



INSTYTUT INFORMATYKI TEORETYCZNEJ I STOSOWANEJ
POLSKIEJ AKADEMII NAUK

THEORY AND APPLICATIONS OF HYBRID
QUANTUM-CLASSICAL OPTIMIZATION ALGORITHMS

ROZPRAWA DOKTORSKA

mgr Ludmila BOTELHO

Promotor:

dr hab. Jarosław MISZCZAK, prof. IITiS

Gliwice, 25.06.2024



INSTITUTE OF THEORETICAL AND APPLIED INFORMATICS, POLISH
ACADEMY OF SCIENCES

THEORY AND APPLICATIONS OF
HYBRID QUANTUM-CLASSICAL
OPTIMIZATION ALGORITHMS

DOCTORAL DISSERTATION

mgr Ludmila BOTELHO

Supervisor:

dr hab. Jarosław MISZCZAK, prof. IITiS

Gliwice, 25.06.2024

Contents

List of publications	9
Abstract in Polish	11
Abstract in English	13
1 Introduction	17
2 Preliminaries	21
2.1 Complexity	21
2.1.1 Complexity Classes	22
2.2 Basics of Graphs	23
2.3 Optimization Programming formulations	24
2.3.1 Integer Linear Programming (ILP)	24
2.3.2 Quadratic Unconstrained Binary Optimization	25
2.4 Combinatorial Problems	26
2.4.1 Max-Cut	26
2.4.2 Travelling Salesperson	27
2.4.3 Job Scheduling Problems	28
2.5 Simulated Annealing	28
2.6 The Ising Model	29
2.7 Basics of Quantum Circuits	30
2.7.1 The qubit	30
2.7.2 Unitary operators	31
2.7.3 Measurement process	32
2.8 Quantum Optimization Algorithms in NISQ era	33
2.8.1 Adiabatic Quantum Computation	34
2.8.2 Quantum Annealing	34
2.8.3 Variational Quantum Algorithms	35
2.8.4 Hamiltonian Simulation	36
2.8.5 Quantum Approximation Optimization Algorithms	37

Contents

2.9	How to compute and evaluate quantum algorithms?	39
2.9.1	Meta-programming	39
2.9.2	Gradient descent and gradient-free methods for classical optimization	39
2.9.3	Input/Output	40
3	Error Mitigation for QAOA using post-selection	41
3.1	Introduction	41
3.2	Error Mitigation	43
3.2.1	Post-Selection by filtering and compression	45
3.3	Post-selection schemes for different encodings	46
3.3.1	k -hot encoding	48
3.3.2	One-hot encoding	48
3.3.3	Domain-wall encoding	49
3.3.4	Binary and Gray encoding	51
3.3.5	One-hot and binary mixed	52
3.3.6	Summary	53
3.4	Application to Quantum Alternating Operator Ansatz	54
3.5	Circuit Design	57
3.6	Results	57
3.7	Conclusion	62
3.8	Chapter Summary	63
4	Hamiltonian-Oriented Homotopy QAOA	65
4.1	Introduction	65
4.2	Energy Landscape	66
4.3	Homotopy optimization method	68
4.4	Hamiltonian-Oriented Homotopy QAOA	69
4.4.1	Initialization strategy	70
4.5	Results	71
4.6	Conclusion	73
4.7	Chapter Summary	76
5	Applications of Quantum Annealing for Music Theory	79
5.1	Introduction	79
5.2	Music generation and optimization problems	80
5.3	Music Generation	82
5.3.1	Melody Generation	82
5.3.2	Rhythm Generation	90
5.4	Music Reduction	92
5.4.1	Phrase identification	93

Contents

5.4.2	Determining the weight for each phrase	93
5.5	Optimization algorithm	94
5.5.1	QUBO formulation	95
5.5.2	Post-processing	96
5.5.3	Enhancing the model	96
5.5.4	Results	97
5.6	Conclusions	101
5.7	Chapter Summary	102
6	Conclusions	105
	Bibliography	119

Contents

List of publications

Publications relevant to this dissertation are highlighted using bold font.

1. A. Kundu, L. Botelho, A. Glos, *Hamiltonian-Oriented Homotopy QAOA*, Physical Review A, Vol. 109, No. 2, pp. 022611 (2024).
DOI:[10.1103/PhysRevA.109.022611](https://doi.org/10.1103/PhysRevA.109.022611) arXiv:[2301.13170](https://arxiv.org/abs/2301.13170)
Source Code: DOI:[10.5281/zenodo.7585691](https://doi.org/10.5281/zenodo.7585691)
2. L. Botelho, A. Glos, A. Kundu, J.A. Miszczak, Ö. Salehi, Z. Zimborás, *Error mitigation for variational quantum algorithms through mid-circuit measurements*, Physical Review A, Vol. 105, No. 2, pp. 022441 (2022)
DOI:[10.1103/PhysRevA.105.022441](https://doi.org/10.1103/PhysRevA.105.022441) arXiv:[2108.10927](https://arxiv.org/abs/2108.10927)
Source Code: DOI:[10.5281/zenodo.5864561](https://doi.org/10.5281/zenodo.5864561)
3. A. Arya, L. Botelho., F. Cañete, D. Kapadia, Ö. Salehi, Z. ; *Applications of Quantum Annealing to Music Theory*; In: Miranda, E.R. (eds), *Quantum Computer Music*, Springer, pp. 373-406, (2022)
DOI:[10.1007/978-3-031-13909-3_15](https://doi.org/10.1007/978-3-031-13909-3_15) arXiv:[2201.10557](https://arxiv.org/abs/2201.10557)
Source Code: DOI:[10.5281/zenodo.5856930](https://doi.org/10.5281/zenodo.5856930)
4. C. Albornoz, G. Alonso, M. Andrenkov, P. Angara, A. Asadi, A. Ballon, S. Bapat, L. Botelho, I. De Vlugt, O. Di Matteo, P. Finlay, A. Fumagalli, A. Gardhouse, N. Girard, A. Hayes, J. Izaac, R. Janik, T. Kalajdzievski, N. Killoran, I. Kurečić, J. Soni, D. Wakeham, *Xanadu Quantum Codebook* (2023). Available at: <https://codebook.xanadu.ai/>.
5. L. Botelho, Ö. Salehi, *Fixed interval scheduling problem with minimal idle time with an application to music arrangement problem* (2023) arXiv:[2310.14825](https://arxiv.org/abs/2310.14825)
Source Code: DOI:[10.5281/zenodo.7410349](https://doi.org/10.5281/zenodo.7410349)
6. M. Atallah, H. Velmurugan, R. Sharma, S. Midha, S. Al Mamun, L. Botelho, A. Glos, Ö. Salehi; *Integer Factorization through Func-QAOA* (2023).
arXiv:[2309.15162](https://arxiv.org/abs/2309.15162)

List of publications

7. M. Koniorczyk, K. Krawiec, L. Botelho, N. Besinovic, K. Domino; *Application of a Hybrid Algorithm Based on Quantum Annealing to Solve a Metropolitan Scale Railway Dispatching Problem* (2023). arXiv:[2309.06763](https://arxiv.org/abs/2309.06763)
8. L. Botelho, R.O. Vianna, *Efficient quantum tomography of two-mode Wigner functions*, The European Physical Journal D 74, 1-7 DOI:[10.1140/epjd/e2020-100649-3](https://doi.org/10.1140/epjd/e2020-100649-3) arXiv:[1912.12222](https://arxiv.org/abs/1912.12222)

Abstract in Polish

W ostatnich dziesięcioleciach społeczność naukowa była świadkiem szybkiej ewolucji komputerów, w wyniku której powstały potężne urządzenia połączone z wyrafinowanymi metodami obliczania trudnych zadań, co pozwoliło zrewolucjonizować sposób, w jaki rozumiemy i przetwarzamy informacje. Jednocześnie społeczność naukowa jest świadkiem pojawienia się nowego paradygmatu: technologia kwantowa to obiecująca dziedzina, która potencjalnie zapewnia korzyści obliczeniowe. W szczególności kultowe algorytmy Shora i Grovera zwróciły uwagę środowiska informatycznego teoretyczną możliwością budowania algorytmów szybszych w porównaniu do obecnie znanych dla komputerów konwencjonalnych.

Oprócz tego, w miarę zbliżania się do fizycznych granic obecnej architektury, poszukiwanie alternatywnych metod obliczeniowych przyczynia się również do badań i rozwoju tak zwanych komputerów kwantowych. Jednak wysiłki mające na celu zbudowanie odpornych na uszkodzenia systemów przetwarzania kwantowego zakończyły się skromnym sukcesem. Wdrażane obecnie urządzenia kwantowe są ograniczone skalą i hałaśliwe i znane są jako komputery kwantowe o średniej skali szumu (NISQ). W tym scenariuszu metody hybryd kwantowo-klasycznych, takie jak wariacyjne algorytmy kwantowe (VQA) i wyżarzanie kwantowe, okazały się kandydatami do wykonywania zadań optymalizacyjnych w tak ograniczonym scenariuszu. W szczególności możliwość zastosowania łagodzenia błędów, późniejszej selekcji i lepsze zrozumienie krajobrazów energetycznych może poprawić jakość wyników VQA dla urządzeń NISQ.

Poza tym można zadać sobie pytanie, co w danej chwili potrafi komputer kwantowy. Na przykład interesujące jest wiedzieć, czy komputer kwantowy jest gotowy do wykonywania zadań związanych ze sztuką i kreatywnością. Mówiąc dokładniej, poprzez adaptację ich do problemów optymalizacyjnych sformułowanych do rozwiązania w urządzeniach do wyżarzania.

W tej pracy zaproponowałem dyskusję na temat ulepszenia protokołów przyjaznych NISQ w podejściu opartym na bramkach. Przedstawiamy dwie metody: pierwsza to schemat łagodzenia błędów w wariacyjnych obwodach kwantowych poprzez postselekcję w obwodzie środkowym. Podejście to opiera się na badaniu prawidłowych podprzestrzeni uzyskanych za pomocą różnych kodowań, takich jak

Polish abstract

kodowanie one-hot, k -hot, kodowanie binarne i kodowanie ścian domenowych, które często pojawiają się w kodowaniu problemów optymalizacji kombinatorycznej oraz w chemii kwantowej. Druga to strategia heurystyczna, która wykorzystuje klasyczną optymalizację homotopii, ponieważ ma potencjalne zastosowanie w radzeniu sobie z funkcjami wysoce nieliniowymi. Strategia ta upraszcza wyszukiwanie dobrych parametrów QAOA, utrzymując jednocześnie PQC na niezmiennym poziomie. Aby zaprezentować to podejście, badamy ważony problem Max-Cut na wykresach Baraba-Asiego-Alberta. Na koniec pracowaliśmy także nad aplikacjami Zastosowania wyżarzania kwantowego w teorii muzyki. Rozważamy problem komponowania muzyki z różnych stron, m.in. kompozycji melodii i rytmu. W przypadku redukcji muzyki potraktowaliśmy problem jako wariant planowania zadań, w którym każda maszyna jest instrumentem, a zadania są frazami muzycznymi. Wykorzystując dostępne na rynku wyżarzacze kwantowe generujemy utwory muzyczne, które prezentowane są w trakcie pracy dyplomowej.

Abstract in English

In last decades, the scientific community witnessed the fast evolution of computers, resulting in powerful devices combined with sophisticated methods for computing difficult tasks, which allowed us to revolutionize the way we understand and process information. Alongside, the scientific community witness the emergence of a new paradigm: the Quantum Technology is a promising field that potentially provides computational advantages. In particular, the iconic Shor's and Grover algorithms, called the attention of computer science community with the theoretical possibility of building algorithms which are faster compared to the currently known for conventional computers.

Alongside, as we approach the physical limits of the current architecture, the search for alternatives computational methods also contributes to the research and development of the so called Quantum Computers. However, the efforts to build a fault tolerant quantum processing systems resulted in modest successes. The quantum devices implemented at the moment are limited by scale and noisy, been known as Noisy Intermediate Scale Quantum (NISQ) computers. In this scenario, quantum-classical hybrids methods, such as Variational Quantum Algorithms (VQAs) and Quantum Annealing, appeared as candidates to perform optimization tasks in such limited scenario. In particular, the possibility of applying error mitigation, post-selection, and a better understanding of the energy landscapes can enhance the quality of VQAs results for NISQ devices.

Besides, one can also asks what a quantum computer can do at moment. For instance, it is interesting to know if a quantum computer is ready to perform tasks related with art and creativity. More specific, by adapting them into optimization problems formulated to be solve into annealing devices.

In this thesis, I proposed a discussion about how to improve NISQ friendly protocols in gate based approach. We introduce two methods: the first one is a scheme for error mitigation in variational quantum circuits through mid-circuit post-selection. The approach is based on investigating valid subspaces obtained through different encodings such as one-hot, k -hot, binary, and domain-wall encoding that frequently appear in encoding combinatorial optimization problems and in quantum chemistry. The second is a heuristic strategy that uses classical homotopy

English abstract

optimization, since it has potential application in dealing with highly non-linear functions. This strategy simplifies the search for good QAOA parameters while keeping the PQC unchanged. To showcase this approach, we investigate the weighted Max-Cut problem on Barabási-Albert graphs. Finally, we also worked on applications Applications of Quantum Annealing for Music Theory. We consider the problem of music composition from various aspects, among them the composition of melody and rhythm. For music reduction, we treated the problem as a variant of job scheduling, where each machine is a instrument and the jobs are music phrases. Using the available commercial quantum annealers, we generate music pieces that are displayed in the course of the thesis.

Acknolegments

I am deeply grateful to my supervisor, dr hab. Jarosław Adam Miszczak, whose unwavering support and insightful guidance have been instrumental in shaping this doctoral thesis. His expertise, encouragement, and mentorship have been invaluable, and I am truly fortunate to have had the opportunity to work under his supervision.

I extend my heartfelt appreciation to my colleagues Akash Kundu, dr Adam Glos, dr Özlem Salehi, whose collaborative efforts, constructive feedback, and camaraderie have enriched this experience immensely. Akash is not a only my fellow Ph.D. colleague, but also an excellent friend. I would like to thank Adam and Özlem for the patience and guidance. I would like to thank Krzysztof Domino for sharing his knowledge and expertise in the railway dispatching field.

I am eternally grateful to my friends, Cavallo, Mateusz, and Karolina, for their constant encouragement, understanding, and for being a source of inspiration during the most challenging times.

I owe a profound debt of gratitude to my family: my beloved wife, Agata, for her unconditional love, patience, and unwavering belief in me; my brother, Abner, for his support and motivation; and my mother, Noely, whose sacrifices and blessings have paved the way for my achievements.

Without the collective contributions of these remarkable individuals, this endeavor would not have been possible. Their invaluable support and encouragement have been the driving force behind this accomplishment.

This project was partially supported by the Polish National Science Centre under the OPUS project 2019/33/B/ST6/02011.

English abstract

Chapter 1

Introduction

In last century, the scientific community witnessed the fast evolution of computers resulting in powerful devices and sophisticated methods for dealing with all sort of tasks. It is an undeniable fact that computers are indispensable in our lives at this point.

In a nutshell, a computer is a programmable device capable of carry on instructions, such as arithmetic or logical operations, to process information. Classical computers are the ones that can be modelled as a deterministic Turing Machine. Up to date, the most common classical computers are electronic digital devices that makes uses of bits as their most basic unit of information. Such devices accept data (input), process that data, produce output, and store the results. For instance, digital computers can read, compare and do bit manipulation very well, since their invention. It is the cornerstone of pretty much all the computation we perform nowadays.

However, it is important to remark that the history of modern computers is not limited to digital devices. In fact, through the 20th century, the scientific community observed the rise and fall of alternative computational architectures and algorithms. One could question the very nature of those machines in general: what are computers rather them glorified calculators?

I would like to point out an alternative and older type of classical computer which was one of most prominent method in past, the so called analogue computers. Initially built to perform very specific tasks, different from a universal computational model, the analogue computers are designed to process continuous data and perform mathematical operations using physical quantities, such as electrical signals, mechanical movements, hydraulics pressure, among them. They were widely used before the advent of digital computers and are still utilized in specific applications today¹. The analogue computers were typically much faster then the

¹Mechanical watches, analogue music synthesizers, and thermometers among them.

Chapter 1. Introduction

digital ones, although, less precise than the former ones. Therefore, the idea of building a *hybrid* computer, i.e., digital-analogue devices, was not uncommon: it appears from the need of solving problem with speed and accuracy.

Eventually, the digital computers evolved becoming considerably cheap, fast and powerful. This process culminated in “retiring” some of the analogue devices. Still, a digital computer is a *classical computer*: they are modelled as deterministic Turing device, which is good enough to perform a tantamount of tasks. However, due the nature of those devices – in both theoretical and physical ways – certain types (or classes) of problems are not efficiently solved in those machines.

Meantime, scientists managed to combine two of the most influential and revolutionary theories from the twentieth century: information theory and quantum mechanics. From this new science, the field of Quantum Computing bring to light the possibility of having algorithmic advantage over classical computation and revolutionize our ways to understand and process information. The most notable example is Shor’s polynomial quantum algorithm for finding the prime factors of a integer² in polynomial time [111]. Also, combinatorial problems, which is a important class of optimization problems in our daily lives, are one of the favourite topics to tackle with Quantum Optimization algorithms

The main difference between a classical and quantum computer is the fact that the later one is based on quantum mechanics phenomena, and make usage of quantum bits or *qubits* as their basic information unity. The qubit is distinguished from a classical due it superposition propriety, making the state of a qubit as convex combination of two probabilities amplitudes values. When measuring a qubit, the result is a probabilistic output of a classical bit. Note that, pretty much as the analogue devices, quantum computational approaches improve upon classical methods for specific a number of tasks.

Whereas the development of *quantum computers* show us very valuable theoretical approaches as new way to solve hard problems, the applicability of quantum computing still under development and research. It is not yet known what quantum computers are capable of, how the explore their full potential or even which task can outperform classical computers. Besides, the efforts to build a fault tolerant quantum processing systems resulted in modest successes. The quantum devices implemented at the moment are limited by scale and noisy, been known as Noisy Intermediate Scale Quantum (NISQ) computers. The possibility of building a fault-tolerant quantum computer with sufficient number of qubits is still an open question.

Once again, it comes as no surprise to approach hybrid computation by combining quantum and classical computers as an attempt to overcome the noise typical from quantum systems and the complexity constraints imposed to classical

²a problem know to be difficult and used as the bases for RSA encryption [76]

Chapter 1. Introduction

machines. In fact, giving instructions and reading the outputs of quantum device requires a classical computer as a mainframe. Alternatively, one can say that a quantum computer by the definition must be a *hybrid* computer. As Scott Aaronson says in his popular science blog [3], and I quote: “every quantum algorithm is a “hybrid quantum/classical algorithm” with classical processors used wherever they can be, and qubits used only where they must be”.

Up to this point, NISQ computation relies heavily in hybrid approaches, using both classical and quantum hardware to process information, and specific algorithms tailored for NISQ devices carry partially the some of the optimization tasks in the classical devices. For example, for some quantum annealing processes, the quantum part acts as guide for the classical heuristics.

Even though the quantum technology still in the very beginning and has many challenges to overcome, it inspire the research community towards improving classical algorithms, for example, quantum resilient or post-quantum protocols are now being developed, based on computational problem that are assumed too be hard to solve using quantum computers. One can say that it is also an opportunity to creating novel methods for art and creative applications.

The main idea of this thesis revolves two simple questions: How to improve the way we process information in quantum computers and what they ready to do? In a attempt to answer those questions, I worked in two different types of quantum computation methods: the gate based and quantum annealing. In both methods, my approach was to treat problems as optimizations, more specific, combinatorial optimization, such as job scheduling. For the gate based computation, my work was about improving the quality of the algorithms by either applying error mitigation schemes or homotopic optimization, in other to make a better search in the feasible space of the problems. For the quantum annealing, I allowed myself to explore the creative problems of music composition and and music reduction, adapting them into optimization problems formulated specifically to be solve into annealing devices.

The rest of the thesis is organized as follows. Chapter 2 initiates by giving a brief introduction to complexity, graphs, combinatorial problems, basics of quantum circuits, quantum annealing, and variational quantum algorithms. In Chapter 3 we discuss about error mitigation by using post-selection strategies to filter out wrong solutions of the travelling salesperson problem, diminishing the depth of the circuit and improving the quality of the results by making a better search in the subspace of the feasible solutions. In Chapter 4, we investigate a variant of the Quantum Approximation Optimization Algorithm using Homotopic optmization, in order to approximate the search into a homotopic path similar to quantum annealing. In Chapter 5, we discuss the usage of quantum annealing as a novel approach for music composition and music reduction, by formulating the problems as Quadratic

Chapter 1. Introduction

Unconstrained Binary Optimization. And finally, in Chapter 6, we summarize and conclude the thesis.

In the end, I found myself up with more questions than before. I hope you can enjoy the reading.

Chapter 2

Preliminaries

2.1 Complexity

One of our primary concerns is the amount of resources used by a computer to solve a problem, the so-called *complexity* of the computation. The most common resources analysed are *time*, i.e., the number of elementary operations that are executed during the computation, and *space* which is defined as the amount of memory required to solve the problem.

Measuring the complexity of an algorithms is not a trivial task. However, one can estimate in a convenient form by analysing its asymptotic behaviour when the algorithm run on large inputs. For this dissertation, we are going to use the *big-O notation*, a useful mathematical definition that assists in characterizing the complexity of an important class of computational problem.

The big-O notation can formalized in the following definition [113]:

Definition 1 *Let R^+ be the set of non-negative real numbers and f and g be functions $f, g : \mathcal{N} \rightarrow R^+$. Say that $f(n) = O(g(n))$ if positive integers c and n_0 exist such that for every integer $n \leq n_0$,*

$$f(n) \leq cg(n).$$

When $f(n) = O(g(n))$, we say that $g(n)$ is an upper bound for $f(n)$, or more precisely, that $g(n)$ is an asymptotic upper bound for $f(n)$, to emphasize that we are suppressing constant factors.

In that sense, a function $f(n)$ is $O(g(n))$ if its leading term grows as $g(n)$ or slower. For example, given a function $f(n) = 5n^3 - 2n^2 + n + 3$. The asymptotic upper bound of $f(n)$ is $O(n^3)$ by selecting the highest order term, $5n^3$, and suppressing the constant 5.

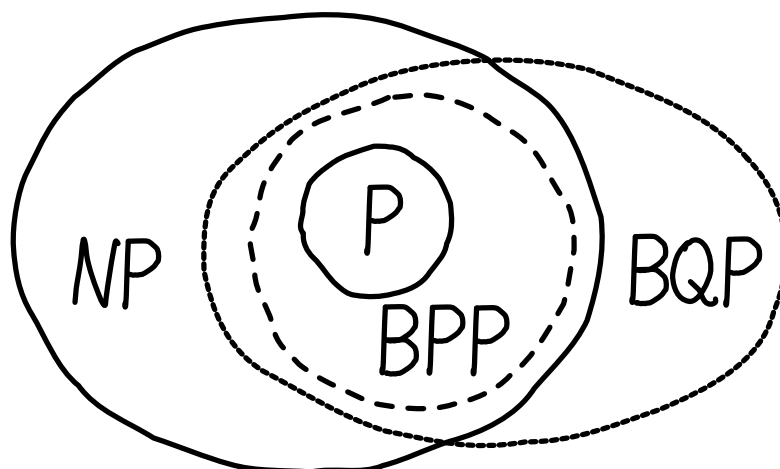


Figure 2.1: A simple representation of the hierarchy complexity classes, assuming that $P \neq NP$.

2.1.1 Complexity Classes

A complexity class is a set of functions that can be computed within a given resource. Using the big-O notation, algorithmic complexities are classified according to the type of function, usually expressed in terms of input size n in bits. This is important in order to measure the efficiency of a given algorithm. For our purposes, polynomial differences in running time are considered to be small, whereas *exponential* differences are considered to be large¹. For convenience, we restrict attention to ‘decision’ problems, where the answer is either ‘yes’ or ‘no’.

Let us define \mathbf{P} as the complexity class that contains all decision problems which are solvable on a *deterministic* classical computer running in polynomial time. This is generalized by the class \mathbf{NP} which contains decision problems whose *solution* can be checked efficiently by classical computers.

One can conceive of computational models that are non-deterministic, for example, by making use of random bits. In that sense, we can define a **bounded-error probabilistic polynomial-time (BPP)** class: the class decision problems solvable by a *probabilistic Turing machine* in polynomial time with an error probability bounded by $1/3$ for all instances.

If we consider the use quantum bits (qubits), we can define a **bounded-error quantum polynomial time (BQP)** class containing decision problems solvable by a quantum computer in polynomial time, with an error probability of at most $1/3$ for all instances [96]. In that sense, BQP is a quantum analogue of BPP. Note also that classical computation, whether deterministic or probabilistic, is a subcase

¹Asymptotically, exponentials grow faster than polynomials.

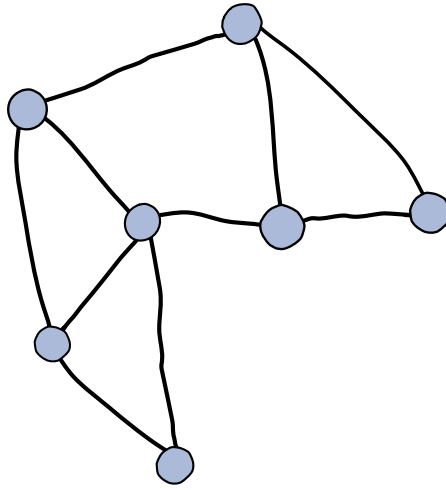


Figure 2.2: A simple graph.

of quantum computation. The hierarchy between classes is described in the Fig. 2.1.

The relationship between BQP and NP class is unknown. However, it is possible that the BQP class contains hard problems for classical computation, which appears to be incomparable to the standard complexity classes. In [2] is suggested that some problems instances existing in BQP could be classified as harder than NP-Complete problems. Paired with the fact that many practical BQP problems are suspected to exist outside of P, this illustrates the potential power of quantum computing compared to classical.

For this dissertation, we are going to take in consideration, for quantum gate-based models, the *circuit depth*, i.e., the number of gates used during the computation, to compute the complexity of quantum algorithms.

2.2 Basics of Graphs

Before describing the problems, we need to recall some aspects of graph theory. A *graph* is a collection of points and lines connecting some pairs of points. Formally, we can say that a graph is then a pair $G = (V, E)$ where V is a set of vertices (also called points or nodes) and E a set of edges, which is the set of pairs of vertices connect by a line. For example, in a graph G that contains nodes i and j , the pair (i, j) represents the edge that connects i and j , as shown in the figure.

In a *simple graph* or undirected graph, the order of i and j does not matter, differently of a *directed graph*, which has arrows instead of lines, as shown in the following figure. The formal description of a directed graph is a pair $\vec{G} = (V, \vec{E})$, where $\vec{E} \subseteq \{(x, y) \mid (x, y) \in V \text{ and } x \neq y\}$.

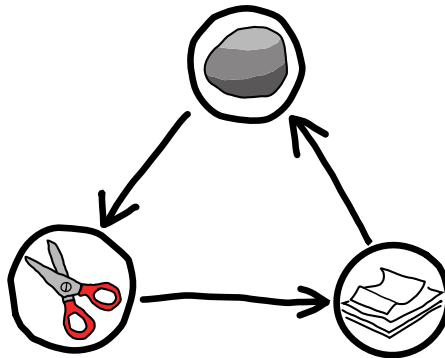


Figure 2.3: The Rock-Paper-Scissor as a direct graph representation. The arrows gives the direction of a move that beats the other.

A path in a graph is a sequence of nodes connected by edges. A simple path is a path that does not repeat any nodes. A graph is connected if every two nodes have a path between them. A path is a cycle if it starts and ends in the same node. A simple cycle is one that contains at least three nodes and repeats only the first and last nodes. When a path or cycle has each node visit only once, it is called Hamiltonian path or Hamiltonian cycle. Finding Hamiltonian paths or cycles in a graph are problems considered to belong to NP-Complete class [48].

2.3 Optimization Programming formulations

Let us now define some possible descriptions for the problems considered in this dissertation as mathematical optimizations, i.e., we select the best conditions to solve the problem given an goal, a set of possible solutions and sometimes, restrictions. In order words, an optimization problem consist in selecting an input such that it returns the maximal (or minimal) value of a real function.

2.3.1 Integer Linear Programming (ILP)

Linear optimization is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. A particular case of LP is the IntegerLinear Programming (ILP), defined over integer variables with a linear objective function and a set of

Chapter 2. Preliminaries

linear constraints. Formally, an ILP is defined as

$$\text{minimize } \sum_i c_i x_i \quad (2.1a)$$

$$\text{subject to } \forall_j \sum_i A_{ij} x_i \leq b_j, \quad (2.1b)$$

$$\forall_i x_i \geq 0, x_i \in \mathcal{Z} \quad (2.1c)$$

where $A_{ij} \in \mathbb{R}$, $b_i \in \mathbb{R}$, $c_j \in \mathbb{R}$ are parameters. Some of the problems are also formulated as Mixed-Integer Linear Programming (MILP), in which some of the decision variables are not constrained to be integer values. ILP are known to be NP-Hard.

