Angles and Devices for Quantum Approximate Optimization

Dissertation
zur Erlangung des Grades
des Doktors der Naturwissenschaften
der Naturwissenschaftlich - Technischen Fakultät
der Universität des Saarlandes

von

David Headley

Saarbrücken
2023

Tag des Kolloquiums:	28. September 2023
Dekan:	Prof. Dr. Ludger Santen
Berichterstatter:	Prof. Dr. Frank Wilhelm-Mauch Dr. Nicholas Chancellor
Akad. Mitglied:	Dr. Adam Miroslaw Wysocki
Vorsitz:	Prof. Dr. Christoph Becher

Abstract

A potential application of emerging Noisy Intermediate-Scale Quantum (NISQ) devices is that of approximately solving combinatorial optimization problems. This thesis investigates a gate-based algorithm for this purpose, the Quantum Approximate Optimization Algorithm (QAOA), in two major themes.

First, we examine how the QAOA resolves the problems it is designed to solve. We take a statistical view of the algorithm applied to ensembles of problems, first, considering a highly symmetric version of the algorithm, using Grover drivers. In this highly symmetric context, we find a simple dependence of the QAOA state's expected value on how values of the cost function are distributed. Furthering this theme, we demonstrate that, generally, QAOA performance depends on problem statistics with respect to a metric induced by a chosen driver Hamiltonian. We obtain a method for evaluating QAOA performance on worst-case problems, those of random costs, for differing driver choices.

Second, we investigate a QAOA context with device control occurring only via single-qubit gates, rather than using individually programmable one- and two-qubit gates. In this reduced control overhead scheme—the digital-analog scheme—the complexity of devices running QAOA circuits is decreased at the cost of errors which are shown to be non-harmful in certain regimes. We then explore hypothetical device designs one could use for this purpose.

Zusammenfassung

Eine mögliche Anwendung für "Noisy Intermediate-Scale Quantum devices" (NISQ devices) ist die näherungsweise Lösung von kombinatorischen Optimierungsproblemen. Die vorliegende Arbeit untersucht anhand zweier Hauptthemen einen gatterbasierten Algorithmus, den sogenannten "Quantum Approximate Optimization Algorithm" (QAOA).

Zuerst prüfen wir, wie der QAOA jene Probleme löst, für die er entwickelt wurde. Wir betrachten den Algorithmus in einer Kombination mit hochsymmetrischen Grover-Treibern für statistische Ensembles von Probleminstanzen. In diesem Kontext finden wir eine einfache Abhängigkeit von der Verteilung der Kostenfunktionswerte. Weiterführend zeigen wir, dass die QAOA-Leistung generell von der Problemstatistik in Bezug auf eine durch den gewählten Treiber-Hamiltonian induzierte Metrik abhängt. Wir erhalten eine Methode zur Bewertung der QAOA-Leistung bei schwersten Problemen (solche zufälliger Kosten) für unterschiedliche Treiberauswahlen.

Zweitens untersuchen wir eine QAOA-Variante, bei der sich die Hardware-Kontrolle nur auf Ein-Qubit-Gatter anstatt individuell programmierbare Ein- und Zwei-Qubit-Gatter erstreckt. In diesem reduzierten Kontrollaufwandsschema—dem digital-analogen Schema—sinkt die Komplexität der Hardware, welche die QAOA-Schaltungen ausführt, auf Kosten von Fehlern, die in bestimmten Bereichen als ungefährlich nachgewiesen werden. Danach erkunden wir hypothetische Hardware-Konzepte, die für diesen Zweck genutzt werden könnten.

Publications

Published

- D. Headley, T. Müller, A. Martin, E. Solano, M. Sanz, F. K. Wilhelm
  Approximating the quantum approximate optimization algorithm with digital-analog interactions
  Phys. Rev. A 106, 042446 (2022) [1]

- D. Headley, F. K. Wilhelm
  Problem-size independent angles for a Grover-driven quantum approximate optimization algorithm
  Phys. Rev. A 107, 012412 (2023) [2]

Preprint

- T. Müller, T. Stollenwerk, D. Headley, M. Epping, F. K. Wilhelm
  Coherent and non-unitary errors in ZZ-generated gates
  arXiv:2304.14212 (2023) [3]

Previous

- J. Roffe, D. Headley, N. Chancellor, D. Horsman, V. Kendon
  Protecting quantum memories using coherent parity check codes
  Quantum Sci. Technol. 3 035010 (2018) [4]

Acknowledgements

This thesis would not have been possible without the support of several academic supervisors throughout and prior to my PhD. I thank Viv Kendon for being a generous Master's advisor and providing a strong introduction to the field. Steve Brierley and Riverlane Research who provided an opening for me to gain experience, before taking on the daunting commitment of a PhD. Markus Leder and my former Mercedes-Benz colleagues for providing many fantastic opportunities for a new PhD student and also for his endless bureaucratic support. I thank Frank for supervising my PhD, for his guidance and opportunities that came with this. Finally, I thank my friends and family who made this time enjoyable and for their continued support.

Contents

List of Figures

Chapter 1

Introduction

In the following chapter, the work of this thesis is motivated. Basic concepts and mathematical language upon which the work is built are introduced alongside the requisite theoretical background. We introduce some important results and concepts in quantum computing alongside classical background on the problems for which we aim to use them.

1.1 Classical and Quantum Computers

The development of classical computers in the previous century has ushered forth a third industrial revolution of development. Following the significant reduction in costs due to advancements in steam technology, standardization, and industrialization applied to consumer goods, artificial lighting, and cost of living, the development of classical computers has led and continues to lead to dramatic reductions in the cost of general computation, communication and the design and simulation of physical structures. The fundamental limitations on the capabilities of classical computation seem unbounded in many domains. For example, the past decade has seen rapid progress in the development of deep learning systems; machine learning algorithms that use many layers of simulated neurons to extract high-level features from raw data. Such approaches have solved problems previously expected to be decades away from solutions, from achieving superhuman performance in Go, and Starcraft, to obviating a research field with the introduction of AlphaFold, predicting 3-d structures of proteins with unprecedented accuracy. Deep learning methods continue to break new ground, with generative models such as GPT-3 [5], a pre-trained transformer model which, when given a prompt, can produce human-like text responses, having been trained on half a trillion examples to optimize its 175 billion parameters.

Despite the countless examples of the tasks at which classical computers excel, the logic on which they fundamentally operate is classical, that is, well described by classical mechanics[1]. Ontologically, classical mechanics suggests the existence of a well-defined state of reality, evolving in time deterministically via some equations of motion. This view of reality, however, was demonstrated to be inconsistent with experimental results. The Stern-Gerlach experiment [6] demonstrated that particles

[1]Modern transistors may be small enough to require the consideration of quantum mechanical effects in their design, but the high-level logic that they implement is entirely classical.

with non-zero magnetic moments are observed to act as if they take discrete orientations in space when passed through a magnetic field gradient—spin is quantized. Quantization of the magnetic moment is inconsistent with classical theories which predict a moment that can take values in a continuous range, with the discretization of states in quantum mechanics providing a resolution to issues faced when explaining natural phenomena with classical mechanics, such as the non-stability of atomic models, in which atoms should emit electromagnetic radiation and collapse, and the ultraviolet catastrophe, in which the energy emitted by black bodies is predicted to become unbounded at low wavelengths. Such ideas of physical reality were tested further with the EPR paradox [7] and Bell tests [8, 9], which, reject the possibility that physical reality has the property of *local realism*—physical observables must either have states that can signal superluminally, or yield measurement outcomes that are not pre-determined and knowable, prior to measurement.

Since classical computers follow the rules of classical mechanics, which are found to differ from those of quantum mechanics, the question of whether a computer operating on the level of quantum mechanics is more powerful than that of a classical device is raised. As such, quantum computing as a concept began to be considered in the 1980s by authors such as Feynman, Deutsch, and Benioff [10, 11, 12]. In short, quantum computers are compelling because they can produce states, which, when measured, produce probability distributions that cannot, in general, be prepared and sampled by classical computers. This is not to say that quantum computers are automatically more powerful than classical computers, though it is currently expected that for very specific tasks such as chemical simulations and factoring, a difference in the resources required exists, which scales exponentially in the size of these problems.

1.1.1 Bits and Qubits

Classical computers of today consist of, at their basic level, components inheriting from the Von Neumann model of computation. A classical computer consists of memory, both for immediate use, and mass storage, a Central Processing Unit in which operations are carried out on stored information, and means of inputting and outputting information to the computer. Almost all modern memory uses information stored on binary registers, typically, the magnetization of small magnetic domains on hard disk drives or the presence of stored charges in solid-state drives. Such binary values may take the states labeled by 0 or 1. The knowledge one has of such a binary system b state can be represented by a probability vector

$$\vec{\psi} = \begin{pmatrix} \mathrm{P}(b=0) \\ \mathrm{P}(b=1) \end{pmatrix} \tag{1.1}$$

where, of course, $\mathrm{P}(b=0) + \mathrm{P}(b=1) = 1$ is the norm of the vector that should be conserved. Operations that conserve such a norm are called right stochastic, in which the elements represent transition probabilities, with rows summing to 1. Therefore, the time evolution of a bit of classical knowledge over some interval can be expressed as:

$$\vec{\psi}_1 = \hat{U}\vec{\psi}_0 \tag{1.2}$$

where

$$\hat{U} = \begin{pmatrix} \mathrm{P}(b_1=0|b_0=0) & \mathrm{P}(b_1=0|b_0=1) \\ \mathrm{P}(b_1=1|b_0=0) & \mathrm{P}(b_1=1|b_0=1) \end{pmatrix} \tag{1.3}$$

which, for non-time-varying $\hat{U}$ is a single step of a Markov process.

