images from left to right, we see that the nodes and their respective distances are shown in the first image. The second image shows the result of the algorithm having a qRobot. In the third and fourth images, we can see two different cases solved by two qRobots. And finally, in the fifth and sixth images, we can see two other issues solved by three qRobots. It is important to highlight that our algorithm in this proof of concept minimises the distance travelled and optimises the number of qRobots necessary to solve the cases presented. If it judges that the task can be performed with a single qRobot, it will not send 2 qRobots.

having a qRobot that centralises all the requests and passes them to the fleet of  $n$  qRobots so that they collect the products belonging to each batch.

We also performed tests and developed a system to model the problem and run it on a Dwave. Despite the optimisation of the algorithm, the number of necessary qubits ( $K(n+1)(n+2)+K\lceil\log_2 M\rceil$ ) and the need for low latency make this code adapted to the Annealing model. For this reason, we have prepared the Raspberry PI to run D'wave directly and under Amazon-braket-ocean-plugin. This scenario could have a "reasonable" latency for low data volume. Things that today, computers based

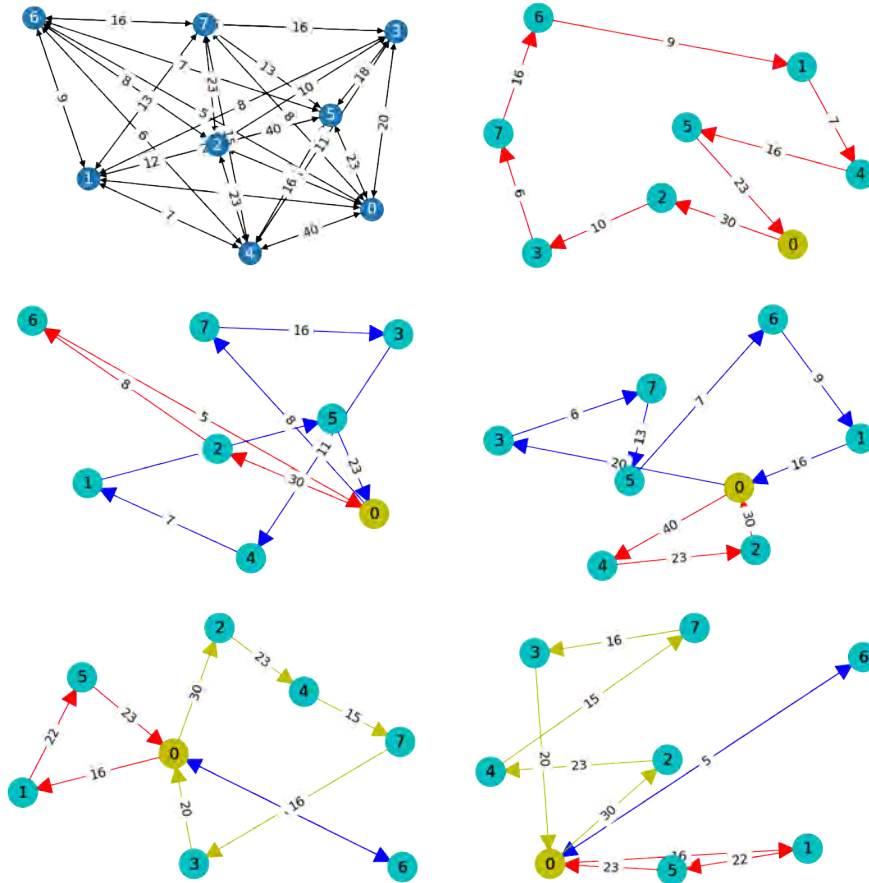


Figure 13.11: In these graphs, we can observe the results of the algorithm in different scenarios. A different color represents each qRobots; qRobot number 1 is red, the next is blue, the third is yellow, and so on, while the depot is the 0 node in yellow color, and the rest of the nodes are represented in blue. The weights of each item (not normalized) in kg are  $w_0 = 0, w_1 = 8, w_2 = 8, w_3 = 3, w_4 = 3, w_5 = 1, w_6 = 2,$  and  $w_7 = 4$ . The maximum capacity of each qRobots is 45. In this case, we have 7 items and the possibility of using up to 4 qRobots. Reading the images from left to right, we see that the nodes and their respective distances are shown in the first image. The second image shows the result of the algorithm having a qRobot. In the third and fourth images, we can see two different cases solved by two qRobots. And, finally, in the fifth and sixth images, we can see two other issues solved by three qRobots. It is important to highlight that our algorithm in this proof of concept minimises the distance travelled and optimises the number of qRobots necessary to solve the cases presented. If it judges that the task can be performed with a single qRobot, it will not send 2 qRobots.

on quantum gates cannot offer.

## 13.6 Conclusions and further work

As we have seen, the problem raised throughout this work offers us an efficient way of managing a series of  $K$  qRobots to collect a set of orders, optimising the number of robots used. The provided approach



# items	Capacity	# qubits	AWS-Braket[57] Avg Time (s)	IBMQ[241] Avg Time (s)	PennyLane[43] Avg Time (s)
2	15	10	11.23	0.053	0.041
3	15	16	22.96	0.40	0.27
4	15	24	33.07	537.46	480
5	15	34	57.93	—	—
6	15	46	118.41	—	—
7	15	60	145.83	—	—
8	15	76	296.81	—	—
9	15	94	335.64	—	—
10	25	115	427.36	—	—
11	25	137	650.25	—	—
12	25	161	908.71	—	—

Table 13.2: Table of the computational results of our picking and batching algorithm on only 1 qRobot. The value of time is an average and includes the waiting time, queue, execution and return of the solution. In the case of  $K$  is equal to 2 for 9 items with the qRobot capacity equal to 15, the number of qubits is 188. The execution time is on AWS Braket and on the D-Wave Advantage\_system1 quantum computer. We can realize that there is a latency time in executing the algorithm from the qRobot to the real quantum computer. In the tests we’ve done, the quantum computer is in the US West (Oregon). Of all the tests that we have done, we have had an average latency time of about 2 plus all the work management processes that rises more or less to about 8 seconds. For the number of qubits exceeding 30, it is very convenient to use AWS-Braket (Advantage\_system1.1)[247] instead of Qiskit or PennyLane. By the number of qubits and the execution time, that is differentially better. This scenario makes the use of quantum in robotics very viable. For the tests with a value of  $M$  lower than those in the table, the number of qubits is relaxed, and the execution time is improved. This leads us to normalize the weights of the batches. Since the number of qubits follows the formula  $K(n + 1)(n + 2) + K \lceil \log_2 M \rceil$ .

# items	Capacity	# qubits	AWS-Braket[57] Avg Time (s)	IBMQ[241] Avg Time (s)	PennyLane[43] Avg Time (s)
2	15	10	0.13	0.053	0.041
3	15	16	0.31	0.40	0.27
4	15	24	1.69	537.46	480
5	15	34	7.93	—	—
6	15	46	11.31	—	—
7	15	60	22.30	—	—
8	15	76	36.11	—	—
9	15	94	54.01	—	—
10	25	115	80.40	—	—
11	25	137	139.67	—	—
12	25	161	195.60	—	—

Table 13.3: In this table, we only consider the running time of the quantum algorithm on the real quantum computer from the qRobot (Advantage\_system1.1[247]), not counting latency time, job creation, and job return time.

applies to a "central computer" capable of carrying out all the calculations and giving each robot orders. However, when we begin to deal with very large problems both in the number of products and the number of robots, the number of qubits required will tend to grow too large. A possible solution is to distribute the calculation of a central computer to each robot so that each one has to calculate

```

pi@raspberrypi:~/qRobot/libcint/build $ python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> tf.__version__
'2.3.1'
>>> import pennylane as qml
>>> qml.__version__
'0.14.1'
>>> import braket_sdk as braket_sdk
>>> braket_sdk.__version__
'1.5.15'
>>> import qiskit
>>> qiskit.__qiskit_version__
{'qiskit-terra': '0.17.1', 'qiskit-aer': None, 'qiskit-ignis': '0.6.0', 'qiskit-ibmq-provider': '0.12.2', 'qiskit-aqua': None, 'qiskit': '0.25.1', 'qiskit-nature': '0.1.1', 'qiskit-finance': None, 'qiskit-optimization': '0.1.0', 'qiskit-machine-learning': None}
>>> exit()
pi@raspberrypi:~/qRobot/libcint/build $

```

Figure 13.12: This figure shows that we judge important environments to carry out quantum computing to robotics and beyond. We can see the correct installation of TensorFlow 3.2.1 as required for all gradient operations; see the installation of PennyLane version 14.1, the installation of the latest version of Amazon Braket, and all the packages of the newest version of Qiskit 0.25 minus the qiskit-machine-learning package.

```

pi@raspberrypi:~/qRobot/libcint/build $ jupyter notebook --version
6.3.0
pi@raspberrypi:~/qRobot/libcint/build $ jupyter --version
jupyter core      : 4.7.1
jupyter-notebook : 6.3.0
qtconsole        : 5.0.3
ipython          : 7.22.0
ipykernel        : 5.5.3
jupyter client   : 6.2.0
jupyter lab      : not installed
nbconvert        : 5.4.1
ipywidgets       : 7.6.3
nbformat         : 5.1.3
traitlets        : 5.0.5
pi@raspberrypi:~/qRobot/libcint/build $

```

Figure 13.13: In this figure, we can see the correct installation of the Jupyter package and the Jupyter notebook that has been our environment of proof of concept. With this, everything is ready to import or write code in the different frameworks mentioned above (IBMQ, AWS-Braket, PennyLane and D-Wave).

its route given a list of products to be collected. In this case, the equations of the problem would not change, just take  $K = 1$  for each qRobot and apply the technique mentioned at the beginning of the discussion. Although it may not be possible to reach the best solutions, this process of distribution of the calculation would suppose a significant computational cost reduction despite the need to create the batches beforehand. This search for batch creation will be studied in future projects. On the other hand, it is important to note that the problem dealt with has a QUBO-type formulation, which allows it to be executed in annealing-type quantum computers. This makes a big difference in today's era (NISQ) as we have managed to work with 200 qubits versus the 30 qubits that we would have with a gate-based quantum computer. Finally, note that the defined problem minimises the total distance the robots travel, making it worthwhile for not all the robots to come out. For future lines, we will address the same problem. Still, We will continue to try to reduce the total times instead of the distance travelled (as done in this previous work[30]) since this situation is also very important in warehouse logistics.

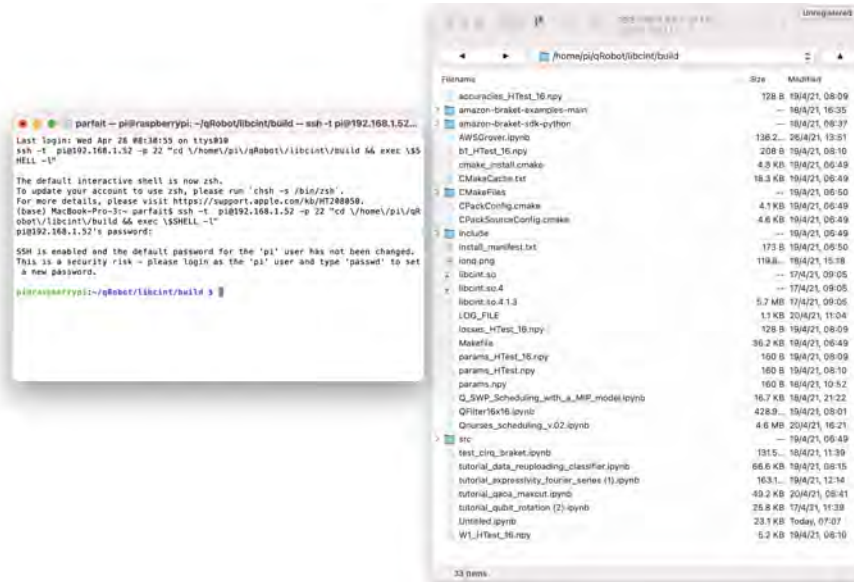


Figure 13.14: This figure shows the files window through the CyberDuck client SSH [122] viewer with the directory and file structure. And on the left, you can see the terminal that gives access to the qRobot. To access the qRobot by SSH, the username and password are required. Everything is configurable [1].

### 13.7 Summary

In this section, we have seen how SWP has become widespread through qRobot. This generalisation was carried out through a proof of concept on a Raspberry Pi 4 and used the D-Wave framework (from AWS-Braket platform), Pennylane and Qiskit, to solve the formulation. This is how we wanted to respond to the universality of the algorithm. In the next section, we will see an overview of this thesis work’s results, discussions, and conclusion, which has led us to formulate a combinatorial optimisation project with hard constraints, its resolution both Top-down and through QML well as its generalisations through the qRobot. All this to answer the initial hypothesis.





## Part III

# Conclusions



# Chapter 14

## Results

### 14.1 Introduction

In this section, we will present the results of this thesis.

Figure (10.1) helps us to understand the results of this thesis work. The results of this thesis work have been broad and positive. In section (11.3), we can see the detailed results of the SWP in the various scenarios designed.

One of the first results that we would like to highlight is the formulation ((11.11) with (11.9) and (11.10)) of the SWP. After its implementation and testing with five different techniques.