2.3.2 Quadratic Unconstrained Binary Optimization

A Quadratic Unconstrained Binary Optimization problem (QUBO) is formally expressed by the optimization problem

$$\min x^T Q x, \quad (2.2)$$

where x is a vector of binary decision variables and Q is a square matrix of real coefficients. By definition, the QUBO model has no constraints other than the requirement for the variables to be binary. However, many combinatorial optimization problems often include additional constraints that must be satisfied besides an objective function to be minimized. Many of these constrained models, such as integer linear programs or integer quadratic programs, can be effectively re-formulated as a QUBO model by introducing quadratic penalties into the objective function as an alternative to explicitly imposing constraints in the classical sense [79, 108].

The significance of the ability of the QUBO model to encompass many problems in combinatorial optimization is enhanced by the fact that the QUBO model can be shown to be equivalent to the Ising model. The transformation between QUBO and Ising model can be performed easily using the mapping $x_i \leftrightarrow \frac{1-s_i}{2}$. QUBO formulations for many optimization problems are presented in [54, 79]. More generally, one can also formulate combinatorial problems with a Higher-Order Binary Optimization (HOBO) formulation.

2.4 Combinatorial Problems

For this dissertation, we study algorithms for combinatorial optimization problems. Combinatorial optimization is a subfield of mathematical optimization that consists of finding an optimal solution from a finite set of possible candidates. This is an essential problem present on the industrial context, arising in resource allocation tasks, scheduling, routing among other logistic problems, which can be hard to find an optimal solution due a large number of constrains and variables. Combinatorial optimization, therefore, has a direct impact in aiding real-world problems and improvement for decision-making processes.

Combinatorial optimization problems are also at the heart of classical theoretical computer science, where they are used to characterize and delineate complexity classes. Typical combinatorial optimization problems have limited structure to exploit, and therefore quantum computing most often good candidates to provide polynomial speed-ups.

We selected three combinatorial problems of interest to exploit both theoretical and applied aspects of quantum computation thought this dissertation, amongst them the Max-Cut problem, the Travelling Salesperson, and variants of Scheduling Problem. Those particular problems have in common the hardness in terms of complexity.

2.4.1 Max-Cut

Given a simple n -node graph $G = (V, E)$, a cut of G is as a partition of the vertices V into two subsets $A = (V_A, E_A)$ and $B = (V_B, E_B)$ such that $V_A \cap V_B = \emptyset$. The size of the cut is the number of edges connecting the two subsets. We can define as maximum cut as the cut of a graph G whose size is at least as large as any other cut. Finding the maximum cut is referred to as the *Max-Cut Problem* which is one of the simplest graph partitioning problems to conceptualize, and yet some variations of this problem are considered difficult. The Max-Cut problem is considered NP-Hard and the weighted version of the decision problem variant was included in the Karp's 21 NP-complete problems [68].

In the weighted version of Max-Cut, we are given an edge weight function where each edge is associated with a real number, $w : E \rightarrow R$, and the problem is to find a cut with maximum weight. We can represent the assignment of vertices to set V_A or V_B using a bit-string, $\mathbf{x} = x_1 \dots x_n$ where $x_i = 0$ if the i -th vertex is in V_A and $x_i = 1$ if it is in V_B . We seek a partition \mathbf{x} which maximizes the following cost function

$$\tilde{C}(\mathbf{x}) = \sum_{i,j} w_{ij} x_i (1 - x_j).$$

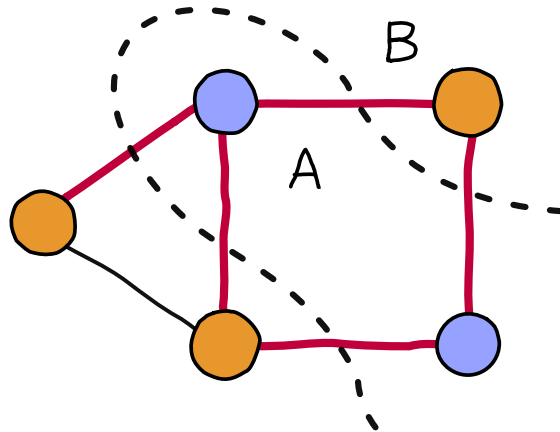


Figure 2.4: A Max-Cut solution for the following graph. Note that this is equivalent of creating a path selecting the maximum number of edges without touching any edge twice.

Note that this can be also written in terms of a Hamiltonian:

$$H_C = \frac{1}{2} \sum_{\langle i,j \rangle} w_{ij}(1 - Z_i Z_j).$$

2.4.2 Travelling Salesperson

The Travelling Salesperson Problem (TSP) starts with a simple statement: given a list of places and the cost for travelling from one location to another, is there a permutation of the places such that the person can visit all with minimal distance travelled? Considered one of the most iconic problem among theoretical computer scientist, the TSP was mathematically formalized around 1930 and has become one of the most employed benchmarking problems in performance analysis of discrete optimization algorithms.

Similar to the case of Max-Cut, the TSP can also be described in terms of graphs. Given a graph $G = (V, E)$, where each edge uv in the graph has a weight W_{uv} associated to it, the goal is to find the shortest Hamiltonian cycle or Hamiltonian path, using all the nodes in the graph, such that the sum of the weights of the edges is minimized. The decision version of the problem, i.e., if there is a path whose total weight is smaller than a threshold value, is NP-complete.

While the problem poses significant computational challenges, numerous heuristic and exact algorithmic approaches have been developed. These approaches enable complete solutions for instances involving tens of thousands of cities. Additionally, for problems involving millions of cities, remarkably accurate approximations [129]. However, it is not known the existence of a classical algorithm that provides a

polynomial time solution to TSP [128], and is possible not expected to exist.

2.4.3 Job Scheduling Problems

Optimal job scheduling is a class of optimization problems that deal with how to appropriately and efficiently assign resources to tasks. Examples for resources can range from machines, processors, people, and rooms, which are available with some limitations. The problems input are the jobs, also called task or activities. The main purposes of job scheduling can be set to shorten the job completion time or optimize resource utilization, by minimizing an objective function. Since many of the scheduling problems difficult to solve, the optimization process often involves complex algorithms and advanced computational techniques to find the most effective scheduling solutions [23].

Considering a specific variant known as job-shop scheduling (JSP), each job consists of a set of operations that must be processed in a specific order, known as precedence constraints. Each operation has a specific machine, which process a single job each time. In the case of a set of identical machines, problem is reduced to a variant known as flexible job shop, where each operation can be processed on any machine of the set. The TSP is a special case of the JSP with a single job, i.e., the cities are the machines and the salesman is the job.

Another variant of the scheduling problems consists in which each job has a fixed start and end time and a value. For given a set of identical machines, the problem can be represented as a coloring interval graphs [55] and it was formulated as binary integer linear programming problem in [9] as

Something is missng here!

2.5 Simulated Annealing

Some of the problems that we studied, along with several problems arising in practice, are difficult, it is unlikely that we can design exact efficient algorithms for them. Therefore, one can alternatively opt for approximate the solution. For instance, a naive approach would be solving the problem by making the locally optimal choice at each stage, know as *greedy algorithm*. However, it does not guaranty an optimal global solution [43].

Other valid approaches to deal with the computational hardness of large combinatorial optimization problems are local meta-heuristic searches, such as Simulated Annealing and Tabu Search, Genetic Algorithms, and Variable Neighbourhood Search. For this dissertation, we approached the Simulated Annealing method for comparison with its quantum version.

Originally introduced by [72] to solve the Travelling Salesman Problem, Simulated Annealing (SA) works by emulating the process of annealing a solid by slowly

lowering the temperature so that when eventually its structure is “frozen”, into a minimum energy configuration. This method can be regarded as a random walk on the search space. The algorithm starts at an initial state x having cost $c = f(x)$ and then iterates to walk through the problem landscape. The unconstrained cost function can be achieved from the original problem in Eq. (2.1). At each iteration k , it selects a random neighbour, which is accepted if it has a lower cost, becoming the current solution c_{new} . If the new solution has a higher cost, it is accepted with a probability determined by the temperature parameter t_k and the difference between the existing and the new costs. The acceptance probability function is usually defined as:

$$P[Accept(t_k, \Delta)] = \min(e^{-\Delta/t_k}), \quad (2.3)$$

where $\Delta = c_{new} - c$. As the cooling process is carried out, t_k is decremented, and the optimal solution is found with the help of thermal fluctuations. The SA is a meta-heuristic method and can be viewed as an adaptation of Metropolis-Hastings algorithm. We refer readers to [75] for a more detailed survey on SA. Simulated annealing has been used to solve various problems including scheduling problems [99, 106, 123], Travelling Salesman Problem and its variants [4, 28], graph colouring [29], and quadratic assignment problem [24].

2.6 The Ising Model

The Ising model is a mathematical model in statistical mechanics that is used to study phase transitions and critical phenomena in physical systems, particularly in the context of magnetism. It was first introduced by the German physicist Ernst Ising in his 1925 doctoral thesis [65]. The problem can be seen as a collection of N particles arranged on the vertices of a graph which is often assumed to be a d -dimensional grid. Each particle can be in one of two states, called spins, represented by a variable $s_i \in \{-1, 1\}$.

The original motivation for the model was the phenomenon of ferromagnetism. The Ising model provides a simplified description of a ferromagnetic material, where the fundamental units are individual magnetic moments or spins. These spins can be thought of as small atomic-scale magnets that can point in one of two directions: up or down. The model is often formulated on a regular lattice, such as a square or a cubic form, where each lattice site hosts a spin. The problem is later generalized in terms of characterization of spin configurations given an external field applied to the system.

The Ising model can be formulated in terms of an undirected graph $G = (V, E)$ with N nodes. Each node $i \in V$ is associated to a spin variable s_i . On each edge $\{i, j\} \in E$, an interaction strength $J_{i,j}$ is assigned. Also, each particle is under the influence of a local magnetic field h_i . Given a spin configuration $\mathbf{s} = (s_1, \dots, s_N)$,

The energy of a given configuration is defined by the following Hamiltonian:

$$H(\mathbf{s}) = \sum h_i s_i + \sum_{i \neq j} J_{ij} s_i s_j, \text{ where } J_{i,j}, h_i \in \mathbb{R}. \quad (2.4)$$

Despite the simple formulation, finding the ground state, i.e., a spin configuration \mathbf{s} that minimizes the Hamiltonian H above can be computationally hard [12]. For NP-Complete problems, it is easy to find a mapping to the decision form of the Ising model. In fact, the Ising model is closely related to two problems on binary variables. These latter formulations are sometimes more convenient to work with, partly because heuristic algorithms are typically implemented to return binary-valued solutions.

2.7 Basics of Quantum Circuits

Quantum circuits are a way to visually depict the sequence of operations that are performed on qubits throughout the course of a computation. You can think of quantum circuits like a recipe, or set of instructions that tells you what to do to each qubit, and when to do it. By placing and performing the operations in a certain way, we can realize different quantum algorithms. In the next subsections we will briefly describe the components of a quantum circuit.

2.7.1 The qubit

The basic unit of information in a quantum circuit is a qubit. Mathematically, the qubit is a state represented by a bidimensional normalized vector in the Hilbert $\mathcal{H} = \mathbb{C}^2$ space. Using the Dirac's notation, the vectors $|0\rangle$ and $|1\rangle$ form a basis in the \mathcal{H} , the so called *computational basis*, defined as the following column vectors:

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (2.5)$$

A general noiseless (pure) state $|\psi\rangle \in \mathcal{H}$ is a linear combination of basis states, and can be written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where } \alpha, \beta \in \mathbb{C} \text{ and } |\alpha|^2 + |\beta|^2 = 1. \quad (2.6)$$

The complex terms α and β are also called *probability amplitudes*. As a consequence of this mathematical description, unlike a classical bit, a qubit is not limited to being defined in terms of a single deterministic state. Instead, qubits can exist as a *superposition* of states. This property is the essence of practically most of quantum

Chapter 2. Preliminaries

computational methods. It is important to emphasise that it a misconception that a qubit in a superposition of two states is in both states at the same time: this is false because the description refers to the probability of observing the state in certain configuration when it is *measured*.

However, it is also necessary to have a more generalized representation, especially when dealing with non-pure or *mixed state*: an *ensemble* of states $\{p_x, |\psi_x\rangle\}$, which describes the statistical uncertainty by considering a weighted mixture of the states in a set $\{|\psi_x\rangle\}$ where corresponding probability of each state $|\psi_x\rangle$ being selected is $p_x \in [0, 1]$. The density operator corresponding to this ensemble is given by the convex combination

$$\rho = \sum p_x |\psi_x\rangle\langle\psi_x|. \quad (2.7)$$

The operator ρ as defined above is known as the density operator because it is the quantum generalization of a probability density function.

To perform a computational tasks, usually more qubits together are used. The Hilbert space for a system composed of multiple qubits is built considering the tensor product of the single-qubit Hilbert spaces. As an example, the two-qubit system lives in the composed Hilbert space $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 = \mathbb{C}^4$. The computational basis is combining, resulting all the possible combinations of $|0\rangle$ and $|1\rangle$: $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

2.7.2 Unitary operators

To manipulate the qubits, it is important that the operation must preserve the normalization of the state. In that case, such operation are described as an unitary transformation $U : \mathcal{H} \rightarrow \mathcal{H}$ and $UU^\dagger = I$. Operations on qubits are *reversible*, and often called *gates*. For example, the Hadamard gate, maps the state in the computation basis to the $+/-$ basis:

$$H|0\rangle = |+\rangle, \quad H|1\rangle = |-\rangle, \quad (2.8)$$

where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. An important class of two-qubit unitary operations are the controlled gates. For example, the CNOT gate, is a controlled NOT gate, where value of the target qubit flips according to the value of the control qubit. Given a two qubit state $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$, where the first qubit is the control and the second one is the target, we have:

$$CNOT|\psi\rangle = CNOT(a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle) \quad (2.9)$$

$$= a|00\rangle + b|01\rangle + c|11\rangle + d|10\rangle. \quad (2.10)$$


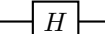
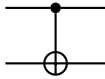
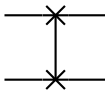
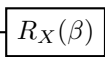
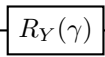
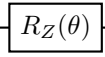
Operator	Gate
X	
H	
CNOT	
SWAP	
Rx	
Ry	
Rz	

Table 2.1: Some examples of quantum gates and their representation in a circuit.

Therefore, when the first qubit state is $|1\rangle$, the action of the CNOT is equivalent to the classical NOT gate in the second qubit.

When dealing with multi-qubit systems, the tensor product formalism also applies to unitary operators. It is important to emphasise that it can be proven that a n -qubit unitary can be decomposed as a product of just single-qubit gates and two-qubit controlled gates. Therefore, a small set of the quantum operators can define the so called universal gate set, which are sufficient to implement arbitrary multi-qubit computation [110].

Some examples of quantum gates with circuit are available on the Table 2.1.

2.7.3 Measurement process

Finally, once the qubits are prepared and manipulated through operations, the measurement is the process of extracting the information about systems. This is done by measuring observables: a set of physical quantities which are variables such as position or momentum of a particle. In the quantum theory, we represent observables as Hermitian operators in part because their eigenvalues are real numbers.

For instance, given a pure state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, suppose that we measure the Z operator. This measurement is also called a “measurement in the computational basis”, because eigenvalues of the Z operator are the elements of the computational basis. The measurement postulate of the quantum theory, also known as the *Born’s*

rule, states that the system reduces to the state $|0\rangle$ with probability $|\alpha|^2$ and reduces to the state $|1\rangle$ with probability $|\beta|^2$.

Expressing the measurement outcome probabilities can be formalized by taking the inner product. Given a quantum state $|\psi\rangle$, the probability that we observe it in state $|\varphi\rangle$ when we measure it with respect to a basis that includes $|\varphi\rangle$ is equal to

$$\text{pr}(\varphi) = |\langle\varphi|\psi\rangle|^2. \quad (2.11)$$

A measurement in quantum computing is probabilistic, therefore, in order to get a clearer picture of the value of an observable for a given qubit state, we need to compute its *expectation value*. The expectation value of an observable O on a quantum system described by a noiseless state $|\psi\rangle$ is

$$\langle O \rangle = \langle\psi|O|\psi\rangle. \quad (2.12)$$

More generally, we can also define the expectation value using the density matrix representation and the trace operation:

$$\langle O \rangle = \text{Tr}[O\rho]. \quad (2.13)$$

2.8 Quantum Optimization Algorithms in NISQ era

The current state of quantum computing is referred to as the Noisy Intermediate-Scale Quantum (NISQ) era. Currently quantum devices are small, i.e., with no more than hundreds to thousands of qubits with considerable noise. Furthermore, it is believed that the new generation of quantum devices will be larger and less noisy, up to such a point that they will outperform classical algorithms and heuristics. Besides, a huge effort is underway for using these devices for real or realistic problems, which include quantum simulation, optimization, and machine learning. In particular, optimization algorithms are expected to be among the first to provide a quantum advantage.

This emerging paradigm relies on harnessing the power of quantum computation with hybrid quantum-classical approaches. Such algorithms delegate the classically difficult parts of the computation to the quantum computer and perform the remaining on a sufficiently powerful classical device. On gate-based quantum computers, the Variational Quantum Algorithms (VQAs), as the name suggests, update variationally the variables of a parametrized quantum circuit. Among them is the Quantum Approximate Optimization Algorithm (QAOA). Other quantum computing paradigms propose different kinds of algorithms. They are inspired and

hybridized with analogue approaches. These include quantum annealing, digital-analogue quantum computation, Gaussian Boson Sampling and analogue quantum computation.

For combinatorial optimization problems, the two leading algorithms are Quantum Annealing and QAOA, both motivated by the quantum adiabatic theorem. In this section, we will cover the recent applications for Quantum Annealing, QAOA, and also Quantum Inspired Algorithms.

2.8.1 Adiabatic Quantum Computation

In the Adiabatic Quantum Computing, the computation proceeds from an initial Hamiltonian whose ground state is easy to prepare, to a final Hamiltonian whose ground state encodes the solution to the computational problem. The adiabatic theorem guarantees that the system will track the instantaneous ground state provided the Hamiltonian varies sufficiently slowly. In the following next sections, it will be described two methods based and inspired on the quantum adiabatic theorem.

2.8.2 Quantum Annealing

Quantum annealing [6, 66] is a heuristic algorithm that runs in the framework of adiabatic quantum computing (AQC) [42], targeting optimization problems. In AQC, a system starting in the lowest energy state (ground state) of some initial Hamiltonian H_0 (a mathematical operator that describes the system's energy) is likely to stay in the ground state throughout the evolution, given that the system is evolved slowly enough. Hence, if some problem Hamiltonian H_P is introduced gradually to the system, it is likely that the system ends up in the ground state of H_P at the end of the evolution time T . Mathematically, the evolution of the system is described by the time-dependent Hamiltonian

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P. \quad (2.14)$$

Note that quantum annealing is a physical process in an analogue quantum device, as opposed to simulated annealing, and exploits quantum phenomena like tunnelling and superposition.

The quantum annealers provided by D-Wave implement a problem Hamiltonian whose energy is expressed by an *Ising model* of the form

$$E(\mathbf{s}) = \sum_i h_i s_i + \sum_{i < j} J_{ij} s_i s_j, \quad (2.15)$$

where \mathbf{s} is a spin configuration of variables $s_i \in \{-1, 1\}$. Thus, one can use quantum annealing to solve optimization tasks expressed in terms of an Ising model or equivalently in the form of QUBO since the transformation between the two can be easily accomplished. Note that finding the minimum energy configuration of an Ising model is known to be NP-Hard.

Besides, the quality of the QA solutions is heavily affected by embedding, which is the process of mapping a specific optimization problem onto the physical qubits and couplers of a quantum annealer. Quantum annealers have a limited number of qubits and a specific connectivity structure, therefore, the connectivity graph of the currently-available quantum annealers is limited.

Quantum annealing has been used to solve optimization problems from different domains, including transportation [38, 95, 108], automotive [52, 130], and scheduling [36, 64, 124]. Recently, it has been also used in the scope of music theory for composing music [10].

The speed-up provided by QA for such problems is under a scientific debate and not evident [18, 59, 83]. Furthermore, while available D-Wave's annealers have thousands of qubits, the topology restrictions may limit the size of tractable problems to cases solvable by classical procedures.

2.8.3 Variational Quantum Algorithms

The Variational Quantum Algorithms (VQAs) are a set of hybrid classical-quantum methods which use a classical optimizer to train a Parameterized Quantum Circuit (PQC). The VQAs have emerged as a leading strategy to address the constraints imposed by current gate-based quantum devices, which include noise and a small number of qubits.

The methods consists in approaching the problem solution by encoded it as the minimum of a cost function which depends on variational parameters. Those parameters are iteratively tuned, usually via gradient-based methods, to find the minimum of the function, hence the solution.

A combinatorial problem can be modeled as a QUBO and consequently as Ising Hamiltonian. The cost function H_P , therefore, can take the shape of the 2-local Ising Hamiltonian. Besides, Higher Order Binary Optimization (HOBO) formulations can also be solved natively using QAOA or VQE as suggested in [41, 51, 108].

The *Variational Quantum Eigensolver* (VQE), introduced by Peruzzo et al. [105], is a technique used for finding the quantum state corresponding to the minimum energy of a Hamiltonian. The expectation value of the Hamiltonian H_P in a state $|\psi(\theta)\rangle$ can be expressed as

$$E(\theta) = \langle \psi(\theta) | H_P | \psi(\theta) \rangle. \quad (2.16)$$

The objective of this method is to search for a trial qubit state of a wave function $|\psi(\theta)\rangle$, which depends on a set of variational parameters, such that the expectation value of the Hamiltonian is minimized.

In order to translate this minimization task into a problem that can be executed on a quantum gate computer, one must start by defining a so-called ansatz wavefunction that can be implemented on a quantum device as a series of quantum gates. From this ansatz or parametrized quantum circuit, which is corresponding to an initial ground state of the system, the target Hamiltonian ground state may be obtained by iterative minimization of the cost function. Therefore, the Hamiltonian should be written in a form that is directly measurable on a quantum computer, as a weighted sum of spin operators or Pauli operators [119].

The optimization is carried out by a classical optimizer which leverages a quantum computer to evaluate the cost function and calculate its gradient at each optimization step. The quantum circuit is optimized using classical procedures like gradient descent or simultaneous perturbation stochastic approximation (SPSA). Due to its generality, VQE is commonly used for molecule Hamiltonians, however, its usability for combinatorial optimization problems may be limited.

Hence, given a properly defined cost function, variational algorithms then proceed by combining quantum and classical resources in an iterative loop as follows:

- On the quantum computer: estimate the cost function for the current values of the parameters via repeated measurements;
- On the classical computer: input the outcome in a classical optimisation algorithm that proposes a new value for the parameters so that the cost is lower;
- repeat previous steps until stop conditions are met (convergence, execution time, etc.).

2.8.4 Hamiltonian Simulation

Given a graph $G(V, E)$, we can design a Hamiltonian so that the ground state gives us an optimal solution to problem we would like to solve on the graph. We can associate each node in V to a qubit, the edges E to interactions. Since the evolution in a quantum circuit takes form of unitary operations, assuming that the Hamiltonian is time independent, we can describe the unitary

$$U = e^{-itH/\hbar}. \quad (2.17)$$

However, designing a quantum gate corresponding to such unitary can be complicated. In that sense, it can be approximated by expansion as series.

Trotter-Suzuki formulation Also known as Trotter formulas or Trotter–Suzuki decompositions is a product expression to simulate the sum-of-terms of a Hamiltonian by simulating each one separately for a small time slice. This can be better understood by taking as example a Hamiltonian that can be written as $-iH/\hbar = A + B$. By using the Baker-Campbell-Hausdorf expansion, we have:

$$e^{t(A+B)} = e^{tA}e^{tB}e^{-\frac{t^2}{2}[A,B]}e^{\frac{t^3}{6}(2[B,[A,B]]+[A,[A,B]])}e^{-\frac{t^4}{24}([[[A,B],A],A]+3[[A,B],A],B)+3[[A,B],B],B)} \dots \quad (2.18)$$

This equation can be truncated, giving the following approximation:

$$e^{t(A+B)} = (e^{At/n}e^{Bt/n})^n + O\left(\frac{1}{n}\right). \quad (2.19)$$

If we consider $H = H_0 + H_1$ and making the proper substitutions, we have

$$U(t) = (e^{-itH_0/n\hbar}e^{-itH_1/n\hbar})^n + O\left(\frac{1}{n}\right). \quad (2.20)$$

By breaking down the evolution into these smaller steps and applying each operator sequentially, you can approximate the time evolution of the quantum system. The accuracy of the approximation depends on the size of the time steps and the complexity of the system’s Hamiltonian.

2.8.5 Quantum Approximation Optimization Algorithms

QAOA is a hybrid quantum/classical introduced by Farhi et al. [41], which combines the idea of the Variational Quantum Eigensolver (VQE) and Quantum Annealing. The approximation is done by constructing a specific variational ansatz through first-order Suzuki-Trotter decomposition approximating adiabatic evolution, and it is designed for the quantum gate-based model. Unlike Quantum Annealing, QAOA can be also used for problems expressed as HOBQ. The algorithm leads to a state prepared by applying a mixer operator $\exp(H_0)$ and a cost (or problem) operator $\exp(H_P)$ applied in alternation, resulting in the state as alternating blocks of the two unitaries applied p times such that

$$|\psi(\beta, \gamma)\rangle = \underbrace{U(\beta)U(\gamma) \cdots U(\beta)U(\gamma)}_{p \text{ times}} |\psi_0\rangle, \quad (2.21)$$

where $|\psi_0\rangle$ is a initial state which is a eigenstate of the mixer Hamiltonian.

Due its compatibility with NISQ devices, QAOA is been used to investigate combinatorial problems and different variants were developed. For instance, the Quantum Alternating Optimization Ansatz constrains the search space in accor-

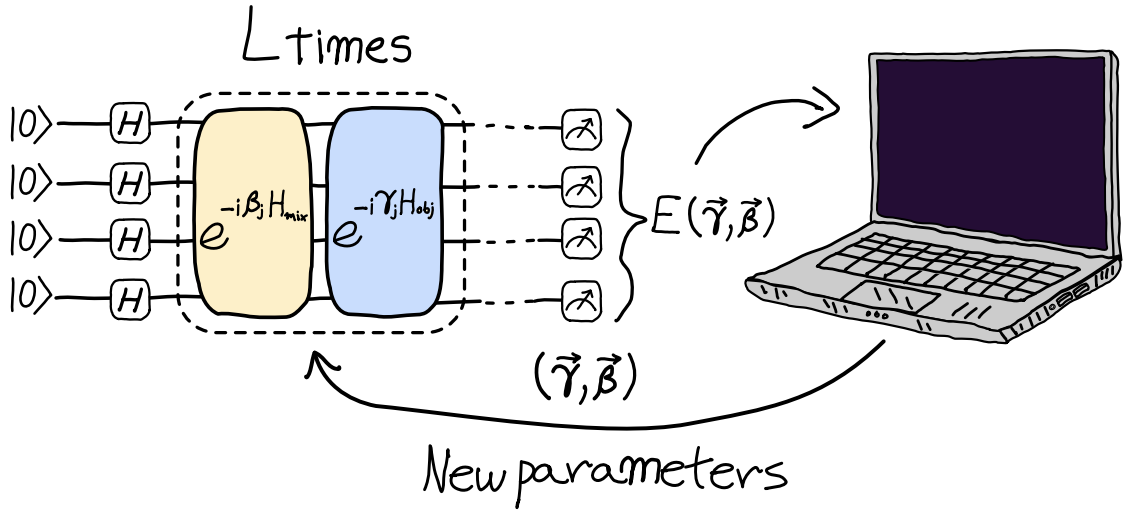


Figure 2.5: Schematic representation of QAOA. On the left is presented a parametrized quantum circuit. A initial state is prepared as the eigenstate of the mixer Hamiltonian H_0 which is equal to a superposition of the computational basis $|+\rangle^{\otimes N}$. The state is evolved by applying alternating $e^{-i\gamma_i H_{mix}}$ and $e^{-i\gamma_i H_{obj}}$. Then, on the right, the result is passed to a classical optimizer, that finds new values for the parameters $\vec{\gamma}, \vec{\beta}$ that minimizes $E(\vec{\gamma}, \vec{\beta}) = \langle H_{obj} \rangle$. The loop is repeated until convergence to a minimum of the cost is reached or stopping criteria are met.

dance with the problem constraints, which are embedded in the initialization and mixer layers. In the original version, constraints are expressed as penalty terms in the objective function and the search is performed on the entire Hilbert space.