Likewise, quantum computers have, as their basic constituent parts, similar two-state systems. However, the rules for such systems—qubits—are different. The Dirac notation $|0\rangle$, $|1\rangle$ denotes two possible states of a qubit—the computational basis. States of a qubit, however, can also take that of linear combinations of the above, a general state of a qubit can be written as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \tag{1.4}$$

with α, β as complex numbers. This is similar to the classical bit case, with instead of the sum of the elements being one, the sum in quadrature evaluating to 1, that is,

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.5}$$

When measured, the probability of measuring the qubit to be in a given configuration is given by the Born rule with

$$\mathrm{P}(b=0), \mathrm{P}(b=1) = |\alpha|^2, |\beta|^2. \tag{1.6}$$

Transformations of such pure states can then be expressed by unitary operations. A unitary one-qubit transformation $\hat{U}$ can be expressed as:

$$\hat{U} = \begin{pmatrix}u_{00} & u_{01}\\ u_{10} & u_{11}\end{pmatrix} \tag{1.7}$$

with

$$|\psi'\rangle = \hat{U}\,|\psi\rangle = u_{00}\alpha\,|0\rangle + u_{01}\beta\,|1\rangle + u_{10}\alpha\,|0\rangle + u_{11}\beta\,|1\rangle \tag{1.8}$$

where $\hat{U}$ is a 2×2 unitary matrix satisfying $\hat{U}^\dagger\hat{U} = \hat{U}\hat{U}^\dagger = \mathbb{I}$ where $\mathbb{I}$ is the identity matrix. As the stochastic matrix acting on probability vectors preserves the 1-norm, a unitary operator acting on a pure state preserves the 2-norm, which, then preserves the 1-norm of measurement outcome probabilities. Such unitary transformations on the state of the qubit are generated by the time-dependent Schrödinger equation:

$$\hat{H}\,|\psi\rangle = \hat{E}\,|\psi\rangle \tag{1.9}$$

which is obeyed for a given Hermitian Hamiltonian $\hat{H}$ and the energy operator $\hat{E} = i\hbar\frac{\mathrm{d}}{\mathrm{d}t}$. Solving the time-dependent Schrodinger equation for generic $\hat{H}$ and an initial state $\psi(0)$ gives:

$$|\psi(t)\rangle = \hat{U}(t,0)\,|\psi(0)\rangle\,,\ \hat{U}(t,0) = \mathcal{T}\exp\left(i\int_{t'=0}^{t}\hat{H}(t')\,\mathrm{d}t'\right) \tag{1.10}$$

which gives the state of a system at any time, given the initial state and time-dependent Hamiltonian are known.

So far, we have that describing a pure state of a qubit requires similar mathematical machinery to that of a bit, for which knowledge only of the probabilities that it takes given values. To introduce this epistemic uncertainty to the qubit state, we may describe states with density matrices. Density matrices are defined as a mixture of quantum states with associated probabilities p_i summing to 1. A density matrix is thus defined generically as

$$\hat{\rho} = \sum_i p_i\,|\psi_i\rangle\,\langle\psi_i|\,. \tag{1.11}$$

where the expectation value of a given observable $\hat{O}$ is produced via

$$\mathrm{Tr}\Big(\hat{O}\hat{\rho}\Big) \tag{1.12}$$

This widens the scope of potential operations acting on a system to those of a probabilistic mixture of unitary operations, described as quantum channels, where for example ε, a channel acting on a density matrix $\hat{\rho}$ can be described by:

$$\varepsilon(\hat{\rho}) = \sum p_i \hat{U}_i^\dagger \hat{\rho} \hat{U}_i \tag{1.13}$$

which is known as the Krauss decomposition of the channel ε.

The state of the qubit can alternatively be written as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \mathrm{e}^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{1.14}$$

where the angles $\theta \in [0, \pi], \phi \in [0, 2\pi]$ are polar coordinates and can be mapped to coordinates on a sphere. This representation of a qubit is referred to as the Bloch sphere, providing a helpful demonstration of single-qubit dynamics and the effects of the single-qubit errors that can occur on qubits, with mixed states represented by points within the Bloch sphere. Valid operations on single qubits then correspond to rotations on the sphere, with error channels corresponding to squashing operations of the sphere along one or more axes. Such a representation can be seen in Figure 1.1. The state of two bits is separable, that is, the state of one bit of information is necessarily independent of the physical state of another. Correlations might exist between observables, but such correlations pertain to the knowledge of such observables, rather than the physical state of the observable itself. The joint state of two qubits is given by the tensor product of the two states. That is, for two qubits described by $|\phi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$, $|\phi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, their joint state is described by:

$$|\phi_1\rangle \otimes |\phi_2\rangle = \alpha_1\alpha_2|0\rangle \otimes |0\rangle + \alpha_1\beta_2|0\rangle \otimes |1\rangle + \beta_1\alpha_2|1\rangle \otimes |0\rangle + \beta_1\beta_2|1\rangle \otimes |1\rangle \tag{1.15}$$

where the squared absolute value of pre-factor to a ket determines the probability of measuring the two qubits in the corresponding configuration. Such states—product states—do not, however, span the space of possible such vectors, thus giving rise to non-classical states which cannot be understood in terms of single-qubit properties alone. Such states are described as entangled, and strike a fine balance between allowing the state of two qubits to seemingly depend on events outside their light cone while forcing such events never to affect a qubits measurement outcome distribution.

But what can you do with quantum states and why should they be more useful than classical bits? As it transpires, the additional correlations afforded to entangled quantum systems, and richer state space, allow for a wider class of algorithms to be performed efficiently than those of classical computers. The following section presents an overview of such algorithms.

1.2 A Brief History of Quantum Algorithms

In the popular consciousness, and to the irritation of physicists attempting to maintain level expectations of the power of quantum computers, it is often believed

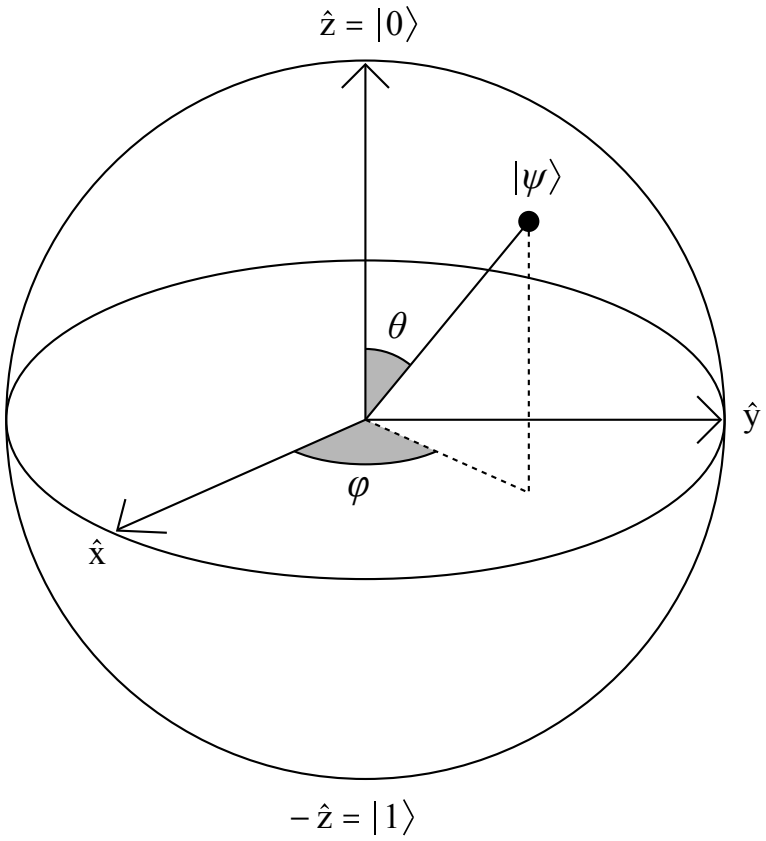

Figure 1.1: Single-qubit states can be mapped to points on the surface of a sphere allowing for an intuitive understanding of single-qubit rotations.

that the power of quantum computers lies in their ability to check every solution to some problem in simultaneity. While partially true, in that for some efficiently computable function f, one can prepare a state $\sum_z \frac{f(z)}{\sqrt{N}} |z\rangle$ on a quantum computer with the same amount[2] of effort one would take to apply a function f to a single input on an equivalent classical computer, this information is not accessible through measurement. While most quantum algorithms do use such operations, obtaining useful measurement outcomes from such states is far more nuanced.