We have tested our algorithm on VQE, QAOA, Numpy Minimum Eigensolver classical, GroverOptimizer, CplexOptimizer, Numpy Minimum Eigensolver for QUBOs, Backtracking and CP-SAT Solver from google on the *ibmq-16-melbourne v1.0.0*, *ibmq-qasm-simulator (up to 32 qubits)* with COBYLA and SLSQP as the classical optimizer. We can consider QAOA as a particular case of VQE. This is one of the reasons we can apply the Hamiltonian of the Ising model of our proposed formulation directly to QAOA almost without modify. The Figures (11.2) to (11.13) show the results of our algorithm once we executed our algorithm under the IBMQ. The optimal visit considers the hours of visits to form the optimal schedule. We have done several experiments with the QML defining different scenes using shot configuration. With our quantum machine, we can configure the number of repetitions of each circuit for sampling we need. With that, we will be doing machine learning on circuit design for each shot. And when the loop ends, we will get to the ground state energy. Consequently, we solved our problem by creating one quantum circuit for each shot, and the best circuit will be the one that optimizes our Social Workers' Problem. We also prepared the solution in QUBO form for the annealing computing.

In section (12.7), we can see the detailed results of the qCBR of the different scenarios. These very promising results lead us to consider our approach to this age of quantum computing. Nevertheless, let us summarise the qCBR results here.

Figures (12.16) to (12.17) offer the implementation outcomes performed in *Qibo*[90] and *Qiskit*[164, 241] to identify the best model architecture and represent functions similar to qCBR.

Tables (12.2) to (12.4) present the global results of qCBR solving the SWP. With table (12.2), the result of the different tested scenarios can be observed by varying the number of patients, social workers, and the quantum circuit's depth to see the global hit number of the qCBR. In table (12.3), we can observe the resolution of the SWP, taking into account five patients, four social workers and



setting the depth of the quantum circuit to eight. With this scenario, it can be observed how the behaviour of the qCBR is a function of the number of cases. It is seen how the system, after a threshold of 240 cases already resolved, begins to give very satisfactory results. Table (12.4) repeats the steps of table (12.3) with the only change of the input data; the number of patients and social workers. Tables (12.5) to (12.8) show the result of the implementation of the classical CBR leveraged on ANN and KNN to solve the SWP.

Tables (12.3) and (12.4) show the best performance at the average accuracy level of qCBR for the classic (Table (12.5) and (12.6)). Table (12.8), (12.7) and (12.2) show the degree of scalability of the qCBR as a function of the variation in the number of patients and social workers. It has also been seen that qCBR is much better shared with overlapping as we wanted to demonstrate.

One of the greatest achievements of this qCBR is the formula for calculating the number of solutions to know the class number of our quantum classifier. This formula has been key for the optimal functioning of our proposal.

$$N_{SOL_{SWP}} = \frac{1}{m!} \sum_{k=0}^{m-1} (-1)^k \binom{m}{m-k} (m-k)^n.$$

With  $n$  the number of patients and  $m$  the number of social workers and knowing that the appearance orders patients in the schedule (from earliest to latest,  $n_1$  the patient with the earliest plan and  $n_k$  the patient with the newest program).

We have designed qRobot to generalize the SWP and complement our learning of the annealing platform, in this case, D-Wave. Since to date, all our experimentation have been on quantum gate based technology (IBM and PennyLane).

In section (13.4), we can also explore here[26], the results of the qRobot as a generalization of the SWP. In this case, we can see in this reference the results to create the platform that has been used to develop the proof of concept.

Let's split the results of this proof of concept in two. 1, the configuration and conversion results of the Raspberry Pi 4 in a quantum computing environment (Fig.(13.12) to Fig.(13.15)) and 2, the picking and batching algorithm results represented by tables (13.1) to (13.3) on one side and Fig.(13.10) and (13.11) on the other.

The block diagram (Fig.(13.9)) summarizes the result of the implementation of the qRobot. The first thing we did was determine the mathematical model of our problem. We then used the Docplex to model our objective function and its constraints. For our proof of concept, we used the Docplex library packages to move from Docplex to QUBO. We had two possible operations according to our objectives from this point on. First, we modelled the problem for computers based on quantum gate technology like IBMQ, and second, for annealing computers like D-Wave. Our experiments used the Exact solver and VQE to test the Qiskit framework as samples based on quantum gates. But before using the VQE, we needed to map the QUBO model to the Ising model. Then, when we used the D-Wave computer, we only needed to reform the QUBO output list from Docplex to the QUBO format of the D-Wave computer.

Let us remember that for the qRobot, we created one universal quantum computer on a Raspberry Pi 4. The main idea is to have a mobile robot for the batching and picking problem. The steps to convert the Raspberry Pi 4 into a "quantum computer" are in the [26].

Table (13.1) shows the experimentation results by setting the number of qRobots as their capacities (maximum load) at 1 and 45, respectively. We compared the execution time of our algorithm with different public access simulators on the market during this experimentation, solving the problem of picking and batching. We observed that, for issues of this nature, especially due to the number of qubits required in each scenario, the behaviour of the D-Wave is the desired one at the temporal level, comparing it with Gate based Quantum Computing. However, it should be taken into account that, for experiments with numbers of qubits less than 20, the behaviour of these simulators is equated with

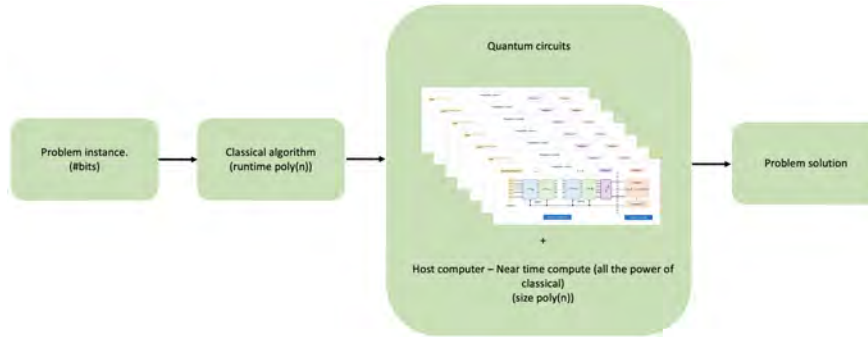


Figure 14.1: In this figure, we can observe a workflow of a quantum algorithm. First, let us remember that a quantum computer or hardware only executes a quantum circuit. But we can use classical algorithms to generate a description of our quantum circuit efficiently, so it would be smart to use it. Our quantum algorithm is born to execute or answer problems/approaches very expensive for classical computing. A good quantum algorithm would be the polynomial scale with qubits. This figure shows that we can clarify it polynomially given a problem and follow these guidelines.

the D-Wave. This experimentation helps to have a clear vision about the feasibility of this proof of concept.

Continuing with the analysis of the results, table (13.2) shows us the computational results of our picking and batching algorithm considering 1 robot through AWS-Braket and on the real quantum computer D-Wave Advantage\_system1[247]. Again, the time value is an average and does not count latency time, job creation, and job return time.

One of the outstanding results of the qRobot is to have achieved a formulation that gives us the minimum (most efficient) number of qubits to carry out the proposed task.  $K(n+1)(n+2)+K \lceil \log_2 M \rceil$ , where the  $K \lceil \log_2 M \rceil$  qubits are needed as ancillaries qubits. With  $M$  the maximum load of the qRobots and  $K$  the number of qRobots available.

We made more experiments, and for more details, we invite the reader to reach them out at (13.4).

We see it as very important to highlight the importance of figure (14.1) as our algorithm's workflow. We have concluded using classical algorithms to efficiently generate a description of our quantum circuit because our quantum algorithm is designed to execute the very expensive approaches for classical computing.

This thesis work has led to two final master's projects. One of the projects was for the computer science master and the other for the mathematics master of the University of Valladolid. The result can be seen in this reference [108]. The objective of the second project was to capture a real scenario based on the VRP problem, in which we proposed a way to solve them based on quantum models. Subsequently, we analyzed the advantages and disadvantages that quantum variational algorithms can offer us compared to the most popular algorithms in classical computing. This master was with the University of Ramon Llull.

## 14.2 Discussion

In these sections (11.4), (12.8) and (13.5), we can see the different discussions made throughout this thesis work. These discussions have motivated us to continue investigating the formulation of the SWP, its implementation, experimentation, and the various comparisons to the generalization of the SWP with the qRobot. In addition, our thesis hypothesis has motivated all these discussions (could quantum computing solve efficiently Hard Constrained Optimization Problems?).

Let us summarise the discussion of this thesis. The first part of this section will be on the validation of the SWP algorithm.

We can observe that, while Backtracking (Fig. (11.21)) and the CPLEX present an exponential behaviour as the number of patients increases, the VQE, QAOA algorithm ((11.22)), without taking into account the cost of evaluation and calibration of the algorithm, have a logarithmic growth. This, as the number of patients grows, will offer more considerable advantages than a classic algorithm, such as Backtracking, since its temporary cost will be much lower for more complex problems (11.21) and (11.22).

We did several tests and comparisons to validate our formulation and benchmark with other optimizers (quantum and classic) that are Backtracking, QAOA, OR\_Tool from google. Many comparisons were also made to fine-tune the algorithm. The results are shown in table (11.5) and figures (11.5) and (11.6).

Since we wanted to make a decent comparison, we mean like many qubit numbers, we had to change the back-end from the *statevector\_simulator* to *qasm\_simulator* and the real quantum computer. We made this change because the superposition calculations of the quantum states or complex amplitudes that the simulator provides to keep track of the algorithm overthrow the computer. It must be said that this has nothing to do with the efficiency of the quantum computer but is a tool that facilitates Qiskit to learn the steps that the algorithm follows.

The results obtained with the *qasm\_simulator* are very similar to the results of the real quantum computer, as seen in the figures (11.12) and (11.13).

We can observe (Fig. (11.19)) in this case that the Quantum exact solver gives the same result as the Backtracking. So instead, the VQE provides a solution with but with a patient change. This can be a problem if the patients will not change social workers daily. Another limitation can be the learning process of the top-down approach. Another improvement is to reduce the computational cost of the top-down algorithm.

The evaluation of the algorithm on an *ibmq\_16\_melbourne v1.0.0* from IBM was fulfilled. With any change in the input, variables are mapped proportionally to our cost variable with a time window. We would like to add that the suggested formulation (11.50) and (11.51) is not only specific to the proposed problem. It can be used to solve any family planning, scheduling and routing problem related to a list of tasks, restrictions, allocation of resources on location and time. The test performed and showed in Fig. (11.27) allows us to see the behaviour of our formulation with the variation of the correction factor  $\varepsilon$ . We understand how our time window  $T_{ij} = (\tau_i - \tau_j)$  adapts perfectly at the extremes to the cost variable in the distance. This achievement is due to the chosen quadratic function (11.51). We want it to be adapted so that our resultant function weights together the short distances and time and the long distances and late times. Other functions can be studied to have a test bench to compare the final results. QAOA, like VQE, takes a qubit operator from the Hamiltonian of the Ising model. The only mapping that gets done is when QAOA builds a variational form based on the qubit operator and what we figured was that does not change the original qubit operator input. Figure (11.25) and (11.26) reveal the comparison work between VQE and QAOA algorithms for the same configuration parameters. After several tests, we confirmed that our algorithm takes less execution time with the QAOA than the VQE and requires fewer samples for optimal solutions. But in many cases, we have to increase the seed value to get a reasonably stable result.

If the reader wants to see the graphs in detail and all the discussions, we invite him to section (11.4).

One of the first things we want to say about the qCBR is that it works very well and is a valid method to solve the SWP through artificial intelligence. However, it must be noted that it is initially required to execute the Top-Down algorithm a few times to have a set of initial cases. For more details, we encourage the reader to refer to the (12.8) section.

One of the issues to comment on is the improvement observed in (figure(12.16)) both two and eight-dimensional classifiers. Due to the small number of depths, but with many more parameters, the 8-dimensional classifiers have an average of about 25% of improvements over the 2-dimensional ones. With this result, in the case of not wanting an accuracy of around 95%, shallow depth could be used, and computation time saved, depending on the problems. Despite all these improvements, it is essential to highlight some improvement aspects. For example, in the intelligence system that allows deciding the proposed solution, now, the average of the *Initial\_point* of each solution class samples' *Initial\_point* is used. However, it could still be seen based on the predicted solution, which *Initial\_point* is the most suitable for the answer to propose. Thus, the cases to be re-used could be better classified.

Also, one of the improvements is to train the classifier with noisy data further so that the qCBR can adapt to real past situations that adjust to the new situation. Because, in practice, there is usually no past case strictly the same as a new one.

The last improvement is to generalize the qCBR to serve various problems (betting, financial, software maintenance, human reasoning, etc.). For this, it would be a success to focus on the design in memory of cases so that different data sizes can be indexed and, above all, train the classifier with several other data models.

In the qRobot's discussions (13.5), we can see that this formulation solves the order in which the robot could manage all the products and make the batches passing through the depot.