However, for both VQAs and QAOAs, due to the nature of the NISQ devices, the quality of the results is presumably reduced with the impact of decoherence. Moreover, it is unlikely to utilize quantum error correction methods with VQAs to overcome the effect of noise due to a large number of qubit requirement. Yet, there are various quantum error mitigation (QEM) techniques suitable for the NISQ era. An extension of QAOA to deal with more general classes of Hamiltonians has been proposed with the name of Quantum Alternating Operator Ansatz [14, 57, 125]. This approach allows us to perform the search into a smaller subspace of the feasible solution.

2.9 How to compute and evaluate quantum algorithms?

As a newborn technology and the close relation with classical computers and lack of large and commercial quantum computers, we need to establish proper tools to design, simulate and analyse quantum algorithms. On this section, we will describe the computational resources we used for this thesis and methods for computing the hybrid-quantum algorithms.

2.9.1 Meta-programming

In general, one can say that meta programming is a umbrella term for techniques that work by manipulating and transforming other programs. For example, using Python’s operator overloading is a basic kind of meta-programming, but many other common techniques such as templates/generics, macros, and code generation broadly fall under the term meta-programming.

Since it is not yet available a way to direct write quantum programs, most of the work done is carried by humans and classical computers in order to map the information as data that can be encoded into quantum devices². Besides, reading the data also requires a classical computer to read the measurement outputs and display it into a coherent format.

Due the most of the popular packages and libraries target for quantum information and quantum computation been available for Python programming language, such as QuTip, Qiskit, Tikket, PennyLane, among them, we opted for designed most of the quantum circuits using Python as a *meta-programming* framework. We also conducted partially the circuit simulation and gradient computation with Julia.

2.9.2 Gradient descent and gradient-free methods for classical optimization

In mathematics, gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. The idea is to take repeated steps in the opposite direction of the gradient of the function at the current point in order to go towards the steepest descent. Conversely, stepping in the direction of the gradient will lead to a local maximum of that function; the procedure is then known

²Note that is also very common to write a quantum algorithm with an *agnostic* approach, i.e., the algorithm is written without taking in consideration physical aspects of the hardware, such as number of gates or qubits, constraints about the type gates or even how to deal with the interaction between the qubits.

as gradient ascent. It is particularly useful for training parametrized quantum circuits, such as VQE and QAOA. Gradient descent should not be confused with local search algorithms, although both are iterative methods for optimization. On this thesis, we used the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm, available on SciPy and Optim.jl.

As an gradient-free alternative, we also considered the constrained optimization by linear approximation (COBYLA), which is an implementation of Powell’s nonlinear derivative-free constrained optimization.

2.9.3 Input/Output

To conclude this section, I would like to briefly describe the sampling process and how to prepare the input of the quantum algorithms described in this thesis. The process starts by sampling the cost function and translating into an Ising Hamiltonian. This is important to cover as much as possible the capacities of the algorithm exploration of different feasible spaces. For quantum annealing, it is sufficient to provide a dictionary where the keys are the QUBO terms and the values are their coefficients. Therefore, the user should generate a QUBO cost function mapping their optimization task. For VQE and variants, the cost Hamiltonian is usually represented as a hash table or dictionary, which maps the terms written as Pauli operators strings and their correspondent float coefficients. The same is true for the mixer Hamiltonian. The parameters $\vec{\gamma}$ and $\vec{\beta}$ can be passed as arrays of floats. The data is used to construct a quantum circuit by using a framework of choice and it is translated into a quantum assembly (QASM) language to communicate with a quantum device. Due the probabilistic nature, the measurement in real quantum devices is perform a number of shots which can be defined by the user, in order to obtain valid statistical results. In simulations, one can also considered the expectation values. Finally, the output of the measurements, can be returned as floats and used to perform classical numerical optimizations to obtain new arrays of $\vec{\gamma}$ and $\vec{\beta}$.

Chapter 3

Error Mitigation for QAOA using post-selection

In this chapter¹, we will discuss about error mitigation via post-selection and by using different encodings schemes for a variation of QAOA.

3.1 Introduction

The Variational Quantum Algorithms (VQAs) are known to be resilient against coherent errors due to their variational nature [87,101]. However, as any other quantum algorithm running in NISQ devices, the quality of their results is presumably reduced with the impact of decoherence. In that sense, it is important to enfacise the usage of error correction hardness the power of the quantum computation and eventually achieve fault-tolerant devices, such that even though part of the qubits experience errors, the system will still return accurate answers. Error correction is a standard technique in classical computing where information is encoded with redundancy so that checks can be made on whether an error has occurred. However, considering the current paradigm of NISQ devices, it is unlikely that VQAs would be enhanced by error correction methods up to this point. In the meantime, one can consider applying quantum error mitigation (QEM) techniques suitable for the NISQ era [16,40]. Note that, QEM aims not to recover the ideal quantum state. Instead, the approach uses post-processing to reach the ideal measurement outcome.

Let us consider the readout error caused by the imperfect measurement devices, which is one of the sources limitations the current quantum hardware. Instead of obtaining the correct probability distribution from the measurement p , the outcome

¹The content of this chapter is based on the author's work [20] and all the figures in this chapter are taken from, or are adaptations of, the figures present in such work.

Chapter 3. Error Mitigation for QAOA using post-selection

becomes a stochastically malformed distribution $S \cdot p$, where S is a stochastic matrix. Many proposed works [22, 49, 80] focus either on building different noise models or schemes to mitigate the noise, for instance, by applying a pseudo-inverse $S^{-1}S \cdot p$ with classical post-processing. Even though the measurement error mitigation is not always sufficient, due the nature of evolution of quantum circuit in a noisy environment, one can be interested in the construction of the algorithm such that the evolution takes place only on a subspace of the full n -dimensional Hilbert space, and measurement outcomes can be classified as valid or not. As invalid outcomes appear due to the effect of noise, once they are removed it could potentially improve the overall fidelity of the measurement statistics.

More specific, in the case of VQAs, such evolution in a subspace of the whole Hilbert space make it is possible to mark many of the measurement outcomes as invalid and removed, as in the case of Variational Quantum Eigensolver [105] and Quantum Alternating Operator Ansatz (QAOA+) [14, 57, 125]. The idea of post-selection performed in the middle of the circuit based on valid states is proposed in previous works with many different approaches.

In the previous studies, it was observed that various valid subspaces appear as a result of the selected encoding scheme when dealing with VQE and QAOA. One the approaches is the binary encoding, used to represent integers and it was recently used to obtain qubit-saving formulations as Travelling Salesman Problem (TSP) [51], graph colouring problem [115], quadratic Knapsack problem [116] and Max k -Cut problem [45]. Another promising approach is the one-hot encoding. Although the already proposed schemes work for one-hot states as well, whether one can further exploit the special property of those states to obtain more efficient error mitigation schemes is unknown. Finally, it is also worth mentioning the domain-wall encoding presented in [30], and the authors provide special mixers preserving the valid subspace of quantum states for QAOA.

For the methods to be NISQ-friendly, they should use as few resources as possible. The resources usually considered are the number of ancilla qubits, the number of gates, and the depth of the circuit. These three, together with the volume, will be considered our main resources for this chapter.

On this chapter, it will be describe a proposed scheme for error mitigation in variational quantum circuits through mid-circuit post-selection. The post-selection is performed by injecting a quantum circuit consisting of both gates and measurements. Our approach is based on investigating valid subspaces obtained through different encodings such as one-hot, k -hot, binary, and domain-wall encoding that frequently appear in encoding combinatorial optimization problems and in quantum chemistry. In particular, the scheme we propose for one-hot encoding works by compressing the valid subspace to the smaller subspace of quantum states and differentiates from the known methods. We also demonstrate the effectiveness of our

approach with an application to QAOA+ for TSP. The proposed error mitigation schemes are suitable, but not limited to NISQ algorithms in principle. Furthermore, they can be currently employed with mid-circuit measurements, recently provided by quantum computers developed by IBM [93] and Honeywell [1].

3.2 Error Mitigation

To fully understand the method described in this thesis, let us discuss briefly about effects of noise on quantum circuits and how error can be mitigated through post-selection performed in the middle of the circuit.

Error mitigation approaches use the outputs of circuit ensembles to reduce or eliminate the effect of noise in estimating expectation values. While fault tolerance quantum devices are still not available, error mitigation is the path that gets quantum computing to usefulness. A simple approach to mitigate the noise is through post-selection: this technique refers to the process of conditioning on the outcome of a measurement on the qubits. Post-selection has been featured in plenty of quantum mechanics experiments, once it facilitates way to a quantum control of the system.

Let U be a quantum circuit with n qubits and initial state $|\psi_0\rangle$. This initial state in the circuit will evolve as $|\psi\rangle = U|\psi_0\rangle$, and it belongs to the subspace spanned by a particular subset $S \subset \{0, 1\}^n$. The final state can be expressed as

$$|\psi\rangle = \sum_{s \in S} \alpha_s |s\rangle. \quad (3.1)$$

Due the effects of noise, instead of a pure quantum state $|\psi\rangle$, it will be returned a mixed state ρ spanned by the whole Hilbert space. The objective of quantum error mitigation is to make ρ as close as possible to the ideal state $|\psi\rangle$. Using post-selection to mitigate the noise, if a measure not comes from from S then it can be discarded.

Consider the projector into the subspace of S as $\Pi_S = \sum_{s \in S} |s\rangle\langle s|$. Our post-selection scheme is based on projecting ρ as

$$\frac{\Pi_S \rho \Pi_S}{\text{tr}(\Pi_S \rho \Pi_S)}, \quad (3.2)$$

and such mixed state is a more reliable representation of the ideal state $|\psi\rangle$ than ρ itself. This is true once that states spanned by S are valid, and the states spanned by the remaining are invalid. In fact, it is important to note that one should consider if correct or incorrect states are detectable or not by post-selection.

With this in mind, we distinguish three orthogonal subspaces defined through

Chapter 3. Error Mitigation for QAOA using post-selection

projections $P_1 := |\psi\rangle\langle\psi|$, $P_2 := \Pi_S - P_1$ and $P_3 := \mathbb{1} - \Pi_S$. Those projections can be interpreted as follows:

- P_1 is the projection onto unknown correct state, which would be measured on the noise-free machine;
- P_2 is the subspace spanned by the incorrect valid states. Those states are not detectable by post-selection applied after the measurement in the computational basis;
- P_3 is the subspace of invalid states, detectable through the post-selection.

The efficiency of post-selection greatly depends on the overlap of the noisy state ϱ with these subspaces: if the overlap $\text{tr}(P_2\varrho)$ is high compared to overlap $\text{tr}(P_1\varrho)$ then we should not expect significant improvement. On the other hand, overlap with $\text{tr}(P_3\varrho)$ only influences the number of circuit runs to get a fixed number of valid samples. Projection $P_1 + P_2$ defines the valid subspace and projection $P_2 + P_3$ defines the incorrect subspace.

Let us consider depolarizing noise, which turns the ideal state $|\psi\rangle$ into noisy state ϱ . A measurement $\{P_1, \mathbb{1} - P_1\}$ would give back the ideal state. Nevertheless, it is unreasonable to expect that such measurement can be implemented in the middle of the circuit in principle, as this would require information about $|\psi\rangle$. On the other hand, performing a measurement $\{\Pi_S, \mathbb{1} - \Pi_S\}$ seems to be much more plausible since S is known. Although this is still not simple for an arbitrary S , it does not require any information other than S .

As an example, suppose that the subspace S consists of quantum states of Hamming weight² 1, so-called one-hot vectors

$$S = \{100\dots 0, 010\dots 0, \dots, 000\dots 1\}, \quad (3.3)$$

and the valid quantum states are those spanned by S . Let us consider a quantum circuit over n qubits consisting of l layers of the ansatz presented in Fig. 3.1. Since the given ansatz does not change the Hamming weight of the state, starting with a valid state, any obtained quantum state throughout the noiseless evolution of the circuit will belong to the subspace spanned by S . The effect of the post-selection discussed above can be improved by performing post-selection in the middle of the circuit by projection onto the subspace P_2 , as the valid states belong to the subspace spanned by S throughout the evolution of the circuit. Let us investigate the effect of mid-circuit post-selection in more detail. The evolutions can be roughly decomposed into amplitude transfer between subspaces defined through P_1, P_2, P_3 , see Fig. 3.2. If the transition was from valid to invalid states only, then we would

²We can say that Hamming weight is number of ones in a bit string.

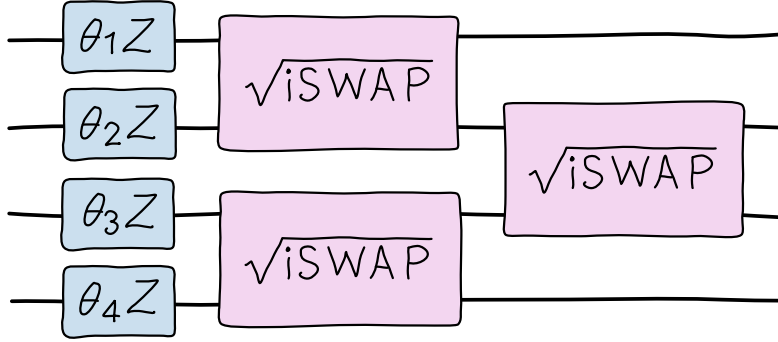


Figure 3.1: Ansatz preserving the subspace of one-hot basis states

not expect any improvement from mid-circuit post-selection compared to the final post-selection. However, the transitions take place also from invalid to valid states. Note that the correct space is only one-dimensional while the dimensionality of the whole valid space usually grows exponentially with the size of the data. Hence, the mid-circuit post-selection attempts to remove the impact of the transitions from invalid states to valid incorrect states mostly.

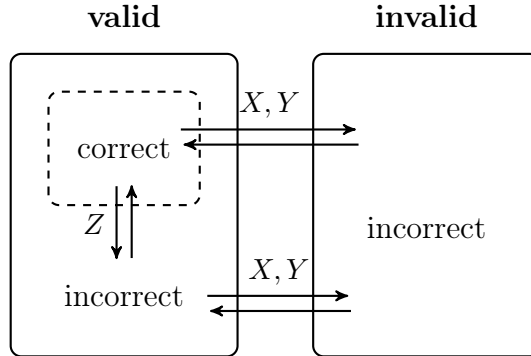


Figure 3.2: A scheme of how X , Y and Z errors changes the subspace of the state.

3.2.1 Post-Selection by filtering and compression

Current quantum devices can only measure qubits independently, and thus measurement $\{\Pi_S, \mathbb{1} - \Pi_S\}$ cannot be applied directly. Even so, we can simulate such measurement. We can distinguish two non-exclusive approaches: *post-selection through filtering* and *post-selection through compression*.

The post-selection through filtering requires ancillary qubits. The idea is to construct a quantum circuit U_{filter} which maps the basis state $|s\rangle$ with $s \in \{0, 1\}^N$

such that

$$U_{\text{filter}}|s\rangle|0\cdots 0\rangle = \begin{cases} |s\rangle|0\cdots 0\rangle, & s \in S, \\ |s\rangle|\varphi_s\rangle, & s \notin S, \end{cases} \quad (3.4)$$

where $|\varphi_s\rangle$ is (preferably) orthogonal to $|0\cdots 0\rangle$ for any $s \in S$. Upon application of U_{filter} , the ancilla can be measured in the computational basis and computation continues only if $0\cdots 0$ was measured.

A second approach, post-selection through compression, does not require extra qubits. Instead, we need a quantum circuit U_{compress} , which compresses valid states to a some smaller subspace $S' \subseteq \mathcal{H}$ such that

$$U_{\text{compress}}|s\rangle = \begin{cases} |\psi_s\rangle|0\cdots 0\rangle, & s \in S, \\ |\varphi_s\rangle, & s \notin S. \end{cases} \quad (3.5)$$

Note that here the only requirement is that some qubits are ‘reset’ to $|0\rangle$ after U_{compress} . Like in post-selection through filtering, the qubits are measured, and the computation continues *iff* all qubits are in state $|0\rangle$. In this case, we uncompute the compression through $U_{\text{compress}}^\dagger$. An evident advantage of this method compared to the previously introduced one is that it can run in-place without extra qubits.

The proposed methods are particularly suitable for quantum devices that allow mid-circuit measurements and can reset qubits to $|0\rangle$. Indeed, in this case, the number of required qubits does not grow with the number of applications of the proposed techniques. Still, it is also possible to harness quantum devices without the mid-circuit measurements feature. It is enough to implement filtering each time with different ancilla and to uncompute the state to a new set of qubits in the compression case. Then, the number of additional qubits will be proportional to the number of corrections applied and the number of measured qubits. However, a large number of mid-measured qubits or the number of post-selections applied makes the approach significantly less NISQ-friendly.

Besides, it is not possible to provide a general description of how to implement U_{compress} or U_{filter} . The reason behind this is that the structure of S depends on the form of the Hamiltonian and the origins of the optimization problem. In the next section, we will discuss the implementations of specific error mitigation schemes for with post-selection circuits for different S , which are specifically useful for various combinatorial and physical optimization problems.

3.3 Post-selection schemes for different encodings

Before moving on to the description of specific error mitigation schemes for different encodings, we would like to recall the circuit counting the electron number from [85].

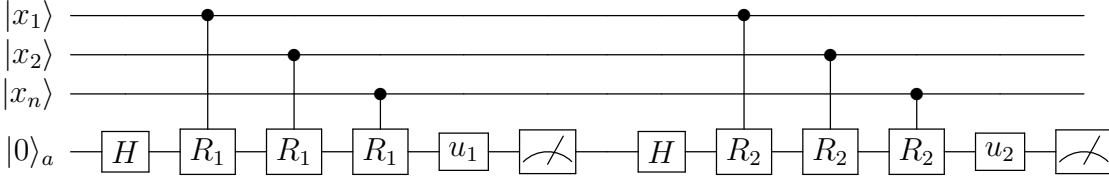


Figure 3.3: An example implementation of the circuit verifying whether the total number of 1’s is equal to κ . The gate R_j is given by $\text{diag}(1, e^{\pi i/2^{j-1}})$ and u_j is given by $\text{diag}(1, e^{-\text{dec}(\kappa_{j-1}\cdots\kappa_1)\pi i/2^{j-1}})$ where $\text{dec}(\kappa_{j-1}\cdots\kappa_1)$ is the decimal representation of the least significant j bits of the binary string $\kappa = \kappa_n\kappa_{n-1}\cdots\kappa_1$. u_1 is defined as the identity operator.

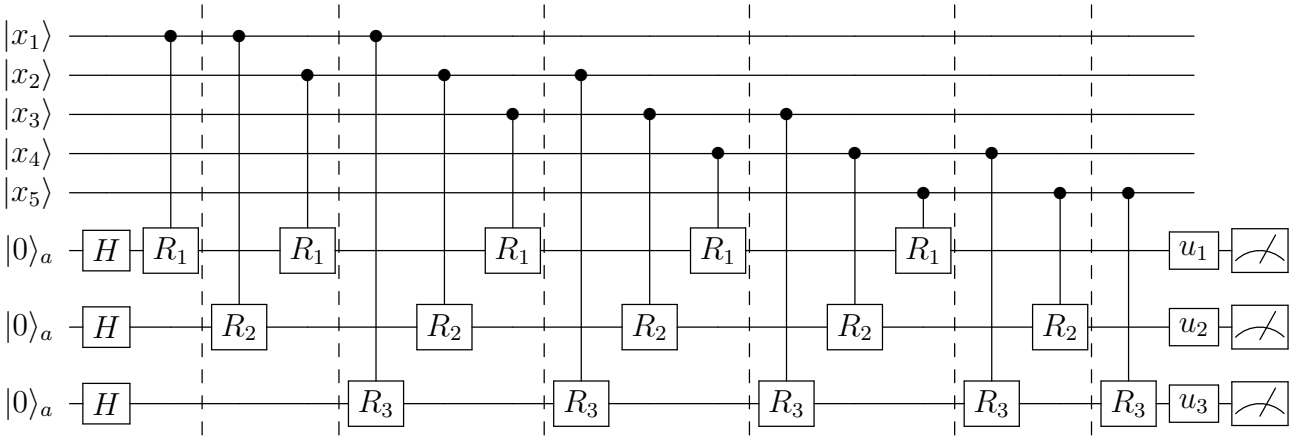


Figure 3.4: An example implementation of alternative circuit verifying whether the total number of 1’s is equal to κ . The idea behind is the same as the one presented in Fig. 3.3 except we have $\sim \log n$ ancilla and we apply gates in parallel.

In a pragmatic sense, the method is based on counting the number of 1’s in a basis state. The circuit described in [85] computes the number of 1’s in binary, one bit at a step, using only a single ancilla. The idea can be used as a subroutine in other circuits to verify whether the total number of 1’s is a particular value.

Let us describe the verification circuit inspired by [85]. Suppose that we want to verify whether the basis state $|x\rangle = |x_1x_2\cdots x_n\rangle$ contains exactly k 1’s. Let κ be the binary representation of k written using $\lceil \log n \rceil$ bits (0’s are padded to the most significant bits if $\lceil \log k \rceil < \lceil \log n \rceil$) and let ξ denote the binary representation of the sum of 1’s in $|x\rangle$. The circuit computes ξ starting from the least significant bit, as long as the measured bits coincide with that of κ ’s. In general, for an n -qubit circuit, there are $\lceil \log n \rceil$ blocks each computing a bit of ξ . After each block, the ancilla qubit is measured in the X -basis. If the measurement result is $|+\rangle$, then it indicates that the bit is 1 and the measurement result $|-\rangle$ indicates that the

bit is 0. Note that there are two possible outcomes when running the verification circuit: If at some stage the measurement outcome does not coincide with κ the computation ends, or all n bits coincide indicating that the verification succeeds. We would like to remark that all $\lceil \log n \rceil$ bits of ξ should be computed since it can be the case that κ and ξ coincide on the first $\lceil \log k \rceil$ bits, although κ and ξ are different.

For example, consider the state $|111\rangle$. For this state, the number of 1's is equal to $k = 3$, which can be represented as the binary string: $\xi = 11$. Given another bit string $\kappa = 01$, the algorithm mentioned above would stop after computing the second bit of ξ since it does not coincide with κ .

In Fig. 3.3, a circuit with $n = 3$ control qubits and a single ancilla qubit is given. Note that there are 2 blocks in the given circuit as the sum can be at most 11_2 . If the first measured bit is not the least significant bit of κ , then the computation ends. Otherwise, the computation continues with the second block.

The overall number of required gates and the depth are $O(n \log n)$. However, one can apply the controlled rotations in parallel, given extra ancilla qubits. The idea for $n = 5$ is presented in Fig. 3.4. This approach requires $\sim \log n$ ancilla and the depth equals $O(n)$. Note that in this case each bit of ξ is stored on a different ancilla qubit.

3.3.1 k -hot encoding

The k -hot states are 0-1 states with Hamming weight k and often appear in physics and computer science: k -hot vectors for $k \geq 2$ are a natural description of quantum k -particle Fock spaces [9, 50, 85]. Dicke states which are the equal superposition of k -hot states are used as the initial state in QAOA [13] for certain problems. k -hot states are also used to encode the feasible states in problems like Max- k Vertex Cover problem [33], and graph partitioning [57].

Post-selection can be applied to k -hot states through filtering by verifying the total number of 1's in the quantum state using the circuits given in Fig. 3.3 or 3.4. The idea was first investigated in [85], in the scope of VQE and particle number preserving ansatz.

3.3.2 One-hot encoding

One-hot encoding is a special case of k -hot encoding, and it is used in literature for encoding various problems like Travelling Salesman Problem, Graph Coloring, and Clique Cover [79]. It is also used for optimization over functions $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$. In the latter case, we specify n quantum registers, and each register consisting of m qubits that encode the values of the function between 1 and m .

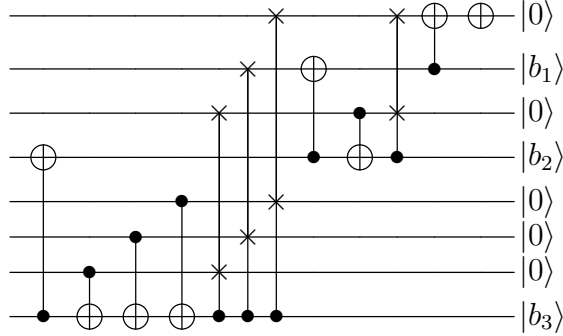


Figure 3.5: An example of the implementation of the map V which transforms the one-hot encoding to binary encoding [109]. Note that the procedure can be adjusted to the case where the maximal stored number is not a power of 2. If an initial state is a superposition of one-hot basis states, then some of the output qubits are set to $|0\rangle$.

We will mention two different approaches for post-selecting one-hot states. Since one-hot vectors are a special case of k -hot vectors with $k = 1$, we can use the filtering approach proposed in the Sec. 3.3.1. Alternatively, one can consider a post-selection through compression with a circuit that converts one-hot representation to binary representation [109]. Let V be the unitary operation implementing this map. For an integer $l \in \{1, \dots, n\}$, let $OH_n(l)$ be the bit assignment for one-hot encoding, i.e. it maps l to the quantum state with a 1 in the l 'th position. Let $B_m(l)$ be the bit assignment function encoding l in binary using exactly m bits. B_m maps l to $b_m \dots b_1$ such that $l = \sum_{i=1}^m 2^{i-1} b_i$. Although the map $V : OH_n(l) \mapsto B_m(l)$ does not preserve the number of qubits, the unoccupied qubits after the transformation are set to $|0\rangle$ as it can be seen in Fig. 3.5. The one-hot to binary conversion leaving some qubits in-state $|0\rangle$ provides a natural scheme for error mitigation.

The circuit implementing V uses $O(n)$ gates, no ancilla, and has $O(n)$ depth [109]. After applying V and measuring the qubits which should be in state $|0\rangle$, V^\dagger should be applied for decompression. Note that the compression approach uses fewer resources when compared to the k -hot filtering approach for one-hot states.

3.3.3 Domain-wall encoding

In domain-wall encoding [30], valid states are of the form $|1 \dots 1 0 \dots 0\rangle$, i.e. the state starts with some number of ones followed by zeros. It requires less connectivity for checking the feasibility condition. For instance, in one-hot encoding, it is required to check whether each pair of qubits are in state $|1\rangle$ or not, while for domain-wall it is sufficient to check only neighboring qubits to see whether a $|0\rangle$ is followed by $|1\rangle$. Note that any problem expressed using n qubits in one-hot encoding can be

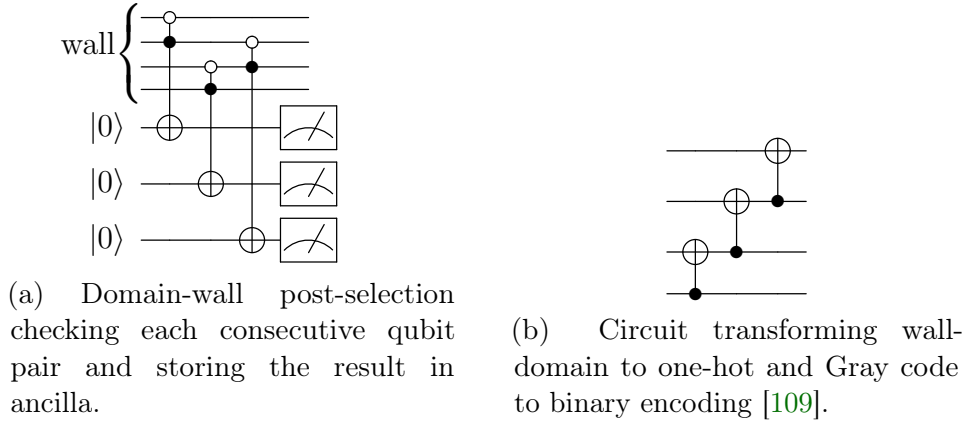


Figure 3.6: Circuits used for postselection for wall-domain encoding.

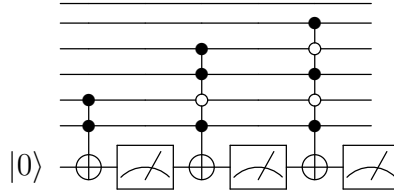


Figure 3.7: Binary exact post-selection for $\mu = 42 = 101001_2$

also expressed by domain-wall encoding, such that integer l is represented with a quantum state where l ones are followed by $n - l$ zeros.

The conditions above also motivate a mid-circuit post-selection scheme through filtering as invalid states can be detected by checking consecutive bits. One approach is to check each neighboring pair of qubits and store the result using $n - 1$ ancilla. While very demanding in the number of qubits, the approach requires only $O(1)$ depth and $O(n)$ gates. An example circuit with 4 qubits can be found in Fig. 3.6a. When the number of qubits is limited, then one can apply the error checking with the output on a single ancilla qubit, and measure it instantly and reset it so that it will be reused for the next condition checking. While the number of ancilla qubits will be only one, the depth will increase to $O(n)$.

Finally, one can use an ancilla-free method by first transforming domain-wall to one-hot encoding using the circuit given in Fig. 3.6b and use the post-selection through compression method described in the previous subsection. In this case, the number of gates and depth is the same as for one-hot vectors which are $O(n)$ for both.