Richard Feynman suggested in a seminal article, "There is plenty of room at the bottom" [13] that the physics of quantum systems is not likely to be efficiently simulable on a classical computer. The Deutsch-Jozsa algorithm [14] and the Bernstein-Vazirani [15] Algorithms provided the first proof that the physics of quantum mechanics allowed for the solution of problems with fewer resources than classically required. It was discovered in 1994 that computing the Fourier transform of a quantum state requires exponentially fewer resources than would be required to compute the Fourier transform of similar classical data [16], with the caveat that such a state must be prepared and read out from the quantum computer constrained by the Holevo bound [17]. Knowing that one can efficiently apply Fourier Trans-

[2]Asymptotically. Operations on classical computers will likely always be less costly in time than those of quantum computers. Classical computers are faster than contemporary quantum computers by a factor of around 1000 without error correction, and perhaps two to three times as many orders of magnitude faster once the overhead of quantum error correction is accounted for.

forms to quantum data introduces the possibility that if one has some efficiently preparable function, it will be efficient to sample from the frequency decomposition of that function applied to a superposition of inputs. From this insight, Shor's algorithm [18] derives, taking advantage of the periodicity in Fermat's little theorem of number theory to efficiently determine prime factors of large numbers. It is believed, but not yet proven, that no classical algorithm will achieve the polynomial-time scaling in factoring provided by Shor's algorithm. Consequently, public key encryption schemes based on the difficulty of factoring may be vulnerable to attacks from quantum computers. In a related development, the quantum phase estimation algorithm [19] was found by Kitaev to be capable of calculating the eigenvalues of an eigenstate for a quantum Hamiltonian.

Soon after the discovery of Shor's factoring algorithm, Grover's algorithm [20]—later generalized to the Amplitude Amplification Algorithm [21]—was discovered. Quantum computers are able to search for *marked* configurations of their state space in time scaling as the square root of the number of possible configurations, whereas classical computers will always need to check half the configurations in the average case and all but one in the worst case.

In the following years, quantum error correction was discovered [22], alongside the surface and topological codes which proved that fault-tolerant quantum computing is a technologically realizable goal.

At the turn of the century, adiabatic quantum computation was discovered [23], alongside quantum computation via quantum random walks [24]. Such algorithms opened up the possibility of solving combinatorial optimization problems on a quantum computer in such a way that is not entirely a Grover-like speedup. Finally, in the past decade, perhaps inspired by a renaissance of the application of neural networks in classical computation, variational quantum algorithms have seen great interest, with algorithms such as the variational quantum eigensolver for finding ground state energies on quantum computers, the quantum approximate optimization algorithm for finding approximate solutions to combinatorial optimization problems, and more general quantum neural networks, aiming to learn high-level features by tuning the angles of quantum gates for classical or quantum learning tasks.

1.3 Fault Tolerance and Near-Term Intermediate-Scale Quantum Computation

For a long time, it was uncertain whether quantum computers could perform useful calculations due to the presence of noise and unwanted interactions with their environment. Error correction in classical computing has been well-established and widely used in applications ranging from error-corrected random access memory to noisy communication with spacecraft. However, the requirements for quantum error correction are more stringent, which has made achieving fault quantum computation more challenging. On quantum information, not only can bit flip-type errors act on the qubits, but also phase flip errors, alongside a continuum of continuous errors in-between. Without quantum error correction, the gate counts required for useful quantum computations without appreciable chance of an error occurring, require gate operation error rates that are likely unreachably low, necessitating the use of quantum error-correcting codes. Such quantum error-correcting codes, do,

fortunately, exist and can be likely realized in practice. The first code, the Shor code [22] encoded one logical qubit onto a subspace of 9 physical qubit, with the surface code [25] later developed, allowing for quantum error correction with local stabilizers.

A headline result from the field of quantum error correction is that of the threshold theorem. For a quantum circuit of n qubits and p gates, there exists an error-correcting code or concatenation thereof using

$$O(\log(p/\epsilon)^c p) \tag{1.16}$$

gates, assuming that p is below the threshold of a given code. That is, if the error rate of the physical qubits on a device is sufficiently low, by applying subsequent layers of error correction, errors can be corrected faster than they are introduced, with the logical error suppressed exponentially in the number of layers of concatenation. Thus, error-corrected quantum computers that take advantage of both the square root speed-up in Grover's algorithm and the exponential speedup of Shor's are physically realizable, though perhaps at an unreasonable scale for the former to come to fruition on the scale of human civilization. Current estimates are that a fault-tolerant quantum computer will require 10^6 physical qubits for each logical qubit encoded therein [26], which accounts for the difficulty of not only applying the error-correcting codes but also that of applying magic state distillation [3].

1.4 Combinatorial Optimization

Combinatorial optimization problems are those of finding the best object from a finite set of discrete objects. The *goodness* of an object is determined by an objective function allocating a value to each element of the set of possible objects. In the notation of this thesis, the discrete objects are labeled by the binary strings $z \in \{0,1\}^n$ with the associated objective function, or solution quality $C(z)$, with

$$C : \{0,1\}^n \to \mathbb{R}. \tag{1.17}$$

Combinatorial optimization problems are ubiquitous in the world from the problem of nurse scheduling, optimizing logistics networks, or finding configurations of inputs in software that lead to bugs.

1.4.1 Decision & Extremization problems

Decision problems are problems with yes or no answers. That is, for a decision problem taking input z, one can imagine the objective function $C(z)$ to return binary values 0 or 1, with 0 corresponding to a non-satisfying string and 1 corresponding to a satisfying string. For example, such a problem might be that of primality testing, in which one determines whether an integer is divisible by a number that is not itself or 1. In such a case, one considers an objective function $C(z)$ to return 0 if a number a has nontrivial factors and 1 otherwise. Other examples of decision

[3] Quantum error-correcting codes such as the surface code can typically apply a restricted set of gates [27]—Clifford gates in the case of the surface code—to which a non-Clifford gate must be added in order to attain universality. Magic state distillation is the process of injecting T-gates into the computation in a fault-tolerant manner.

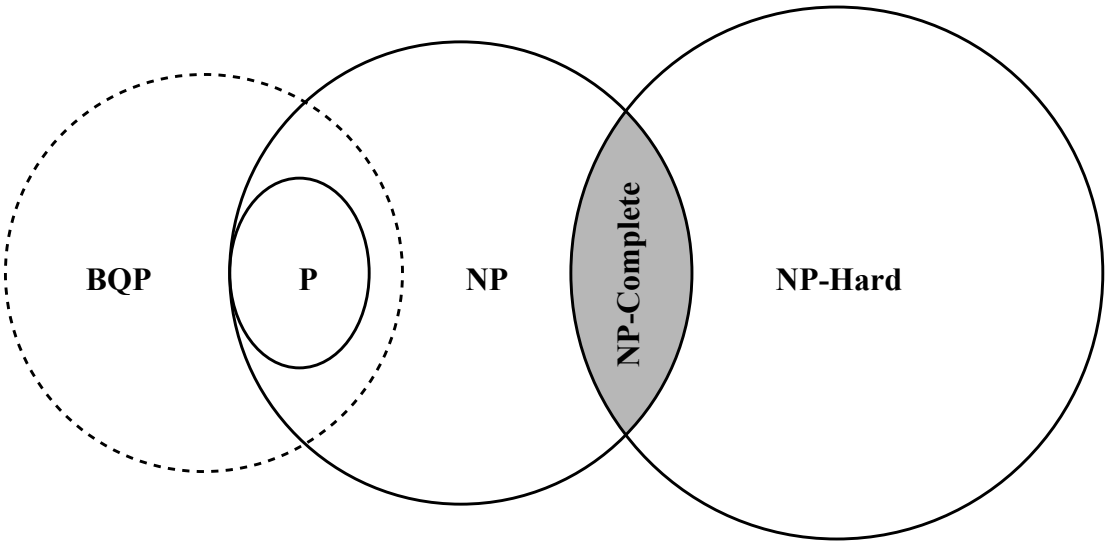

Figure 1.2: Problems can be grouped by their complexity class. Problems in **P** can be solved efficiently on a Turing machine. Problems in **NP** can be solved efficiently by a nondeterministic Turing machine. The complexity class **BQP** is the class of problems that can be solved efficiently with bounded error on a quantum computer, which, importantly, is larger than **P**. **NP-Complete** problems are those to which any other problem in **NP** can be reduced, with **NP-Hard** problems being those that are at least as hard as any problem in **NP**.

problems include those of determining whether a formula is satisfiable or whether a graph has certain properties, for example, whether a Hamiltonian cycle is possible in a given graph.

Decision problems have historically been important to computer scientists because they provide a precise and simple definition of certain problems, with definite true or false answers. The framing of problems as decision problems allows for sets of problems to be collected in complexity classes according to how difficult they are to solve and the resources in time and space one needs to do so. Some examples of these classes are:

P: The class of decision problems that can be solved in time scaling at most polynomially with the size of the problem to be solved. For example, finding the shortest path between two nodes in a graph is in **P**, as Dijkstra's algorithm can provide a path in time scaling at most as the square of the number of nodes, which is polynomial in the size of the problem.