Another important achievement this approach offers is that each robot makes a single trip. However, it is possible to band the code so that if we find ourselves in a situation where there are many batches to create and only a few robots to do the picking, these robots can be made to make the necessary trips if we have  $k$  qRobots that make at most one trip (we will never need more with  $n$  batches). In this way, we will obtain all the packages for trips we are interested in doing. A more understandable way of explaining it would be that when the first qRobot has finished its journey, it should only be ordered to do the one that would have made the qRobot  $k + 1$ , which does not exist and so on with all the qRobots  $k + 2$ ,  $k + 3$ ,  $k + 4$  etc. until all scheduled batches are finished.

One of the most important points to comment on here is that our formulation minimizes the distance. But in the case of wanting to reduce the time, we should use a strategy facing the qRobots, but make some small adjustments in the formulation.

We encourage the reader to review each discussion to get a broader idea of the work developed.

One topic of discussion is *Quantum Error Correction* also known as QEC. In this thesis work, we focus on implementing our quantum algorithms that obey some transpiler of each framework. Thus, leaving us with little margin in correcting errors. A future approach would be to see how we base our algorithms on the design of all quantum circuits. So, we can control even more the issue of error controls.

The formulations proposed for both the SWP and qRobot can be seen as an answer to a class of generic problems written in the form of an objective function with some restrictions. This methodology followed during this thesis can be encapsulated as a framework proposal to solve a large part of these problems. But for this, it would be necessary to study the Lagrange multiplier and different forms of penalty function in detail, inequalities constraints. The latter will be the future lines of this work.

Also, qCBR can be seen as a framework to solve any case-based optimization problem based on a decision-making solver. I want to say that instead of the problem that we are solving here, we can use the qCBR as proposed in this thesis to solve issues like Max-Cut, Max-Clique, etc. Only the classifier should be well trained with the new data sample space.

## 14.3 Conclusions and Further Work

### 14.3.1 Conclusions

Before describing the conclusion of this thesis summarised by Figure (10.1), I would like to highlight my satisfaction with the positive results. From the hypothesis question to the SWP[29, 31, 30, 33, 178, 25, 9, 67, 10, 7, 8] and qRobot[32] proofs of concepts. The gold brush has been put in the search for temporary efficiency with the work of EVA [15].

With this PhD work, we have responded to the initial hypothesis (could quantum computing solves CSP?) carried this research work in the field where the scientific community is debating. Join the ML with quantum computing to create more powerful QML tools. We have developed several quantum algorithms for this NISQ era, hybrid algorithms that use the classical part's potential, and quantum mechanics' power to solve combinatorial optimisation problems, such as the Social Workers' Problem. We have seen that in addition to solving this problem in a Top-Down way, we have proposed a methodology based on Artificial Intelligence to solve and offer the resolution of these problems with our qCBR as human beings do.

A qRobot to generalise an SWP and as a quantum computing approach in mobile robot order picking and batching problem solver optimisation is proposed and tested.

Let us dive deeply into the conclusion of this work. First, we recall that this thesis aims to study the state of the art and design of a series of quantum algorithms on combinatorial optimisation problems. To do this, we define and create a combinatorial optimisation problem with hard restrictions and solve it in two different approaches. In a Top-Down approach, we formulate the problem mathematically, both classical and quantum. Our proposed formulation allows us to design a strategy to take advantage of this NISQ era (Few useful qubits, decoherence, limitations in quadratic techniques with inequality constraints, etc.). Although it seems specific, Said formulation is later generalised to encompass more generic applications. We also made a comparative analysis of 5 different implementations and discussed them during this thesis work. Continuing under the Top-Down approach, we have thoroughly analysed the Ising model and the QUBO model to solve quadratic problems. We have also studied Cplex optimisation tools such as IBM's docplex under their Qiskit framework. The Qiskit optimisation tool included the generic quadratic programs that help to model any optimisation problems. Many converters are also involved in mapping the problem to solve it in the correct input format. Converters like *Inequality to equality* to map inequality constraints to equality constraints with additional slack variables. The *Integer To Binary* converter is useful in the case of the need to convert integer variables into binary variables and the corresponding coefficients. *Linear Equality to Penalty* helps to convert equality constraints into additional terms of the SWP's objective function. Finally, the *Quadratic QUBO* converter is used as a container for all the above converters. We have thoroughly analysed the Minimum Eigen Optimizer and the ADMM optimiser as alternatives to VQE and QAOA. We have also developed some guidelines to make it easier for prospective PhD or graduate students to take their first firm steps in quantum computing.

On the other hand, we have statistically solved the problem with the qCBR algorithm machine learning technique.

To do this, we have developed the classical and quantum CBR and observed the outstanding performance of qCBR compared to its classical counterpart in precision, scalability and average tolerance to an overlapping dataset. We've mitigated some of the standard and classic CBR issues. With the design that has been proposed, it has been possible to measure situations of difficult similarity between cases. Despite the non-linear and overlapping attributes, the classifier has been endowed with characteristics that arrive at two similar topics that may seem quite different because they have different values in the features but are not very significant. For example, in the VQE with *Initial\_point*, we can have different *Initial\_point* associated with each training class sample with the same class. With the "*Initial\_point*" averaging technique, it is possible to solve this problem by providing the qCBR

to distinguish the similarity between cases. Another problem that qCBR mainly solves is the time required to classify a new topic.

With the results of the two implementations (classical and quantum CBR), it is observed that the classical CBR designed with the KNN behaves better for some determined cases (table (12.8)). It is seen that the system has not finished learning thoroughly (table (12.5) and (12.6)) contrary to qCBR (table (12.3) and (12.4)). This is due to the precision of its classifier, without forgetting the important contribution of its synthesis system.

Another improvement that qCBR introduces is when it comes to retaining cases, implementing a retention system that maintains case models and that, together, synthesise the real and most important information. This would not be possible if a variational quantum classifier were not designed.

Following the thread of this thesis work, we generalise our formulation of the SWP to carry out efficient management of robots by substituting robots for social workers and pick-up requests for patients. We call this generalisation qRobot.

The problem raised throughout this work offers us an efficient way to manage a series of  $K$  qRobots (instead of social workers) to collect a set of orders, optimising the number of robots used. The approach provided is applied to a "central computer" capable of doing all the calculations and then giving the orders to each qRobot. However, when we start to deal with very big problems both in the number of products and the number of robots, the number of required qubits will tend to grow too large. One possible solution is to distribute the calculation from a central computer to each robot so that each has to calculate its route given a list of products to collect. In this case, the equations in the problem would not change, just take  $K = 1$  for each qRobot and apply the technique mentioned at the beginning of the discussion. Although it may not be possible to arrive at the best solutions, this process of distributing the calculation would mean a significant reduction in computational cost despite the need to create batches in advance. This search for batching will be explored in future projects. On the other hand, it is important to point out that the problem dealt with has a QUBO-type formulation, which allows its execution in annealing-type quantum computers. This makes a huge difference in the current era (NISQ), as we have managed to work with 200 qubits versus the 30 qubits we would have with a gate-based quantum computer. Finally, note that the defined problem minimises the total distance the robots travel, making it worthwhile for not all robots to exit. For future lines, we will tackle the same problem. Still, we will continue to try to reduce total times instead of distance travelled (as was done in this previous job [30]) as this situation is also very important in warehouse logistics.

We can also conclude that the scenario designed to answer our thesis question is adequate. It is true that, with quantum computing, we can efficiently solve combinatorial optimisation problems with hard constraints and the qRobot, as a generalisation of the SWP, as the Top-Down resolution approach, and the qCBR as the resolution approach based on quantum machine learning has been validated by the quantum scientific community.

To finish this thesis work, I would like to highlight part of my pedagogical contribution to my research group (Data Science 4 Digital Society) and the quantum community as part of the objective of this thesis. Also, how could you monetise this thesis work. Also, I would like to highlight all that I have learned since one should not end any trip without knowing from it. Thank you, Elisabet Golobardes i Ribé! Thank you, Xavier Vilasis Cardona!

### 14.3.2 Further Work

From a general point of view, we will continue investigating optimisation problems and creating quantum algorithms to benefit society and our research group for the future line. Unfortunately, error correction is expensive at the level of the number of qubits. So, there is a very large gap between what is needed and what is currently available to build a useful quantum computer. One line of the future that we set ourselves is to investigate more about Quantum Error Correction (QEC) and find

more efficient codes.

Before diving into details into each part of our contribution in this work, one line of the future that we find interesting is to develop all our libraries and tutorials for the PennyLane, Qiskit, Qibo and AWS-Braket platforms and frameworks in the form of a license.

One of our future lines for the SWP is to take advantage of the improvements in the number of qubits of quantum computers to validate, based on more tests, the computational complexity that we achieved. This validation should confirm if we have achieved exponential improvement comparable to the tests done with Backtracking shown from figure (11.22) and (11.21).

One of the qCBB future lines is when retaining cases, implementing a retention system that maintains model cases and that, together, synthesise the real and most important information. One of the improvements to consider is the implementation of quantum ICA. In this way, the classical ICA analysis's complexity cost will be significantly reduced. Also, counting that the PCA is saved since we have an 8-dimensional classifier, the complexity of the qCBB would be that of the classifier plus some setup constants. The other exciting line of the future is to design the memory of cases using the quantum technique of random-access memory (qRAM) [50] to improve the memory of stored instances.

One of the future lines for the qRobot is to adapt the formulation of the qRobot for optimisation of time, not distance. This means minimising the time it takes for the robots to collect all the orders or, what is equivalent, minimising the maximum of the lengths that each robot travels. Another line of the future is to develop an algorithm *Quantum Annealing*, focusing on exposing the mathematics that helps to understand how to perform good modelling. Another line of the future is to try to apply all the improvements achieved in this work to the IBMQ *Docplex* tool [120] as a library.

A clear future line is to design a framework to encapsulate the methodology we defined here to solve optimisation problems. But for this, it would be necessary to study the Lagrange multiplier and different forms of penalty function in detail, inequalities constraints.

### 14.3.3 International contribution and talent pool

This section will detail our contributions to journals, conferences, and publications of the scientific community and the group of talents we met to develop this project.

In the SWP, we have contributed with the formulation ((11.2.1), (11.2.3) and its first intent of generalisation (11.2.10)) of the SWP mathematically, both classically and quantum, and made a comparative study on the IBMQ quantum computer. In this paper, we have validated our formulation, both for QUBO and Ising in the scientific community:

- ◇ *Atchade-Adelomou P., Golobardes Ribé E., Vilasis Cardona X. Formulation of the Social Workers' Problem in quadratic unconstrained binary optimisation form and solve it on a quantum computer. Journal of Computer and Communications. 2020 Nov 5. DOI: 10.4236/jcc.2020.811004*

**Abstract** The problem of social workers visiting their patients at home is a class of combinatorial optimisation problems and belongs to the type of problems known as NP-Hard. These problems require heuristic techniques to provide an efficient solution in the best of cases. In this article, in addition to providing a detailed resolution of the Social Workers' Problem using the Quadratic Unconstrained Binary Optimization Problems (QUBO) formulation, an approach to mapping the inequality constraints in the QUBO form is given. Finally, we map it in the Hamiltonian of the Ising model to solve it with the Quantum Exact Solver and Variational Quantum Eigensolvers (VQE). The quantum feasibility of the algorithm will be tested on IBMQ computers.

This paper uses the VQE to create a solution based on the machine learning technique. In addition, we have participated in the international conference on Hybrid Artificial Intelligent Systems. Our most relevant contributions in this work have been put in the following paper.

- ◇ *Atchade-Adelomou P., Golobardes Ribé E., Vilasís Cardona X. (2020) Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver. In: de la Cal E.A., Villar Flecha J.R., Quintián H., Corchado E. (eds) Hybrid Artificial Intelligent Systems. HAIS 2020. Lecture Notes in Computer Science, vol 12344. Springer, Cham. [https://doi.org/10.1007/978-3-030-61705-9\\_21](https://doi.org/10.1007/978-3-030-61705-9_21)*

**Abstract** The scheduling problem of social workers is a class of combinatorial optimisation problems that can be solved in exponential time at best. Because it belongs to a class of the issues known as NP-Hard, which have a huge impact on our society, nowadays, the focus on the quantum computer should no longer be just for its enormous computing capacity but also for the use of its imperfection (Noisy Intermediate-Scale Quantum (NISQ) era) to create a powerful machine learning device that uses the variational principle to solve the optimisation problem by reducing their complexity's class. We propose a formulation of the Vehicle Routing Problem (VRP) with time windows to efficiently solve the social workers' schedule problem using Variational Quantum Eigensolver (VQE). The quantum feasibility of the algorithm will be modelled with docplex and tested on IBMQ computers.

In this work, we have worked with other techniques based on parameterised circuits to create an Intelligent social workers' schedule problem solver. The following paper has put our most relevant contributions to this work.