3.3.4 Binary and Gray encoding

Binary encoding of an integer l using n bits is obtained by the assignment $B_n(l)$ as discussed in Sec. 3.2. It is used in QUBO formulations to save qubits while representing slack variables as discussed in [79]. It is also used while formulating qubit efficient higher-order unconstrained binary optimization formulations (HOBO) for problems like TSP [51] and graph coloring [115].

Using n bits, the numbers $0, 1, \dots, 2^{n-1} - 1$ are naturally encoded. If not all of the integer values encoded using n bits are admissible, then some of the encoded integers will be invalid and this will increase the infeasible space. There are several workarounds to solve this issue. One approach is to use the knowledge about the maximal attainable value \bar{x} [67], and update the encoding as

$$\sum_{i=1}^n 2^{i-1} b_i + \left(\bar{x} - \sum_{i=1}^{\lceil \log(\bar{x}) \rceil} 2^{i-1} \right) b_n, \quad (3.6)$$

which introduces bias for higher values. However, when the numbers encoded are the slack variables turning inequality $f(b) \geq 0$ into $f(b) + x_i = 0$, usually small values of x_i are encountered so that the original inequality is satisfied tightly or almost tightly. For this reason, introducing bias for higher values may have a negative effect on the optimization. Furthermore, in recent HOBO formulations using binary encoding [51, 115], quantum states which encode too large values have to be penalized unlike the method above. However one may expect a variation of this algorithm with QAOA+ which will forbid (up to noise) quantum state from evolving into too large numbers, for example, a particular version of QAOA+. Motivated by this, and also for completeness, we describe a filtering scheme below.

Suppose the valid integer can be at most $\mu = b'_n b'_{n-1} \dots b'_1$. Let I'_0 be the collection of indices i for which $b'_i = 0$. The bit assignment is invalid (*ie.* encodes a larger integer than μ) if at some bit at which it should be zero, it is one *and* all of the more significant bits are the same as that of μ . For example, if we have $\mu = 42 = 101001_2$, then incorrect numbers are of the form $11b_3 b_2 b_1 b_0$, $1011b_1 b_0$ and $10101b_0$, where b_i are arbitrary. Hence, we need to verify whether any such situation occurs. An exemplary post-selecting circuit proposed in Fig. 3.7 for $\mu = 42$.

In the worst case, for instance when $\mu = 10 \dots 0_2$, one may need to check $n - 1$ invalid forms. Each check requires implementation of multi-controlled NOT gates. To implement a multi-controlled NOT gate controlled by n qubits, we will consider two different methods: The first method described in [107] uses no ancilla, requires $O(n^2)$ gates and has $O(n)$ depth. The second method proposed in [60] uses $O(n)$ ancilla, $O(n)$ gates and has $O(\log n)$ depth.

Hence, if one wants to save ancilla qubits, then the ancilla-free implementation

of multi-controlled NOT gate is more convenient, the overall approach requiring a single ancilla qubit, $O(n^3)$ gates and the circuit has $O(n^2)$ depth. Using the second method, overall circuit requires $O(n)$ ancilla, $O(n^2)$ gates and has $O(n \log n)$ depth.

In general, checking all invalid forms might be costly depending on the value of μ as the error mitigation itself might introduce some errors. For instance, when $\mu = 10 \cdots 0$, checking only the most significant bit that should be 0 is enough to eliminate half of the invalid cases. In general, this would require only a single application of multi-controlled NOT gate, and in the worst case when $\mu = 11 \cdots 10$ there will be $n - 1$ control qubits. In such a case, it may not be efficient to use an error mitigation circuit only to eliminate a single invalid state. However, if there are multiple registers, say k , encoding numbers in binary, then the proportion of the feasible to all states equal

$$\left(\frac{n-1}{n}\right)^k \approx e^{-\frac{k}{n}}. \quad (3.7)$$

Even for this extreme case, for $k \approx n$ we already have a constant fraction of the mitigated cases. This scenario appears in [51]. So we can say it might be infeasible to eliminate an error for a single number, but, it still may be beneficial for multiple registers.

Note that this approach can also be used for one-hot encoding in combination with post-selection through compression scheme discussed in Sec. 3.2. One can check if the compressed number in binary is representing a number greater than or equal to n in an n -qubit circuit. In this case, the depth and the number of gates will be $O(n)$.

In addition, the proposed approach can be applied to Gray-code encoding [109], after transforming it to binary encoding using the circuit given in Fig. 3.6b. The transformation has no impact on any of the resource measures.

3.3.5 One-hot and binary mixed

Finally, let us consider a combination of one-hot encoding and binary encoding proposed in [51, 109]. In such cases, bits encoding a single number are partitioned into l groups, each group consisting of m qubits, and only one of the groups has nonzero bits. If \bar{l} -th group is the one with nonzero bits, then the bits of \bar{l} -th group encodes the number $x_{\bar{l}}$ in binary or Gray-code encoding, and the value of the encoded number is $(\bar{l} - 1)(2^m - 1) + x_{\bar{l}} - 1$. For example for 4 groups, each with 2 bits, for the sequence 00 00 10 00 we have $\bar{l} = 3$ and thus the value encoded is $(3 - 1)(2^2 - 1) + 10_2 - 1 = 7$. Note that two conditions can be asserted: Exactly one group consists of nonzero bits, and the last group may only attain some values due to the redundancy of binary encoding described in the previous paragraph.

The latter can be solved the same way as it was solved for purely binary encoding in Sec. 3.3.4. For the former, we need to check whether the number of groups in which all consecutive bits in the group are all zeros is equal to $l - 1$.

One approach is to count the number of such groups using the verification idea from Fig. 3.3. To implement the circuit, we need to implement a rotation gate controlled by m qubits for each one of the l groups. Using the ancilla-free and non-ancilla-free implementations of multi-controlled NOT gate, this would require $O(lm^2)$ and $O(lm)$ gates, respectively. Recall that there are two different approaches for verification, one using 1 ancilla qubit and the other using $\log l$ qubits. Single-ancilla verification idea is visualized in Fig. 3.8a. To save qubits, one may prefer ancilla-free multi-controlled NOT gate and single ancilla verification, overall which would require 1 ancilla qubit, $O(lm^2 \log l)$ gates and has $O(lm \log l)$ depth. To have a circuit with smaller depth, one can use non-ancilla-free multi-controlled NOT gate and verification with $\log l$ ancilla, resulting in $O(m \log l)$ ancilla, $O(lm \log l)$ gates and $O(l \log m)$ depth.

In the second approach, the idea is to store the information whether each group consists of all zeros or not in an ancilla qubit. To implement this idea we use l ancilla qubits, and save the required information. After applying NOTs on those qubits we can check whether the resulting l -qubit state is an one-hot state. Then using the compression scheme for one-hot encoding from Sec. 3.3.2, we can check if the resulting state is one-hot. Using the ancilla-free implementation of multi-controlled NOT gate, this would require $O(l)$ ancilla, $O(lm^2)$ gates and $O(l + m)$ depth. When we use non-ancilla-free implementation of multi-controlled NOT gate, then we have two options. We can use different ancilla for each multi-controlled NOT gate requiring $O(lm)$ ancilla overall, and we can implement the circuit using $O(l + \log m)$ depth, or using the same $O(m)$ ancilla for each multi-controlled NOT gate, we can have a circuit with $O(l + m)$ ancilla and $O(l \log m)$ depth. For both approaches, the number of required gates is $O(lm)$.

This scheme can be also used for the mixture of Gray and one-hot encoding [109] after it is translated into binary encoding using the circuit given in Fig. 3.6b.

3.3.6 Summary

We present a summary of the resource requirements for the methods discussed so far. Depending on whether we use the 1- or $\log n$ ancilla counting method, we use notation Σ_1 and Σ_{\log} respectively. T_{free} denotes the ancilla-free implementation of multi-controlled NOT gate, while T_{ancilla} denotes the $O(n)$ ancilla implementation. $T_{\text{multi-anc}}$ is the case where it is guaranteed that for each implementation of the multi-controlled NOT gate, different ancilla qubits are available. Finally, M denotes more subtle implementation details. For wall-domain encoding, it differentiates between the parallel checking with $O(n)$ and 1 ancilla. For the mixed encoding, M_{store}

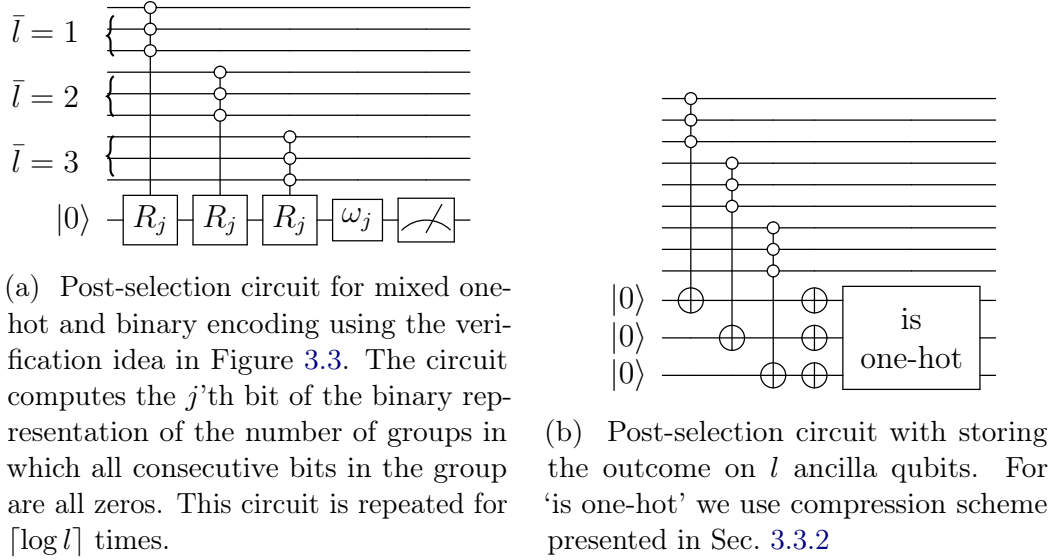


Figure 3.8: Post-selection circuits for binary encoding and mixed encoding.

denotes the approach of constructing one-hot vector and using the compression scheme. For all of the encodings, we assume that the analyzed system consists of n qubits. For the mixed encoding, this also gives an identity $n = lm$.

In addition to the resources considered in the previous sections, we also present the volume of the encoding. The volume is defined as the product of depth and the number of qubits. For the number of qubits, we used the sum of ancilla qubits and n .

3.4 Application to Quantum Alternating Operator Ansatz

In this section, we will consider the Travelling Salesman Problem (TSP). The QUBO formulation for TSP over N cites is given as

$$A \sum_{t=1}^N \left(1 - \sum_{i=1}^N b_{t,i} \right)^2 + A \sum_{i=1}^N \left(1 - \sum_{t=1}^N b_{t,i} \right)^2 + \sum_{\substack{i,j=1 \\ i \neq j}}^N W_{ij} \sum_{t=1}^N b_{t,i} b_{t+1,j}, \quad (3.8)$$

where W is the cost matrix, and $b_{t,i}$, is the binary variable such that $b_{t,i} = 1$ iff the i -th city is visited at time t [79]. A is a constant which needs to be adjusted so that the optimal solution of QUBO encodes the optimal solution for TSP. The formulation uses N^2 qubits which produces a large infeasible space *ie.* there

Encoding	Info.	Ancilla	Gates	Depth	Volume
<i>k</i>-hot	Σ_1 [85]	1	$O(n \log n)$	$O(n \log n)$	$O(n^2 \log n)$
	Σ_{\log}	$O(\log n)$	$O(n \log n)$	$O(n)$	$O(n^2)$
Domain-wall	$M_{\text{inductive}}$	1	$O(n)$	$O(n)$	$O(n^2)$
	M_{parallel}	$O(n)$	$O(n)$	$O(1)$	$O(n^3)$
Binary/Gray	T_{free}	1	$O(n^3)$	$O(n^2)$	$O(n^3)$
	T_{anc}	$O(n)$	$O(n^2)$	$O(n \log n)$	$O(n^2 \log n)$
Mixed	$\Sigma_1 T_{\text{free}}$	$O(1)$	$O(lm^2 \log l)$	$O(lm \log l)$	$O(l^2 m^2 \log l)$
	$\Sigma_1 T_{\text{anc}}$	$O(m)$	$O(lm \log l)$	$O(l \log l \log m)$	$O(l^2 m \log l \log m)$
	$\Sigma_{\log} T_{\text{free}}$	$O(\log l)$	$O(lm^2 \log l)$	$O(lm)$	$O(l^2 m^2)$
	$\Sigma_{\log} T_{\text{anc}}$	$O(m \log l)$	$O(lm \log l)$	$O(l \log m)$	$O(l^2 m \log m)$
	$M_{\text{store}} T_{\text{free}}$	$O(l)$	$O(lm^2)$	$O(l + m)$	$O(l^2 m + lm^2)$
	$M_{\text{store}} T_{\text{anc}}$	$O(l + m)$	$O(lm)$	$O(l \log m)$	$O(l^2 m \log m)$
	$M_{\text{store}} T_{\text{multi-anc}}$	$O(lm)$	$O(lm)$	$O(l^2 m + \log m)$	$O(l^2 m + lm \log m)$

Table 3.1: Summary of the resource requirements of the post-selection circuits for different encoding using filtering.

Encoding	Gates	Depth
1-hot	$O(n)$	$O(n)$
Domain-wall	$O(n)$	$O(n)$

Table 3.2: Summary of the resource requirements of the post-selection circuits for different encodings using compression.

are 2^{N^2} possible solutions to QUBO model, while the number of routes is only $N! = 2^{O(N \log N)}$. To reduce the infeasible space, one possible approach is to encode the problem using less number of qubits as proposed in [51]. Another approach is to reduce the effective space of the evolution, which is the idea behind the Quantum Alternating Operator Ansatz (QAOA+) [57].

QAOA+ is considered as an extension of QAOA that allows more general families of mixing operators. In QAOA+, the initial state is usually a feasible solution to the problem, and the mixer operator restricts the search to the feasible subspace by mapping feasible states to other feasible states. Unlike the regular version of QAOA, the evolution takes place in a smaller subspace of the full Hilbert space. On this chapter we will consider a special case of QAOA+ called XY-QAOA.

Chapter 3. Error Mitigation for QAOA using post-selection

In XY-QAOA, the mixer is chosen as XY-Hamiltonian

$$\sum_{i=1}^N X_i X_{i+1} + Y_i Y_{i+1}, \quad (3.9)$$

applied on every one-hot register, which preserves the Hamming weight of the quantum states [125]. In the case of TSP over N cities, N registers each with N qubits are used such that if $b_{t,i} = 1$, then register t encodes i using one-hot encoding. The initial state can be prepared as the Kronecker product of W -states which can be efficiently implemented [125]. Note that the choice of the initial state is particularly suitable for XY-mixer as XY-mixer maps one-hot states to one-hot states. Although the generated subspace contains some infeasible states as well, it contains the whole feasible space for the TSP problem and is significantly smaller than the full Hilbert space. More precisely, the evolution takes place in $N^N = 2^{O(N \log N)}$ dimensional subspace of the full N^2 -qubit Hilbert space.

As the post-selection scheme, we use the compression scheme for one-hot vectors proposed in Sec 3.3.2. We consider a noise model where every gate is affected by a random unitary channel applied after each quantum gate, including gates from the post-selection. In particular, we will consider depolarizing noise, amplitude damping noise, and random X noise with parameter γ reflecting the strength of the noise: the smaller the value of gamma, the least is the effect of the noise on the evolution. We assume that the initial state is $|0 \cdots 0\rangle$ and measurements (both final and in the middle) are implemented perfectly. Ideal initial state preparation is justified as any digression into infeasible subspace will be detected by mid-circuit post-selection, or will produce some bias for QAOA+, which may be corrected by adjusting the parameters of the ansatz. For measurements, we note that the noise is highly biased. States $|0\rangle$ are much less prone to error compared to $|1\rangle$ so that it is unlikely that 1 is measured when one is expecting to measure 0 [81].

A simplified version of the circuit is visualized in Fig.... After a fixed number of QAOA layers we apply the compression scheme presented in Sec.... We continue computation iff all measurements result in 0 states. For the final measurements, we post-select only those measurement samples which would appear in the error-robust computation. Note that in fact the mid-circuit post-selection can be also applied in the middle of the objective Hamiltonian application—this Hamiltonian is implemented by consecutively applying diagonal matrices, which does not change the space over which the states is defined. However, we apply mid-circuit post-selection after at the end of layers only for simplicity.

3.5 Circuit Design

We designed and implemented our circuits using Qiskit software development kit (SDK). One can say that the circuit for XY-QAOA can be break down in the following steps:

1. Preparation of the W -states;
2. implement QUBO objective Hamiltonian followed by the mixer XY Hamiltonian (QAOA layer);
3. change the encoding from Binary to Unary;
4. mid circuit measurement and post-selection;
5. repeat 2 to 4 until reach the maximal number of layers.

The first step is prepared the ground state of interest. For the regular QAOA, this process consists in applying Hadamard gates in all the qubits, generating an equal superposition of the computational basis elements,

$$|0\rangle^{\otimes N} \rightarrow H^{\otimes N} 0^{\otimes N}. \quad (3.10)$$

In the case of QAOA+, the initial state is Kronecker product of W -state.

When implementing the objective Hamiltonian, we only included the part for computing the cost routes and for verifying whether at different time-points we have distinct cities. Note that the part which checks whether at given time point only one city is visited is guaranteed by the algorithm itself.

At the time, in order to keep the depth of the circuit at the minimal, we had to enforce some ways to organize the distribution of gates. For instance, we set the gates fo QUBO usign round robyn scheduling based on [51].

3.6 Results

We start by investigating the effect of the post-selection for randomly chosen angles. We sample 100 instances of TSP for $n = 3, 4$ cities. Cost matrix W is a random matrix with elements sampled i.i.d. from the range $\{1, \dots, 9\}$. The penalty value equals $A = 2 \max_{i,j} W_{ij}$. In the rest of the discussion, any considered energy will be for rescaled QUBO, such that the corresponding (attainable) maximal value of the pseudo-Boolean function is 1, and the smallest value is 0. Let E be the true energy coming from the noise-robust evolution and let $E_{\text{no-mid}}$ be the energy obtained from the noisy evolution but with post-selection applied on the final outcomes only. Finally, let E_{mid} be the energy with the post-selection applied both in the

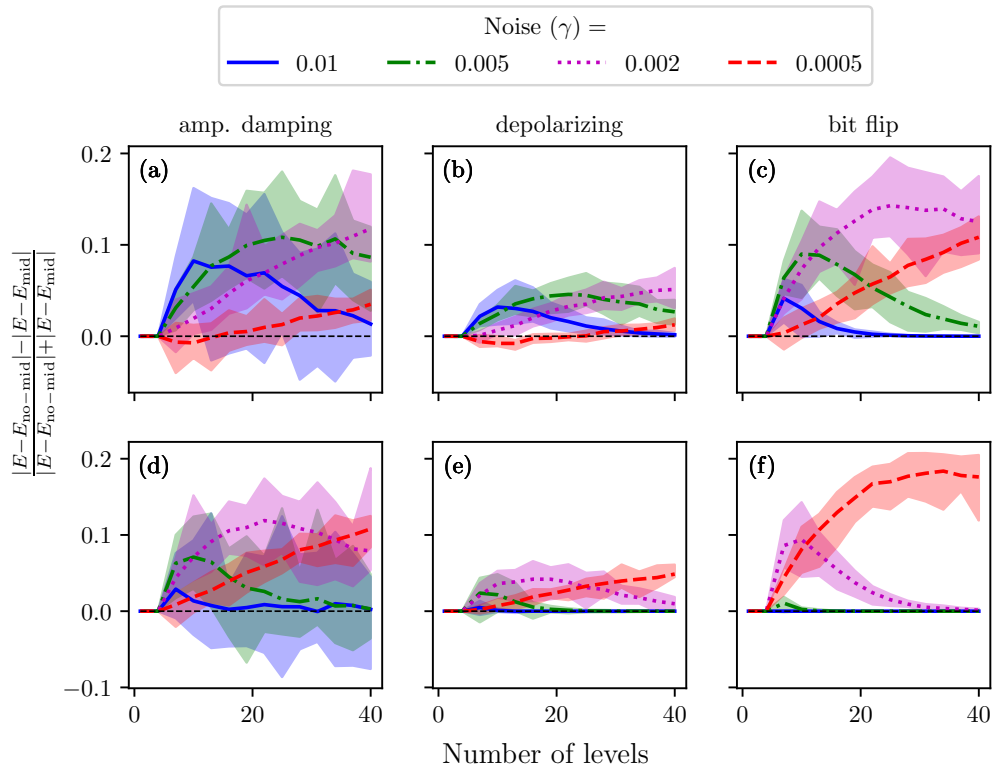


Figure 3.9: The efficiency of mid-circuit post-selection against final circuit post-selection only. The subplots (a), (b) and (c) are for 3 cities, and (d), (e), (f) are showing the results for 4 cities.

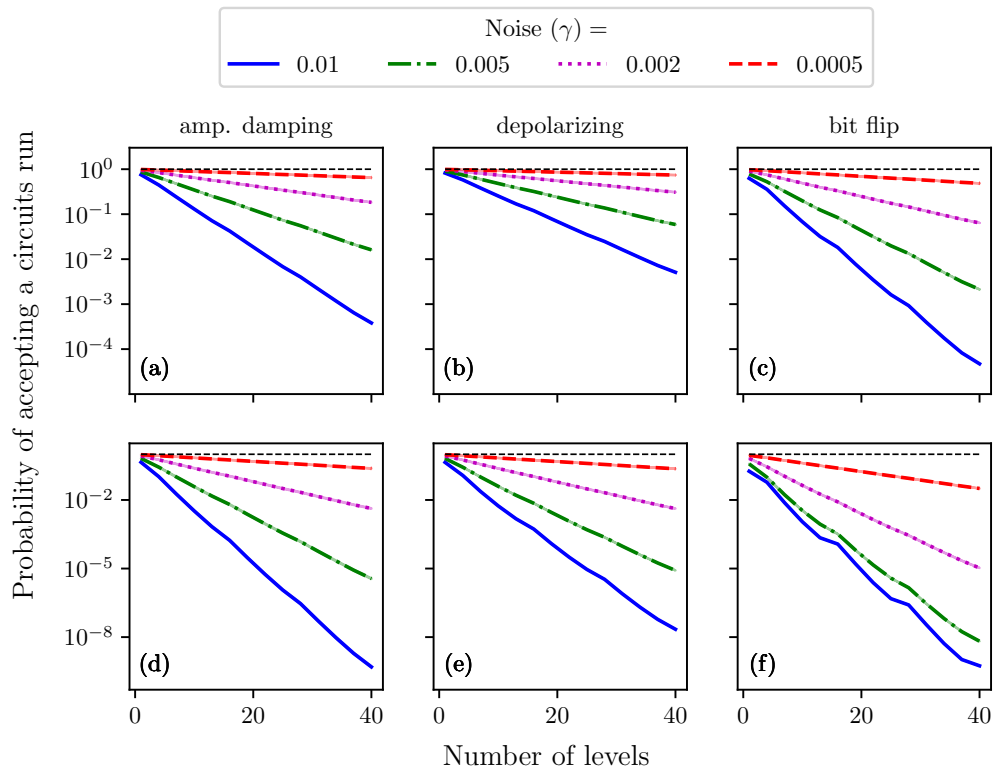


Figure 3.10: Acceptance probability, defined as the probability of accepting the circuit run. The lines represent the mean value of the relative acceptance probability over 100 samples. The (hardly visible) areas represent the samples between 10th and 90th percentile. The subplots (a), (b) and (c) are showing the results for 3 cities instances, and (d), (e), (f) are showing the results for 4 cities.

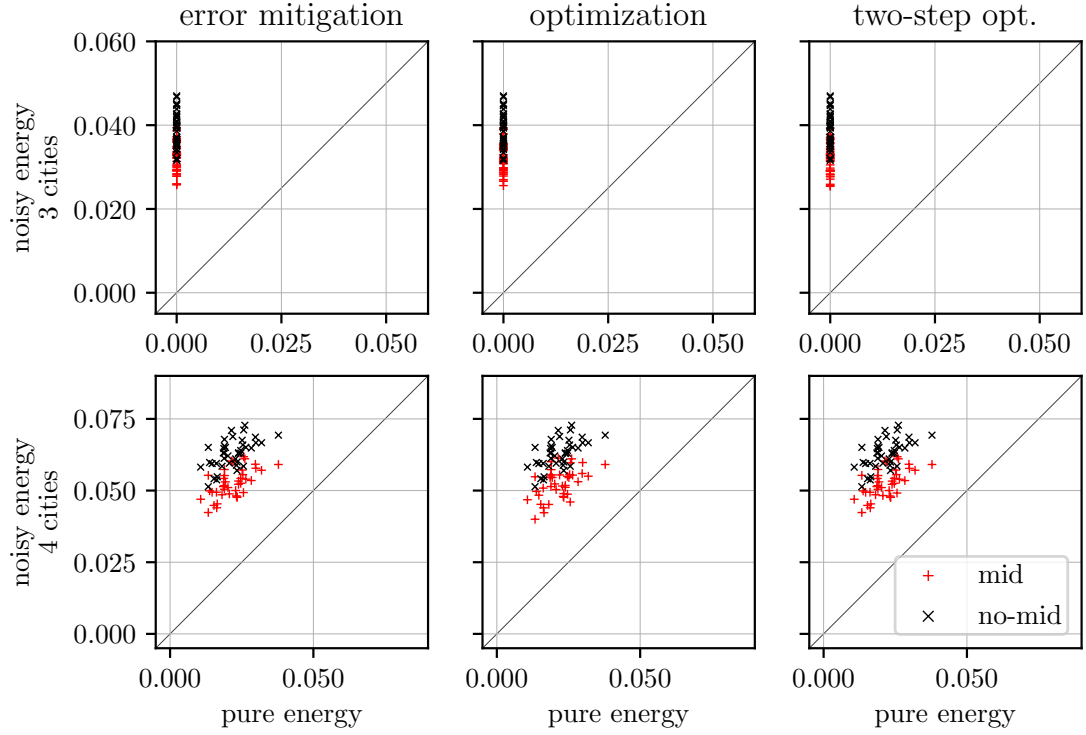


Figure 3.11: Effect of post-selection on QAOA optimization. In the first column, we simply correct the optimal angles obtained through regular optimization with the post-selection applied at every 2 layers. In the second column, we compare optimization with and without mid-circuit post-selection, starting with the same angles. Finally, in the last column, we take the optimal angles obtained through regular optimization and repeat the optimization with mid-circuit post-selection. The solid black line is the $y = x$ and denotes the ‘no difference’ case. The red solid line is the $y = 0.85x$.

middle (at every 4th layer) and on the final outcomes. Our measure of quality is $\Delta E^{(i)} = |E^{(i)} - E_{\text{no-mid}}^{(i)}| - |E^{(i)} - E_{\text{mid}}^{(i)}|$, where i stands from the i -th TSP instance. Note that the larger the value, the more positive impact the mitigation scheme has on the output.

The results presented in Fig 3.9 show that the effect of post-selection strongly depends not only on the noise impact γ , but also on the type of the noise. This is expected, as the noise structure also affects the way the amplitude is transferred between valid and invalid states. For example, for random Z noise our method (the same as the classical post-selection) cannot detect any deviation. However, for all combinations of noise strength and noise models, we see that the post-selection has mostly a positive effect on the evolution.

The mean of ΔE is usually detached from zero. However, the area denoting the space within plus/minus standard deviation highly deviates from the mean. Yet, in most of the cases, the difference of the mean and standard deviation is close to 0, which shows that our method has likely no negative effect against final post-selection only.

The error mitigation process requires extra measurements some circuits are discarded. In Fig. 3.10, we show the probability of accepting circuit run. The probability decreases exponentially with the number of levels, as we discard a fixed portion of the measurements every few layers, every four layers in our case. We observe that the increase in the noise strength diminishes further the probability.

Let us now consider the effect of post-selection on the optimization process. We considered 40 TSP instances generated as described above, with 8 layers and random X noise with $\gamma = 0.002$. We consider 3 scenarios here. In all of them, we use the classical post-selection of the final outcomes, as classical post-selection can be implemented efficiently using classical computing. In the first scenario, we optimize the circuit without mid-circuit post-selection. Then we inject inside this circuit a mid-circuit post-selection procedure and compare the obtained energy with the previous one. This is the most efficient method, as the mid-circuit part of the circuit does not take part in the optimization process, which may slightly decrease the time required for the optimization. In the first column of Fig3.11, we can see that this approach provides stable improvement of around 15% for both 3 and 4 cities case.

One may expect that correcting the energy via post-selection, through correcting the energy, will provide an alternative, more faithful energy landscape. To analyse this, we used mid-circuit post-selection also in the middle of optimization. We considered two approaches. In the first, we compare the energy obtained through circuits with and without post-selection, starting from the same initial angles. In the second, we first optimized the circuit without mid-circuit post-selection, and then we took the final circuit and re-optimized it with the circuit containing

mid-circuit post-selection. Both are presented in Fig. 3.11 We can see that there is almost no difference between these two approaches, which indicates that the landscape may be very similar for these cases. This in turn implies, that it may be sufficient to optimize the circuit without the mid-circuit post-selection, and only then apply the mid-circuit post-selection.