NP: The class of problems that can be solved by a non-deterministic Turing machine[4] in polynomial time. Given a solution, a deterministic Turing machine[5] can verify that this solution does or does not satisfy the problem in time

[4] A nondeterministic Turing machine is a model for a theoretical computer allowing for unlimited parallelization, as is often the mental model for the mechanism by which quantum computers outperform classical computers in popular understanding.

[5] A theoretical model of what constitutes a computer that is sufficiently general in its programmability.

scaling at most polynomially. For example, factoring is in **NP**, since a nondeterministic Turing machine can simultaneously check all factors, with a Turing machine able to check that a given individual number is a factor in polynomial time.

NP-Hard: These are problems that are at least as hard as any problem in **NP**. These include totally intractable problems such as the halting problem[6].

NP-Complete: Problems that are both **NP** and **NP-Hard**. These are the hardest problems solvable by nondeterministic Turing machines and exist on the interface between **NP** and **NP-Hard**. **NP-Complete** problems have the property of being reducible to other **NP** complete problems, that is, by increasing the size of the problem by an amount scaling only polynomially in the size of the problem to be solved, one can reduce it to another **NP-Complete** problem. The archetypal example of an **NP-Complete** problem is 3-SAT, proven by Cook [28], to which Karp then famously reduced 21 problems [29].

Typically more difficult than decision problems, minimization or maximization problems are those of finding a string that extremizes an objective function C, given that it takes real values on some range. Such problems can readily be converted to decision problems, by asking the question:

Does there exist a string with a value greater/less than x?

and asking such a question for a number of different values. Therefore, the same complexity classes can be applied to extremization versions of problems, with some popular examples of extremization problems being the knapsack problem, the traveling salesperson problem, and the number partitioning problem, all of which are examples of **NP** complete problems.

1.4.2 Approximate Optimization

While **NP-Complete** problems necessarily require exponentially growing resources on a classical computer[7]—to determine whether they are satisfiable or not, in the case of decision problems, or to find the extremizing input in the case of extremization problems—the time required to find an approximate solution is not so severely constrained. For example, the Knapsack problem—that of maximizing the total value of a selected set of items of given value and weight, given that their individual weights do not exceed some maximum capacity—can be approximated to multiplicative error ϵ time scaling as polynomially in ϵ, and, therefore, one can produce an answer arbitrarily close to the optimal in polynomial time [30]. Typically, in approximate optimization one aims to maximize the approximation ratio, that is, the expected attained solution value divided by the expected optimum value of the cost function.

[6]The halting problem is that of determining whether an algorithm will terminate or not. It was notably proven undecidable by Turing, and therefore, cannot be solved on a deterministic, nor a non-deterministic, Turing machine.

[7]Assuming $\mathbf{P} \neq \mathbf{NP}$, which is a typical and generally uncontroversial assumption in the field of computer science.

1.5 Problem Types

Problem types in combinatorial optimization will typically fall into two categories, constrained and unconstrained problems. Constrained problems consist of two sets of states, a feasible subspace, and a non-feasible subspace. Where the fact that there is a non-feasible subspace for a problem is usually the consequence of ensuring that the can be expressed conveniently at the cost of not all states being feasible. Constraints can be considered in two main ways, firstly, by ensuring that an algorithm exploring the space of possible solutions only makes moves between feasible states, or, more crudely, by allocating large penalties to states that do not satisfy constraints. In this thesis, we consider unconstrained problems, though, some results could be generalized to the constrained setting.

Maximum cut MAX-CUT problems are defined by a graph G with vertices and edges, V, E. The decision version of the problem is to determine whether there exists a bisection of the graph splitting the vertices into two sets with a given number l of edges between two sets. This can be shown to be an **NP-Complete** problem.

For the maximization version of the problem, the objective function takes the form

$$C(z) = \sum_{(i,j)\in E(G)} z_i \oplus z_j \tag{1.18}$$

where z_i is the ith bit in z, which describes a state, or partition, of the graph. One can complicate this problem further by applying it to a weighted graph, with each term acquiring a multiplicative factor w_{ij}. Notably, the Goemans-Williamson algorithm [31] approximately solves MAX-CUT instances, attaining an approximation ratio of at least 0.878 times the optimal value.

k-Exclusive-OR k-XOR problems generalize MAX-CUT problems to hyper-graphs. To do so, clauses cover k variables and evaluate the parity of the sum of such variables. Therefore, the objective function can be written as:

$$C(z) = \sum_{\text{edge}\in E(G)} \bigoplus_{i\in\text{edge}} z_i \tag{1.19}$$

in which the clauses are local to a greater number of variables and the notation $\bigoplus$ refers to the modulo 2 sum. In this case, the edges are taken from a hyper-graph G with hyper-edges containing k variables.

k-Satisfiability SAT problems are typically those formed with Boolean formulas involving variables and their negations. k-SAT problems consist of OR clauses between k different variables. So, a clause may take the form:

$$C_a(z) = \neg z_{i_1} \vee z_{i_2} \vee \ldots \vee z_{i_{k-1}} \vee \neg z_{i_k} \tag{1.20}$$

Where $\neg$ applies a not gate to the binary variable z_i. In the case of the XOR version, the OR is replaced by the exclusive OR. For the decision version of the problem, asking whether every clause is satisfiable, k-SAT problems are **NP-Complete** for $k > 2$, with the $k = 2$ case being efficiently decided in polynomial time. For the XOR version, the decision problem is in P. When considering the maximization version, the $k = 2$ version alongside the XOR version becomes **NP-Hard**.

Sherrington-Kirkpatrick & Mixed-Spin Glasses While the aforementioned problems gain their complexity from the presence or non-presence of possible clauses, Sherrington-Kirkpatrick mixed-spin problems have every possible clause present, with, however, instance-to-instance variation entering in the form of Gaussian distributed weights. In lay terms, the Sherrington-Kirkpatrick problem is that of splitting a group of n people into two groups which maximize the preferences of all n people, given each pair, or k-sized group of people has a given allocated preference value, which could be any real number. The question of whether all clauses can be satisfiable is, given the number of clauses, for modest problem sizes, certain to be negative. That is, if the locality of the clauses, k, is greater than the trivial case of $k = 1$. The clauses are of the XOR, parity type, without the possibility for negated variables. As such, clauses take the form:

$$C_a(z) = g_a \bigoplus_{i \in S_a} z_i \tag{1.21}$$

with $g_a \sim \mathcal{N}(\mu, \sigma^2)$, being typically Gaussian distributed with given parameters. This would, in a physics context, usually be re-centered and rescaled with no fundamental change to the important properties of the problem, to correspond with the energies of fully connected spin glasses.

1.5.1 Random Graph Types

When sampling from randomly generated problem types, one has options as to how such graphs are generated. Problems such as MAX-CUT have simple graph interpretations, with edges directly associating with clauses in the problem. SAT problems, on the other hand, have more complexity in that there are 2^k clause types for each possible hyperedge, due to the possibility for negated variables.

Erdős Rényi Each possibly occurring clause is selected with a filling factor, or probability, f. For a problem-size-independent phenomenon, often such a filling factor is selected such that the ratio of clauses to variables remains constant.

Regular Graphs Ideally, each possible k-regular graph—graphs where each vertex is connected to exactly k neighbors—is selected with equal probability. Such unbiased selection is nontrivial for large graph sizes.

Computational complexity classes typically refer to worst-case complexity, so not all random graphs for a specific problem type will yield computationally hard instances. For many problems, such as k-SAT and others, there exists a deep connection to the thermodynamic concept of a phase transition. When considering random SAT instances, there exists a critical ratio of clauses to variables at which the probability that a SAT instance is satisfiable rapidly drops from unity to zero. Such phase transitions have the quality of becoming sharper with increasing problem size, and discontinuous in the limit of large problem sizes, as one finds in the thermodynamic theory of phase transitions. Knowledge of where such transitions lie for a given class of problems is instructive as it allows for a method to generate ensembles of problems that are not decidable in polynomial time, with the implication that such

parameters also result in hard maximization variants. Such hard problems are good targets for the use of heuristic algorithms.

1.6 Classical Algorithms for Combinatorial Optimization

Combinatorial optimization as previously introduced is an area of study within the field of optimization that deals with problems involving discrete choices. These problems often arise in a variety of practical contexts, such as scheduling, routing, and resource allocation. In this section, we will briefly introduce some of the common methods used by classical computers to solve such problems, including exact algorithms, approximation algorithms, and heuristics. We will also discuss the strengths and limitations of these approaches, and, later, explore the potential for quantum computers to improve upon these methods. By understanding the various ways in which classical computers can solve combinatorial optimization problems, we can gain insight into the potential benefits and challenges of using quantum computers for optimization in the future.