- ◇ *Atchade-Adelomou P., Golobardes Ribé E., Vilasís Cardona X. Using the Parameterized Quantum Circuit combined with Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers' schedule problem solver. arXiv preprint arXiv:2010.05863. 2020 Oct 12.*

**Abstract** The social worker scheduling problem is a class of combinatorial optimisation problems that combines scheduling with routing issues. These types of problems with classical computing can only be solved, in the best of cases, in an approximate way and significantly when the input data does not grow considerably. Today, the focus on the quantum computer should no longer be only on its enormous computing power but also on the use of its imperfection for this era (Noisy Intermediate-Scale Quantum (NISQ)) to create a powerful optimisation and learning device that uses variational techniques. We had already proposed a formulation and solution of this problem using the capacity of the quantum computer. In this article, we present some broad results of the experimentation techniques. And above all, we propose an adaptive and intelligent solution that efficiently recalculates the schedules of social workers—taking into account new restrictions and changes in the initial conditions, using a case-based reasoning system and the variational quantum eigensolver based on a finite-depth quantum circuit. That encodes the ground state of the Hamiltonian of social workers. The quantum feasibility of the algorithm will be modelled with docplex and tested on IBMQ computers.

In the case of qCBR's contributions, we have created a dataset (from the SWP formulation) with an overfitting and overlapping problem creating a quantum classifier based on the universality theorem. We had to perform a very efficient quantum multiclass classifier and synthesiser to do this. We have also developed a formula for calculating the number of classes of the multiclass classifier. As a sum of our contributions, we have proposed a quantum CBR (qCBR) to solve optimisation problems as we usually solve them as human beings. The most relevant contributions have been put in the following paper. Nevertheless, we can find details of the qCBR at reference (12.6).

- ◇ *Atchade-Adelomou P, Casado-Fauli D, Golobardes Ribé E, Vilasís Cardona X. quantum Case-Based Reasoning (qCBR). arXiv preprint arXiv:2104.00409. 2021 Apr 1.*

**Abstract** Case-Based Reasoning (CBR) is an artificial intelligence approach to problem-solving with a good record of success. This article proposes using Quantum Computing to improve some of the key processes of CBR defining so a Quantum Case-Based Reasoning (qCBR) paradigm. The focus is on designing and implementing a qCBR based on the variational principle that improves its classical



counterpart in terms of average accuracy, scalability and tolerance to overlapping. A comparative study of the proposed qCBR with a classic CBR is performed for the case of the Social Workers' Problem as a sample of a combinatorial optimisation problem with overlapping. The algorithm's quantum feasibility is modelled with docplex, tested on IBMQ computers, and experimented with the Qibo framework.

In the case of qRobot, we have contributed both to the creation of a "universal" platform to create a quantum computer in a raspberry PI 4 and, on the other hand, to create an efficient formulation for this quantum era that generalises the SWP and solves the problem of batching and picking within from a warehouse. We are pleased to say that the paper, in addition to being published in MDPI magazine, Algorithms was also on the cover. It is worth saying that these improvements could lead to improvements over TSP and VRP. Our most relevant contributions over the qRobot have been put in the following paper. Nevertheless, we can find details of our formulation at reference (13.3.1) and we can also find how we have turned the Raspberry PI 4 into a universal computer.

The description step by step into how installing and running PennyLane, AWS-Braket, D-Wave-Ocean, Qiskit, on a Raspberry Pi 4 under the ARM64[129] operating system turn it into a quantum computing simulator and use it to access real quantum computers from IBMQ[241, 164], AWS-Braket[57], D-Wave[77], and Regetti[215]. These frameworks and packages are required for the proof of concept that we propose [26].

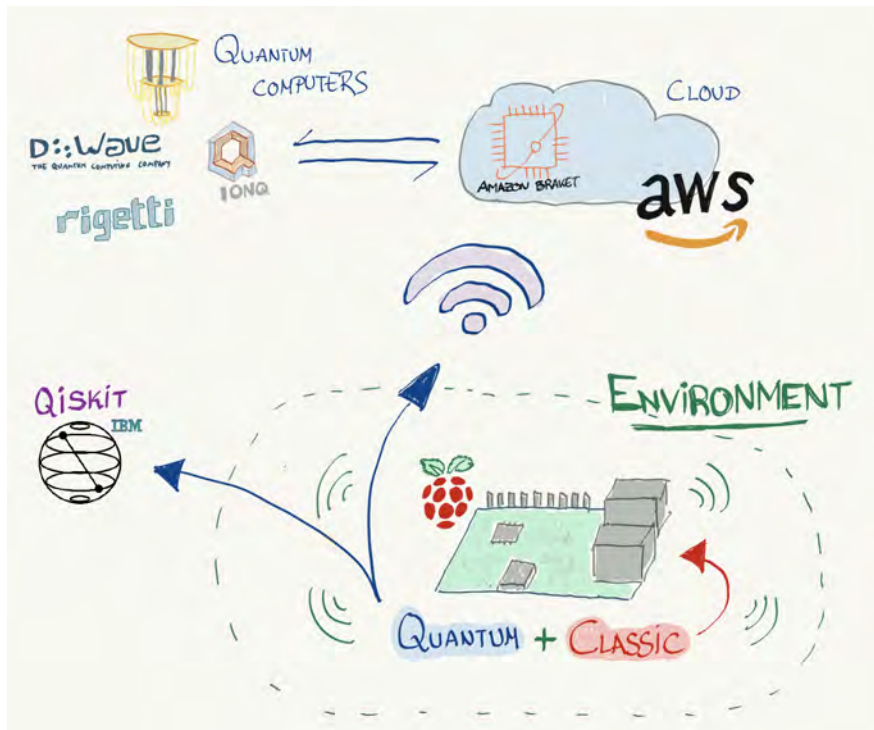


Figure 14.2: We have installed the following frameworks successfully (Qiskit, PennyLane, AWS-Braket) on our Raspberry Pi 4 under the ARM64 operating system. More information about the qRobot platform can be found at Ref. [26]

◇ *Atchade-Adelomou, Parfait, Guillermo Alonso-Linaje, Jordi Albo-Canals, and Daniel Casado-Fauli. 2021. "qRobot: A Quantum Computing Approach in Mobile Robot Order Picking and*

**Abstract** This article aims to bring quantum computing to robotics. A quantum algorithm is developed to minimise the distance travelled in warehouses and distribution centres where order picking is applied. For this, a proof of concept is proposed through a Raspberry Pi 4, generating a quantum combinatorial optimisation algorithm that saves the distance travelled and the batch of orders to be made. In case of computational need, the robot will be able to parallelise part of the operations in hybrid computing (quantum + classical), accessing CPUs and QPUs distributed in a public or private cloud. Before this, we must develop a stable environment (ARM64) inside the robot (Raspberry) to run gradient operations and other quantum algorithms on IBMQ, Amazon Braket, D’wave and PennyLane locally or remotely. The proof of concept will run in such quantum environments above.

In the case of QFilter, we have contributed to quantum computing by proposing a method to, on the one hand, bring quantum computing closer to the community of classical computing (machine learning) and, on the other, a new approach to solving convolutional networks (CNN) by changing only the classical scalar product by a quantum and keeping the rest of the classical code. This approach and contribution have made us a finalist of about 2000 participants of the Xanadu Hackathon (with the AWS-Braket contest, Google). The following paper has put our most relevant contributions over the QFilter.

◇ *Atchade-Adelomou P., Alonso-Linaje G. Quantum Enhanced Filter: QFilter. arXiv preprint arXiv:2104.03418. 2021 Apr 7.*

**Abstract** Convolutional Neural Networks (CNN) are used mainly to treat problems with many images characteristic of Deep Learning. This work proposes a hybrid image classification model to take advantage of quantum and classical computing. The method will use the potential that convolutional networks have shown in artificial intelligence by replacing classical filters with variational quantum filters. Similarly, this work will compare other classification methods and the system’s execution on different servers. Finally, the algorithm’s quantum feasibility is modelled and tested on Amazon Braket Notebook instances and experimented on PennyLane’s philosophy and framework.

In the case of EVA, we have contributed to quantum computing by proposing a new method to calculate the quantum expected value based on a single quantum circuit, compared to the VQE technique that divides a given Hamiltonian into several small ones of very shallow depth. The importance of this contribution is enormous in this quantum age, but it will be even more so for the future since we only make one call to the quantum computer. Also, it is worth saying that EVA is intended for chemical applications. With EVA, we could efficiently calculate the energy of some molecules to facilitate the discovery of new drugs. Our most relevant contributions over EVA have been put in the following paper.

◇ *Alonso-Linaje G, Atchade-Adelomou P. EVA: a quantum Exponential Value Approximation algorithm. arXiv preprint arXiv:2106.08731. 2021 Jun 16.*

**Abstract** VQE is currently one of the most widely used algorithms for optimising quantum computers’ problems. However, a necessary step in this algorithm is calculating the expectation value given a state, which is calculated by decomposing the Hamiltonian into Pauli operators and obtaining this value for each of them. We have designed an algorithm capable of figuring this value using a single circuit in this work. In addition, a time cost study has been carried out, and it has been found that in certain more complex Hamiltonians, it is possible to obtain a good performance over the current methods.

### 14.3.4 Schedule/Work plan

In this session, we will observe a series of tasks and time management that summarise this thesis work. Like any project, this had needed a work plan, specific planning even though part of its tasks could not be limited due to the subject's demand and, above all, because of the little information found when we started this journey in the centre of the quantum computing world. Table (14.1) shows this PhD task development. Also, we can observe below the detail of the tasks that have been carried out.

Table 14.1: PhD Tasks development

month	0	3	6	9	12	15	18	21	24	27	30	33	36
T1	█	█	█										
T2				█	█								
T3						█							
T4							█						
T5							█						
T6							█						
T7								█	█				
T8										█	█		
T9									█	█	█	█	
T10							█	█	█	█			
T11							█	█	█	█	█	█	
T12							█	█	█	█	█	█	

- ◇ **Task 1:** This task took place during the first six months of the work. Studying the Strengths, Weaknesses, Opportunities, and Threats (SWOT) was crucial for developing this work.
- ◇ **Task 2:** This task took place during the first six months of the work. This task began with a solid study of the existing CSP algorithms and models.
- ◇ **Task 3:** This task took place during the first three months of the work. Study for better comprehension of the linear, quadratic programming solver's functioning.
- ◇ **Task 4:** This task took place during the first one months of the work. Study TSP, JSP and VRP. State of the art of Heuristic.
- ◇ **Task 5:** This task took place during the first one months of the work. State of the art of Systems and Hamiltonians concepts, Ising Hamiltonian Model, The Hamiltonian of a TSP, The Hamiltonian of a JSP and VRP.
- ◇ **Task 6:** This task took place during the first one months of the work. State of the art of the complexity class and Quantum Complexity class.
- ◇ **Task 7:** This task took place during the first six months of the work. Design and Model, mathematically the Social Workers' Problem (SWP) as CSP according to state of the art and the ones proposed in the scope of the PhD work. Study the players involved in SWP, namely quantum computing frameworks, quantum computers, and quantum research groups. Third, study and Map the SWP in quantum.
- ◇ **Task 8:** This task took place during the first six months of the work. First, developing and implementing the SWP in this NISQ era. Solve the SWP with QML. This methodology will be based on clustering and classification techniques, resulting in a rule base concerning the SWP. An extensive set of simulation results will be used as the basis for applying the proposed QML

method. Third, designing and implementing a quantum Case-Based Reasoning (qCBR) and testing and validating the developed models and approaches.

- ◇ **Task 9:** Writing at least two papers for top-level conferences and six articles for SCI journals.
- ◇ **Task 10:** Development and implementation of the QFilter.
- ◇ **Task 11:** Development and implementation of the qRobot Platform and the picking and batching formulation.
- ◇ **Task 12:** Development and implementation EVA.

### 14.3.5 Pedagogical contribution

As a PhD work, one of the most important parts besides validating the hypothesis, writing the contributions in scientific articles, is self-explanatory and can be reproduced; our job is self-consistent. That is why we have made our contributions academic and catalysts to attract new talent in collaborations from our group and team.

This section will highlight part of my pedagogical contribution to my research group.

- ◇ Some guidelines and steps to solve combinatorial optimization problems in quantum.
- ◇ A documentation of the state of the art and compilation and access to the main tutorials of quantum frameworks
- ◇ a GitHub with the codes and experimentation environments
- ◇ Complete formulation of a real combinatorial optimization problem (SWP) with restrictions and mapping it in quantum.
- ◇ Step-by-step development of how to bring a classical to a quantum optimization problem (SWP)
- ◇ Design, implementation and results of the SWP Top-down resolution and quantum Case-Based Reasoning
- ◇ Design, implementation and comparisons of a variational quantum classifier.
- ◇ Design, implementation of a universal quantum platform using Raspberry PI.
- ◇ Design, implementation and comparisons of picking and batching problem.
- ◇ co-accompany a group of collaborators in quantum computing for DS4DS.
- ◇ Co-direction of a TFGs in quantum computing.

### 14.3.6 Contribution to business vision

Today, we can already see a lot of movements in quantum computing companies. Several companies are already using this technology to offer consulting tasks at the optimization level or training at the introductory level. Our work goes a little further. However, the group could particularize one of my formulations as a Fintech product and base it on the cloud. Developing a subscription/use business model; Pay peruse.

### 14.3.7 Lessons learned (Hard Skills)

Before ending this session, I would like to leave the evidence of what I learned at the cognitive level and, above all, to record what I have acquired at the skill level. I feel very fortunate to write what I have learned in this section.