3.7 Conclusion

There have been some recent attempts to mitigate errors in Variational Quantum Algorithms (VQAs) through mid-circuit post-selection. Following this line of work, we presented post-selection schemes for various encodings and different valid subspaces of quantum states, which can be used with VQA while solving particular combinatorial optimization problems and problems from quantum chemistry. We implemented the one-hot to binary post-selection through compression scheme to solve the Travelling Salesman Problem (TSP) using the Quantum Alternating Operator Ansatz (QAOA+) algorithm. The experiment results show that for amplitude damping, depolarizing, and bit-flip noises, the mid-circuit post-selection has a positive impact on the outcome compared to final post-selection only. The schemes we propose are qubit efficient, do not need classical operation, and use only mid-circuit measurements and reset. Hence, with the emerging technology of mid-circuit measurements [32, 33], the presented methods are currently applicable to NISQ algorithms. Finally, our method can also be used in principle outside the scope of VQA.

Although we have only considered the TSP problem in our numerical experiments, it is worth noting that the proposed schemes can be used with different objective Hamiltonians. Our ancilla-free post-selection through compression scheme can be applied to any problem where the feasible states are one-hot, including the problems defined over permutations such as Vehicle Routing Problem [19], variations of TSP [103, 108], Railway Dispatching Problem [37, 38], Graph Isomorphism Problem [25], Flight Gate Assignment Problem [114].

There are several research directions that can be pursued requiring further investigation. First of all, in general, the optimal number of post-selections to apply is not evident. Many factors should be considered here, including the complexity of the post-selection, the form of the feasible subspace S , the strength and form of the noise affecting the computation. It is desirable to design methods that would choose the optimal number (and perhaps the position) of mid-circuit post-selections to be applied.

3.8 Chapter Summary

On this chapter, we discussed how to enhance the performance and reliability of VQAs by investigating the error mitigation strategy utilizing mid-circuit measurements. The key concepts are:

- Variational Quantum Algorithms (VQAs): These algorithms leverage classical-quantum hybrid approaches where a parametrized quantum circuit is optimized using a classical optimizer to solve problems. It is a crucial class of algorithms for quantum computing since it is particularly suited NISQ devices.
- Error Mitigation: Techniques aimed at reducing the impact of noise and errors in quantum computations without requiring fault-tolerant quantum error correction.
- Mid-Circuit Measurements: The process of measuring certain qubits at intermediate steps during the execution of a quantum circuit. This technique allows for dynamic error detection and correction, improving the overall fidelity of quantum operations.

We introduce a error mitigation approach using a post-selection scheme. The post-selection is performed by injecting a quantum circuit consisting of both gates and measurements in between the layers of a parametrized circuit with the intuit to filter out the states that don't belong to the feasible state. We consider the different post-selection strategies for various valid subspaces obtained through different encodings that frequently appear in encoding combinatorial optimization problems and in quantum chemistry. Among them:

- k -hot: The k -hot states are 0-1 states with Hamming weight k and often appear in physics and computer science. Post-selection can be applied to k -hot states through filtering by verifying the total number of 1's in the quantum state
- One-hot: this encoding is a special case of k -hot encoding, and it is used in literature for encoding various problems like Travelling Salesperson Problem, Graph Coloring, and Clique Cove. Since one-hot vectors are a special case of k -hot vectors with $k = 1$, it can use the post-selection strategy mentioned above. Alternatively, one can consider a post-selection through compression with a circuit that converts one-hot representation to binary representation.
- Domain-wall: valid states are of the form $|1 \dots , 10, \dots 0\rangle$, i.e. the state starts with some number of ones followed by zeros. The particularities of

this encoding scheme motivate a mid-circuit post-selection scheme through filtering as invalid states can be detected by checking consecutive bits.

- Binary and Gray encoding: a binary encoding consist in encoding integer l using n qubits. For example, the number 3 can be mapped in a quantum state prepared as $|011\rangle$. The Gray encoding, also known as reflected binary, is an ordering of the binary numeral system such that two successive values differ in only one bit. Therefore, the state representing the number 3 in Gray encoding is $|101\rangle$.

The post-selection scheme for binary encoding consists in checking if the bit assignment is invalid, for instance, if at some bit at which it should be zero, it is one and all of the more significant bits are the same. Each check requires implementation of multi-controlled NOT gates. Implementing multiple multi-controlled NOT gates can be done by either using ancilla or not. Note that this approach can also be used for one-hot encoding in combination with post-selection through compression scheme. This proposed approach can be applied to Gray-code encoding, after transforming it to binary encoding, once that the transformation has no impact on any of the resource measures.

In particular, the scheme we propose for one-hot encoding works by compressing the valid subspace to the smaller subspace of quantum states and differentiates from the known methods. We then proceed by implementing the one-hot to binary post-selection through to solve the Travelling Salesman Problem (TSP) using the Quantum Alternating Operator Ansatz (QAOA+) algorithm.

The proposed error mitigation strategy using mid-circuit measurements offers a promising avenue for enhancing the performance of VQAs on NISQ devices. The experiment results show that for amplitude damping, depolarizing, and bit-flip noises, the mid-circuit post-selection has a positive impact on the outcome compared to final post-selection only. The schemes we propose are qubit efficient, do not need classical operation, and use only mid-circuit measurements and a reset. Hence, with the emerging technology of mid-circuit measurements the presented methods are currently applicable to NISQ algorithms.

Chapter 4

Hamiltonian-Oriented Homotopy QAOA

The classical homotopy optimization approach has the potential to deal with highly nonlinear landscapes, such as the energy landscape of QAOA problems. In this chapter¹, we introduce Hamiltonian-Oriented Homotopy QAOA (HOHo-QAOA), a heuristic method for combinatorial optimization using QAOA, based on classical homotopy. We will briefly discuss the Homotopy method and compare with other approaches for QAOA considering the Max-Cut problem.

4.1 Introduction

Given the limited resources of quantum computers, it is essential to effectively explore the landscape of cost function when implementing VQA's Parametrized Quantum Circuits. Particularly, in the case of QAOA, the landscape of energy function is highly non-linear. Therefore, sophisticated methods are necessary to deal with such complicated landscapes.

This motivate to investigate the QAOA cost functions by formulating a heuristic strategy that uses classical homotopy optimization, since it has potential application in dealing with highly non-linear functions [126]. Such method comprises a homotopy map, taking each value of interpolating parameter $\alpha \in [0, 1]$ and outputs an optimization problem. In particular, for $\alpha = 0$, the problem is easy-to-solve, and for $\alpha = 1$ the homotopy map returns the problem of interest. During the interpolation process, which changes the value of α from 0 to 1, the solution continuously changes and is expected to be optimal, or close to optimal for the intermediate problems. If the intermediate optimization succeed, in the end we

¹The content of this chapter is based on the author's work and all the figures in this chapter are taken from, or are adaptations of, the figures present in such work.

obtain the optimum of the target problem. One can see quantum annealing as a particular type of homotopy optimization. A homotopy optimization for VQE was already proposed in [47] and improved in [58, 87]. However, its applicability for QAOA was only briefly mentioned in [86].

The introduced Hamiltonian-Oriented Homotopy QAOA (HOHo-QAOA), illustrated in Fig. 4.1, decomposes the optimization into several loops. The homotopy map provides a smooth transition between the mixer Hamiltonian and the problem Hamiltonian during the optimization and each loop uses a combination of these two Hamiltonians for cost function evaluation and the quantum state is optimized with respect to such intermediate cost functions. This strategy simplifies the search for good QAOA parameters while keeping the PQC unchanged. To showcase this approach, we investigate the weighted Max-Cut problem on Barabási-Albert graphs. First, we empirically analysed the impact of the choice of the homotopy parameters: the initial α_{init} value and the step parameter α_{step} which defines the difference between two consecutive α values. Setting α_{init} and α_{step} close to zero provides a better approximation to the optimal solution in theory, we empirically show that one can still get a good approximation even if α_{init} and α_{step} are detached from zero. This hugely reduces the computational cost of HOHo-QAOA. Finally, we compare HOHo-QAOA with other commonly used QAOA optimization strategies [41, 132].

By studying the role of the adiabatic path in the QAOA landscape, we aim to uncover insights that could lead to improved optimization strategies or better convergence properties of the algorithm.

4.2 Energy Landscape

Let us take a briefly look at the QAOA, the energy landscape considering a single parameter θ . Given an initial quantum state, if applicable, all the unitary operations that precede the θ -dependent operation are applied, which maps the initial state into a different one. Afterwards, under an assumption of noiseless evolution, an unitary $\exp(-i\theta H)$ corresponding to a mixer or objective Hamiltonian H is applied, followed by the the remaining operations. At the end, the energy estimation with respect to the observable is measured. Given a Hamiltonian H with spectrum set $\{E_1, \dots, E_k\}$, and O be an arbitrary observable. The energy function with respect to θ and a arbitrary state ρ can be computed as

$$\text{tr}(\exp(-i\theta H)\rho \exp(i\theta H)O) = C + \sum_{i>j} A_{i,j} \cos(\theta(E_i - E_j) + B_{i,j}), \quad (4.1)$$

in which $\{E_i\}$ is the set of all eigenvalues of the operator H , and real parameters $C, A_{i,j}, B_{i,j}$ depend on the initial state, observable, and θ -independent quantum

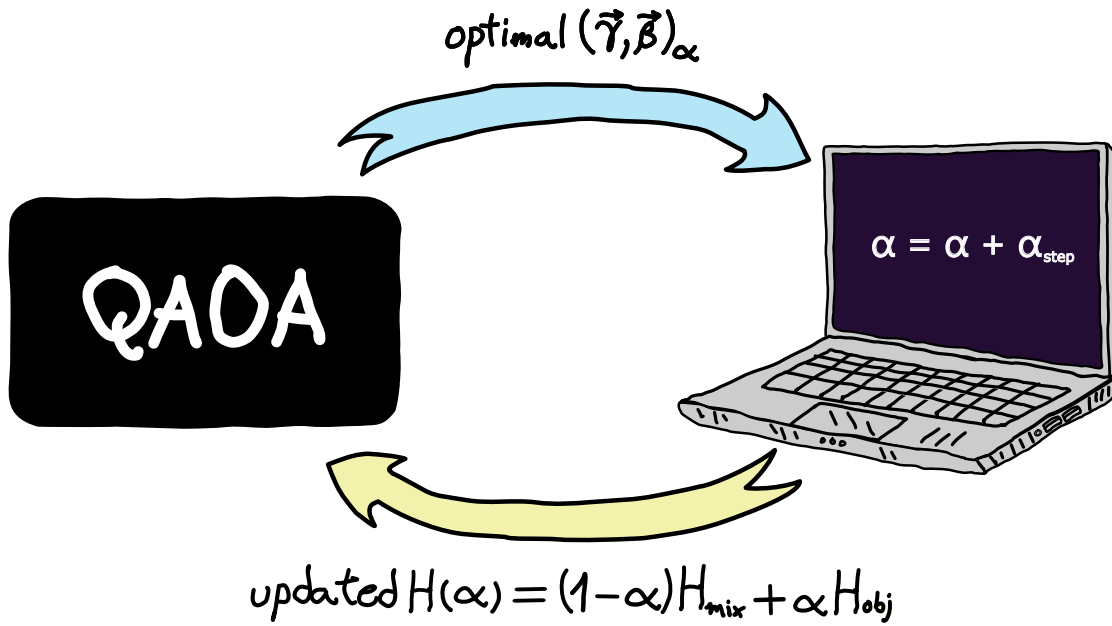


Figure 4.1: Schematic representation of HOHo-QAOA. The algorithm starts with choosing an initial value of parameters (γ, β) according to some probability distribution P , and optimizing them for the initial Hamiltonian H for $\alpha = \alpha_{\text{init}}$ with the chosen classical optimization procedure. Then, the optimal parameters for the ansatz are iteratively used as the initial parameters for the consecutive optimization routines for H with an increased value of α , i.e. $\alpha = \alpha + \Delta\alpha$. The procedure stops at $H = H_{\text{obj}}$ for $\alpha_{\text{init}} = 1$, which is the objective Hamiltonian. Throughout the chapter, α is referred to as a homotopy parameter.

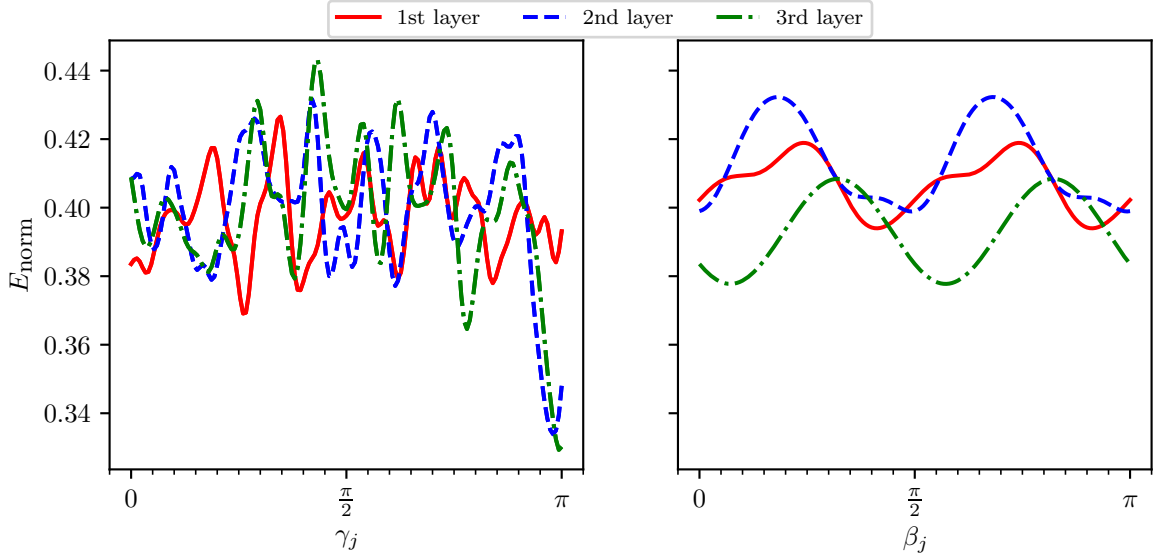


Figure 4.2: Illustration of the highly nonlinear energy landscape of QAOA for Max-Cut for 10 nodes with weighted Barabási-Albert graph for (a) objective Hamiltonian and (b) mixer Hamiltonian. E_{norm} is a normalized energy of the objective Hamiltonian, described in Eq. 4.8, so that the eigenvalues lie in $[0, 1]$.

operations. Note that Eq. (4.1) is highly non-linear, therefore its optimization may be difficult in practice. This is in contrast to typically used VQE approaches, in which the parameter-dependent unitary can be reduced to a single-qubit gate, which in turn may result in a simple, yet powerful gradient-free optimization technique [91, 100].

Unfortunately, the number of cosines in Eq. (4.1) may grow quadratically with a number of distinct eigenvalues of the considered Hamiltonian. In the case of the objective Hamiltonian, the number may be particularly high. While for many simple problems like unweighted Max-Cut or Max-SAT the number of different eigenvalues usually grows polynomially with the size of the data, for weighted Max-Cut each partition may result in a different objective value, which may give $O(2^n)$ different energies in general. A complicated energy landscape can be seen already even for a small and simple instance, see Fig. 4.2. For problems generating such complicated landscapes, more sophisticated methods may be at hand.

4.3 Homotopy optimization method

One of the well-known methods to solve a system of highly nonlinear problems is *homotopy optimization*, where a homotopy map is constructed between two systems.

The solution corresponding to one of the systems is transformed into the solution of the other system. For example, consider the function $f_{\text{targ}}(x)$ which encodes a computationally hard problem and $f_{\text{init}}(x)$ which is a problem with an easy-to-find solution. Then the particular homotopy map between the systems can be given as

$$\mathcal{F}(\alpha, x) = g_1(\alpha)f_{\text{targ}}(x) + g_2(\alpha)f_{\text{init}}(x), \quad 0 \leq \alpha \leq 1, \quad (4.2)$$

where

$$\begin{aligned} g_1(0) &= 0, & g_2(0) &= 1, \\ g_1(1) &= 1, & g_2(1) &= 0. \end{aligned} \quad (4.3)$$

Here, we get a family of problems corresponding to $\min_x \mathcal{F}(\alpha, x) = 0$ for each α value from 0 to 1. We track the optimized solutions starting from $(\alpha, x) = (0, x_0)$, as α moves from 0 to 1, which for a successful homotopy map leads to $(\alpha, x) = (1, x_1)$, where x_1 is ideally the optimal solution of f_{targ} .

The state-of-the-art approach is to start from $(\alpha_{\text{init}}, x_{\text{init}})$ with x_{init} minimizing $\mathcal{F}(0, x) = f_{\text{init}}(x)$. Then the problem $\min_x \mathcal{F}(\alpha + \alpha_{\text{step}}, x) = 0$ is iteratively solved using the solution of $\min_x \mathcal{F}(\alpha, x)$ as a starting point, for sufficiently small $\alpha_{\text{step}} > 0$ [126].

4.4 Hamiltonian-Oriented Homotopy QAOA

The Hamiltonian-oriented homotopy QAOA decomposes the optimization process of the objective Hamiltonian into several optimization loops. Each loop optimizes the energy

$$E_\alpha(\vec{\gamma}, \vec{\beta}) = \langle \vec{\gamma}, \vec{\beta} | H(\alpha) | \vec{\gamma}, \vec{\beta} \rangle, \quad (4.4)$$

where $H(\alpha)$ encodes the homotopy map

$$H(\alpha) = g_1(\alpha)H_{\text{mix}} + g_2(\alpha)H_{\text{obj}}, \quad 0 \leq \alpha \leq 1. \quad (4.5)$$

For $\alpha = 1$ the expectation value in Eq. (4.4) is the energy corresponding to the H_{obj} . While there is a freedom in the choice of g_1 and g_2 , throughout the paper we use a simple case

$$g_1(\alpha) = 1 - \alpha, \quad g_2(\alpha) = \alpha. \quad (4.6)$$

During the optimization process, we choose an initialization of mixer and objective parameters (at $\alpha = 0$) in such a way that the parameters corresponding to the mixer are sampled from the uniform random distribution $U(a, b)$ in an interval $[a = 0, b = 2\pi]$ and the objective parameters are all set to 0. With this initialization we make sure that the homotopy starts from the exact ground state of the mixer

on a noise-free setting, as application of mixer on its eigenstate does not change the state. For $\alpha' > \alpha \geq 0$ the initial parameters are chosen as

$$(\vec{\gamma}, \vec{\beta})_{\alpha'}^{\text{init}} = (\vec{\gamma}, \vec{\beta})_{\alpha}^*, \quad (4.7)$$

here $*$ denotes the optimal parameters for α .

It should be noted that each run of HOHo-QAOA follows the generic structure of homotopy process as in Eq. (4.5) where the ‘‘run-time’’ of HOHo-QAOA is characterized by the α_{step} , for a fixed α_{init} . The parameter α_{init} fixes the initial α value. Generally, it can be inferred that better approximation to the optimal solution can be achieved if we choose a sufficiently small value of α_{step} and α_{init} . They can be described in a more elaborated way as follows. Small value of α_{step} helps us realizing the homotopy of Eq. (4.5) and at the same time if we initiate with $\alpha_{\text{init}} \rightarrow 0$, it becomes easier to find the ground state for the first step. To show this, throughout the paper, we investigate the normalized energy

$$E_{\text{norm}}(E_{\alpha}(\vec{\gamma}, \vec{\beta}), \alpha) = \frac{E_{\alpha}(\vec{\gamma}, \vec{\beta}) - \min H(\alpha)}{\max H(\alpha) - \min H(\alpha)}, \quad (4.8)$$

with respect to parameters of HOHo-QAOA, where $E_{\text{norm}}(\alpha) = 0$, is the normalized ground energy for any $\alpha \in [0, 1]$, and $\min H(\alpha)$ ($\max H(\alpha)$) denotes minimum (maximum) of $H(\alpha)$.

4.4.1 Initialization strategy

In the following, first we numerically discuss proposed settings for initial QAOA parameters $(\vec{\gamma}, \vec{\beta})^{\text{init}}$. With this setting we show that the homotopy parameters i.e. α_{init} , α_{step} can be chosen detached from zero without compromising the efficiency of the method. We consider and optimized energy E_{norm}^* , or in the case of HOHo-QAOA also an intermediate optimized step energy $E_{\text{norm}}^*(\alpha)$. In the numerical results the E_{norm}^* is averaged over 100 experiments. Details of the experiment can be found in Appendix....

For the numerical investigation of optimal QAOA parameters, which is illustrated in Figure 4.3, we consider three possible initialization choices of the mixer and objective parameters at $\alpha = \alpha_{\text{init}}$:

1. RR (Random Random): When the parameters corresponding to mixer and objective Hamiltonians are chosen from a uniform random distribution $U(0, 2\pi)$ i.e. $\gamma_j^{\text{init}} \sim U(0, 2\pi)$, $\beta_j^{\text{init}} \sim U(0, 2\pi)$.
2. NZR (Near-Zero Random): The parameters corresponding to mixer Hamiltonian are chosen from $U(0, 2\pi)$ but objective parameters are sampled from the

values very close zero i.e. $\gamma_j^{\text{init}} \sim U(0, v)$, $\beta_j^{\text{init}} \sim U(0, 2\pi)$, where v is 0.05.

3. ZR (Zero Random): Mixer parameters are sampled from $U(0, 2\pi)$ chosen and objective is all zeros i.e. $\gamma_j^{\text{init}} = 0$, $\beta_j^{\text{init}} \sim U(0, 2\pi)$ as proposed before.

4.5 Results

In this section we analyze the performance of the introduced algorithm with respect to other optimization strategies introduced above.

In order to enable the simple reproduction of our results, we publish our code in <https://doi.org/10.5281/zenodo.7585691>. The algorithms for generating data and plotting were implemented in Julia and Python programming languages. Versions of the software and additional packages are listed in <https://github.com/iitis/hoho-qaoa-code>.

In Figure [4.4](b) we investigate how the efficiency of the optimization depends on the α_{step} . During this investigation, we take 10 layers of HOHo-QAOA. We observe that in the range $10^{-4} \leq \alpha_{\text{step}} < 0.5$ the approximation to the ground energy and the corresponding standard deviation with increasing $\alpha_{\text{step}} \rightarrow 0$ remains almost unchanged, giving rise to a region of stability with respect to α_{step} . This behavior of E_{norm}^* with α_{step} is similar to what we can observe for α_{init} . This lead us to a conclusion that one can choose α_{step} detached from zero for HOHo-QAOA. It should be noted that due to high simulation cost the experiment for 16 qubits is halted at the $\alpha_{\text{step}} = 10^{-2}$, whereas the investigation for 6, 16 qubits is extended to 10^{-4} .

The discussion and numerical results from the previous paragraphs give us the following initialization rules of HOHo-QAOA, which leads to a high efficiency of the method:

1. The parameters of mixer and objective should be initialized with ZR setting i.e. $\gamma_j^{\text{init}} = 0, \beta_j^{\text{init}} \sim U(0, 2\pi)$,
2. Although one can infer that $\alpha_{\text{init}} \rightarrow 0$ along with $\alpha_{\text{step}} \rightarrow 0$ gives the best result, our investigations show that one can choose the homotopy parameters detached from zero. This greatly reduces the cost of simulating HOHo-QAOA

While it is natural for HOHo-QAOA to initialize using ZR strategy, it is unclear whether this choice will improve or worsen the results for QAOA or T-QAOA. Therefore before comparing state-of-the-art methods to the introduced one, we verify whether there is any difference in the performance for QAOA and T-QAOA with respect to the initialization of the optimized angles. In Fig. ... we investigate state-of-the art methods for parameters $(\gamma_j, \beta_j)^{\text{init}}$ initialized with RR

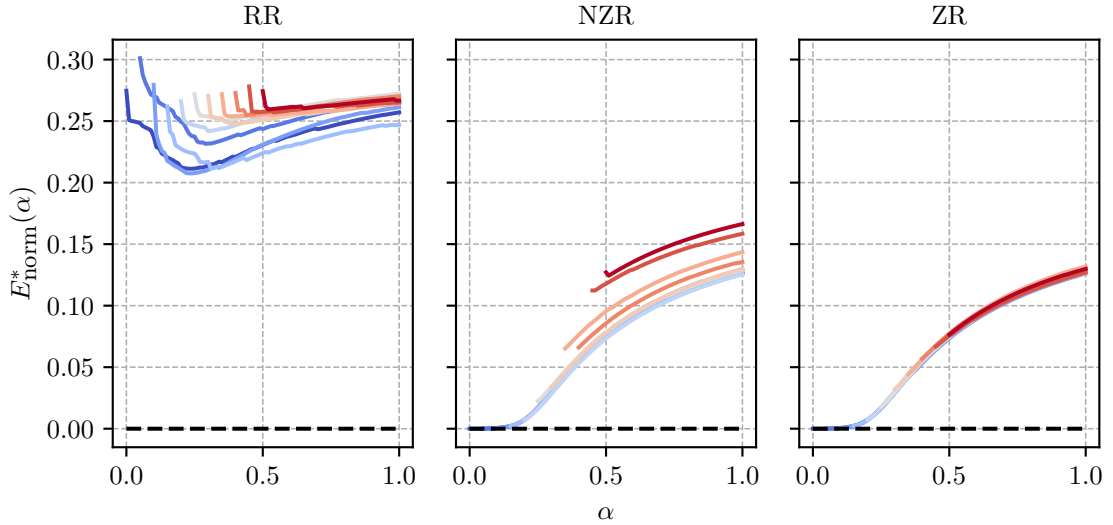


Figure 4.3: The impact of different methods of initialization of γ_j, β_j on HOHo-QAOA. The left, the middle and the right figures are representing the convergence for RR (Random Random), NZR (Near-Zero Random) with parameter $v = 0.05$, and ZR (Zero Random) initialization respectively, see Sec. 4.4.1 for details. It is visible that the ZR is outperforming the other two initializations. Although for $\alpha_{\text{init}} \leq 0.2$ the performance of NZR and ZR are comparable but as we tune $\alpha_{\text{init}} > 0.2$, the minima for NZR scatters in region $0.10 < E_{\text{norm}} < 0.15$ whereas the minima for ZR clusters in a very narrow E_{norm} -width.

and ZR strategy. We observe that the performance of QAOA and T-QAOA is not influenced by the chosen strategies. This justifies using ZR strategy when comparing QAOA, T-QAOA and HOHo-QAOA.

Note that for QAOA we are observing undesired non-monotonic behavior with respect to the number of layers. We claim that this is caused because of a complicated landscape of the energy function, which makes difficult to optimize it if no information about the problem instance is used during the initialization from large number of nodes. This argument is complies with good performance of T-QAOA where the initial parameters of $(L + 1)$ -layer step is evaluated based on local optimal solutions of the L -layers step.

In Fig... we compare the performance of HOHo-QAOA with the other variants when $(\gamma_j, \beta_j)^{\text{init}}$ are initialized using ZR setting. In the first experiment we run the algorithms with a fixed number of nodes while increasing number layers. In the second experiment the number of layers is fix while we vary the number of nodes. The plots present optimized energy values, averaged respectively over 100 and 50 instances. The data shows that the introduced HOHo-QAOA gives us significantly smaller energy in both experiment setups. Good improvements remains as more

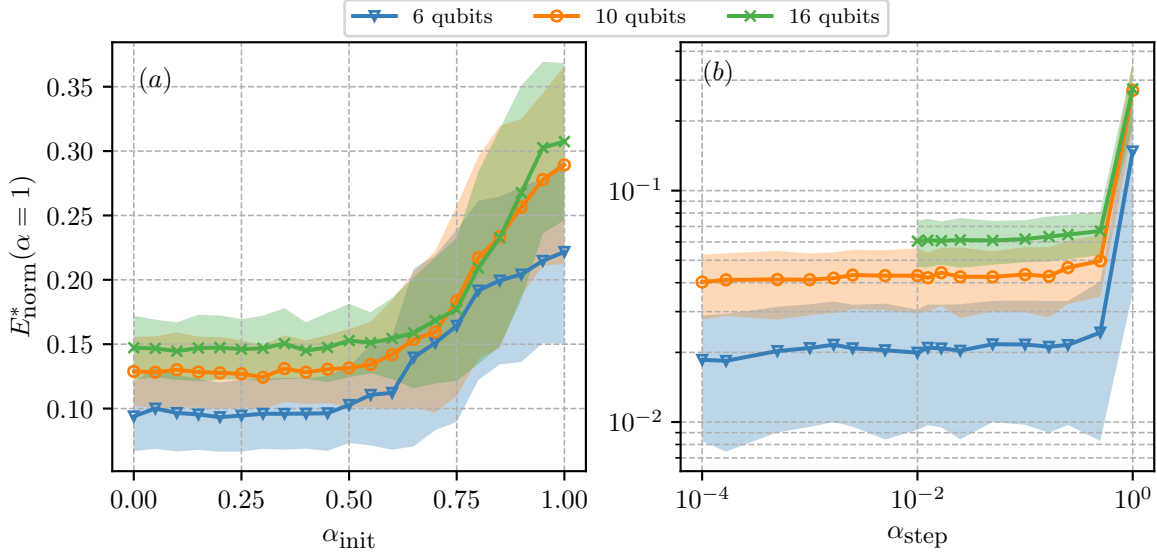


Figure 4.4: We illustrate the dependency of E_{norm}^* with α_{init} and α_{step} . In (a) the variation of E_{norm}^* with α_{init} for 3 layers of HOHo-QAOA is presented, with $\gamma_j^{\text{init}} = 0$, $\beta_j^{\text{init}} \sim U(0, 2\pi)$. In the figure, we see a *region of stability* of HOHo-QAOA in respect with α_{init} in the range 0.0 to 0.50. In (b) we present E_{norm}^* vs α_{step} using 10 layers of HOHo-QAOA. Just like in the case of α_{init} , for α_{step} a same *region of stability* can be observed. This gives us the preference on the choice of *step parameter* while utilizing HOHo-QAOA. It should be noted that the y-axis in (a) is in **linear** scale and whereas in (b) it is in **log** scale. The lines in both the plots are taken α_{init} and α_{step} -wise and is the mean of 100 experiments. The area under the plots are standard deviation of energies.

layers of the HOHo-QAOA are used and also outperforms the other variants of QAOA for higher number of nodes. This conclusions remain valid also for the best sample solution chosen (dashed line). It should be noted that the HOHo-QAOA outperforms QAOA and the T-QAOA in each and every layer starting from initial layer 5 to final layer 100.