1.6.1 Hill Climbing

Hill climbing is the simplest heuristic method one could employ to find a solution to a combinatorial optimization problem. Such an algorithm on starts at a string z_0 and evaluates the objective function for neighboring states. Selecting the highest value neighbor, one repeats this process until a maximum is found (No neighbors are better than the currently occupied state). This strategy fails for problems in which local maxima are present, with there being no mechanism by which one may avoid such features.

1.6.2 Simulated Annealing

Simulated Annealing is a stochastic metaheuristic algorithm inspired by the behavior of physical systems undergoing cooling. Namely, crystal structures such as glasses and metals, or even chocolate, when cooled, find different crystalline structures depending on the rate of cooling. Fast rates of cooling result in coarse-grained crystalline structure without long-range order, with slow annealed crystals exhibiting long-range order and a more global minimum of free energy.

To simulate this process, and to find a good solution to an extremization problem, one takes a problem characterized by an objective function $C(z)$ and a start point z_0, alongside an acceptance probability, normally taken to be:

$$\mathrm{P(acceptance)} = 1 - e^{\frac{C(z_{\mathrm{new}}) - C(z)}{T}}. \tag{1.22}$$

The algorithm takes its active position z, initially chosen to be z_0, and randomly chooses a neighbor (typically flipping one bit in z at random). The value of the objective function is then evaluated for this new value. If the new objective function value is lower than that of the active position, the new position is taken as the active position and the process repeats. If the new position is worse than the active position, the new position is taken only with probability P(acceptance). The key

idea here is that the temperature T should be started high and gradually reduced. As such, early in the annealing process, more global exploration of the landscape takes place as transitions increasing the objective function are allowed to occur. As the temperature decreases to near zero, the algorithm transitions to local hill climbing.

1.6.3 Backtracking

Backtracking is a popular algorithm that offers a more effective approach than the brute-force method of exhaustively checking every solution to a problem until a satisfactory answer is found. This technique is particularly useful in decision problems that consist of a series of clauses that must all be satisfied. The algorithm begins by randomly assigning a value to the first variable, and then checks if all clauses are still feasible. If they are, it assigns a value to the next variable and repeats the process. This leads to a depth-first exploration of the tree of potential allocations. This is more efficient than brute-force search because when a partial allocation of values to variables is determined to be infeasible, multiple potential solutions are automatically eliminated from the space of possible solutions. Backtracking algorithms are the foundation of most modern SAT solvers, such as DPLL [32] and conflict-driven clause learning [33].

1.7 Quantum Algorithms for Combinatorial Optimization and Search

In 2014, it was proposed that discrete, gate-based quantum computers might be good at solving combinatorial optimization problems, with the help of classical optimizers [34]. At this time, it was already known that quantum annealers might provide efficient solutions for optimization problems, by encoding problems into the system Hamiltonian, but, demonstrating that this technique can provide significant speedups over classical techniques has proven difficult. Gate-based quantum computers are expected to have the advantage of quantum error correction allowing levels of noise to be reduced arbitrarily in future machines, whereas it is likely that creating fault-tolerant annealing machines is much more difficult. Thus, one might expect that gate-based algorithms might provide a clearer path to usefulness as quantum systems scale to large sizes and with the assistance of error correction, detection, or mitigation techniques.

1.7.1 Grover's Algorithm

Grover's algorithm [20] is an algorithm for searching an unstructured database on a quantum computer for a marked element denoted z^*. In its original formulation, the algorithm takes access to an *oracle* to whom the question may be asked,

> Is element x a marked element?

To which the oracle responds with a yes or no answer. Assuming that there are N possible elements of which one may be marked, a classical computer operating on classical information must query the oracle $N/2$ times in the average case before they

receive an affirmative answer. This increases to $N-1$ times in the worst possible case, in which every element with the exception of the marked element is verified not to be so.

Quantum computers dealing in quantum information leverage the possibility of querying the oracle with a quantum superposition state of possible elements. That is, the oracle is represented by a procedure that applies an operator $U_G = |z\rangle \rightarrow (-1)^{f(z)} |z\rangle$ to an array of qubits in a state describing the selected element. Where, the function f is determined by:

$$f(z) = \begin{cases} 1, \text{ if } z = z^* \\ 0, \text{ otherwise.} \end{cases} \tag{1.23}$$

and so

$$\hat{U}_{\text{Oracle}} = \mathbb{I} - 2 |z^*\rangle \langle z^*| \tag{1.24}$$

which is a reflection about the marked state $|z^*\rangle$. Grover's algorithm also makes use of the Grover diffusion operator. This operator, reflecting the state about the initial, superposition state over all possible computational basis states,

$$|\text{Initial}\rangle = \sum_z \frac{1}{\sqrt{N}} |z\rangle = |+\rangle^{\otimes n}, \tag{1.25}$$

is given by:

$$\hat{U}_{\text{D}} = \mathbb{I} - 2 |+\rangle^{\otimes n} \langle +|^{\otimes n}. \tag{1.26}$$

Repeated application of these operators induces an oscillation between an initial state, which is a superposition state over all possible elements

$$|\text{Initial}\rangle = \sum_z \frac{1}{\sqrt{N}} |z\rangle = |+\rangle^{\otimes n} \tag{1.27}$$

and a state which has high overlap with the target state $|z^*\rangle$, encoding the solution to the search problem. With the repeated application of these operators, the overlap of the resulting state increases linearly with the number of applications. Since the probability of measuring the marked state is proportional to the square of this overlap $\langle \psi | z^* \rangle$, this probability, therefore, initially increases as the square of the number of steps, leading to an asymptotic time complexity of $O(\sqrt{N})$ required queries, rather than $O(N)$ queries for the classical case of manually checking individual elements. Grover's algorithm, while asymptotically optimal, can also be improved with the use of classical optimizers [35] which is an area of study especially relevant to Chapter 3.

Realizations of Grover Oracles

While described as an algorithm for searching an unstructured database, such a database should be queryable by a quantum computer, meaning such an algorithm is unlikely to be useful for a naive database search, however, there are less obvious uses of Grover's algorithm for which a useful implementation could be derived. Potential examples of Grover oracles that could yield useful implementations could be:

- *Cryptographic hash functions*—Hash functions such as SHA-3 [36] are designed to be one-way. It is easy to compute the *hash* of a message, but hard to reverse engineer a *hash* to find a corresponding message—a collision. By applying a hash function to a quantum array, and applying a phase shift if the array takes some value, one can realize a Grover oracle for this context. This allows one to find collisions in hash functions asymptotically faster than possible than with random guessing on a classical computer, since, by design, such functions do not have an easily exploitable structure.

- *Threshold functions for combinatorial optimization problems*—One can evaluate whether a given state of a quantum computer has an objective function exceeding or lesser than some threshold and apply a π phase on this condition, thus producing a Grover oracle-type operation.

- *Decision problems*—One can realize a Grover oracle that applies a phase on the condition that all clauses in a combinatorial optimization problem are satisfiable. This is a specific example of a threshold function set to the maximum value.

1.7.2 Adiabatic Quantum Computing

The Adiabatic theorem, coined in 1928 by Max Born and Vladimir Fock [37], is stated as follows:

> A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian spectrum.

70 years later, it was noted [23] that this could be exploited to perform quantum computations. By initializing a system in the ground state of some known Hamiltonian one slowly turns off, while turning on a Hamiltonian for which the ground state encodes potential solutions to a combinatorial optimization problem, one can measure this ground state and read out a solution. The caveat here is that the interpolation between the initial and final Hamiltonian must be performed sufficiently slowly to suppress leakage of probability amplitude from the instantaneous ground state. That is, for time evolution in which the minimum gap between the instantaneous ground state and the first excited state takes value ΔE, the time taken for the adiabatic algorithm to find a ground state of the problem Hamiltonian scales as:

$$T = O\left(\frac{1}{\Delta E^2}\right). \tag{1.28}$$

This means that for increasingly large problem sizes, the time taken by an adiabatic quantum computer to find a ground state scales exponentially in the number of qubits if typical problems have a minimum gap that scale as such. Therefore, the coherence time required for a device to solve useful problems becomes extremely large, and error correction would certainly be required. As of yet, a practical scheme for error-correcting the adiabatic algorithm has not been presented, with prospects for error-correcting schemes that can preserve the usefulness of square-root speedups generally considered to be poor.

Adiabatic quantum computing and Grover's algorithm, applied to the search problem, represent an (asymptotically) optimal way of searching and solving the Grover problem [38]. The Grover problem, however, by design, does not have any exploitable structure. But what of structured problems[8]? Post-Grover algorithms for searching for needles in haystacks focus on solving problems that are messier. Rather than simply marking a set of correct answers, optimization problems have a number associated with each possible element, leading to significantly increased complexity. Such problems can be harder, in that even if one finds the element with optimal cost, one might not know that this is the best solution without exhaustively checking every other possible solution. Problems of this type might be totally unstructured, in the case of random cost models, studied in the context of the quantum approximate optimization algorithm in Chapters 3 and 4, or highly structured, in the case of MAX-CUT, and SAT and SK problems studied throughout the thesis. These problems are structured, that is, if you take two similar states—states that share common bits—such states are likely to have similar values for the problem's objective function. There are now several candidates for quantum algorithms that could exploit such structured combinatorial problems. These are summarized in the following.