- ◇ Quantum mechanics applied to quantum computing
- ◇ Qiskit, Qibo, PennyLane, AWS-Braket, D-Wave, etc. programming environments
- ◇ Mathematical formulations of optimization problems
- ◇ Mapping of classical combinatorial optimization problems in quantum
- ◇ Design and implement classical and quantum classifiers
- ◇ Understand quantum Machine Learning.
- ◇ Write articles
- ◇ Scientific rigour
- ◇ Quantum computing.
- ◇ The era in which quantum computing is found.

### 14.3.8 Consolidated skills (Soft Skills)

Quantum computing is not one of the most intuitive concepts and one of the easy ones, but behind its complexity, I did during this 3-year journey to help me develop and consolidate the soft skills that I list in this section.

- ◇ Self-motivation
- ◇ Overcome the obstacles
- ◇ Find my life

# Bibliography

- [1] Raspberry .org. *Configuration of the Raspberry Pi*. 2021. URL: <https://www.raspberrypi.org/documentation/remote-access/ssh/passwordless.md>.
- [2] Salama A. Mostafa, Mohd Sharifuddin Ahmad, and M. A. Firdaus M. “A Soft Computing Modeling to Case-based Reasoning Implementation”. In: *International Journal of Computer Applications* 47.7 (June 2012), pp. 14–21. DOI: 10.5120/7199-9976. URL: <https://doi.org/10.5120/7199-9976>.
- [3] Agnar Aamodt and Enric Plaza. “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches”. In: *Artificial Intelligence Communications* 7 (1994), pp. 39–59.
- [4] Scott Aaronson. “The limits of quantum”. In: *Scientific American* 298.3 (2008), pp. 62–69.
- [5] Abdiansah Abdiansah and Retantyo Wardoyo. “Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM”. In: *International Journal of Computer Applications* 128.3 (Oct. 2015), pp. 28–34. DOI: 10.5120/ijca2015906480. URL: <https://doi.org/10.5120/ijca2015906480>.
- [6] Qingfeng Wang Tauqir Abdullah. “An Introduction to Quantum Optimization Approximation Algorithm”. In: *arXiv* (2018).
- [7] Atchade Parfait Adelomou, Elisabet Golobardes Ribe, and Xavier Vilasis Cardona. *Using the Parameterized Quantum Circuit combined with Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers’ schedule problem solver*. 2020. arXiv: 2010.05863 [quant-ph].
- [8] Atchade Parfait Adelomou, Elisabet Golobardes Ribé, and Xavier Vilasis Cardona. “Formulation of the Social Workers’ Problem in Quadratic Unconstrained Binary Optimization Form and Solve It on a Quantum Computer”. In: *Journal of Computer and Communications* 08.11 (2020), pp. 44–68. DOI: 10.4236/jcc.2020.811004. URL: <https://doi.org/10.4236/jcc.2020.811004>.
- [9] Parfait Atchade Adelomou. “Quantum Social Workers’ Problem all the code”. In: (Oct. 2021). URL: <https://github.com/pifparfait/QSWP>.
- [10] Parfait Atchade Adelomou, Elisabet Golobardes Ribé, and Xavier Vilasis Cardona. “Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 245–260. DOI: 10.1007/978-3-030-61705-9\_21. URL: [https://doi.org/10.1007/978-3-030-61705-9\\_21](https://doi.org/10.1007/978-3-030-61705-9_21).
- [11] Simone Agostinelli, Andrea Marrella, and Massimo Mecella. “Towards intelligent robotic process automation for BPMers”. In: *arXiv preprint arXiv:2001.00804* (2020).

- [12] Maria Albareda-Sambola et al. “The multi-period incremental service facility location problem”. In: *Computers & Operations Research* 36.5 (2009), pp. 1356–1375.
- [13] Fowler Alex. “Improve QUBO formulation for D-Wave Quantum Computing”. In: *Research Gate* (2017).
- [14] “Alibaba Cloud Quantum Development Platform: Large-Scale Classical Simulation of Quantum Circuits”. In: (2019).
- [15] Guillermo Alonso-Linaje and Parfait Atchade-Adelomou. *EVA: a quantum Exponential Value Approximation algorithm*. June 2021. arXiv: 2106.08731 [quant-ph].
- [16] Guillermo Alonso-Linaje and Parfait Atchade-Adelomou. *EVA: a quantum Exponential Value Approximation algorithm*. June 2021. URL: <https://github.com/pifparfait/EVA>.
- [17] M. V. Altaisky, N. E. Kaputkina, and V. A. Krylov. “Quantum neural networks: Current status and prospects for development”. In: *Physics of Particles and Nuclei* 45.6 (Nov. 2014), pp. 1013–1032. DOI: 10.1134/s1063779614060033. URL: <https://doi.org/10.1134/s1063779614060033>.
- [18] Unai Alvarez-Rodriguez et al. “Quantum artificial life in an IBM quantum computer”. In: *Scientific reports* 8.1 (2018), pp. 1–9.
- [19] Khaled Amaïef and Jie Lu. “Ontology-supported case-based reasoning approach for intelligent m-Government emergency response services”. In: *Decision Support Systems* 55.1 (Apr. 2013), pp. 79–97. DOI: 10.1016/j.dss.2012.12.034. URL: <https://doi.org/10.1016/j.dss.2012.12.034>.
- [20] Andris Ambainis. “Quantum walk algorithm for element distinctness”. In: *SIAM Journal on Computing* 37.1 (2007), pp. 210–239.
- [21] G.E. Anaya Fuentes et al. “Solución al Problema de Secuenciación de Trabajos mediante el Problema del Agente Viajero”. In: *Revista Iberoamericana de Automática e Informática Industrial RIAI* 13.4 (2016), pp. 430–437. ISSN: 1697-7912. DOI: <https://doi.org/10.1016/j.riai.2016.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S1697791216300280>.
- [22] Andra Angeleanu. “New technology trends and their transformative impact on logistics and supply chain processes”. In: *International Journal of Economic Practices and Theories* 5.5 (2015), pp. 413–419.
- [23] Eva Armengol, Albert Palaudaries, and Enric Plaza. “Individual prognosis of diabetes long-term risks: A CBR approach”. In: *Methods of Information in Medicine-Methodik der Information in der Medizin* 40.1 (2001), pp. 46–51.
- [24] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510. DOI: 10.1038/s41586-019-1666-5. URL: <https://doi.org/10.1038/s41586-019-1666-5>.
- [25] Parfait Atchade. “Quantum Social Workers’ Problem code v1”. In: (Oct. 2020). URL: [https://github.com/pifparfait/QSWP/blob/master/Quantum\\_Social\\_Workers\\_Problem\\_v1.0.ipynb](https://github.com/pifparfait/QSWP/blob/master/Quantum_Social_Workers_Problem_v1.0.ipynb).
- [26] Parfait Atchade-Adelomou. *qRobot Platform for picking and batching problem*. May 2021. URL: [https://github.com/pifparfait/qRobot\\_Platform/blob/main/RaspberryPi\\_ARM64\\_for\\_QML.ipynb](https://github.com/pifparfait/qRobot_Platform/blob/main/RaspberryPi_ARM64_for_QML.ipynb).
- [27] Parfait Atchade-Adelomou. *quantum Case Based Reasoning*. 2021. URL: <https://github.com/pifparfait/qCBR>.
- [28] Parfait Atchade-Adelomou and Guillermo Alonso-Linaje. *Quantum Enhanced Filter: QFilter*. 2021. arXiv: 2104.03418 [quant-ph].

- [29] Parfait Atchade-Adelomou, Elisabet Golobardes-Ribe, and Xavier Vilasis-Cardona. “Using the Parameterized Quantum Circuit combined with Variational-Quantum-Eigensolver (VQE) to create an Intelligent social workers’ schedule problem solver”. In: *Arxiv* (2020). arXiv: 2010.05863 [quant-ph].
- [30] Parfait Atchade-Adelomou, Elisabet Golobardes-Ribé, and Xavier Vilasis-Cardona. “Formulation of the Social Workers’ Problem in Quadratic Unconstrained Binary Optimization Form and Solve It on a Quantum Computer”. In: *Journal of Computer and Communications* 8.11 (2020), pp. 44–68.
- [31] Parfait Atchade-Adelomou, Elisabet Golobardes-Ribé, and Xavier Vilasis-cardona. “Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver”. In: *International Conference on Hybrid Artificial Intelligence Systems*. 2020, pp. 245–260.
- [32] Parfait Atchade-Adelomou et al. “qRobot: A Quantum Computing Approach in Mobile Robot Order Picking and Batching Problem Solver Optimization”. In: *Algorithms* 14.7 (2021). ISSN: 1999-4893. DOI: 10.3390/a14070194. URL: <https://www.mdpi.com/1999-4893/14/7/194>.
- [33] Parfait Atchade-Adelomou et al. *quantum Case-Based Reasoning (qCBR)*. 2021. arXiv: 2104.00409 [cs.AI].
- [34] Amir Hossein Azadnia et al. “Order batching in warehouses by minimizing total tardiness: a hybrid approach of weighted association rule mining and genetic algorithms”. In: *The Scientific World Journal* 2013 (2013).
- [35] Bhowmik B. “Dynamic Programming—Its Principles, Applications, Strengths, and Limitations”. In: *International Journal of Engineering Science and Technology* 2 (9 2010), pp. 4822–4826.
- [36] Ryan Babbush, Bryan O’Gorman, and Alán Aspuru-Guzik. “Resource efficient gadgets for compiling adiabatic quantum optimization problems”. In: *Annalen der Physik* 525.10-11 (2013), pp. 877–888.
- [37] Claudio Baccigalupo and Enric Plaza. “Case-Based Sequential Ordering of Songs for Playlist Recommendation”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 286–300. DOI: 10.1007/11805816\_22. URL: [https://doi.org/10.1007/11805816\\_22](https://doi.org/10.1007/11805816_22).
- [38] Vijay Balasubramanian et al. “Quantum complexity of time evolution with chaotic Hamiltonians”. In: *Journal of High Energy Physics* 2020.1 (2020), pp. 1–44.
- [39] Panagiotis Kl. Barkoutsos et al. “Improving Variational Quantum Optimization using CVaR”. In: *Quantum* 4 (Apr. 2020), p. 256. ISSN: 2521-327X. DOI: 10.22331/q-2020-04-20-256. URL: <http://dx.doi.org/10.22331/q-2020-04-20-256>.
- [40] Kerstin Beer et al. “Training deep quantum neural networks”. In: *Nature Communications* 11.1 (Feb. 2020). DOI: 10.1038/s41467-020-14454-2. URL: <https://doi.org/10.1038/s41467-020-14454-2>.
- [41] R. Bellman. “Dynamic Programming”. In: *Science* 153.3731 (July 1966), pp. 34–37. DOI: 10.1126/science.153.3731.34. URL: <https://doi.org/10.1126/science.153.3731.34>.
- [42] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2020. arXiv: 1811.04968 [quant-ph].
- [43] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2020. arXiv: 1811.04968 [quant-ph].
- [44] Ville Bergholm et al. “PennyLane: Automatic differentiation of hybrid quantum-classical computations”. In: *arXiv preprint arXiv:1811.04968* (2018).



- [45] Ville Bergholm et al. “PennyLane: Automatic differentiation of hybrid quantumclassical computations”. In: *arXiv* (2020).
- [46] KOEN Bertels et al. “Quantum computer architecture: Towards full-stack quantum accelerators”. In: *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1–6.
- [47] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (2017), pp. 195–202.
- [48] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (Sept. 2017), pp. 195–202. ISSN: 1476-4687. DOI: 10.1038/nature23474. URL: <http://dx.doi.org/10.1038/nature23474>.
- [49] JD Biamonte. “Nonperturbative k-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins”. In: *Physical Review A* 77.5 (2008), p. 052331.
- [50] Miles Blencowe. “Quantum RAM”. In: *Nature* 468.7320 (Nov. 2010), pp. 44–45. DOI: 10.1038/468044a. URL: <https://doi.org/10.1038/468044a>.
- [51] Christian Blik1ú, Pierre Bonami, and Andrea Lodi. “Solving mixed-integer quadratic programming problems with IBM-CPLEX: a progress report”. In: *Proceedings of the twenty-sixth RAMP symposium*. 2014, pp. 16–17.
- [52] Eva Borbely. *Grover search algorithm*. 2007. arXiv: 0705.4171 [cs.DS].
- [53] Hammer Endre Boros and Peter L. “Discrete Applied Mathematics”. In: *Science Direct* Volume 123, 15 (Issues 1–3 Nov. 2002), pp. 155–225.
- [54] J Eli Bourassa et al. “Blueprint for a scalable photonic fault-tolerant quantum computer”. In: *Quantum* 5 (2021), p. 392.
- [55] Wojciech Bożejko, Jarosław Pempera, and Czesław Smutnicki. “Parallel Simulated Annealing for the Job Shop Scheduling Problem”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 631–640. DOI: 10.1007/978-3-642-01970-8\_62. URL: [https://doi.org/10.1007/978-3-642-01970-8\\_62](https://doi.org/10.1007/978-3-642-01970-8_62).
- [56] Amazon Braket. *Amazon Braket Services*. [https://aws.amazon.com/braket/?nc1=h\\_ls](https://aws.amazon.com/braket/?nc1=h_ls). 2021 (accessed May 5, 2021).
- [57] Amazon Braket. *github Amazon Braket*. <https://github.com/aws/amazon-braket-sdk-python>. 2021 (accessed February 26, 2021).
- [58] Adam R Brown and Leonard Susskind. “Second law of quantum complexity”. In: *Physical Review D* 97.8 (2018), p. 086015.
- [59] Michael W Browne. “Cross-Validation Methods”. In: *Journal of Mathematical Psychology* 44.1 (Mar. 2000), pp. 108–132. DOI: 10.1006/jmps.1999.1279. URL: <https://doi.org/10.1006/jmps.1999.1279>.
- [60] Jan Carl Budich, Stefan Walter, and Björn Trauzettel. “Failure of protection of Majorana based qubits against decoherence”. In: *Physical Review B* 85.12 (Mar. 2012). ISSN: 1550-235X. DOI: 10.1103/physrevb.85.121405. URL: <http://dx.doi.org/10.1103/PhysRevB.85.121405>.
- [61] Ciro Fernando Bustillo-Lecompte and Mehrab Mehrvar. “Slaughterhouse wastewater characteristics, treatment, and management in the meat processing industry: A review on trends and advances”. In: *Journal of environmental management* 161 (2015), pp. 287–302.
- [62] Richard H. Byrd et al. “A Limited Memory Algorithm for Bound Constrained Optimization”. In: *SIAM Journal on Scientific Computing* 16.5 (Sept. 1995), pp. 1190–1208. DOI: 10.1137/0916069. URL: <https://doi.org/10.1137/0916069>.
- [63] Spall James C. *SPSA COMMUNITY*. Johns Hopkins University, Jan. 2021. URL: <https://www.jhuapl.edu/SPSA/>.