4.6 Conclusion

In the article, we present a novel algorithm for combinatorial optimization. The method is a combination of homotopy optimization with an application in QAOA. In our method, the observable used for computing the energy is changed during the optimization process. The process starts with the observable being a mixer, for which the initial state of QAOA is a grounds state, and is slowly moved into the objective Hamiltonian. In addition, we verify that, although traditionally in

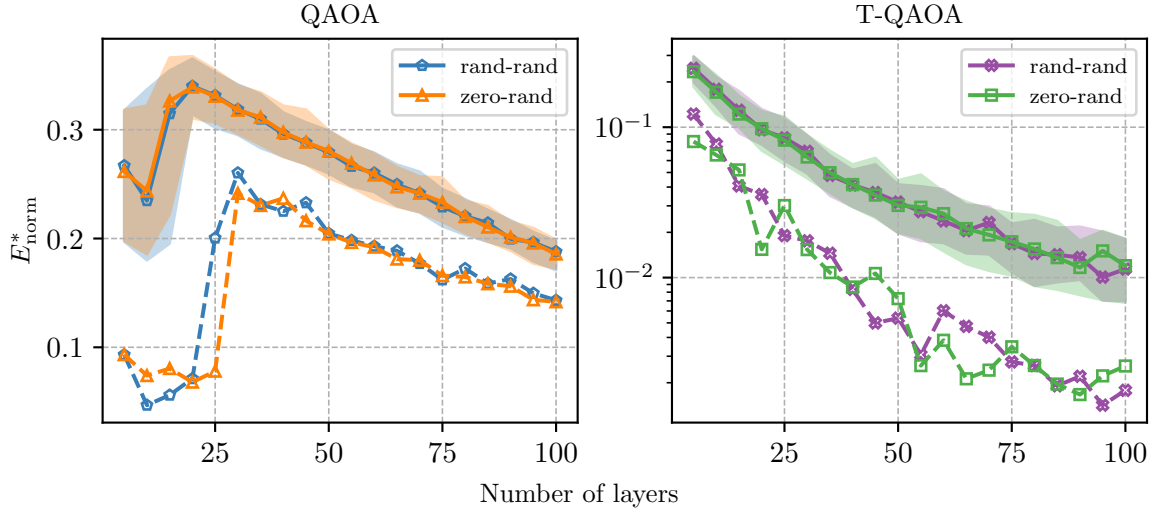


Figure 4.5: Comparison of different initialization for QAOA and T-QAOA. In the left (right) figure we illustrate how the E_{norm}^* changes with increasing number of layers in QAOA (T-QAOA) under the RR and ZR settings. The solid line is the median energy over 100 experiments, meanwhile the dashed line represents the best sample, taken layer-wise and node-wise by choosing the minimum energy among all the experiments. The areas are delimited by the first and third quartile.

the homotopy method, the initial value of transition parameter α should be 0 and the step should be as small as possible, for QAOA the value of considered parameters can be detached from 0. Our investigation of different initializations of HOHo-QAOA helped us to conclude that the Zero Random (ZR) initialization is the optimal choice for the weighed Max-Cut problem. However, the optimal choice of the hyperparameters, or whether we can indeed detach α_{init} from 0, may depend on the particular problem at hand and the size of its instances.

A homotopy optimization is an algorithm dedicated to nonlinear optimized functions, and since even a simple QAOA landscape is a linear combination of many – for some problems exponentially many – sinusoidal functions, our approach is well motivated for such energy function. This is in contrast to the typical VQE optimization process, in which the function landscape with respect to a single parameter is just a sine. By comparing our approach and QAOA algorithm with the typical choice of optimization strategies we numerically confirmed that our method outperforms state-of-the-art approaches.

While our algorithm was only presented for QUBO and X -mixer, it is not restricted to it. In particular, if the transition function is of the form $H(\alpha) = g_1(\alpha)H_{\text{mix}} + g_2(\alpha)H_{\text{obj}}$, we only require the energy of the H_{mixer} to be efficiently

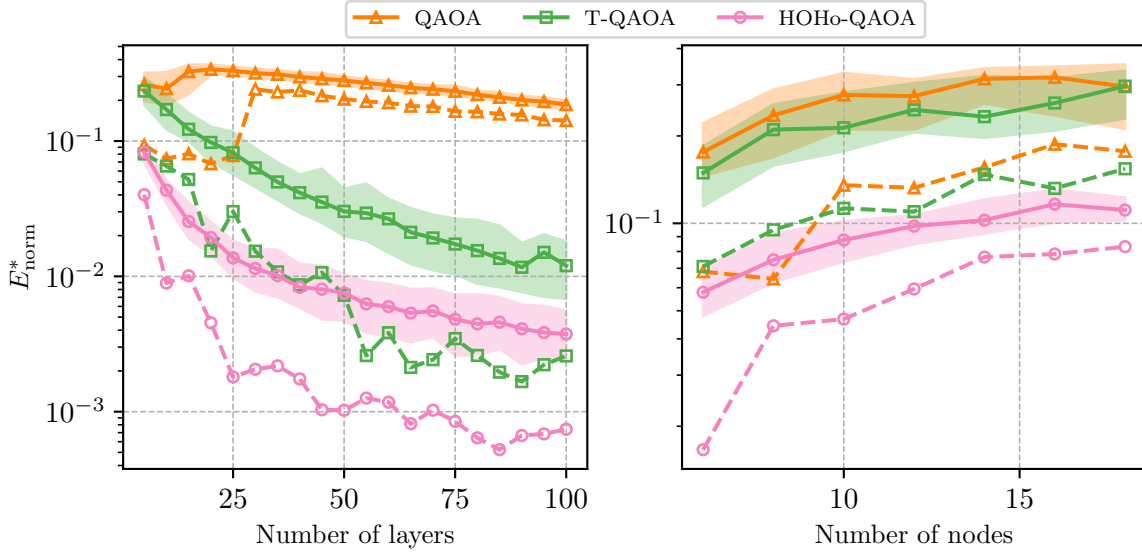


Figure 4.6: Performance of HOHo-QAOA compared to QAOA and T-QAOA. On both figures, for all the QAOA methods, we applied the ZR settings. The areas are delimited by the first and third quartile. The solid line presents E_{norm}^* median over 100 experiments for the left figure and 50 experiments for the right figure, and the dashed line represents the best sample, taken layer-wise and node-wise by choosing the minimum energy among all the experiments. On the left figure, the number of nodes is fixed to 10. On the right, the number of layers is fixed to 5 and the energy is sampled within 6 to 18 nodes. The homotopy parameters are set as $\alpha_{\text{init}} = 0$ and $\alpha_{\text{step}} = 0.01$. One can see that in both cases the averaged energy as well as the best sample of HOHo-QAOA outperforms the other variants of QAOA.

computable and the ground state to be easily prepared. Both the XY-mixer [125] and the Grover mixer [14] satisfy these conditions. Moreover, our approach remains also valid for higher-order binary problems [27, 51, 115] and more advanced pseudo-code-based QAOA Hamiltonian implementation [11].

The introduced HOHo-QAOA uses the same quantum circuit as the standard QAOA, yet it allows reaching quantum states with much lower expectation values. Hence, using this method does not increase the impact of the noise. Compared to T-QAOA, where new layers are added one by one, our algorithm is from the very beginning working on the full circuit. One could expect therefore that HOHo-QAOA overshoots with the number of QAOA layers, as it is not chosen adaptively as for T-QAOA. However, as we observed for the given number of layers, HOHo-QAOA explores the ansatz much better than T-QAOA. Hence, repeating HOHo-QAOA from scratch with a gradually increasing number of layers may lead in fact to

shorter (and thus more noise-robust) quantum circuits compared to the T-QAOA method.

One could expect that the HOHo-QAOA algorithm will be the slowest method of all the considered ones because of the classical optimization being executed for all intermediate α . However, if the functions g_1, g_2 which defines the combination of mixer and objective Hamiltonians are not varying extensively, one should expect that the ground states for most of the intermediate Hamiltonians should be close to neighbouring ones. Therefore, the number of steps to be taken by the classical optimizer with each α change is expected to be much smaller compared to adding a new layer as in T-QAOA with randomly chosen QAOA parameters. Whether this phenomenon will make up for, so that the total optimization time will be comparable to the time required for T-QAOA would require investigating larger instances.

4.7 Chapter Summary

On this chapter, we present a variant of QAOA, the Hamiltonian-oriented homotopy QAOA (HOHo-QAOA): a heuristic method based on classical homotopy optimization. The motivation lies in the ability of homotopic optimization to handle highly non-linear landscapes, such as the energy landscape of QAOA problems. By decomposing the optimization of QAOA into multiple loops using a homotopy map, HOHo-QAOA effectively navigates the non-linear energy landscape to search for low-energy states. The trick is to mimic the search that happens in a adiabatic path, which is known to reach a global minimal. The key concepts are:

- **QAOA:** The QAOA is a variational quantum algorithm dedicated to combinatorial optimization problems which applies a parametrized quantum circuit based on a trotterized adiabatic evolution, i.e., the circuit consists of interchangeably applied so-called mixer and problem Hamiltonians.
- **Energy Landscape:** Considering that, for QAOA, the circuit evolution is done by applying the unitarie $\exp -i\theta H$, where θ is a set of parameters of the circuit and H the Hamiltonian which described the energy of the system, the energy function with respect to θ takes a high non-linear form, making the optimization of QAOA particularly difficult, prone to stuck in local minima.
- **Homotopic Optimization:** One of the well-known methods to solve a system of highly non-linear problems is homotopy optimization, where a homotopy map is constructed between two systems. The solution corresponding to one of the systems is transformed into the solution of the other system by smoothly

Chapter 4. Hamiltonian-Oriented Homotopy QAOA

transitioning between two systems by introducing an interpolating parameter, which we denoted in the chapter as *alpha*, that varies from 0 to 1.

In that sense, we can say that HOHo-QAOA utilizes a homotopy map to create an optimization problem for each value of the interpolating parameter α , enabling the algorithm to iterate through different loops using a combination of mixer and objective Hamiltonians for cost function evaluation. This approach smoothens the transition between the mixer and problem Hamiltonians during optimization, simplifying the search for good QAOA parameters while keeping the quantum circuit unchanged, i.e., HOHo-QAOA uses the same quantum circuit as the standard QAOA, yet it allows reaching quantum states with much lower expectation values. Hence, using this method does not increase the impact of the noise.

We empiric analysed the impact of homotopy parameters, specifically the initial α value and step parameter *alpha* step. It demonstrates that even when these parameters are not close to zero, HOHo-QAOA can still provide a good approximation to the optimal solution, reducing computational costs. By iteratively navigating through these loops, HOHo-QAOA improves the search for low-energy states, outperforming the state-of-the-art QAOA and a variant of heuristic learning of QAOA, know as trajectories QAOA (T-QAOA).

Chapter 4. Hamiltonian-Oriented Homotopy QAOA

Chapter 5

Applications of Quantum Annealing for Music Theory

In this chapter¹, we will discuss about formulating music composition and reduction problems, and solving it using quantum annealing devices.

5.1 Introduction

Music composition can be thought of, in a very simplistic way, as a creative process where one puts combines sound and silences resulting in a sequence that is aesthetic interesting and/or pleasing hear. Over the years, it has been discovered that music commonly associated with soothing and pleasing feelings follows some rules and possess common patterns. Those rules have evolved and solidified up to some extent over the centuries. Yet, there is still certain flexibility keeping open room for creativity.

Being able to identify some rules and common patterns to guide the music composition process is one of the keystones of music theory, therefore, is extensible applied to algorithmic music composition. The seeds of computer-generated music were sown at the end of the nineteenth century by Ada Lovelace, the first computer programmer and also one of the first to recognize the usage of machines beyond calculations. Lovelace put forward the idea that Babbage’s prototype computer, the analytical engine, “might compose elaborate and scientific pieces of music of any degree of complexity or extent” [88]. Yet, this dream was not realized until the 1950s, when the first computer-generated music piece, the Iliac Suite, was composed [61]. Following the advancements in computer science, various methodologies have been explored for generating music, including stochastic approaches, rule-based systems,

¹The content of this chapter is based on the author’s works [10,21] and all the figures in this chapter are taken from, or are adaptations of, the figures present in such work.

evolutionary algorithms, and machine learning [102]. The emergence of quantum computers heralds a new addition to this sequel.

With quantum computers starting to become accessible to the scientific community and enthusiasts, we are witnessing the growth of a new field referred to as quantum computer music [90]. Although the term comprises using quantum computers to generate, perform, and listen to music, we will focus on this chapter on music generation and music reductions, which are problems that could benefit of a new computation paradigm.

Most of the work undertaken so far on music generation using quantum computing is based on the gate-based model. In [69], a simple algorithm named Gatemel is developed to generate music using IBM quantum computers. A classical-quantum algorithm is introduced in [70], which uses Grover's search and follows a rules-based approach for composing music. For quantum algorithms, it was also proposed using quantum walks and quantum natural language processing to compose music.

In this chapter, it is proposed new methodologies for dealing with generating and reducing music using quantum annealing by approaching the problems as an optimization tasks. We consider the problem of music composition from various aspects, among them the composition of melody and rhythm. For music reduction, we treated the problem as a variant of job scheduling, where each machine is a instrument and the jobs are music phrases. Using D-Wave² quantum annealers, we generate music pieces that are displayed in the course of the text.

The rest of the chapter is structured as follows. The first part describes how to formulate the music composition as an optimization problem. We begin with a review of the classical approaches for music composition in the framework of optimization in Sec. 5.2. We describe the fundamental techniques and formulations for music composition using quantum annealing in Sec5.3. In Sec. 5.4 we present the music reduction problem. Finally, we conclude with a discussion and suggestions on future work in Sec. 5.6.

5.2 Music generation and optimization problems

Computational music generation is a field that emerged alongside the rise of digital computers in the late 1940s. The first computer to play a music piece was CSIR Mark-1 [39]. In the initial period, most studies focused reproducing an existing music piece using computers rather than composing a new music piece. Later with the increase of computational power, the focus of researchers shifted to creating new music with the help of computers [17]. Programs and a set of programming languages known as MUSIC-N were developed by Max Mathews at

²D-Wave is the company that produces publicly available quantum annealers. <https://www.dwavesys.com/>

Bell Laboratories in 1957 [84]. Besides, the popularization of video games and synthesizers played a huge role in computational music. This field is also known as algorithmic music [112]. Various tools and techniques for algorithmic music generation have been thoroughly discussed in [89].

Optimization is a widely used technique for computational music generation. As mentioned previously, any piece of music can be seen as a sequence of sounds and rests. These sequences can be recognized by their conformity to a specific musical style or any other musical attribute. Consequently, any divergence from this established proprieties in the generated music leads to a cost or penalty. Therefore, through the optimization process, it is possible to generate a sequence of music elements that minimizes the deviation expected for a pre-identified music style. Now, we will briefly summarize some of the techniques used for computational music generation, where the new composition is achieved through optimizing some parameters related to the music piece.

Statistical modelling is a common technique used in computational music generation. In this approach, the existing music corpus is analysed and its statistical properties are derived. To generate a new music, the process starts with a given note. The next note is chosen by calculating the pitch with the highest probability given the set of parameters based on the previously derived proprieties. The following notes are selected in the same fashion. Thus, sequentially the entire new music piece is obtained. Here it is important to observe that the newly generated music piece follows the statistical properties of the music piece from which the probabilities were calculated [82]. *Illiad Suite* is considered to be the first musical score generated by a computer in 1957 [46,61], using a statistical method known as Markov chains.

Constraint programming is a programming paradigm where instead of steps to solve a problem, the properties of the solution are specified by declaring a set of constraints that must be satisfied. Mainly, it is an expansion of constraint logic programming, which in turn is an expansion of satisfiability problem (SAT) [7]. The set of constraints with a general-purpose search algorithm can solve large practical combinatorial and scheduling problems [122]. Since music composition is a process where at every step, the choices available for the composer are in the form of combinations of notes, chords, or intervals; these combinations are constrained naturally by the rules of music, such as melody or harmony generation rules [121]. This resemblance in the process of music generation and constraint programming has been used in modeling and generation of various musical forms such as counterpoint, harmony, rhythm, form, and instrumentation [5].

Integer linear programming (ILP), which is often referred to as integer programming (IP) in the literature, is another main framework for solving combinatorial problems. Though it shares a lot in common with constraint programming, ILP

only uses variables with integer values to represent the problem as will be described in the following section. Based on our knowledge, integer programming has not been used extensively in the literature in the scope of music generation. Some works in this line are the following. The natural one-to-one mapping between integers and pitch values of the 12-tonal music system has been exploited by Tsubasa Tanaka et al. [117] for musical motif analysis of existing masterpieces of music. Nailson dos Santos Cunha et al. have generated guitar solo music using integer programming [34].

The bottleneck in the ILP is their hardness: ILP are NP-Complete. The usual approach for solving a combinatorial problem via ILP could be via exact algorithms, such as branch and bound, cutting planes method among them. Alternatively, several heuristic methods such as hill-climbing, simulated annealing, ant colony optimization, Hopfield neural networks, tabu search have been used to reach the solution set [32, 53]. The next imperative step would be using quantum methods for solving combinatorial problems, which can be initially formulated as ILP problems, which can be easily converted into a form that is suitable for quantum annealing.

5.3 Music Generation

Now we are going to briefly introduce how to map Melody generation into a problems that can be solved in a quantum annealing device. In the next subsections, we will defining a model for a sequence of notes, followed by the rules to select which notes should be selected and rests. Then, we can define a objective function and the constrains for the problem to be optimized using a QUBO formulation. To conclude the piece, after describing how to generate the pitches for the melody, we take into account the rhythm generation.

5.3.1 Melody Generation

As described in the previously, quantum annealing is a approach to find the optimal solution of a problem encoded as a Ising Hamiltonian. We will start by investigating different ways of formulating the process of melody generation as an optimization problem. When we talk about melody generation, we refer to the generation of the pitches only.

Model

Suppose that we want to generate a melody consisting of n notes, where the pitches belong to the set $P = \{p_1, p_2, \dots, p_k\}$. We will define the binary variables $x_{i,j}$ for

$i \in [n]$ and $j \in P$ as in Eq. 5.1.

$$x_{i,j} = \begin{cases} 1 & \text{note at position } i \text{ is } j \\ 0 & \text{otherwise.} \end{cases} \quad (5.1)$$

The total number of variables required in this formulation is $n|P|$. Note that we aim to generate n notes simultaneously. At the end of the optimization process, we will obtain an assignment to the binary variables $x_{i,j}$, which will indicate the pitch of the note selected for each position.

Next, we will define some constraints which can be included in the objective function using the penalty method. The first rule we need to incorporate is that only one of the variables $x_{i,j}$ is equal to 1 for each position i . The rule is necessary as exactly one pitch should be selected for each time point. This is equivalent to having the constraint defined in Eq. 5.2 for each $i \in [n]$,

$$\sum_{j \in P} x_{i,j} = 1. \quad (5.2)$$

This constraint is the backbone of our formulation, and it should be included in the QUBO using a sufficiently large penalty coefficient as it should never be violated.

Let us go through an example. Let $P = \{\mathbf{C4}, \mathbf{D4}, \mathbf{E4}, \mathbf{G4}\}$ and $n = 5$. The QUBO formulation has 20 variables defined through Eq. 5.1 and penalty terms of the form presented in Eq. 5.3 for each $i = 1, \dots, 5$, that are obtained from Eq 5.2.

$$\left(1 - \sum_{j \in P} x_{i,j}\right)^2. \quad (5.3)$$

At this point, any sequence of 5 notes has 0 energy and is equally likely to be produced and there are in total 4^5 such sequences. The list of the first 5 samples obtained from D-Wave as a result of running the QUBO formulation described above is given in Table 5.1.

The energies of all the samples in Table 5.1 are 0 as no constraint is violated, i.e., precisely one of the variables is 1 for each i . The resulting note sequences are given in Table 5.2

So far, we have only defined a single rule ensuring a single note at each time point. In general, one would like to introduce some more rules while composing a melody as we will discuss next.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
Sample 1	0 0 1 0	0 0 1 0	0 0 0 1	0 1 0 0	0 1 0 0
Sample 2	0 1 0 0	0 0 0 1	1 0 0 0	1 0 0 0	0 1 0 0
Sample 3	0 1 0 0	0 1 0 0	0 0 0 1	0 0 0 1	0 0 1 0
Sample 4	0 0 1 0	0 0 0 1	0 0 1 0	1 0 0 0	0 0 0 1
Sample 5	0 1 0 0	0 0 1 0	1 0 0 0	0 0 1 0	0 1 0 0

Table 5.1: The list of first 5 samples obtained from D-Wave. The columns represent the values of the variables $x_{1,C4}, x_{1,D4}, x_{1,E4}, x_{1,G4}, x_{2,C4}, \dots, x_{5,G4}$ in the given order.

Sample 1	E4 – E4 – G4 – D4 – D4
Sample 2	D4 – G4 – C4 – C4 – D4
Sample 3	D4 – D4 – G4 – G4 – E4
Sample 4	E4 – G4 – E4 – C4 – G4
Sample 5	D4 – E4 – C4 – E4 – D4

Table 5.2: The list of note sequences corresponding to the samples given in Table 5.1.

Rules About Consecutive Notes

Suppose that we want to add a restriction that the note p_l does not appear after the note p_k . This is useful for avoiding particular intervals and amending our model. Such a restriction can be incorporated into the QUBO formulation by adding the term defined in Eq. 5.4 to the objective function multiplied with a suitable penalty coefficient C for each $i \in [n - 1]$,

$$x_{i,p_k} x_{i+1,p_l}. \quad (5.4)$$

Note that when both variables equal 1 simultaneously, a penalty of C is added to the objective function. Alternatively, we can express the same rule using the constraint defined in Eq. 5.5 as

$$x_{i,p_k} + x_{i+1,p_l} \leq 1. \quad (5.5)$$

In this case, the inequality should be first transformed into equality by using slack variables and then added to the objective function.

Now let us consider a rule saying that the same note does not appear more than twice in a row. Similar to what we had above, the term $x_{i,p_j} x_{i+1,p_j} x_{i+2,p_j}$ can

be added to the objective for each $i \in [n - 2]$ and $p_j \in P$. However, this is not a quadratic term, and quadratization is needed to obtain a QUBO. Alternatively, the rule can be expressed by the constraint given in Eq. 5.6, which forces that at most two of the variables are equal to 1 simultaneously,

$$x_{i,p_j} + x_{i+1,p_j} + x_{i+2,p_j} \leq 2. \quad (5.6)$$

Taking the previous example and assuming that $P = \{\mathbf{C4}, \mathbf{D4}, \mathbf{E4}, \mathbf{G4}\}$ and $n = 5$, let us also add the rule that $\mathbf{G4}$ does not follow $\mathbf{D4}$ using Eq. 5.4 and include Eq. 5.6 in our formulation as well. The first 5 samples from the experiment results are listed in Table 5.3. All the sequences in the table have 0 energy and obey the rules we have incorporated.

Sample 1	$\mathbf{G4} - \mathbf{D4} - \mathbf{E4} - \mathbf{G4} - \mathbf{C4}$
Sample 2	$\mathbf{E4} - \mathbf{E4} - \mathbf{C4} - \mathbf{E4} - \mathbf{G4}$
Sample 3	$\mathbf{E4} - \mathbf{D4} - \mathbf{C4} - \mathbf{E4} - \mathbf{G4}$
Sample 4	$\mathbf{D4} - \mathbf{E4} - \mathbf{G4} - \mathbf{D4} - \mathbf{C4}$
Sample 5	$\mathbf{D4} - \mathbf{E4} - \mathbf{G4} - \mathbf{D4} - \mathbf{E4}$

Table 5.3: The list of note sequences obtained from D-Wave after incorporating constraints given in Eq. 5.4 and Eq. 5.6.

Semitones and Augmented Intervals

Now let us investigate the different ways we can choose set P . We can identify the notes through the number of semitones between the lowest pitch and the pitch in consideration. The binary variables $x_{i,j}$ are defined as in Eq. 5.7 for each $i \in [n]$ and $j \in P$.

$$x_{i,j} = \begin{cases} 1, & \text{note at position } i \text{ is } j \text{ semitones} \\ & \text{apart from the lowest pitch of the sequence,} \\ 0, & \text{otherwise.} \end{cases} \quad (5.7)$$

When P is selected as $\{0, 1, 2, \dots, 12\}$, then it represents the notes from a chromatic scale. Note that this representation is independent of the key chosen as the result may be interpreted in any key. For instance, if we let $x_{i,0} = \mathbf{C4}$, subsequently the resulting P is the set of notes from the chromatic scale of \mathbf{C} .

Identifying the notes through semitones, it will be easier for us to model some rules like avoiding particular intervals. Let A be the list of intervals in semitones, that we would like to avoid. This rule can be incorporated by adding the term

defined in Eq. 5.8 to the objective function for all $i \in [n]$, $|j' - j| \in A$,

$$x_{i,j}x_{i+1,j'}. \quad (5.8)$$

Alternatively, it translates to the constraint given in Eq. 5.9, so that whenever an unallowed interval is used, we have a penalty,

$$x_{i,j} + x_{i+1,j'} \leq 1 \text{ for all } i \in [n], |j' - j| \in A. \quad (5.9)$$

We are also capable of taking P as a subset of the chromatic scale. For instance, the set $P = \{0, 2, 4, 5, 7, 9, 11, 12\}$ represents the notes from a major scale. Let us also assume that we would like to set the first and the last pitch of the generated music piece as the first degree of the scale. This can be incorporated by simply adding the terms given in Eq. 5.10 to the objective.

$$(1 - x_{1,0}), \quad (1 - x_{n,0}) \quad (5.10)$$

Let us go over an example. We will let $A = [6, 8, 10, 12]$, so that we would like to avoid the intervals tritone, augmented fifth, augmented sixth, and augmented seventh (octave). Assuming that $P = \{0, 2, 4, 5, 7, 9, 11, 12\}$, $n = 20$ and incorporating rules Eq. 5.2, Eq. 5.6, Eq. 5.8, and Eq. 5.10, we obtain a QUBO formulation. Due to the increased number of constraints, the solution returned by D-Wave QPU violates some constraints and fails to return the ground state. Using D-Wave hybrid solver, the obtained melody interpreted in **C Major** is given in Figure 5.1. Note that not all the constraints are satisfied in this sample as well. This can be viewed both as a caveat and a feature, as the violation of some constraints introduces some randomness to the process.



Figure 5.1: The sequence of semitones is obtained from D-Wave hybrid solver and it is interpreted in **C Major**.

Diatonic Scale and Tendency Notes

Previously, we investigated how one can identify the notes through semitones. All music pieces have a definite key signature. Therefore it is often more suitable to work with pitches from a particular scale.

Chapter 5. Applications of Quantum Annealing for Music Theory

For simplicity, let us consider the 8 degrees of a diatonic scale. In this case, the set P consists of d_1, d_2, \dots, d_8 where d_j is the j 'th degree of the scale. Instead of defining the variables by the pitches, we will define them through degrees for each $i \in [n]$ and $d_j \in P$ as given in the following equation:

$$x_{i,j} = \begin{cases} 1 & \text{note at position } i \text{ is } d_j, \\ 0 & \text{otherwise.} \end{cases} \quad (5.11)$$

As a result of the optimization procedure, we will obtain a degree sequence, which can then be translated into a note sequence based on the chosen scale. Hence, our model is readily adaptable to different scales. Furthermore, the rules described in the previous subsections are still applicable.