1.7.3 Quantum Random Walks

Developed as an alternative approach to quantum computing in the mid-2000s, quantum random walks saw attention due to the success of classical random algorithms which perform well on problems such as high-k k-SAT [39] and computing matrix permanents [40]. Quantum walks were eventually found to be a separate method of expressing a universal model [41] of quantum computation and are a quantum analog of classical random walks, in which the addition of quantum mechanical phase allows for more complex dynamics than the Markovian dynamics one finds in the classical context. To summarize the field, a quantum random walker, by means of quantum interference, travels further from an initial position than a classical walker by means of steps that are discrete or continuous. Childs et. al. [42] found that a problem of traversing a specifically designed graph, accessed by the quantum algorithm via oracle queries, can be solved exponentially faster by a continuous-time quantum random walk than by any possible classical walker. Using a search oracle to obtain square-root speedups in such problems was also shown possible in the quantum random walk context [43]. The algorithms of quantum random walks, adiabatic quantum computing, Grover's algorithm, and the quantum approximate optimization algorithm, are all examples of algorithms taking advantage of a problem and driver Hamiltonian, and using some schedule to apply such Hamiltonians to the aim, to find good solutions to a problem. The thing that differs in the algorithms, is the choice of schedule used, whether continuous or discrete, and, the choice of driver and problem.

[8]The question of what structure is in this context is explored in Chapter 4, and ultimately means that taking some move on some small move on a relevant metric on the space of possible states results in a correspondingly small change in energy or combinatorial optimization objective function.

1.8 Realizing Qubits with Superconductors

There are many candidates for the realization of qubits. The most viable options for this are currently those of superconducting circuits, trapped ions, and quantum computing with photon-based qubits. Each of such options has advantages and disadvantages. For example, all trapped ions or photons are indistinguishable from other trapped ions or photons of the same species, bypassing any issue of fabrication defects that plague the design of superconducting qubits, which must be precisely etched or deposited onto a surface with lithographic techniques. Trapped ions and photonic implementations have their own issues, with the challenges of keeping individual ions trapped alongside others, to be addressed by lasers, or the challenge of inducing interactions between photons, which at low energies, is an uncommon phenomenon.

Non-linearity is a required property for optical quantum computers and superconducting quantum computers, though, for different reasons[9]. When quantum computing with photons—which do not readily interact with other photons—one requires a medium facilitating such interactions. An option for this is the use of Kerr media, in which the index of refraction is not independent of the power of incident light or the number of photons transiting. In linear optical quantum computing, such non-linearity is attained via the post-selection to generate entangled pairs of photons, which can then be used in a measurement-based quantum computing scheme. In superconducting quantum computing devices, non-linear elements are needed in order to ensure that the ground and first excited states of the system are addressable independently from an upper spectrum of states. Without non-linear elements, one simply derives the physics of a quadratic harmonic oscillator in which energy levels are spaced equally, with external influences acting on all transitions between adjacent states. In superconducting qubits, the non-linear element available is the Josephson Junction, for which the phenomenon of superconductivity yields a nonlinear inductance.

1.8.1 The Josephson Junction

Josephson Junctions consist of two superconducting regions separated by a non-superconducting insulating medium. Each superconducting region is described by a wave function associated with the non-commuting phase and number observables. The wave function on one side of such an insulator interacts with that of the other side and for low energies tends to equalize the phase on either side. Differences in the phase of such wave functions, therefore, incur an energetic cost. The current flowing across a Josephson tunnel is predicted and measured to be

$$I = I_{\mathrm{C}} \sin \phi \tag{1.29}$$

where ϕ is the difference in phase of the two superconducting materials. I_{C} is the maximum current sustained without entering the normal phase of the superconductors. The voltage across the junction is given by:

$$V = \Phi_0 \partial_t \phi \tag{1.30}$$

[9]For trapped ions the Hamiltonian is already strongly non-harmonic, and therefore no extra non-linear element is needed.

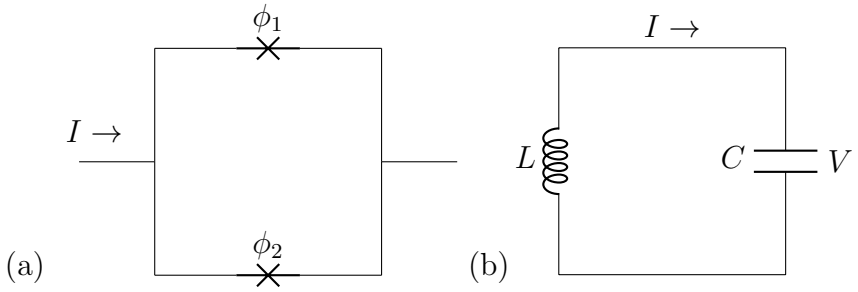

Figure 1.3: (a) A circuit diagram for a DC-SQUID, with two Josephson junctions arranged in a loop. SQUIDs act as Josephson junctions with tunable Josephson energy. (b) A circuit diagram for an LC circuit.

with $\Phi_0 = \frac{h}{2e}$ is the superconducting flux quantum for cooper pairs of electrons. Combined, these equations describe an energy-conserving circuit element storing energy:

$$U = E_J \cos\phi, \tag{1.31}$$

where $E_J = \frac{I_C \Phi_0}{2\pi}$. Since the junction also bears resemblance to a capacitor, with differing charge density on either side of the insulating junction, a capacitance C_J is also associated with the circuit element with energy $C_J V^2$ stored.

1.8.2 The SQUID

SQUIDs, or Superconducting QUantum Interference Devices are loops of superconducting material containing two Josephson junctions. A circuit diagram showing a SQUID schematic can be seen in Figure 1.3a. If the current across a DC-SQUID is I then the phases across the two junctions are matched such that the sum of the Josephson currents equals I, so, one can write

$$I = I_C \sin(\phi_1) + I_C \sin(\phi_2). \tag{1.32}$$

It is required that the phase of the macroscopic wave function satisfies the boundary condition of its looped topology. As such the sum of the changes in phase at each junction should be equal to the external flux-applied phase. The desired properties of a Josephson junction can be engineered by varying the material used for the junction, alongside the geometry of the junction. For example, the greater the surface area of the junction, the greater the critical current, and conversely for the thickness of the junction.

1.8.3 Deriving Hamiltonians from Circuit Diagrams

A standard method to determine the Hamiltonian of a superconducting device is to write down the classical equations of motion of a circuit and associate such equations with a Lagrangian and therefore Hamiltonian. One can then apply canonical quantization to obtain a quantum description. The energy stored in such a linear inductor and capacitor can be written as:

$$E_L = \frac{L}{2} I^2, \;\; E_C = \frac{C}{2} V^2. \tag{1.33}$$

The charge $Q(t)$ across the capacitor and the flux threading the inductor $\Phi(t)$ can be written as

$$Q(t) = \int_{-\infty}^{t} I(t')\,\mathrm{d}t', \;\; \Phi(t) = \int_{-\infty}^{t} V(t')\,\mathrm{d}t' \tag{1.34}$$

since the current through a linear inductor is proportional to the threaded flux via the inductance, $\Phi = LI$ and $Q = CV$, one can combine the above equations to write:

$$\Phi(t) = LC\ddot{\Phi}(t) \tag{1.35}$$

This equation of motion can be associated with the Euler-Lagrange equation

$$\frac{\partial \mathcal{L}}{\partial \Phi} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = 0 \tag{1.36}$$

for the Lagrangian

$$\mathcal{L}(\Phi, \dot{\Phi}) = \frac{C\dot{\Phi}^2}{2} - \frac{\Phi^2}{2L} \tag{1.37}$$

from which the Hamiltonian can be derived as

$$\mathcal{H} = \dot{\Phi}\frac{\partial \mathcal{L}}{\partial \dot{\Phi}} - \mathcal{L} = \frac{Q}{2C} + \frac{\Phi^2}{2L} \tag{1.38}$$

with the conjugate momentum

$$Q = \frac{\partial \mathcal{L}}{\partial \dot{\Phi}} = C\dot{\Phi}. \tag{1.39}$$

These canonical coordinates can be promoted to quantum observables with

$$Q \to \hat{Q}, \;\; \Phi \to \hat{\Phi} \tag{1.40}$$

where the observables obey their commutation relations

$$\left[\hat{\Phi}, \hat{Q}\right] = i\hbar \tag{1.41}$$

which implies that the observables are well described by a quantum mechanical wavefunction. In the usual way, a quantum harmonic oscillator can be expressed as a linear combination of ladder operators rather than its classical canonical coordinates. As such one can write:

$$\hat{\Phi} = \sqrt{\frac{\hbar Z}{2}}(\hat{a} + \hat{a}^\dagger), \;\; \hat{Q} = i\sqrt{\frac{\hbar}{2Z}}(\hat{a} - \hat{a}^\dagger) \tag{1.42}$$

where $Z = \sqrt{L/C}$ is the characteristic impedance of the LC oscillator with frequency $\omega = 1/\sqrt{LC}$.