- [64] Benioff Peter Cabauy and Paul. “Cyclic Networks of Quantum Gates”. In: (2020).
- [65] Yu-Dong Cai and Kuo-Chen Chou. “Nearest neighbour algorithm for predicting protein sub-cellular location by combining functional domain composition and pseudo-amino acid composition”. In: *Biochemical and biophysical research communications* 305.2 (2003), pp. 407–411.
- [66] A Robert Calderbank et al. “Quantum error correction and orthogonal geometry”. In: *Physical Review Letters* 78.3 (1997), p. 405.
- [67] Xavier Vilasis Cardona. “Using the Variational-Quantum-Eigensolver (VQE) to Create an Intelligent Social Workers Schedule Problem Solver”. In: *Hybrid Artificial Intelligent Systems: 15th International Conference, HAIS 2020, Gijón, Spain, November 11-13, 2020, Proceedings*. Vol. 12344. Springer Nature. 2020, p. 245.
- [68] Çağla Cergibozan and A Serdar Tasan. “Order batching operations: an overview of classification, solution techniques, and future research”. In: *Journal of Intelligent Manufacturing* 30.1 (2019), pp. 335–349.
- [69] Wanqing Chen et al. “Cancer statistics in China, 2015”. In: *CA: a cancer journal for clinicians* 66.2 (2016), pp. 115–132.
- [70] Peng Cheng and Steven M LaValle. “Resolution complete rapidly-exploring random trees”. In: *Proceedings 2002 IEEE international conference on robotics and automation (cat. no. 02CH37292)*. Vol. 1. IEEE. 2002, pp. 267–272.
- [71] Pinar Civicioglu. “Backtracking Search Optimization Algorithm for numerical optimization problems”. In: *Applied Mathematics and Computation* 219.15 (Apr. 2013), pp. 8121–8144. DOI: 10.1016/j.amc.2013.02.017. URL: <https://doi.org/10.1016/j.amc.2013.02.017>.
- [72] G. Clarke and J. W. Wright. “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”. In: *Operations Research* 12.4 (1964), pp. 568–581. ISSN: 0030364X, 15265463. URL: <http://www.jstor.org/stable/167703>.
- [73] Claude Cohen-Tannoudji, Bernard Diu, and Frank Laloe. “Quantum Mechanics, Volume 2”. In: *Quantum Mechanics* 2 (1986), p. 626.
- [74] Brian Cornet, Hua Fang, and Honggang Wang. “Overview of Quantum Technologies, Standards, and their Applications in Mobile Devices”. In: *GetMobile: Mobile Computing and Communications* 24.4 (2021), pp. 5–9.
- [75] Lilia M. Cortina et al. “Incivility in the workplace: Incidence and impact.” In: *Journal of Occupational Health Psychology* 6.1 (2001), pp. 64–80. DOI: 10.1037/1076-8998.6.1.64. URL: <https://doi.org/10.1037/1076-8998.6.1.64>.
- [76] Daniele Cuomo, Marcello Caleffi, and Angela Sara Cacciapuoti. “Towards a distributed quantum computing ecosystem”. In: *IET Quantum Communication* 1.1 (2020), pp. 3–8.
- [77] D-Wave. *D-Wave computer*. 2021. URL: <https://www.dwavesys.com/>.
- [78] BOVET D. and Pierluigi CRESCENZI. “Introduction to the theory of complexity”. In: *Introduction to the theory of complexity*. xi. New York: Prentice Hall international series in computer science, 1994, ISBN 0-13- 915380-2 (1994).
- [79] Alexander Shen Daniel Hammer Andrei Romashchenko and Nikolai Vereshchagin. “Inequalities for Shannon Entropy and Kolmogorov Complexity”. In: *Journal of Computer and System Sciences* (60 2000), pp. 442–464.
- [80] George B Dantzig, Alex Orden, Philip Wolfe, et al. “The generalized simplex method for minimizing a linear form under linear inequality restraints”. In: *Pacific Journal of Mathematics* 5.2 (1955), pp. 183–195.

- [81] George B. Dantzig and Wm. Orchard-Hays. “The Product Form for the Inverse in the Simplex Method”. In: *Mathematical Tables and Other Aids to Computation* 8.46 (Apr. 1954), p. 64. DOI: 10.2307/2001993. URL: <https://doi.org/10.2307/2001993>.
- [82] Ronald De Wolf. “Quantum computing: Lecture notes”. In: *arXiv preprint arXiv:1907.09415* (2019).
- [83] Shantanu Debnath et al. “Demonstration of a small programmable quantum computer with atomic qubits”. In: *Nature* 536.7614 (2016), pp. 63–66.
- [84] Jeroen Dehaene and Bart De Moor. “Clifford group, stabilizer states, and linear and quadratic operations over  $GF(2)$ ”. In: *Physical Review A* 68.4 (Oct. 2003). ISSN: 1094-1622. DOI: 10.1103/physreva.68.042318. URL: <http://dx.doi.org/10.1103/PhysRevA.68.042318>.
- [85] Huamin Zhang Feng Ding. “On the Kronecker Products and Their Applications”. In: *Hindawi Publishing Corporation Journal of Applied Mathematics* (2013).
- [86] František Duchoň et al. “Path planning with modified a star algorithm for a mobile robot”. In: *Procedia Engineering* 96 (2014), pp. 59–69.
- [87] Dmitri Maslov Gerhard W. Dueck and D. Michael Miller. “Toffoli Network Synthesis With Templates”. In: *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS* 24 (6 2005), pp. 807–817.
- [88] Denning Dorethy E. “A lattice model of secure information flow”. In: *Communications of ACM* 19 (5 1976), pp. 236–243.
- [89] Harper R. Grimsley Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. In: *Nature Communications* (2019).
- [90] Stavros Efthymiou et al. *Qibo: a framework for quantum simulation with hardware acceleration*. 2020. arXiv: 2009.01845 [quant-ph].
- [91] Robert Eisberg and Robert Resnick. *Quantum physics of atoms, molecules, solids, nuclei, and particles*. 1985.
- [92] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph].
- [93] Edward Farhi and Aram W Harrow. *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*. 2019. arXiv: 1602.07674 [quant-ph].
- [94] Sebastian Feld et al. “A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer”. In: *arXiv* (2019).
- [95] Cucker Felipe. “A Theory of Complexity, Condition and Roundoff”. In: *ArXiv* (2014).
- [96] Mark Fingerhuth, Tomáš Babej, and Peter Wittek. “Open source software in quantum computing”. In: *PLOS ONE* 13.12 (Dec. 2018). Ed. by Leonie AnnaEditor Mueck, e0208561. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0208561. URL: <http://dx.doi.org/10.1371/journal.pone.0208561>.
- [97] Carro Antonio Carlos Schneider Beck Fl and Luigi. *Dynamic Optimization Techniques*. Brazil: springer, 2010.
- [98] Raspberry Pi Foundation. *Raspberry Pi 4*. 2021. URL: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>.
- [99] J.H. Friedman, F. Baskett, and L.J. Shustek. “An Algorithm for Finding Nearest Neighbors”. In: *IEEE Transactions on Computers* C-24.10 (Oct. 1975), pp. 1000–1006. DOI: 10.1109/t-c.1975.224110. URL: <https://doi.org/10.1109/t-c.1975.224110>.

- [100] Hill Michael C. Fu and Stacy D. “Optimization of discrete event systems via simultaneous perturbation stochastic approximation”. In: 29 (1997), pp. 233–243.
- [101] K. Fukunaga and P.M. Narendra. “A Branch and Bound Algorithm for Computing k-Nearest Neighbors”. In: *IEEE Transactions on Computers* C-24.7 (July 1975), pp. 750–753. DOI: 10.1109/t-c.1975.224297. URL: <https://doi.org/10.1109/t-c.1975.224297>.
- [102] N Gademann. “Van de velde, S.,(2005)”. In: *Order batching to minimize total travel time in a* ().
- [103] Michael R Garey. “A Guide to the Theory of NP-Completeness”. In: *Computers and intractability* (1979).
- [104] Elizabeth Gibney. “D-wave upgrade: How scientists are using the world’s most controversial quantum computer”. In: *Nature News* 541.7638 (2017), p. 447.
- [105] Elizabeth Gibney. “Inside Microsoft’s quest for a topological quantum computer”. In: *Nature* (2016).
- [106] Kalai Gil. *The Argument against Quantum Computers, the Quantum Laws of Nature, and Google’s Supremacy Claims*. Herzliya, 2018-2019.
- [107] Teun van Gils, An Caris Katrien Ramaekers, and Rene BM de Koster. “Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review.” In: *European Journal of Operational Research* 267.1 (2018), pp. 1–15.
- [108] Saúl González-Bermejo, Guillermo Alonso-Linaje, and Parfait Atchade-Adelomou. “GPS: Improvement in the formulation of the TSP for its generalizations type QUBO”. In: *arXiv preprint arXiv:2110.12158* (2021).
- [109] D Gottesman. “The Heisenberg representation of quantum computers”. In: *arXiv:quant-ph/9807006* 1.1 (June 1998). URL: <https://www.osti.gov/biblio/319738>.
- [110] Daniel Gottesman. “Theory of fault-tolerant quantum computation”. In: *Physical Review A* 57.1 (Jan. 1998), pp. 127–137. DOI: 10.1103/physreva.57.127. URL: <https://doi.org/10.1103/physreva.57.127>.
- [111] Harper R Grimsley et al. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. In: *Nature communications* 10.1 (2019), pp. 1–9.
- [112] G. G. Guerreschi and A. Y. Matsuura. “QAOA for Max-Cut requires hundreds of qubits for quantum speed-up”. In: *Scientific Reports* 9.1 (May 2019). DOI: 10.1038/s41598-019-43176-9. URL: <https://doi.org/10.1038/s41598-019-43176-9>.
- [113] G. G. Guerreschi and A. Y. Matsuura. “QAOA for Max-Cut requires hundreds of qubits for quantum speed-up”. In: *Scientific Reports* 9.1 (May 2019). DOI: 10.1038/s41598-019-43176-9. URL: <https://doi.org/10.1038/s41598-019-43176-9>.
- [114] Gregory Gutin, Anders Yeo, and Alexey Zverovich. “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP”. In: *Discrete Applied Mathematics* 117.1-3 (Mar. 2002), pp. 81–86. DOI: 10.1016/s0166-218x(01)00195-0. URL: [https://doi.org/10.1016/s0166-218x\(01\)00195-0](https://doi.org/10.1016/s0166-218x(01)00195-0).
- [115] Gregory Gutin, Anders Yeo, and Alexei Zverovitch. “Exponential neighborhoods and domination analysis for the TSP”. In: *The traveling salesman problem and its variations*. Springer, 2007, pp. 223–256.
- [116] Vojtěch Havlíček et al. “Supervised learning with quantum-enhanced feature spaces”. In: *Nature* 567.7747 (Mar. 2019), pp. 209–212. DOI: 10.1038/s41586-019-0980-2. URL: <https://doi.org/10.1038/s41586-019-0980-2>.
- [117] Daochen Wang Oscar Higgott and Stephen Brierley. “Accelerated Variational Quantum Eigensolver”. In: *arXiv* (2019).