Some notes in the scale are less stable than the others which are known as the *tendency notes* and tend to resolve to the stable ones. Let us examine how to incorporate rules about tendency notes. According to the rule, degrees 2, 4, and 6 resolve down by one step, and degree 7 resolves to the octave. To reflect the rule about tendency notes, for each $i \in [n - 1]$, we will add the terms defined in Eq. 5.12 to the objective,

$$x_{i,2}(1 - x_{i+1,1}), \quad x_{i,4}(1 - x_{i+1,3}), \quad x_{i,6}(1 - x_{i+1,5}), \quad x_{i,7}(1 - x_{i+1,8}). \quad (5.12)$$

Using equations (5.10) to (5.12) to formulate our model and setting $n = 20$, the degree sequence obtained from D-Wave corresponding to one of the samples with the lowest energy is given in Figure 5.2. The degree sequence is interpreted for different scales.



Figure 5.2: The degree sequence is obtained from D-Wave and it is interpreted in C Major and G Minor (natural) in the presented music scores, respectively.

Objective Function

Having discussed how to incorporate different constraints into the model, we can now explore how we can modify the objective function to differentiate between the feasible solutions. When we have only the constraints, all sequences of notes that do not violate any of the constraints are feasible solutions, and they are equally likely to be sampled since they have the lowest possible energy. Any violated constraint increases the energy value by the penalty value of the constraint. Note that as mentioned earlier, sometimes we would like particular constraints to be never violated (like Eq. 5.2), while a solution in which some constraints are violated can still be desirable (avoidance of particular intervals).

In order to differentiate between the feasible solutions, we can give some “rewards” to a particular sequence of notes, i.e. decrease their energy. For instance, we might give a higher reward for pitch D4 following C4 vs. pitch E4 following C4. The way to accomplish this is to have the term given in Eq. 5.13 in the objective function,

$$- \sum_{\substack{i \in [n-1] \\ j, j' \in P}} W_{j,j'} x_{i,j} x_{i+1,j'}. \quad (5.13)$$

Here, $W_{j,j'}$ is the weight associated with having note j' after note j . The larger the weight, the higher the reward we have in the objective function. Note that we have a negative sign in front in Eq. 5.13, as we have a minimization problem and want to decrease the energy. The weights can be determined by analyzing some music pieces and forming a transition matrix of weights examining the consecutive notes. Below, we illustrate this idea through a simple music piece.



Figure 5.3: An excerpt from Beethoven’s Ode to Joy.

In Figure 5.3, an excerpt from Beethoven’s Ode to Joy is given. To identify the weights, we count the occurrences of consecutive pairs. To start with, we identify the occurrences of each note and then count the number of times the note is followed by another particular note. For instance, the note F#4 appears at positions 1, 2, 7, 12, 13. It is followed by F#4 and E4 twice, and by G4 once. Hence, we can deduce the weights given as

$$W_{F\#4,E4} = 2, \quad W_{F\#4,F\#4} = 2, \quad W_{F\#4,G4} = 1. \quad (5.14)$$

As the selected music piece is a short one, it does not contain all the notes from P and for those notes i , the corresponding weight $W_{i,j} = 0$ for all j . We would like to remark that having 0 as the weight does not imply that the corresponding note tuple is always avoided but means that we don't give additional rewards to such pairs. The list of all non-zero weights is defined as

$$\begin{aligned}
 W_{F\#4,E4} &= 2, & W_{F\#4,F\#4} &= 2, & W_{F\#4,G4} &= 1, & (5.15) \\
 W_{G4,F\#4} &= 1, & W_{G4,A4} &= 1, \\
 W_{A4,G4} &= 1, & W_{A4,A4} &= 1, \\
 W_{E4,D4} &= 1, & W_{E4,E4} &= 1, & W_{E4,F\#4} &= 1, \\
 W_{D4,D4} &= 1, & W_{D4,E4} &= 1.
 \end{aligned}$$

Note that taking the number of occurrences as the weights, we are also giving rewards to pitches that appear more frequently than the other. For instance, F# appears five times, and the overall reward is higher when more F#'s appear in the sequence.

Now the question is how should we select set P . We can use Eq. 5.11 to define our binary variables as the degrees from a scale. Hence, the matrix W defines the transition weights between the scale degrees. Note that the newly generated music piece mimics the one from which the transition weights are obtained. The longer the music piece, the better estimates are obtained for the weights. Multiple pieces can be selected as well, paying attention that they are from the same scale, or in case considering pieces from different scales, one should take degrees of the scale instead of the pitches when calculating the weights.

We define the QUBO formulation defined through binary variables given in Eq. 5.11 using the constraints defined in Eq. 5.2, Eq. 5.10, Eq. 5.6, Eq. 5.12 and Eq. 5.13 as the objective function with the weights obtained from Eq. 5.15. Note that in this case, one needs to properly set the penalty coefficients; in case the constraint is violated and there is a reward, an increase in the energy due to the penalty should be larger than the decrease in the energy due to reward. The resulting music piece obtained from the D-Wave hybrid solver is given in Figure 5.4. We interpreted the obtained degree sequence in D Major.

Alternatively, instead of generating weights for the transitions between individual degrees, we can collect statistics about the different intervals used in the music piece and how often they appear. Then accordingly, we can reward the intervals that occur more frequently.



Figure 5.4: The degree sequence is obtained from D-Wave using the transition weights from Ode to Joy and it is interpreted in D Major.

Rests

The sets we have considered so far only consisted of the pitches; however, we may want to include rests in the music piece as well. In this case, a rest element can be included in the set P with appropriate rules. For instance, we may want to avoid two consecutive rests, which we can easily accomplish with the rules we have shown for consecutive notes. In addition, we can set the number of rests used in the music piece by introducing the constraint given in Eq. 5.16.

$$\sum_i x_{i,r} = k, \quad (5.16)$$

where k is the total number of rests we want in the music piece and $x_{i,r}$ denotes that note at position i is a rest.

5.3.2 Rhythm Generation

So far, we have discussed generating the pitches of the melody. In this section, we additionally take into account the rhythm.

Rhythmic Sequence

When generating a music piece, one option would be to generate the pitch sequence and the rhythmic sequence separately. In such a case, the idea will be similar to what we had previously. The set S will consist of possible durations for the notes, such as whole, half, quarter, etc. The binary variables $y_{i,j}$ for $i \in [n]$ and $j \in D$ will be defined as

$$y_{i,j} = \begin{cases} 1 & \text{note at position } i \text{ has duration } j, \\ 0 & \text{otherwise.} \end{cases} \quad (5.17)$$

Similarly, the first rule we need to incorporate is that only one of the variables $y_{i,j}$ is equal to 1 for each position i , which is expressed using the constraint given

takes into account the duration of each type of note as given as

$$\sum_{\substack{i \in [n] \\ j \in D}} d(j)y_{i,j} = L, \quad (5.21)$$

where $d(j)$ is the duration of j and L is the total length of the music piece in terms of eighth notes so that $d(\mathbf{E}) = 1$, $d(\mathbf{Q}) = 2$, $d(\mathbf{DQ}) = 3$ and $d(\mathbf{H}) = 4$. We discretize the durations in terms of eighth notes as it is the note with shortest duration in our example.

Rhythm and Pitch Generated Together

Another alternative is to consider pitches together with their durations. If P is the set of possible pitches, and D is the set of possible durations, then overall, there will be $|D||P|$ possibilities for each note. Assuming there are n notes, the number of variables we need significantly increases to $n|D||P|$ in this case. The binary variables for $i \in [n]$, $j \in P$ and $k \in D$ take the form presented in Eq. 5.22.

$$x_{i,j,k} = \begin{cases} 1 & \text{note at position } i \text{ is pitch } j \text{ and has duration } k, \\ 0 & \text{otherwise.} \end{cases} \quad (5.22)$$

The previous rules we defined about the consecutive notes and intervals apply here too. Those constraints are included independent of k , the note's duration. Likewise, we can still take an objective function based on another music piece. This time, we inspect the number of occurrences of consecutive pitch-duration pairs. However, we would like to remark that the performance of the quantum annealers decreases as the number of variables increases. Hence, it is often desirable to have models with a smaller number of variables.

5.4 Music Reduction

Music reduction is the task of selecting particular parts - often called phrases - of a music piece involving multiple instruments to be played with a smaller number of instruments. By mapping phrases to jobs and the number of instruments desired in the final music piece to the number of machines, we obtain a problem where the starting and ending time of each job are fixed and the goal is to assign jobs to the machines. In this scenario, idle time means that some particular instrument is not playing any music for that duration, which should be minimized to obtain a final music piece that captures the original music piece as much as possible. Consequently, the weight of each phrase reflects how important and rich a particular phrase is in

reflecting the main melody of the music piece.

In previous research, several approaches to the automatic arrangement of ensemble or orchestral pieces for single instruments have been considered. A variety of methods for music reduction and phrase selection were applied previously such as machine learning [31], state-transition models [98], hidden Markov model [63], entropy optimization [62], and local boundary detection model [78]. Besides the extraction of information, those works also consider the playability of the newly arranged pieces.

To build the problem instance, the first step is the identification of the musical phrases and their weights. Subsequently, we present some experimental results.

5.4.1 Phrase identification

A phrase in a melody is a sequence of notes that express a musical idea on its own. The ending of a phrase may be marked with relative lengthening of the last note, intensity or timbral change, or the presence of rests or pauses [26, 97, 120]. Identification of musical phrases has been considered an essential part of music perception by humans [74, 94]. Meanwhile, the automated detection of phrases has been considered by many researchers. Some approaches make use of machine learning techniques such as deep neural networks [56] or use statistical modelling [104] and rule-based approaches which aim to identify the points of change in different musical parameters such as pitch and rhythm [77, 92]. In [78], the phrase selection is based on computing the cost function with the information entropy of the phrase.

We used the Local Boundary Detection Model (LBDM) proposed by [26] for identifying musical phrases. First, sequences of pitch, interonset (duration between the beginning of two consecutive notes), and rest intervals are obtained from a given melodic sequence. For each interval, boundary strength values are calculated and combined to obtain the melody's overall local boundary strength profile. Boundary strength values are proportional to the degree of change between two consecutive intervals, and the boundary introduced on the larger interval is proportionally stronger. The peaks of this sequence indicate the local boundaries.

To identify the peaks of the sequence, one needs to set a threshold value. However, based on the threshold value, the phrases' length may become too small or too large. To overcome this issue, we developed the algorithm described in Algorithm 1.

5.4.2 Determining the weight for each phrase

The goal of music reduction is to highlight a song's most distinctive features and capture the main melody while reducing the number of instruments. The main

melody usually consists of pitch and rhythm rich in information. More formally, such melodic phrases have higher entropy regarding information theory. In order to quantify the amount of information in a phrase, we used the approach presented in [78], which identifies pitch and interonset interval (IOI) entropy. The pitch entropy corresponds to the frequency variety in a scale occurring in a sequence of notes comprising a phrase. In the case of chords, which contain multiple pitches, we only take into account the highest note in the chord for entropy calculation because listeners tend to focus on the notes with the highest pitches.

Mathematically, the entropy for a random variable X with possible values x_1, \dots, x_n is given by

$$H(X) = - \sum_{i=1}^n P(x_i) \log_2 P(x_i), \quad (5.23)$$

where $P(x_i)$ is the probability that X takes value x_i . Suppose that we represent with x_i the possible pitch values that can occur in a phrase. Then, the probabilities $P(x_i)$ are computed by

$$P(x_i) = \frac{n_i}{N}, \quad (5.24)$$

where n_i is the number of occurrences of the i th pitch in the phrase and the phrase contains N notes. The melody entropy for each phrase is calculated based on this probability calculation.

In addition to pitch, rhythm is also a factor in determining the amount of information a phrase contains. Compared with the duration of notes, the IOI is easier to perceive. Similarly to the pitch, we calculate the IOI entropy based on the its probability, with x_i as the possible IOI value in a set of N different IOIs with n_i occurrences in a phrase.

The underlying assumption is that compared to the accompaniment, the rhythm in the main melody is more complicated and more difficult to predict. That means the combined pitch and IOI entropy of the phrase in the main melody part should be the largest.

5.5 Optimization algorithm

In this section, we will first describe the QUBO formulation for OFISP_{min-i} problem and then discuss the classical post-processing step.

5.5.1 QUBO formulation

Let us define the binary variable x_i such that

$$x_i = \begin{cases} 1, & \text{if job } b_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \quad (5.25)$$

for $i = 1, \dots, N$.

Assuming that the time intervals for jobs are given by discrete time units (minutes, seconds), we need the following constraint:

$$\sum_{i: k \in [s_i, e_i]} x_i \leq M \quad \text{for } k = 0, \dots, K. \quad (5.26)$$

This constraint ensures that more than M jobs are not selected for any time point and can be considered a hard constraint. Any solution violating this condition would be an infeasible solution.

We also would like to have exactly M jobs assigned to each time point to minimize idle time, although this may not always be possible. Hence, we introduce the following constraint:

$$\sum_{i: k \in [s_i, e_i]} x_i = M \quad \text{for } k = 0, \dots, K. \quad (5.27)$$

Using the first constraint together with this additional condition which can be considered as a soft constraint, we penalize the cases where the number of selected jobs x_i exceeds M with a larger penalty than the case fewer jobs are selected. That also complies with the goal of minimization of idle time in case it is only possible to find a solution with idle time.

The goal is to maximize the total weight of the selected jobs:

$$\sum_{i=1}^N w_i x_i. \quad (5.28)$$

Incorporating the constraints presented in Eq. (5.27) and Eq. (5.26), we present the following QUBO formulation:

$$\sum_{i=1}^N w_i x_i + P_1 \sum_{k=1}^K \left(\sum_{i: k \in [s_i, e_i]} x_i + E(\xi_k) - M \right)^2 + P_2 \sum_{k=1}^K \left(\sum_{i: k \in [s_i, e_i]} x_i - M \right)^2. \quad (5.29)$$

In the given formulation, $E(\xi_k)$ is the binary representation of the integer

slack variable ξ_k , which is needed for converting the inequality constraint given in Eq. (5.26) into equality. P_1 and P_2 are the penalty coefficients and P_1 should be selected larger than P_2 . For the given formulation, the number of variables required is upper bounded by $O(N + K \log M)$: we need N binary variables to represent x_i , and $K \log M$ variables to represent K slack variables each ranging from 0 to M .

5.5.2 Post-processing

Note that the obtained solution does not give us information about the job machine assignment but only the list of selected jobs. Nevertheless, given the list of selected jobs, the assignment can be determined using a classical greedy algorithm typically used for *interval partitioning*. The original problem aims to find an assignment with the minimum possible number of classrooms to schedule all the lectures with fixed starting and ending times. If we adapt this to our case, the lectures become jobs, and classrooms become machines. In this approach, the jobs are sorted according to their starting times and assigned greedily to machines. The algorithm has time complexity $O(N \log N)$ [73], where N is the number of jobs. We describe the algorithm in detail in Algorithm 1 for completeness.

Algorithm 1 Pseudocode for job machine assignment

Input: J – the list of selected jobs
Sort J by starting time in non-decreasing order
 $m \leftarrow 1$
for j in J **do**
 if j is compatible with some machine $k \in \{1, \dots, m\}$ **then**
 Assign j to k
 else
 $m \leftarrow m + 1$
 Assign j to m

5.5.3 Enhancing the model

On top of the formulation presented above, one can introduce further constraints. For instance, it might be the case that we do not want particular jobs to be selected simultaneously. In such a case, it would be enough to include the term $Px_i x_j$ for the specific jobs b_i and b_j , where P is a suitable penalty coefficient. Note that one can prioritize some jobs by increasing their weights further.

In the presented QUBO formulation, we assumed that all the machines were identical. The case of unidentical machines is commonly considered in the literature [8]. In that case, we have the restriction about machines on which a job can be

processed. Let us denote the set of machines on which job b_i can be processed by $R_i \subseteq R$. To model the problem, we will introduce the binary variables indexed by the job number and the assigned machine, as in the case of the ILP for OFISP. Let x_{ij} be the binary variable such that

$$x_{ij} = \begin{cases} 1, & \text{if job } b_i \text{ is assigned to machine } r_j \\ 0, & \text{otherwise.} \end{cases} \quad (5.30)$$

for $i = 1, \dots, N$ and $j = 1, \dots, M$. We need Eq., which forces each job to be assigned to a single machine, and the objective stays as in Eq.. Additionally, we need the following constraints:

$$\sum_{i: k \in [s_i, e_i]} \sum_{j=1}^M x_{ij} \leq M \quad \text{for } k = 0, \dots, K, \quad (5.31)$$

$$\sum_{i: k \in [s_i, e_i]} \sum_{j=1}^M x_{ij} = M \quad \text{for } k = 0, \dots, K. \quad (5.32)$$

Now, to make sure that a job is not assigned to an incompatible machine, we add the terms Px_{ij} , for those i, j pairs where $r_j \notin R_i$. Hence, any incompatible assignment is penalized. Note that the number of variables required for this formulation is $N \cdot M$ instead of N , without counting the slack variables.

5.5.4 Results

We chose two classical compositions for our experiments: Suite No. 3 in D major, BWV 1068, Air, and Beethoven's Symphony No. 7 in A major, Op. 92, Second Movement. The music sheets are encoded as MIDI files with public domain licenses and parsed with `music21` [35] library. Our goal was to reduce the given music piece to two tracks. To start with, we classically processed the phrase selection algorithm, identified the phrases, and created the QUBO formulation using `pyqubo` [131] library.

We conducted the annealing experiments with the available solvers provided by D-Wave, which included simulated, quantum, and hybrid annealers. The parameters for simulated annealing used in our experiment are the number of reads $n_r = 1000$ and the number of sweeps $n_s = 1000$. We used two quantum processing units (QPUs) in the quantum annealing experiments. The first one, `Advantage_system4.1`, provides 5627 qubits interconnected by Pegasus graph topology. The second device, `Advantage2_prototype1.1`, has 563 qubits with the underlying Zephyr topology. The tested parameters are discussed further below.

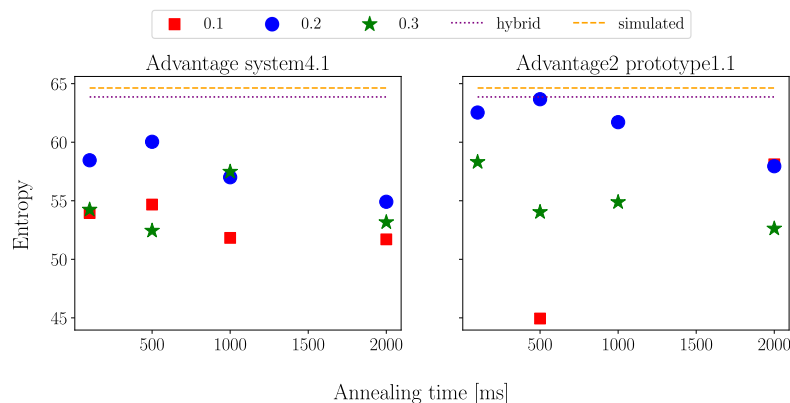


Figure 5.6: Experiment results for entropy varying chain strength and annealing times. The dashed lines correspond to the simulated annealing best result and the dotted lines to the hybrid solver solution. The square, round, and star dots correspond to chain strength values 0.1, 0.2, and 0.3, respectively

For the hybrid solver, no parameters are set, and it returns a single solution.

When running a problem on D-Wave QPUs, the variables should be mapped to the QPU architecture, as the underlying graph representing the interactions in the QPU is not fully connected. This process is known as minor embedding, where multiple physical qubits represent a single logical qubit named a chain. The coupling between those qubits is called the chain strength, which should be accordingly set so that it is not too large to override the actual problem, yet it is not small so that the chain is not broken [127]. The problems were automatically embedded via D-Wave’s native probabilistic embedding algorithm.

For the quantum annealing solvers, we vary the parameters for chain strength as $c_r = 0.1, 0.2, 0.3$ and the annealing time as $t = 100, 500, 1000, 2000$. The number of reads and the annealing time are subject to the relation $n_r \cdot t < 10^6$, and we picked the maximum possible number of reads based on this relation.

The solutions in which hard constraints defined in Eq. 5.26 are violated are considered infeasible, and the violations of Eq.5.27 are deemed as soft violations. The results point to two solutions; the first is based on the sample with the highest entropy, and the second is based on the sample with the minimum number of soft violations. We decided to proceed with higher entropy solutions to maximize the information from the original song. The code and its description used to generate the experiment is available on <https://doi.org/10.5281/zenodo.7410349>.

Suite No. 3 in D major, BWV 1068, Air For this composition, which comprehends four tracks with a length of 19 measures, we identified the set of jobs from the 41 identified phrases. From this, the number of QUBO variables is 80.

Advantage_system4.1			
	Chain Strength		
Annealing Time	0.1	0.2	0.3
100	0	0	0
500	4	0	1
1000	3	0	1
2000	1	0	1

Table 5.4: Number of soft constraint violations for Suite No. 3 in D major, BWV 1068, Advantage_system4.1.

The results are presented in Fig 5.6. We compared the QPU results with simulated annealing and hybrid solver, which returned feasible solutions with no soft constraint violation and higher entropy. Feasible solutions were obtained for both Advantage_system4.1 and Advantage2_prototype1.1 QPUs, with soft violations. The number of physical variables for Advantage_system4.1 varied in a range of 107 to 130 qubits and for Advantage2_prototype1.1 from 106 to 126. The performances of both QPUs for chain strength $c_r = 0.2$ were absent of soft constraint violations and yielded the highest entropy.

Symphony No. 7 in A major, Op. 92, Second Movement We decided to use for the second experiment a more challenging composition with 12 tracks of length 276 measures. We identified 591 phrases to build the set of jobs and, respectively, 1056 QUBO variables.

The results for simulated annealing and Hybrid Solver returned feasible solutions with entropy values of 463.05 and 441.99, respectively. The solutions contain around the same amount of soft constraint violations, around 16% of the measures. Due to the number qubit requirements, the problem did not fit into Advantage2_prototype1.1. The number of physical variables required to solve the problem lay in the range of 2700 to 3020 for Advantage_system4.1; however, no feasible solutions were obtained. For more details about the number of violated constraints, we refer to Tables tables 5.4 to 5.7.

Advantage2_prototype1.1			
	Chain Strength		
Annealing Time	0.1	0.2	0.3
100	4	0	0
500	2	0	0
1000	5	0	0
2000	0	0	1

Table 5.5: Number of soft constraint violations for Suite No. 3 in D major, BWV 1068, Advantage2_prototype1.1.

Soft Violations			
	Chain Strength		
Annealing Time	0.1	0.2	0.3
100	149	157	163
500	153	162	167
1000	151	161	161
2000	149	162	170

Table 5.6: Number of soft constraint violations for for Symphony No. 7 in A major, Op. 92, Second Movement, Advantage_system4.1.

Hard Violations			
Chain Strength			
Annealing Time	0.1	0.2	0.3
100	26	40	33
500	25	45	40
1000	34	42	54
2000	26	41	54

Table 5.7: Number of hard constraint violations for for Symphony No. 7 in A major, Op. 92, Second Movement, `Advantage_system4.1`.

5.6 Conclusions

The main goal of this chapter was to lay the groundwork for music composition and music reduction using quantum annealing. We introduced various formulations and demonstrated how music rules and styles could be incorporated into the model. Prior to this work, the majority of the studies on music composition and music reduction have focused on gate-based quantum computers. Notwithstanding the previous efforts [44, 71], this is the first time quantum annealing is used for melodic music composition.

The presented methodologies will be intriguing for composers, quantum computing researchers, and the community of algorithmic music generation and allow interested readers to build upon the given notions. Being the first comprehensive study on the nascent field of music composition using quantum annealing, we believe that it will foster the development of the area. As a side contribution, a novel application area for quantum annealing is demonstrated within this chapter.

The popularity of quantum annealing relies on the promise that one may gain speedup by exploiting quantum effects when solving optimization problems. We would like to emphasize that creativity is in the foreground instead of potential quantum speedup throughout the chapter. Nonetheless, it is envisaged that it will become possible to solve larger-scale problems and obtain higher quality results faster than the known classical techniques with technological advancements. As the technologies evolve, the ideas discussed here can be leveraged for building larger-scale and more sophisticated models, and it may also be possible to exploit quantum annealing for speedup when composing music.

The study has opened up a vast amount of directions to investigate. First of all, the models can be ameliorated to incorporate further rules. This might be beneficial for harmony generation in particular, as the traditional rules for harmonization have complex structures. In addition, one may adopt the approach we have taken for defining the objection function while generating a melody, for the Markov random fields. The harmonic progressions can be identified from an existing music piece, and the potentials can be set in accordance. One may focus on generating specific music types, such as counterpoint music, which heavily depends on rules. As a natural progression, a target problem would be music completion. Music completion refers to the task of replacing the missing notes in a given music piece. It is a suitable candidate problem as it can be defined as the problem of maximizing the likelihood that a sequence of notes is selected to replace the missing ones. Besides music creation, one may consider other problems from the domain of music, such as music arrangement, where the aim is to arrange a given music piece to a given list of instruments by reduction. The list can be expanded to include any problem that can be expressed as an optimization task.

Regarding the music reduction problem, observations in the conducted experiments suggest that one can further tune the model. Taking into account the enhancements proposed in Sec. 5.5.3, one can avoid particular phrases being played simultaneously, which might create dissonance, or give higher weights to phrases from specific instruments. Furthermore, for a given phrase, one can indicate which instruments can play the phrase at the cost of having a larger number of variables.

5.7 Chapter Summary

For this chapter, we proposed a formulation of music composition and music reduction as optimization problems to be solved using quantum annealing.

- Music generation is not usually treated as ILP problem, but this formulation can be easily mapped into a QUBO form, which is suitable for quantum annealing devices.
- We can treat the choice of selecting a notes and pauses based on a set of rules, which can be translated into a model of constraints.
- The constraints are mapped into penalties, and incorporated to a QUBO formulation.
- The QUBO is then passed to a quantum annealing devices and we can obtain the a sequence of notes and pauses that are optimal to the cost function, generating a melody.

Chapter 5. Applications of Quantum Annealing for Music Theory

- Music reduction can be seen as an special case of a job scheduling with fixed time interval.
- The goal of this optimization is to diminish the idle time in the machines (instruments) while also selecting the most informational rich phrases, which gives more identity to the song.
- The problem is formulated also as QUBO.
- For larger instances, the problem seems underperform compared to the classical counterpart simulated annealing method.

Chapter 5. Applications of Quantum Annealing for Music Theory

Chapter 6

Conclusions

In this thesis, we investigated the application of hybrid quantum computing techniques for solving combinatorial problems with two distinct families of NISQ friendly algorithms: the Quantum Approximation Optimization Algorithm and the Quantum Annealing methods. For the gate based approaches, we focus on improving the performance of the algorithms with error mitigation or homotopy optimization. And for quantum annealing, we explored the possibility of performing creative and artistic work.

Our results can be summarized as:

- The variations of QAOA proposed in this thesis showed improvement compared with state of the art version;
- Quantum Annealing can provide solutions for music composition and reduction, once they are formulated as optimization problems;
- None of the methods investigated shows advantage over classical algorithms for solving combinatorial problems. In fact, for some instances of music reduction, specially the most challenging ones, the Quantum Annealing approached returned worst results compared to Simulated Annealing.

We believed that those results are expected in the scope of a early stage technology. The possible drawbacks to mention are:

- Problems with parsing and encoding data to quantum devices. The choice can heavily affect both size and depth of resulting quantum circuit, choice of gates and some architecture a more sensible affect by those constrains;
- The problem should take in consideration the architectures nuances, and are mostly ignored during the simulations; since most of the quantum algorithms design so far are device agnostic;

Chapter 6. Conclusions

- The cost of running the algorithms in theory versus real devices can diverge significantly. Therefore, minimizing the advantaged since the number of operations can be in reality drastically different from the device agnostic approaches, which takes in consideration only the number of gates and operations applied in logical qubits.

One can say that it is also worth investigating if VQA and Quantum Annealing would ever provide advantage even in the situation of a fault-tolerant quantum device. Contrary to other conventional quantum algorithms, such as Phase Estimation, Amplitude Amplification and Quantum Fourier Transform, VQAs and QA have no substantial proof of advantage. In the case of QA, once the problem is encoded as a QUBO or Ising Hamiltonian, it should also take in consideration embedding it to the qubits topology. Such problem is known to be NP-Hard. In other words, if one wishes to solve a NP problem, it is also creating another, that could be hard as the original one, just to run the solution in a QA device.

Due to its nature, *quantum computers are hybrid* by default. However, at the current paradigm, the computation heavily relies on the classical side. The classical computer not only processes substantial part of the data, going even further as actually computing several steps of the solutions, but also acts as the subtract to set the instructions as a meta-programming platform.