1.8.4 Superconducting Qubit Varieties

There is a continuum of possible designs for superconducting qubits, each with drawbacks and advantages, here we list the types that have seen the greatest attention in the past two decades of development.

Phase Qubits

A phase qubit is perhaps the simplest superconducting qubit one could produce, consisting of simply a Josephson junction, biased by an applied current. The first experiments demonstrating this qubit type occurred in 2002 [44], with a coherence time of just 10 ns. A demonstration of coupled phase qubits followed in 2006 [45] when various improvements to the design had been made. The phase qubit design results in a tilted washboard potential with the two qubit states localized to a well along this potential. Single qubit gates can be performed by DC and microwave pulses of the applied bias current, with readout being particularly easy in this case. Since in the case that the qubit is excited from the first excited state to a higher state, the well no longer traps the phase and the device behaves as an open circuit, with the associated appearance of a measurable voltage. The Hamiltonian can be written as:

$$\hat{H}_{\text{Phase}} = E_{\text{C}}\hat{Q}^2 - E_{\text{J}}\cos\hat{\phi} - \frac{\Phi_0}{2\pi}I\hat{\phi} \tag{1.43}$$

where I is an applied bias current, from which the tilted washboard potential can be observed in the phase observable $\hat{\phi}$. The associated circuit diagram can be seen in Figure 1.4

Flux Qubits

Flux qubits [46, 47] are characterized by mesoscopic persistent currents. While the simplest incarnation of a flux qubit can consist of just a single Josephson junction and inductor, a radio frequency SQUID (rf-SQUID) qubit, flux qubits can also be produced with any odd number of junctions. A popular sub-variant is that of a three-junction qubit, in which splitting one junction to form a SQUID results in additional tune-ability. Regardless of the components used, and the degrees of freedom by which external flux may influence the Hamiltonian, all flux qubits are characterized by double-well potential, which, away from the symmetric degeneracy point, has ground and first excited states well-characterized by clockwise and anticlockwise persistent currents. One of the two Junctions is fabricated with Josephson energy greater than the other two. For flux qubits in which many junctions are utilized, the name fluxonum [48] has been coined for inductively shunted versions that reduce sensitivity to charge noise. The Hamiltonian for a one-junction flux qubit typically depicted in Figure 1.4c takes the form:

$$H_{\text{Flux}} = E_{\text{C}}\hat{Q}^2 + E_{\text{L}}(\hat{\phi} + \phi_{\text{ext}})^2 - E_{\text{J}}\cos\left(\hat{\phi}\right) \tag{1.44}$$

where the constant E_{J} can be made to depend on an external control parameter with the use of a SQUID in place of a Josephson junction. Control of the qubit is achieved via the external flux parameter. Qubits can be operated at the degeneracy point, for which the qubit's energy gap becomes first-order insensitive to external magnetic fields [49, 50] or in a de-tuned configuration, in which strong qubit-qubit inductive interactions can be engineered, at the cost of a high exposure to environmental flux noise[10].

[10]This is a tradeoff made by the designers of the d-wave quantum annealers, who produce large devices with high connectivity and qubit counts, but, have never published data on coherence times for qubits therein.

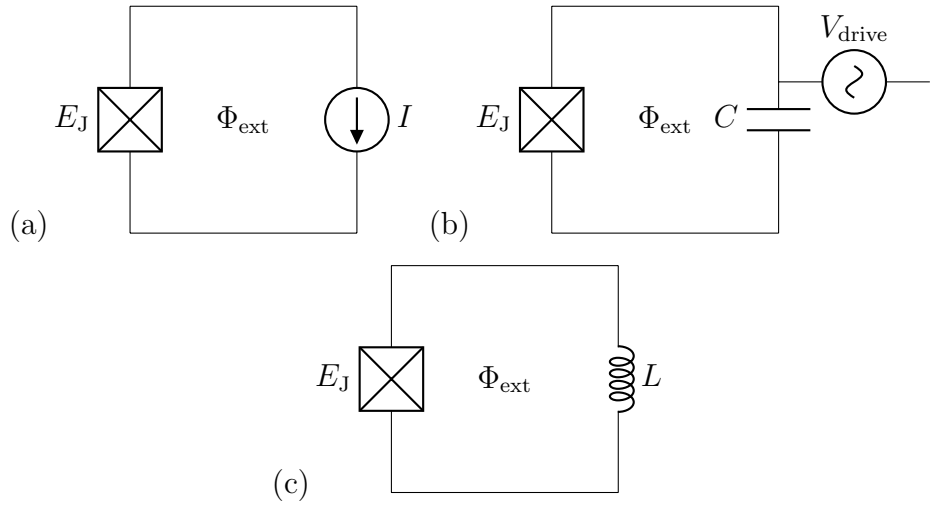

Figure 1.4: (a) A circuit diagram for a phase qubit, consisting simply of a current source biasing a single Josephson Junction. (b) A circuit diagram for a charge or transmon qubit, consisting of an island on which a number of charges can be trapped. (c) A circuit design for a one-junction flux qubit, in which a double well potential is formed in the phase variable of the qubit, resulting in ground and first excited states corresponding to oppositely propagating persistent currents.

Charge & Transmon Qubits

Charge or Cooper pair box qubits and their descendants the transmon qubit are designed such that their Hamiltonians lie close to those of the quantized LC circuits, with non-quadratic perturbing terms from Josephson junctions allowing the interrogation of only the lowest lying two states [51]. First investigated in 1998 [52], charge qubits presented an alternative design choice to the previously investigated flux qubits, and, were thought to potentially have more precisely controllable properties. Namely, it was expected that precise voltage control might be more easily facilitated than precise flux control, which is subject to spurious magnetic field fluctuations. The Hamiltonian takes the general form:

$$\hat{H}_{\text{Charge}} = E_{\text{C}}(\hat{Q} - Q_0)^2 - E_{\text{J}} \cos\hat{\phi}, \tag{1.45}$$

with the control parameter entering in the form of the offset charge, Q_0, or an externally applied voltage. There also exists the possibility to promote the Josephson junction to a SQUID, allowing an additional control parameter, resulting in a qubit type known as a split cooper pair box. In their initial incarnation, charge qubits carried a ratio of $E_{\text{J}}/E_{\text{C}}$ around unity. In this configuration, however, the devices suffer from poor coherence times due to sensitivity to charge noise. To address this sensitivity, the transmon qubit was developed. Transmon qubits use large capacitors to shunt the intrinsic capacitance of the Josephson junction, increasing the ratio $E_{\text{J}}/E_{\text{C}}$ to values in excess of 50. Modern transmon designs use large X-shaped designs, thereby known as xmons, to facilitate the design feature of extra capacitance, while also offering a natural form factor for the tiling of qubits to facilitate interactions. Transmons can achieve coherence times in the hundreds of microseconds [53], due to their designs, and also other developments in relevant materials

science, allowing for the rapid pace of system size increases. This has resulted in systems producing the landmark result of quantum advantage [54].

1.8.5 Restriction to Qubit Basis

The qubits introduced here have an infinite number of energy levels that can be occupied. Assuming that the qubit can be initialized in the ground or first excited state and that operations that we apply to qubit do not result in an appreciable transfer of population from the lowest two lying states to higher energy levels, one can restrict the operators to their action on the two lowest-lying states. Ensuring operations are restricted to the qubit subspace is achieved practically by applying the tools of optimal control theory. The restriction of the Hamiltonian can be achieved mathematically by projecting the infinite-dimensional Hamiltonian onto the qubit basis so

$$\hat{H} \to \sum_{i,j \in \{0,1\}^2} |i\rangle \langle i| \hat{H} |j\rangle \langle j| \tag{1.46}$$

where $|0\rangle$, $|1\rangle$ are the ground and first excited states of the qubit. The single-qubit Hamiltonian then becomes expressible in the desired basis as:

$$\hat{H} = \frac{a}{2} Z + \frac{b}{2} X \tag{1.47}$$

which is readily diagonalizable to produce eigenenergies and the operating frequency of the qubit. In the case of transmon qubits, the computational basis is taken to be that of the qubit eigenbasis, so $a = \omega_{12}, b = 0$, where ω_{12} is the qubit frequency. For Flux qubits, the eigenbasis is taken to be that of the persistent current basis, and as such both a splitting a and a tunneling b term are present in the computational basis. For a Hamiltonian with both a and b present, the qubit frequency in its eigenbasis is given by,

$$\omega_{12} = \sqrt{a^2 + b^2} \tag{1.48}$$

which can be derived via the diagonalization of the single-qubit Hamiltonian.

1.8.6 Interactions Between Qubits

To generate interactions between physical qubits, one should couple the degrees of freedom of individual systems together with Hamiltonian terms. Depending on the form of physical coupling and qubit type used, the coupling terms take different forms when expressed in the single-qubit eigenbasis. For a more complete reference, the author refers the reader to [55].