- [118] Chih-Ming Hsu, Kai-Ying Chen, and Mu-Chen Chen. “Batching orders in warehouses by minimizing travel distance with genetic algorithms”. In: *Computers in industry* 56.2 (2005), pp. 169–178.
- [119] Ekeland I. “On the variational principle *Journal of Mathematical Analysis and Applications*”. In: (1974).
- [120] IBM. *DOcplex Python Modeling API*. 2021. URL: <https://www.ibm.com/docs/en/icos/12.9.0?topic=docplex-python-modeling-api>.
- [121] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. “Which problems have strongly exponential complexity?” In: *Journal of Computer and System Sciences* 63.4 (2001), pp. 512–530.
- [122] Iterate. *cyberduck - SSH*. 2021. URL: <https://cyberduck.io/>.
- [123] Djordjevic Ivan. *Quantum Information Processing and Quantum Error Correction: An Engineering Approach*. ELSEVIER, 2012.
- [124] Vanderbei Robert J. *Linear Programming. Foundations and Extensions*. Springer, 2014.
- [125] Lars Jaeger. *The Second Quantum Revolution*. Springer International Publishing, 2018. DOI: 10.1007/978-3-319-98824-5. URL: <https://doi.org/10.1007/978-3-319-98824-5>.
- [126] Dave Jaggard. “ARM architecture and systems”. In: *IEEE Annals of the History of Computing* 17.04 (1997), pp. 9–11.
- [127] Soran Jahangiri et al. “Quantum algorithm for simulating molecular vibrational excitations”. In: *Physical Chemistry Chemical Physics* 22.44 (2020), pp. 25528–25537. ISSN: 1463-9084. DOI: 10.1039/d0cp03593a. URL: <http://dx.doi.org/10.1039/D0CP03593A>.
- [128] Roman Slowinski James H. Patterson F. Brian Talbot and Jan Weglarz. “Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems”. In: *European Journal of Operational Research - North-Holland* 49 (1990), pp. 68–79.
- [129] Qingye Jiang, Young Choon Lee, and Albert Y Zomaya. “The Power of ARM64 in Public Clouds”. In: *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE. 2020, pp. 459–468.
- [130] Stuart Johns and Peter Brucker. “Scheduling Algorithms.” In: *The Journal of the Operational Research Society* 47.8 (Aug. 1996), p. 1077. DOI: 10.2307/3010416. URL: <https://doi.org/10.2307/3010416>.
- [131] Tadashi Kadowaki and Hidetoshi Nishimori. “Quantum annealing in the transverse Ising model”. In: *Physical Review E* 58.5 (Nov. 1998), pp. 5355–5363. ISSN: 1095-3787. DOI: 10.1103/PhysRevE.58.5355. URL: <http://dx.doi.org/10.1103/PhysRevE.58.5355>.
- [132] Hemant M Kakde. “Range searching using KD tree”. In: *from the citeseerx database on the World Wide Web: <http://citeseerx.ist.psu.edu/viewdoc/summary>* (2005).
- [133] Peter J Karalekas et al. “A quantum-classical cloud platform optimized for variational hybrid algorithms”. In: *Quantum Science and Technology* 5.2 (Apr. 2020), p. 024003. DOI: 10.1088/2058-9565/ab7559. URL: <https://doi.org/10.1088/2058-9565/ab7559>.
- [134] Richard M Karp. “Reducibility among combinatorial problems”. In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.
- [135] Tosio Kato. “On the Trotter-Lie product formula”. In: *Proceedings of the Japan Academy* 50.9 (1974), pp. 694–698.
- [136] Viv Kendon. “Quantum computing using continuous-time evolution”. In: *Interface Focus* 10.6 (Oct. 2020), p. 20190143. DOI: 10.1098/rsfs.2019.0143. URL: <https://doi.org/10.1098/rsfs.2019.0143>.

- [137] Nader Khammassi et al. “QX: A high-performance quantum computer simulation platform”. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE. 2017, pp. 464–469.
- [138] Christina Knapp et al. “Dephasing of Majorana-based qubits”. In: *arXiv* (2018).
- [139] Sören Koch and Gerhard Wäscher. “A grouping genetic algorithm for the order batching problem in distribution warehouses”. In: *Journal of Business Economics* 86.1-2 (2016), pp. 131–153.
- [140] Gary Kochenberger et al. “The unconstrained binary quadratic programming problem: a survey”. In: *Journal of Combinatorial Optimization* 28.1 (2014), pp. 58–81.
- [141] Kuzmak and Tkachuk. “Preparation and study of the entanglement of the Schrödinger cat state on the ibmq-melbourne quantum computer”. In: *Condensed Matter Physics* 23.4 (Dec. 2020), p. 43001. DOI: 10.5488/cmp.23.43001. URL: <https://doi.org/10.5488/cmp.23.43001>.
- [142] Katarzyna Anna Kuźmicz. “Benchmarking in omni-channel logistics”. In: *Research in logistics & production* 5 (2015).
- [143] Thi Ha Kyaw et al. *Quantum computer-aided design: digital quantum simulation of quantum processors*. 2020. arXiv: 2006.03070 [quant-ph].
- [144] Edward O. Wilson Rolf Landauer. *Fundamental concepts of Hamiltonian Pauli Terms Quantum Computation*.
- [145] Gilbert Laporte. “The vehicle routing problem: An overview of exact and approximate algorithms”. In: *European Journal of Operational Research* 59.3 (June 1992), pp. 345–358. DOI: 10.1016/0377-2217(92)90192-c. URL: [https://doi.org/10.1016/0377-2217\(92\)90192-c](https://doi.org/10.1016/0377-2217(92)90192-c).
- [146] Steven M LaValle et al. “Rapidly-exploring random trees: A new tool for path planning”. In: (1998).
- [147] Steven M LaValle, James J Kuffner, BR Donald, et al. “Rapidly-exploring random trees: Progress and prospects”. In: *Algorithmic and computational robotics: new directions* 5 (2001), pp. 293–308.
- [148] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. “A quantum annealing architecture with all-to-all connectivity from local interactions”. In: *Science advances* 1.9 (2015), e1500838.
- [149] Yang T.D. Lee and C. N. “Statistical Theory of Equation of State and Phase Transitions. Lattice Gas and Ising Model”. In: *Physical Review* 87 (3 1952), pp. 410–419.
- [150] Jan Karel Lenstra and AHG Rinnooy Kan. “Some simple applications of the travelling salesman problem”. In: *Journal of the Operational Research Society* 26.4 (1975), pp. 717–733.
- [151] Hui Li and Jie Sun. “Ranking-order case-based reasoning for financial distress prediction”. In: *Knowledge-Based Systems* 21.8 (Dec. 2008), pp. 868–878. DOI: 10.1016/j.knsys.2008.03.047. URL: <https://doi.org/10.1016/j.knsys.2008.03.047>.
- [152] Salim Al Isma’ili Mengxiang Li et al. “African Societal Challenges Transformation through IoT”. In: *Association for Information Systems* (2017).
- [153] Min Liang and Li Yang. *A note on threshold theorem of fault-tolerant quantum computation*. 2010. arXiv: 1006.4941 [quant-ph].
- [154] Laura Lozano and Javier Fernández. “Razonamiento basado en casos: Una visión general”. In: *Recuperado el* (2008).
- [155] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014), p. 5.
- [156] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014). DOI: 10.3389/fphy.2014.00005. URL: <https://doi.org/10.3389/fphy.2014.00005>.

- [157] Andrew Lucas. “Ising formulations of many NP problems”. In: *Frontiers in Physics* 2 (2014), p. 5.
- [158] Eduardo Lupiani et al. “Monitoring elderly people at home with temporal Case-Based Reasoning”. In: *Knowledge-Based Systems* 134 (Oct. 2017), pp. 116–134. DOI: 10.1016/j.knosys.2017.07.025. URL: <https://doi.org/10.1016/j.knosys.2017.07.025>.
- [159] Anuradha Mahasinghe et al. “Solving the Hamiltonian Cycle Problem using a Quantum Computer”. In: *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, Jan. 2019. DOI: 10.1145/3290688.3290703. URL: <https://doi.org/10.1145/3290688.3290703>.
- [160] CPLEX User’s Manual. “Ibm ilog cplex optimization studio”. In: *Version 12* (1987), pp. 1987–2018.
- [161] John Metcalfe Mark Lewis Gary Kochenberger. “Robust Optimization of Unconstrained Binary Quadratic Problems”. In: *Inder Science Online* (2019).
- [162] de Matthew Sands Richard Feynman and Robert Leighton. *The Feynman Lectures on Physics*. The New Millennium Edition, 2011.
- [163] Catherine C. McGeoch. “Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice”. In: *Synthesis Lectures on Quantum Computing* 5.2 (July 2014), pp. 1–93. DOI: 10.2200/s00585ed1v01y201407qmc008. URL: <https://doi.org/10.2200/s00585ed1v01y201407qmc008>.
- [164] David C McKay et al. “Qiskit backend specifications for openqasm and openpulse experiments”. In: *arXiv preprint arXiv:1809.03452* (2018).
- [165] Yeray Mezquita et al. “A Review of k-NN Algorithm Based on Classical and Quantum Machine Learning”. In: *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*. Springer International Publishing, July 2020, pp. 189–198. DOI: 10.1007/978-3-030-53829-3\_20. URL: [https://doi.org/10.1007/978-3-030-53829-3\\_20](https://doi.org/10.1007/978-3-030-53829-3_20).
- [166] Andrew C. Doherty Michael A. Nielsen Mark R. Dowling Mile Gu. “Quantum Computation as Geometry”. In: *Science* 311 (2006), pp. 1133–1135.
- [167] Microsoft. *Microsoft Quantum Development Kit*. June 2021. URL: <https://azure.microsoft.com/es-es/resources/development-kit/quantum-computing/>.
- [168] Microsoft. *qsharp by Microsoft*. June 2021. URL: <https://qsharp.community/>.
- [169] Alán Aspuru-Guzik Mikhail Smelyanskiy Nicolas P. D. Sawaya. “qHiPSTER: The Quantum High Performance Software Testing Environment”. In: *arXiv* (2016).
- [170] Stephen L. Morgan and Stanley N. Deming. “Simplex optimization of analytical chemical methods”. In: *Analytical Chemistry* 46.9 (Aug. 1974), pp. 1170–1181. DOI: 10.1021/ac60345a035. URL: <https://doi.org/10.1021/ac60345a035>.
- [171] Ana Moura. “A Multi-Objective Genetic Algorithm for the Vehicle Routing with Time Windows and Loading Problem”. In: *Intelligent Decision Support*. Gabler, pp. 187–201. DOI: 10.1007/978-3-8349-9777-7\_11. URL: [https://doi.org/10.1007/978-3-8349-9777-7\\_11](https://doi.org/10.1007/978-3-8349-9777-7_11).
- [172] Hazarath Munaga and Venkata Jarugumalli. “Performance Evaluation: Ball-Tree and KD-Tree in the Context of MST”. In: *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer Berlin Heidelberg, 2012, pp. 225–228. DOI: 10.1007/978-3-642-32573-1\_38. URL: [https://doi.org/10.1007/978-3-642-32573-1\\_38](https://doi.org/10.1007/978-3-642-32573-1_38).
- [173] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.
- [174] Michael A. Nielsen, Isaac Chuang, and Lov K. Grover. “Quantum Computation and Quantum Information”. In: *American Journal of Physics* 70.5 (May 2002), pp. 558–559. DOI: 10.1119/1.1463744. URL: <https://doi.org/10.1119/1.1463744>.

- [175] William S Noble. “What is a support vector machine?” In: *Nature Biotechnology* 24.12 (Dec. 2006), pp. 1565–1567. DOI: 10.1038/nbt1206-1565. URL: <https://doi.org/10.1038/nbt1206-1565>.
- [176] Hugo Daniel Macedo José Nuno Oliveira. “Typing linear algebra: A biproduct-oriented approach”. In: *Science of Computer Programming* 78 (2012), pp. 2160–2191.
- [177] Guido Pagano et al. “Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator”. In: *Proceedings of the National Academy of Sciences* 117.41 (2020), pp. 25396–25401.
- [178] Atchade-Adelomou Parfait, Elisabet Golobardes-Ribé, and Xavier Vilasis-Cardona. “Formulation of the Social Workers’ Problem in Quadratic Unconstrained Binary Optimization Form and Solve It on a Quantum Computer”. In: *Journal of Computer and Communications* 8.11 (2020), pp. 44–68. DOI: 10.4236/jcc.2020.811004.
- [179] Adrian Parra-Rodriguez et al. “Digital-analog quantum computation”. In: *Physical Review A* 101.2 (Feb. 2020). ISSN: 2469-9934. DOI: 10.1103/physreva.101.022305. URL: <http://dx.doi.org/10.1103/PhysRevA.101.022305>.
- [180] Adrián Pérez-Salinas et al. “Data re-uploading for a universal quantum classifier”. In: *Quantum* 4 (Feb. 2020), p. 226. ISSN: 2521-327X. DOI: 10.22331/q-2020-02-06-226. URL: <http://dx.doi.org/10.22331/q-2020-02-06-226>.
- [181] Adrián Pérez-Salinas et al. *One qubit as a Universal Approximant*. 2021. arXiv: 2102.04032 [quant-ph].
- [182] Lounesto Pertti. *Clifford Algebras and Spinors*. Cambridge University Press, 1997.
- [183] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Communications* 5.1 (July 2014). DOI: 10.1038/ncomms5213. URL: <https://doi.org/10.1038/ncomms5213>.
- [184] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature Communications* 5.1 (July 2014). DOI: 10.1038/ncomms5213. URL: <https://doi.org/10.1038/ncomms5213>.
- [185] Wittek Peter. *Quantum Machine Learning What Quantum Computing Means to Data Mining*. Academic Press, 2014.
- [186] Kok Pieter. *A First Introduction to Quantum Physics*. Springer, 2018.
- [187] Enrico Prati et al. “From the Quantum Moore’s Law toward Silicon Based Universal Quantum Computing”. In: *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–4.
- [188] William K. Pratt and John Wiley. “A Wiley-Interscience Publication”. In: *Digital Image Processing*. John Wiley & Sons, INC, 1978, pp. 636–646.
- [189] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. DOI: 10.22331/q-2018-08-06-79. URL: <https://doi.org/10.22331/q-2018-08-06-79>.
- [190] Paolo Priore et al. “Utilización del razonamiento basado en casos en la toma de decisiones. Aplicación en un problema de secuenciación”. In: *Dirección y Organización* 28 (2002).
- [191] Qiskit. “docplex (high-level modeling language)”. In: *Medium* (May 2020). URL: [https://qiskit.org/documentation/tutorials/optimization/3\\_minimum\\_eigen\\_optimizer.html](https://qiskit.org/documentation/tutorials/optimization/3_minimum_eigen_optimizer.html).
- [192] IA Eckley R Killick P Fearnhead. “Optimal Detection of Changepoints With a Linear Computational Cost”. In: *Journal of the American Statistical Association* 107 (2012), pp. 1590–1598.