It is safe to say that, even though quantum devices at this point do not match for classical devices, they do push the later to evolve and adapt. Inspired by quantum algorithms, a trend of “dequantizing” quantum algorithm, the classical methods benefit with more efficient and resilient approaches inspired by the quantum counterparts. Attempts to “dequantizing” claims of quantum advantage [118] and post-quantum cryptography [15] are the most prominent examples.

Sometimes we have to stop or even step back if we want to move forward.

Bibliography

- [1] Honeywell System model H1, 2020. <https://www.honeywell.com/us/en/company/quantum/quantum-computer>.
- [2] S. Aaronson. Bqp and the polynomial hierarchy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 141–150, 2010.
- [3] S. Aaronson. Shtetl-optimized, The blog of Scott Aaronson, 2023. <https://scottaaronson.blog/?p=4447>.
- [4] A. S. Alfa, S. S. Heragu, and M. Chen. A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers & Industrial Engineering*, 21(1-4):635–639, 1991. doi:10.1016/0360-8352(91)90165-3.
- [5] T. Anders. 133 Compositions Created with Constraint Programming. In *The Oxford Handbook of Algorithmic Music*. Oxford University Press, 02 2018. doi:10.1093/oxfordhb/9780190226992.013.5.
- [6] B. Apolloni, C. Carvalho, and D. De Falco. Quantum stochastic optimization. *Stochastic Processes and their Applications*, 33(2):233–244, 1989. doi:10.1016/0304-4149(89)90040-9.
- [7] K. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003. doi:10.1017/cbo9780511615320.
- [8] E. M. Arkin and E. B. Silverberg. Scheduling jobs with fixed start and end times. *Discrete Applied Mathematics*, 18(1):1–8, 1987. doi:10.1016/0166-218X(87)90037-0.
- [9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, et al. Hartree-Fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, 2020. doi:10.1126/science.abb9811.

Bibliography

- [10] A. Arya, L. Botelho, F. Cañete, D. Kapadia, and Ö. Salehi. *Applications of Quantum Annealing to Music Theory*, pages 373–406. Springer International Publishing, Cham, 2022. doi:[10.1007/978-3-031-13909-3_15](https://doi.org/10.1007/978-3-031-13909-3_15).
- [11] B. Bakó, A. Glos, Ö. Salehi, and Z. Zimborás. Near-optimal circuit design for variational quantum optimization, 2022. [arXiv:2207.06294](https://arxiv.org/abs/2207.06294).
- [12] F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, oct 1982. doi:[10.1088/0305-4470/15/10/028](https://doi.org/10.1088/0305-4470/15/10/028).
- [13] A. Bärttschi and S. Eidenbenz. Deterministic preparation of Dicke states. In *International Symposium on Fundamentals of Computation Theory*, pages 126–139. Springer, 2019. doi:[10.1007/978-3-030-25027-0_9](https://doi.org/10.1007/978-3-030-25027-0_9).
- [14] A. Bärttschi and S. Eidenbenz. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 72–82. IEEE, 2020. doi:[10.1109/qce49297.2020.00020](https://doi.org/10.1109/qce49297.2020.00020).
- [15] D. J. Bernstein and T. Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017. doi:[10.1038/nature23461](https://doi.org/10.1038/nature23461).
- [16] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022. doi:[10.1103/revmodphys.94.015004](https://doi.org/10.1103/revmodphys.94.015004).
- [17] V. Bogdanov, C. Woodstra, J. Bush, and S. T. Erlewine. *All Music Guide to Electronica: the Definitive Guide to Electronic Music*. CMP Media, 2001.
- [18] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218–224, 2014. doi:[10.1038/nphys2900](https://doi.org/10.1038/nphys2900).
- [19] M. Borowski, P. Gora, K. Karnas, M. Błajda, K. Król, A. Matyjasek, D. Burczyk, M. Szewczyk, and M. Kutwin. New hybrid quantum annealing algorithms for solving Vehicle Routing Problem. In *International Conference on Computational Science*, pages 546–561. Springer, 2020. doi:[10.1007/978-3-030-50433-5_42](https://doi.org/10.1007/978-3-030-50433-5_42).
- [20] L. Botelho, A. Glos, A. Kundu, J. A. Mischczak, Ö. Salehi, and Z. Zimborás. Error mitigation for variational quantum algorithms through mid-circuit

Bibliography

- measurements. *Physical Review A*, 105(2):022441, 2022. doi:[10.1103/physreva.105.022441](https://doi.org/10.1103/physreva.105.022441).
- [21] L. Botelho and Ö. Salehi. Fixed interval scheduling problem with minimal idle time with an application to music arrangement problem, 2023. arXiv:[2310.14825](https://arxiv.org/abs/2310.14825).
- [22] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta. Mitigating measurement errors in multiqubit experiments. *Physical Review A*, 103(4):042605, 2021. doi:[10.1103/physreva.103.042605](https://doi.org/10.1103/physreva.103.042605).
- [23] P. Brucker and S. Knust. On the complexity of scheduling. In *Introduction to Scheduling*, pages 21–42. CRC Press, 2009.
- [24] R. E. Burkard and T. Bönniger. A heuristic for quadratic boolean programs with applications to quadratic assignment problems. *European Journal of Operational Research*, 13(4):374–386, 1983. doi:[10.1016/0377-2217\(83\)90097-8](https://doi.org/10.1016/0377-2217(83)90097-8).
- [25] C. S. Calude, M. J. Dinneen, and R. Hua. QUBO formulations for the graph isomorphism problem and related problems. *Theoretical Computer Science*, 701:54–69, 2017. doi:[10.1016/j.tcs.2017.04.016](https://doi.org/10.1016/j.tcs.2017.04.016).
- [26] E. Cambouropoulos. The local boundary detection model (lbdm) and its application in the study of expressive timing. In *ICMC*, page 8, 2001.
- [27] C. Campbell and E. Dahl. QAOA of the highest order. In *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*, pages 141–146. IEEE, 2022. doi:[10.1109/icsa-c54293.2022.00035](https://doi.org/10.1109/icsa-c54293.2022.00035).
- [28] V. Černý. Thermodynamical approach to the Traveling Salesman Problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.
- [29] M. Chams, A. Hertz, and D. De Werra. Some experiments with simulated annealing for coloring graphs. *European Journal of Operational Research*, 32(2):260–266, 1987. doi:[10.1016/s0377-2217\(87\)80148-0](https://doi.org/10.1016/s0377-2217(87)80148-0).
- [30] N. Chancellor. Domain wall encoding of discrete variables for quantum annealing and QAOA. *Quantum Science and Technology*, 4(4):045004, 2019. doi:[10.1088/2058-9565/ab33c2](https://doi.org/10.1088/2058-9565/ab33c2).
- [31] S.-C. Chiu, M.-K. Shan, and J.-L. Huang. Automatic system for the arrangement of piano reductions. In *2009 11th IEEE International Symposium on Multimedia*, pages 459–464. IEEE, 2009. doi:[10.1109/ism.2009.105](https://doi.org/10.1109/ism.2009.105).

Bibliography

- [32] M. Conforti, G. Cornuéjols, G. Zambelli, et al. *Integer Programming*, volume 271. Springer, 2014. doi:[10.1007/978-3-319-11008-0](https://doi.org/10.1007/978-3-319-11008-0).
- [33] J. Cook, S. Eidenbenz, and A. Bärtschi. The quantum alternating operator ansatz on maximum k-vertex cover. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 83–92. IEEE, 2020. doi:[10.1109/qce49297.2020.00021](https://doi.org/10.1109/qce49297.2020.00021).
- [34] N. d. S. Cunha, A. Subramanian, and D. Herremans. Generating guitar solos by integer programming. *Journal of the Operational Research Society*, 69(6):971–985, 2018.
- [35] M. S. Cuthbert and C. Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In J. S. Downie and R. C. Veltkamp, editors, *ISMIR*, pages 637–642. International Society for Music Information Retrieval, 2010. URL: <http://dblp.uni-trier.de/db/conf/ismir/ismir2010.html#CuthbertA10>.
- [36] B. Denkena, F. Schinkel, J. Pirnay, and S. Wilmsmeier. Quantum algorithms for process parallel flexible job shop scheduling. *CIRP Journal of Manufacturing Science and Technology*, 33:100–114, 2021. doi:[10.1016/j.cirpj.2021.03.006](https://doi.org/10.1016/j.cirpj.2021.03.006).
- [37] K. Domino, M. Koniorczyk, K. Krawiec, K. Jałowiecki, and B. Gardas. Quantum computing approach to railway dispatching and conflict management optimization on single-track railway lines. *arXiv:2010.08227*, 2020.
- [38] K. Domino, A. Kundu, Ö. Salehi, and K. Krawiec. Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing. *Quantum Information Processing*, 21(9):1–33, 2022. doi:[10.1007/s11128-022-03670-y](https://doi.org/10.1007/s11128-022-03670-y).
- [39] P. Doornbusch. Early computer music experiments in Australia and England. *Organised Sound*, 22(2):297–307, 2017. doi:[10.1017/s1355771817000206](https://doi.org/10.1017/s1355771817000206).
- [40] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, 2021. doi:[10.7566/jpsj.90.032001](https://doi.org/10.7566/jpsj.90.032001).
- [41] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm, 2014. Technical Report MIT-CTP/4628. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [42] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. Technical Report MIT-CTP-2936, Massachusetts Institute of Technology, 2000. doi:[10.48550/arXiv.quant-ph/0001106](https://doi.org/10.48550/arXiv.quant-ph/0001106).

Bibliography

- [43] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, jul 1998. doi:[10.1145/285055.285059](https://doi.org/10.1145/285055.285059).
- [44] A. Freedline. Algorhythms: Generating music with D-Wave’s quantum annealer. (Accessed on 12/15/2021). URL: <https://link.medium.com/tmpnTELgIKb>.
- [45] F. G. Fuchs, H. Ø. Kolden, N. H. Aase, and G. Sartor. Efficient encoding of the weighted MAX- k -CUT on a quantum computer using QAOA. *SN Computer Science*, 2(2):1–14, 2021.
- [46] T. Funk. *Zen and the Art of Software Performance: John Cage and Lejaren A. Hiller Jr.’s HPSCHD (1967-1969)*. PhD thesis, University of Illinois at Chicago, 2016. doi:[10.26209/ahd](https://doi.org/10.26209/ahd).
- [47] A. Garcia-Saez and J. Latorre. Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers, 2018. arXiv:[1806.02287](https://arxiv.org/abs/1806.02287).
- [48] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1979.
- [49] M. R. Geller and M. Sun. Efficient correction of multiqubit measurement errors. *Quantum Science and Technology*, 6(2):025009, 2020. doi:[10.1088/2058-9565/abd5c9](https://doi.org/10.1088/2058-9565/abd5c9).
- [50] I. M. Georgescu, S. Ashhab, and F. Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153, 2014. doi:[10.1103/revmodphys.86.153](https://doi.org/10.1103/revmodphys.86.153).
- [51] A. Glos, A. Krawiec, and Z. Zimborás. Space-efficient binary optimization for variational quantum computing. *npj Quantum Information*, 8(1):1–8, 2022. doi:[10.1038/s41534-022-00546-y](https://doi.org/10.1038/s41534-022-00546-y).
- [52] A. Glos, A. Kundu, and Ö. Salehi. Optimizing the production of test vehicles using hybrid constrained quantum annealing, 2022. doi:[10.1007/s42979-023-02071-x](https://doi.org/10.1007/s42979-023-02071-x).
- [53] F. Glover. Tabu search—part II. *ORSA Journal on computing*, 2(1):4–32, 1990. doi:[10.1287/ijoc.2.1.4](https://doi.org/10.1287/ijoc.2.1.4).
- [54] F. Glover, G. Kochenberger, and Y. Du. A tutorial on formulating and using qubo models, 2018. arXiv:[1811.11538](https://arxiv.org/abs/1811.11538).
- [55] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Elsevier, 2004.

Bibliography

- [56] Y. Guan, J. Zhao, Y. Qiu, Z. Zhang, and G. Xia. Melodic phrase segmentation by deep neural networks, 2018. [arXiv:1811.05688](https://arxiv.org/abs/1811.05688).
- [57] S. Hadfield, Z. Wang, B. O’Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms*, 12(2):34, 2019. [doi:10.3390/a12020034](https://doi.org/10.3390/a12020034).
- [58] S. M. Harwood, D. Trenev, S. T. Stober, P. Barkoutsos, T. P. Gujarati, S. Mostame, and D. Greenberg. Improving the variational quantum eigensolver using variational adiabatic quantum computing. *ACM Transactions on Quantum Computing*, 3(1):1–20, 2022. [doi:10.1145/3479197](https://doi.org/10.1145/3479197).
- [59] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver. Perspectives of quantum annealing: Methods and implementations. *Reports on Progress in Physics*, 83(5):054401, 2020. [doi:10.1088/1361-6633/ab85b8](https://doi.org/10.1088/1361-6633/ab85b8).
- [60] Y. He, M.-X. Luo, E. Zhang, H.-K. Wang, and X.-F. Wang. Decompositions of n -qubit Toffoli gates with linear circuit complexity. *International Journal of Theoretical Physics*, 56(7):2350–2361, 2017. [doi:10.1007/s10773-017-3389-4](https://doi.org/10.1007/s10773-017-3389-4).
- [61] L. A. Hiller and L. M. Isaacson. *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc., 1979.
- [62] G. Hori and S. Sagayama. Hmm-based automatic arrangement for guitars with transposition and its implementation. In *Music Technology meets Philosophy - From Digital Echos to Virtual Ethos: Joint Proceedings of the 40th International Computer Music Conference, ICMC*, 2014.
- [63] J.-L. Huang, S.-C. Chiu, and M.-K. Shan. Towards an automatic music arrangement framework using score reduction. *ACM Trans. Multimedia Comput. Commun. Appl.*, 8(1), feb 2012. [doi:10.1145/2071396.2071404](https://doi.org/10.1145/2071396.2071404).
- [64] K. Ikeda, Y. Nakamura, and T. S. Humble. Application of quantum annealing to nurse scheduling problem. *Scientific Reports*, 9(1):1–10, 2019. [doi:10.1038/s41598-019-49172-3](https://doi.org/10.1038/s41598-019-49172-3).
- [65] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925. [doi:10.1007/BF02980577](https://doi.org/10.1007/BF02980577).
- [66] T. Kadowaki and H. Nishimori. Quantum annealing in the transverse Ising model. *Physical Review E*, 58(5):5355, 1998. [doi:10.1103/physreve.58.5355](https://doi.org/10.1103/physreve.58.5355).

Bibliography

- [67] S. Karimi and P. Ronagh. Practical integer-to-binary mapping for quantum annealers. *Quantum Information Processing*, 18(4):1–24, 2019. doi:[10.1007/s11128-019-2213-x](https://doi.org/10.1007/s11128-019-2213-x).
- [68] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 2010.
- [69] A. Kirke. Programming gate-based hardware quantum computers for music. *Musicology*, (24):21–37, 2018. doi:[10.2298/muz1824021k](https://doi.org/10.2298/muz1824021k).
- [70] A. Kirke. Applying quantum hardware to non-scientific problems: Grover’s algorithm and rule-based algorithmic music composition, 2019. arXiv:[1902.04237](https://arxiv.org/abs/1902.04237).
- [71] A. Kirke and E. R. Miranda. Experiments in sound and music quantum computing. In *Guide to unconventional computing for music*, pages 121–157. Springer, 2017. doi:[10.1007/978-3-319-49881-2_5](https://doi.org/10.1007/978-3-319-49881-2_5).
- [72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. doi:[10.1016/b978-0-08-051581-6.50059-3](https://doi.org/10.1016/b978-0-08-051581-6.50059-3).
- [73] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson Education India, 2006.
- [74] T. R. Knösche, C. Neuhaus, J. Haueisen, K. Alter, B. Maess, O. W. Witte, and A. D. Friederici. Perception of phrase structure in music. *Human Brain Mapping*, 24(4):259–273, 2005. doi:[10.1002/hbm.20088](https://doi.org/10.1002/hbm.20088).
- [75] C. Koulamas, S. Antony, and R. Jaen. A survey of simulated annealing applications to operations research problems. *Omega*, 22(1):41–56, 1994. doi:[10.1016/0305-0483\(94\)90006-x](https://doi.org/10.1016/0305-0483(94)90006-x).
- [76] A. K. Lenstra. *Integer Factoring*, pages 611–618. Springer US, Boston, MA, 2011. URL: https://doi.org/10.1007/978-1-4419-5906-5_455, doi:[10.1007/978-1-4419-5906-5_455](https://doi.org/10.1007/978-1-4419-5906-5_455).
- [77] F. Lerdahl and R. S. Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1996. doi:[10.7551/mitpress/12513.001.0001](https://doi.org/10.7551/mitpress/12513.001.0001).
- [78] Y. Li, C. M. Wilk, T. Hori, and S. Sagayama. Automatic piano reduction of orchestral music based on musical entropy. In *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5, 2019. doi:[10.1109/CISS.2019.8693036](https://doi.org/10.1109/CISS.2019.8693036).

Bibliography

- [79] A. Lucas. Ising formulations of many NP problems. *Frontiers in Physics*, 2:5, 2014. [doi:10.3389/fphy.2014.00005](https://doi.org/10.3389/fphy.2014.00005).
- [80] F. B. Maciejewski, F. Baccari, Z. Zimborás, and M. Oszmaniec. Modeling and mitigation of cross-talk effects in readout noise with applications to the Quantum Approximate Optimization Algorithm. *Quantum*, 5:464, 2021. [doi:10.22331/q-2021-06-01-464](https://doi.org/10.22331/q-2021-06-01-464).
- [81] F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec. Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography. *Quantum*, 4:257, 2020. [doi:10.22331/q-2020-04-24-257](https://doi.org/10.22331/q-2020-04-24-257).
- [82] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero. A corpus-based hybrid approach to music analysis and composition. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 839. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.
- [83] S. Mandra and H. G. Katzgraber. A deceptive step towards quantum speedup detection. *Quantum Science and Technology*, 3(4):04LT01, 2018. [doi:10.1088/2058-9565/aac8b2](https://doi.org/10.1088/2058-9565/aac8b2).
- [84] P. Manning. *Electronic and Computer Music*. Oxford University Press, 2013. [doi:10.1093/acprof:oso/9780199746392.001.0001](https://doi.org/10.1093/acprof:oso/9780199746392.001.0001).
- [85] S. McArdle, X. Yuan, and S. Benjamin. Error-mitigated digital quantum simulation. *Physical Review Letters*, 122(18):180501, 2019. [doi:10.1103/physrevlett.122.180501](https://doi.org/10.1103/physrevlett.122.180501).
- [86] J. R. McClean, M. P. Harrigan, M. Mohseni, N. C. Rubin, Z. Jiang, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Low-depth mechanisms for quantum optimization. *PRX Quantum*, 2(3):030312, 2021. [doi:10.1103/prxquantum.2.030312](https://doi.org/10.1103/prxquantum.2.030312).
- [87] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016. [doi:10.1088/1367-2630/18/2/023023](https://doi.org/10.1088/1367-2630/18/2/023023).
- [88] L. F. Menabrea and A. Lovelace. *Sketch of the Analytical Engine invented by Charles Babbage, Esq.* Association for Computing Machinery and Morgan & Claypool, 2015. [doi:10.1145/2809523.2809528](https://doi.org/10.1145/2809523.2809528).

Bibliography

- [89] E. R. Miranda. *Composing Music with Computers*. CRC Press, 2001. doi:
[10.4324/9780080502403](https://doi.org/10.4324/9780080502403).
- [90] E. R. Miranda. Quantum computer: Hello, music!, 2020. arXiv:2006.13849,
doi:[10.1007/978-3-030-72116-9_34](https://doi.org/10.1007/978-3-030-72116-9_34).
- [91] K. M. Nakanishi, K. Fujii, and S. Todo. Sequential minimal optimization for
quantum-classical hybrid algorithms. *Physical Review Research*, 2(4):043158,
2020. doi:[10.1103/physrevresearch.2.043158](https://doi.org/10.1103/physrevresearch.2.043158).
- [92] E. Narmour. *The analysis and cognition of basic melodic structures: The
implication-realization model*. University of Chicago Press, 1990. doi:[10.2307/897927](https://doi.org/10.2307/897927).
- [93] P. Nation and B. Johnson. How to measure and reset a qubit in the middle
of a circuit execution, 2021. [https://www.ibm.com/blogs/research/2021/
02/quantum-mid-circuit-measurement/](https://www.ibm.com/blogs/research/2021/02/quantum-mid-circuit-measurement/).
- [94] C. Neuhaus, T. R. Knösche, and A. D. Friederici. Effects of musical expertise
and boundary markers on phrase perception in music. *Journal of Cognitive
Neuroscience*, 18(3):472–493, 2006. doi:[10.1162/jocn.2006.18.3.472](https://doi.org/10.1162/jocn.2006.18.3.472).
- [95] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and
B. Parney. Traffic flow optimization using a quantum annealer. *Frontiers in
ICT*, 4:29, 2017. doi:[10.3389/fict.2017.00029](https://doi.org/10.3389/fict.2017.00029).
- [96] M. A. Nielsen and I. L. Chuang. Quantum computation and quantum
information. *Phys. Today*, 54:60–2, 2001. doi:[10.1017/cbo9780511976667](https://doi.org/10.1017/cbo9780511976667).
- [97] K. N. Olsen, R. T. Dean, and Y. Leung. What constitutes a phrase in sound-
based music? a mixed-methods investigation of perception and acoustics.
PloS one, 11(12):e0167643, 2016. doi:[10.1371/journal.pone.0167643](https://doi.org/10.1371/journal.pone.0167643).
- [98] S. Onuma and M. Hamanaka. Piano arrangement system based on composers’
arrangement processes. In *Proceedings of International Computer Music
Conference 2010, ICMC*, 2010.
- [99] I. H. Osman and C. Potts. Simulated annealing for permutation flow-shop
scheduling. *Omega*, 17(6):551–557, 1989. doi:[10.1016/0305-0483\(89\)
90059-5](https://doi.org/10.1016/0305-0483(89)90059-5).
- [100] M. Ostaszewski, E. Grant, and M. Benedetti. Structure optimization for
parameterized quantum circuits. *Quantum*, 5:391, 2021. doi:[10.22331/
q-2021-01-28-391](https://doi.org/10.22331/q-2021-01-28-391).

Bibliography

- [101] P. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al. Scalable quantum simulation of molecular energies. *Physical Review X*, 6(3):031007, 2016. doi:[10.1103/physrevx.6.031007](https://doi.org/10.1103/physrevx.6.031007).
- [102] G. Papadopoulos and G. Wiggins. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, volume 124, pages 110–117. Edinburgh, UK, 1999.
- [103] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti. A QUBO model for the Traveling Salesman Problem with Time Windows. *Algorithms*, 12(11):224, 2019. doi:[10.3390/a12110224](https://doi.org/10.3390/a12110224).
- [104] M. T. Pearce, D. Müllensiefen, and G. A. Wiggins. A comparison of statistical and rule-based models of melodic segmentation. In *ISMIR*, pages 89–94, 2008.
- [105] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):1–7, 2014. doi:[10.1038/ncomms5213](https://doi.org/10.1038/ncomms5213).
- [106] C. Potts and L. N. Van Wassenhove. Single machine tardiness sequencing heuristics. *IIE Transactions*, 23(4):346–354, 1991. doi:[10.1080/07408179108963868](https://doi.org/10.1080/07408179108963868).
- [107] M. Saeedi and M. Pedram. Linear-depth quantum circuits for n -qubit Toffoli gates with no ancilla. *Physical Review A*, 87(6):062318, 2013. doi:[10.1103/PhysRevA.87.062318](https://doi.org/10.1103/PhysRevA.87.062318).
- [108] Ö. Salehi, A. Glos, and J. A. Mischczak. Unconstrained binary models of the travelling salesman problem variants for quantum optimization. *Quantum Information Processing*, 21(2):1–30, 2022. doi:[10.1007/s11128-021-03405-5](https://doi.org/10.1007/s11128-021-03405-5).
- [109] N. P. Sawaya, T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi. Resource-efficient digital quantum simulation of d -level systems for photonic, vibrational, and spin- s hamiltonians. *npj Quantum Information*, 6(1):1–13, 2020. doi:[10.1038/s41534-020-0278-0](https://doi.org/10.1038/s41534-020-0278-0).
- [110] A. Sawicki, L. Mattioli, and Z. Zimborás. Universality verification for a set of quantum gates. *Phys. Rev. A*, 105:052602, May 2022. doi:[10.1103/PhysRevA.105.052602](https://doi.org/10.1103/PhysRevA.105.052602).

Bibliography

- [111] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
- [112] Short history of computer music. <http://artsites.ucsc.edu/EMS/Music/equipment/computers/history/history.html>. (Accessed on 15 December 2021).
- [113] M. Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012. URL: <https://books.google.pl/books?id=1aMKAAAQBAJ>.
- [114] T. Stollenwerk, E. Lobe, and M. Jung. Flight gate assignment with a quantum annealer. In *International Workshop on Quantum Technology and Optimization Problems*, pages 99–110. Springer, 2019. doi:10.1007/978-3-030-14082-3_9.
- [115] Z. Tabi, K. H. El-Safty, Z. Kallus, P. Haga, T. Kozsik, A. Glos, and Z. Zimboras. Quantum optimization for the graph coloring problem with space-efficient embedding. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 56–62. IEEE, 2020. doi:10.1109/qce49297.2020.00018.
- [116] K. Tamura, T. Shirai, H. Katsura, S. Tanaka, and N. Togawa. Performance comparison of typical binary-integer encodings in an Ising machine. *IEEE Access*, 9:81032–81039, 2021. doi:10.1109/access.2021.3081685.
- [117] T. Tanaka, B. Bemman, and D. Meredith. Integer programming formulation of the problem of generating milton babbitt’s all-partition arrays. In *The 17th International Society for Music Information Retrieval Conference*, 2016.
- [118] E. Tang. *Quantum Machine Learning Without Any Quantum*. PhD thesis, 2023. URL: <http://hdl.handle.net/1773/50751>.
- [119] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022. The Variational Quantum Eigensolver: a review of methods and best practices. doi:10.1016/j.physrep.2022.08.003.
- [120] N. Todd. A model of expressive timing in tonal music. *Music Perception*, 3(1):33–57, 1985. doi:10.2307/40285321.
- [121] C. Truchet, C. Agon, and P. Codognet. A constraint programming system for music composition, preliminary results. In *In the Seventh International*

Bibliography

Conference on Principles and Practice of Constraint Programming, Musical Constraints Workshop, Paphos, 2001.

- [122] P. Van Beek and X. Chen. CPlan: A constraint programming approach to planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 585–590, 1999. URL: <https://aaai.org/papers/083-aaai99-083-cplan-a-constraint-programming-approach-to-planning/>.
- [123] P. J. Van Laarhoven, E. H. Aarts, and J. K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1):113–125, 1992. doi: [10.1287/opre.40.1.113](https://doi.org/10.1287/opre.40.1.113).
- [124] D. Venturelli, D. J. Marchand, and G. Rojo. Quantum annealing implementation of job-shop scheduling. *arXiv:1506.08479*, 2015.
- [125] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel. XY-mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Physical Review A*, 101(1):012320, 2020.
- [126] L. T. Watson and R. T. Haftka. Modern homotopy methods in optimization. *Computer Methods in Applied Mechanics and Engineering*, 74(3):289–305, 1989. doi: [10.1016/0045-7825\(89\)90053-4](https://doi.org/10.1016/0045-7825(89)90053-4).
- [127] D. Willsch, M. Willsch, C. D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson, and K. Michielsen. Benchmarking advantage and D-Wave 2000Q quantum annealers with exact cover problems. *Quantum Information Processing*, 21(4):1–22, 2022. doi: [10.1007/s11128-022-03476-y](https://doi.org/10.1007/s11128-022-03476-y).
- [128] G. J. Woeginger. *Exact Algorithms for NP-Hard Problems: A Survey*, pages 185–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. doi: [10.1007/3-540-36478-1_17](https://doi.org/10.1007/3-540-36478-1_17).
- [129] World tsp. <https://www.math.uwaterloo.ca/tsp/world/>. Accessed: 2024-03-24.
- [130] S. Yarkoni, A. Alekseyenko, M. Streif, D. Von Dollen, F. Neukart, and T. Bäck. Multi-car paint shop optimization with quantum annealing. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 35–41. IEEE, 2021. doi: [10.1109/qce52317.2021.00019](https://doi.org/10.1109/qce52317.2021.00019).
- [131] M. Zaman, K. Tanahashi, and S. Tanaka. Pyqubo: Python library for mapping combinatorial optimization problems to qubo form. *IEEE Transactions on Computers*, 71(4):838–850, 2021. doi: [10.1109/tc.2021.3063618](https://doi.org/10.1109/tc.2021.3063618).

Bibliography

- [132] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10:021067, Jun 2020. [doi:10.1103/PhysRevX.10.021067](https://doi.org/10.1103/PhysRevX.10.021067).

Bibliography