- [193] Berman Paul R. *Introductory Quantum Mechanics. A Traditional Approach Emphasizing Connections with Classical Physics*. Springer, 2018.
- [194] Bes Daniel R. *Quantum Mechanics. A Modern and Concise Introductory Course*. New York: Springer, 2012.
- [195] Shankar R. *Principles of Quantum Mechanics*. Springer, 1980.
- [196] Nicholas Rawlinson and Malcolm Sambridge. “The fast marching method: an effective tool for tomographic imaging and tracking multiple phases in complex layered media”. In: *Exploration Geophysics* 36.4 (2005), pp. 341–350.
- [197] Nikolay Raychev. “Quantum circuit for spatial optimization”. In: *International Journal of Scientific and Engineering Research* 6.6 (2015), pp. 1365–1368.
- [198] Bhat RB. “Natural frequencies of rectangular plates using characteristic orthogonal polynomials in Rayleigh-Ritz method”. In: *Journal of Sound and Vibration* 102 (4 1985), pp. 493–499.
- [199] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum Support Vector Machine for Big Data Classification”. In: *Physical Review Letters* 113.13 (Sept. 2014). ISSN: 1079-7114. DOI: 10.1103/physrevlett.113.130503. URL: <http://dx.doi.org/10.1103/PhysRevLett.113.130503>.
- [200] Patrick Rebentrost et al. “Quantum gradient descent and Newton’s method for constrained polynomial optimization”. In: *New Journal of Physics* 21.7 (July 2019), p. 073023. DOI: 10.1088/1367-2630/ab2a9e. URL: <https://doi.org/10.1088/1367-2630/ab2a9e>.
- [201] Alistair P. Rendell et al. “Computational chemistry on Fujitsu vector± parallel processors: Development and performance of applications software”. In: (2000).
- [202] Harvey S Leff Andrew F Rex. *Maxwell’s Demon 2, Entropy, Classical and Quantum Information, Computing*. Harvey S Leff, Andrew F Rex, 2002.
- [203] Joschka Roffe. “Quantum error correction: an introductory guide”. In: *Contemporary Physics* 60.3 (July 2019), pp. 226–245. ISSN: 1366-5812. DOI: 10.1080/00107514.2019.1667078. URL: <http://dx.doi.org/10.1080/00107514.2019.1667078>.
- [204] Manuel de la Rosa Fernández. “Spin glasses”. In: (2018).
- [205] Katarzyna Roszak, Radim Filip, and Tomáš Novotný. “Decoherence control by quantum decoherence itself”. In: *Scientific Reports* 5.1 (June 2015). DOI: 10.1038/srep09796. URL: <https://doi.org/10.1038/srep09796>.
- [206] PJY Wong; RP Agarwal M Bohner. “Sturm-Liouville eigenvalue problems on time scales”. In: *Applied Mathematics and Computation* 99 (1999), pp. 153–166.
- [207] Dreyfus S. “Richard Bellman on the Birth of Dynamic Programming”. In: *Institute for Operations Research and the Management Sciences* (2002).
- [208] Martin Savelsbergh and Tom Van Woensel. “50th anniversary invited article—city logistics: Challenges and opportunities”. In: *Transportation Science* 50.2 (2016), pp. 579–590.
- [209] Douglas C. Schmidt and Larry E. Druffel. “A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices”. In: *Journal of the ACM* 23.3 (July 1976), pp. 433–445. DOI: 10.1145/321958.321963. URL: <https://doi.org/10.1145/321958.321963>.
- [210] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
- [211] Maria Schuld and Nathan Killoran. “Quantum Machine Learning in Feature Hilbert Spaces”. In: *Physical Review Letters* 122.4 (Feb. 2019). ISSN: 1079-7114. DOI: 10.1103/physrevlett.122.040504. URL: <http://dx.doi.org/10.1103/PhysRevLett.122.040504>.

- [212] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. “An introduction to quantum machine learning”. In: *Contemporary Physics* 56.2 (Oct. 2014), pp. 172–185. ISSN: 1366-5812. DOI: 10.1080/00107514.2014.964942. URL: <http://dx.doi.org/10.1080/00107514.2014.964942>.
- [213] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. “The quest for a Quantum Neural Network”. In: *Quantum Information Processing* 13.11 (Aug. 2014), pp. 2567–2586. DOI: 10.1007/s11128-014-0809-8. URL: <https://doi.org/10.1007/s11128-014-0809-8>.
- [214] Engineering National Academies of Sciences (and Medicine) et al. *Quantum computing: progress and prospects*. National Academies Press, 2019.
- [215] Eyob A Sete, William J Zeng, and Chad T Rigetti. “A functional architecture for scalable quantum computing”. In: *2016 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE. 2016, pp. 1–6.
- [216] 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), pp. 1484–1509. ISSN: 1095-7111. DOI: 10.1137/S0097539795293172. URL: <http://dx.doi.org/10.1137/S0097539795293172>.
- [217] Julia Siderska. “Robotic Process Automation—a driver of digital transformation?” In: *Engineering Management in Production and Services* 12.2 (2020).
- [218] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. “Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms”. In: *Advanced Quantum Technologies* 2.12 (Oct. 2019), p. 1900070. ISSN: 2511-9044. DOI: 10.1002/qute.201900070. URL: <http://dx.doi.org/10.1002/qute.201900070>.
- [219] Claudio Gambella Andrea Simonetto. “Multi-block ADMM Heuristics for Mixed-Binary Optimization on Classical and Quantum Computers”. In: *IBM Research* (2020).
- [220] Ali Hassan Sodhro, Sandeep Pirbhulal, and Arun Kumar Sangaiah. “Convergence of IoT and product lifecycle management in medical health care”. In: *Future Generation Computer Systems* 86 (Sept. 2018), pp. 380–391. DOI: 10.1016/j.future.2018.03.052. URL: <https://doi.org/10.1016/j.future.2018.03.052>.
- [221] Robert Stárek et al. “Experimental quantum decoherence control by dark states of the environment”. In: *New Journal of Physics* 22.9 (Sept. 2020), p. 093058. DOI: 10.1088/1367-2630/abb47d. URL: <https://doi.org/10.1088/1367-2630/abb47d>.
- [222] Hardy Willi-Hans Steeb and Yorick. *Matrix Calculus and Kronecker Product A Practical Approach to Linear and Multilinear Algebra*. 2019.
- [223] Adam R. Brown Leonard Susskind. “Second law of quantum complexity”. In: *INSPIRE* (2018).
- [224] Roubíček T. *Relaxation in optimization theory and variational calculus*. 2011.
- [225] McDonald Kirk T. *Ph410 Physics of Quantum Computation1*. Princeton University, 2017.
- [226] Ewin Tang. *Quantum-inspired classical algorithms for principal component analysis and supervised clustering*. 2019. arXiv: 1811.00414 [cs.DS].
- [227] community The SciPy. *COBYLA*. The SciPy community, Jan. 2021. URL: <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-cobyla.html>.
- [228] James A. Tompkins et al. *Facilities planning*. John Wiley and Sons, 2010.
- [229] Paolo Toth and Daniele Vigo, eds. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Jan. 2002. DOI: 10.1137/1.9780898718515. URL: <https://doi.org/10.1137/1.9780898718515>.
- [230] Tony T Tran et al. “A hybrid quantum-classical approach to solving scheduling problems”. In: *Ninth annual symposium on combinatorial search*. 2016.

- [231] C-Y Tsai, James JH Liou, and T-M Huang. “Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time”. In: *International Journal of Production Research* 46.22 (2008), pp. 6533–6555.
- [232] Edward Tsang. *Foundations of constraint satisfaction: the classic text*. BoD–Books on Demand, 2014.
- [233] Prasanta K. Panigrahi Utkarsh Azad Bikash K. Behera. “Solving Vehicle Routing Problem using Quantum Approximate Optimization Algorithm”. In: *Research Gate* (2019).
- [234] Wil MP Van der Aalst, Martin Bichler, and Armin Heinzl. *Robotic process automation*. 2018.
- [235] Francisca Vasconcelos. “Quantum computing@ MIT: the past, present, and future of the second revolution in computing”. In: *arXiv preprint arXiv:2002.05559* (2020).
- [236] Eleanor G. Rieffel Davide Venturelli et al. “A case study in programming a quantum annealer for hard operational planning problems”. In: *arXiv* (2014).
- [237] Tomas Vyskocil and Hristo Djidjev. “Embedding Equality Constraints of Optimization Problems into a Quantum Annealer”. In: *Algorithms* 12.4 (Apr. 2019), p. 77. DOI: 10.3390/a12040077. URL: <https://doi.org/10.3390/a12040077>.
- [238] Wu Chai Wah. “On Rayleigh–Ritz ratios of a generalized Laplacian matrix of directed graphs”. In: *Linear Algebra and its Applications* 402 (2005), pp. 207–227.
- [239] Simon Marvin While Aidan H. and Mateja Kovacic. “Urban robotic experimentation: San Francisco, Tokyo and Dubai”. In: *Urban Studies* 58.4 (2021), pp. 769–786.
- [240] Wikiwand. *The Lattice graph*. May 2021. URL: [https://www.wikiwand.com/en/Lattice\\_group](https://www.wikiwand.com/en/Lattice_group).
- [241] Robert Wille, Rod Van Meter, and Yehuda Naveh. “IBM’s Qiskit Tool Chain: Working with and Developing for Real Quantum Computers”. In: *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2019, pp. 1234–1240.
- [242] D. Willsch et al. “Support vector machines on the D-Wave quantum annealer”. In: *Computer Physics Communications* 248 (Mar. 2020), p. 107006. DOI: 10.1016/j.cpc.2019.107006. URL: <https://doi.org/10.1016/j.cpc.2019.107006>.
- [243] D. Bank X. Wang V.A. Truong. *Online Advance Admission Scheduling for Services*. New York, 2018.
- [244] JuaJo Rué Sebastián Xambó. *MATHEMATICAL ESSENTIALS OF QUANTUM COMPUTING*. 2013.
- [245] Lin Xie, Hanyi Li, and Laurin Luttmann. *Formulating and solving integrated order batching and routing in multi-depot AGV-assisted mixed-shelves warehouses*. 2021. arXiv: 2101.11473 [math.OC].
- [246] N. Young. *An Introduction to Hilbert Space*. Cambridge University Press, July 1988. DOI: 10.1017/cbo9781139172011. URL: <https://doi.org/10.1017/cbo9781139172011>.
- [247] Tristan Zaborniak and Rogerio de Sousa. “Benchmarking Hamiltonian Noise in the D-Wave Quantum Annealer”. In: *IEEE Transactions on Quantum Engineering 2* (2021), pp. 1–6. ISSN: 2689-1808. DOI: 10.1109/tqe.2021.3050449. URL: <http://dx.doi.org/10.1109/TQE.2021.3050449>.
- [248] Zee Ma. *A Tutorial on Principal Component Analysis*. en. 2014. DOI: 10.13140/2.1.1593.1684. URL: <http://rgdoi.net/10.13140/2.1.1593.1684>.
- [249] Zhi-jin Zhao and Hai-chao Lai. “A cognitive engine based on case-based reasoning quantum genetic algorithm”. In: *2012 IEEE 14th International Conference on Communication Technology*. IEEE, Nov. 2012. DOI: 10.1109/icct.2012.6511219. URL: <https://doi.org/10.1109/icct.2012.6511219>.

- [250] Linghua Zhu et al. *An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer*. 2020. arXiv: 2005.10258 [quant-ph].
- [251] Peter Zoller et al. “Quantum information processing and communication”. In: *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics* 36.2 (2005), pp. 203–228.