Capacitive coupling can be achieved by placing a capacitor between two qubits, either directly, or indirectly in the case that one requires a tunable coupler. Qubit-qubit interactions via capacitors take the form:

$$H_{\text{int}} = C_{12} V_1 V_2 \tag{1.49}$$

where C_{12} is the capacitance of the capacitor linking qubits 1 and 2. If the coupling capacitor has a capacitance much lower than the qubit capacitance, so as to not

affect the properties of the qubit, the voltages can be expressed as:

$$V_i \to \frac{\hat{Q}_i}{C_i} \tag{1.50}$$

as the classical voltage-charge relation would suggest. This coupling gives rise to a term of the form:

$$\hat{H}_{12} = \frac{\hat{Q}_1 \hat{Q}_2 C_g}{C_1 C_2} \tag{1.51}$$

In the case of the transmon Hamiltonian, one has $\hat{n} \propto \hat{a} - \hat{a}^\dagger$ so, the term takes the form proportional to:

$$(\hat{a}_1 - \hat{a}_1^\dagger)(\hat{a}_2 - \hat{a}_2^\dagger) \to X_1 X_2 \tag{1.52}$$

when projected into the low energy subspace spanned by the qubit, which is possible when the anharmonicity of the qubit is such that the upper levels remain irrelevant to the dynamics. This coupling is described as transverse since the coupling terms have no diagonal elements in the qubit basis.

Inductive coupling Similarly, coupling via electromagnetic inductance can be achieved by bringing parts of a qubit into the vicinity of another, such that current in one induces current in another via Lenz's law. The interaction term from such a design takes the form:

$$H_{12} = M_{12} I_1 I_2 \tag{1.53}$$

where I_1, I_2 are the currents flowing across the qubit inductors and M_{12} is the mutual inductance between the two qubits. For sufficiently small inductive coupling, one produces a coupling term well approximated by

$$M_{12} I_1 I_2 \to M_{12} I_{1c} I_{2c} \sin\hat{\phi}_1 \sin\hat{\phi}_2. \tag{1.54}$$

away from this limit of low coupling, considerations must be made to correct for the loading effect, in which effective inductances are shifted from their isolated values in the presence of others. To make this correction, one inverts an inductance matrix, which is a procedure followed, for a very symmetric case, in Chapter 6.

Inductively coupled flux qubits, assuming the barrier between well is sufficiently large such that the ground and first excited state are well described by superpositions of phase-localized persistent current states, yield transverse couplings. This means, that, the interaction takes the form with

$$\hat{\phi}_1 \to Z_1, \ \hat{\phi}_2 \to Z_2 \tag{1.55}$$

and so the flux qubit Hamiltonian takes the form

$$\hat{H} = \sum_{i=1}^{2} \frac{a}{2} X_i + \frac{b}{2} Z_i + g Z_1 Z_2 \tag{1.56}$$

in the persistent current basis. With a sufficiently high inter-well barrier, and, a non-zero difference in barrier height, the flux qubit Hamiltonian becomes close to diagonal in the persistent current basis.

Chapter 2

The Quantum Approximate Optimization Algorithm

In this chapter, the Quantum Approximate Optimization Algorithm (QAOA), alongside the existing relevant literature, is introduced, setting the scene for the work presented in this thesis. QAOA is a gate-based variational quantum algorithm, designed for use on near-term intermediate-scale quantum computers, to solve problems in combinatorial optimization. It is a member of a class of algorithms, that if quantum computers are to be useful before the large cost of quantum error correction is required, might be capable of maximally exploiting such usefulness to solve problems for which classical computers suffer poorer asymptotic scaling than possible on a quantum computer.

QAOA was introduced by Farhi et. al. [34] in 2014 as an adaptation of the adiabatic model of quantum computing to contemporary, gate-based devices. As in similar quantum algorithms, QAOA consists of a combinatorial optimization problem, entering into the algorithm as a phase separation operator, alongside a mixing operator inducing transitions between states. Unique to QAOA, is the schedule in which these two Hamiltonians are applied, with QAOA being analogous to a bang-bang control version of quantum annealing or quantum walk-based algorithms. Shortly after its discovery, Farhi et. al. found that QAOA provided greater approximation ratios than the previously best-known classical algorithm, on the specific problem type of MAX-E-3-LIN-2 [56]. Following this result, a quantum-inspired classical algorithm [57] was discovered, finding greater approximation ratios for the problem in question. As is the case for similar algorithms such as adiabatic quantum computing and the quantum random walk algorithms, it was demonstrated by Jiang et al. [58] that QAOA can recover the square-root scaling of Grover's search algorithm, replicating Grover's speed-up without the need for Grover's mixing operator. Hadfield et al. discovered that QAOA driving operators can be modified such that a wide variety of problems can be solved without resorting to high-order penalty terms usually considered in an annealing or adiabatic-based approach [59], this is is a feature, enabled by the gate-based nature of QAOA, since continuous-time algorithms face an inability to compile operations from multiple gates. QAOA can be shown to be a universal context for quantum computation [60] in that angles can be found that allow any diagonal two-qubit unitary to be generated. That is, given a well-populated problem Hamiltonian with non-rationally-related pre-factors, it is possible for any general quantum computation to theoretically be performed.

In this section, we break QAOA up into parts and introduce these individually. We then consider some of the existing research in the field and the open questions that still exist.

2.1 Initial State

There are three constituent parts of QAOA. First, there is the initial state of the quantum register used in the algorithm. Typically, as in Grover's algorithm or adiabatic quantum computing, a quantum register is initialized to an unbiased mix of all possible basis states, given by:

$$|\text{Initial}\rangle = H^{\otimes n} |0\rangle^{\otimes n} = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle \,. \tag{2.1}$$

Where H is the usual Hadamard gate. While not explored in this thesis, alternative starting states could be utilized in which only states that satisfy hard constraints are present [59]. In this case, one could initialize the quantum register in some uniform superposition over a set of hard constraint-satisfying states. If such states S satisfy a hard constraint, such an initial state could take the form:

$$|\text{Initial}\rangle = \frac{1}{\sqrt{|S|}} \sum_{z \in S} |z\rangle \,. \tag{2.2}$$

Strategies encoding hard constraints are often pursued in the context of adiabatic quantum computing, for example, the LHZ scheme in which a plaquette system is used to spread logical qubits over multiple physical qubits, mapping two-body interactions to single-qubit terms [61]. Such strategies can also be employed when variables in problems are encoded as unary rather than binary values. Doing so can be advantageous as it can simplify interactions, at the cost of introducing non-valid states which need to be discouraged with penalty terms.

2.2 The Problem Hamiltonian

A second component is that of the phase separation operator. This is an operatorized version of the problem objective function, applying a phase to each computational basis state proportional to its cost as defined by the problem objective function. The phase separation operator takes the form:

$$\hat{U}_{\text{P}}(\gamma) = \mathrm{e}^{i\gamma \hat{H}_{\text{P}}} \tag{2.3}$$

and where the problem Hamiltonian $\hat{H}_{\text{P}}$ is such that

$$\hat{H}_{\text{P}} = \sum_{z=0}^{2^n-1} C(z) \, |z\rangle \langle z| \,, \tag{2.4}$$

where $C(z)$ is specified by a combinatorial optimization problem's objective function as introduced in Section 1.4. In terms of Z-based interactions on qubits, a generic cost function $C(z)$ can be decomposed via a Fourier transform as:

$$\hat{H}_{\text{P}} = \sum_{k=0}^{2^n-1} g(k) Z^{k_1} Z^{k_2} \ldots Z^{k_n} \tag{2.5}$$

where k_i is the ith bit of the binary representation of k, which indexes a possible Z-based interaction. Typical problems for QAOA should be such that g is sparse, being expressible with only a number of non-zero elements growing polynomially in the number of qubits. As such, compilation to near-term quantum computers should allow for QAOA state preparation in such a way that preserves Grover-style square-root speedups. The descriptions of a problem given by $g(k)$ and $C(z)$ are related via an n-dimensional binary-variable Fourier transform, otherwise known as the Walsh-Hadamard transform [62]. This transformation can be succinctly written as:

$$\vec{C} = \sqrt{2}^n H^{\otimes n} \vec{g} \tag{2.6}$$

for vectorized versions of g and C. Such a decomposition can be useful in the simulation of small circuits for QAOA and is further discussed in Appendix A.

2.3 The Driver Hamiltonian

Finally, the QAOA circuit makes use of a driver, or mixing operator, given by:

$$\hat{U}_{\mathrm{D}} = \mathrm{e}^{i\beta\hat{H}_{\mathrm{D}}}. \tag{2.7}$$

$\hat{H}_{\mathrm{D}}$ typically consists of single-qubit-X operators and can then be written as

$$\hat{H}_{\mathrm{D}} = \sum_{i=0}^{n-1} X_i \tag{2.8}$$

but can be extended to a wider class of operators in the interests of improving performance or keeping the state of the quantum computer inside a viable subspace of a problem. In this thesis, for example, we consider the extension of this operator to the specific case of the full-weight Grover driver, and doing so find analytical results for QAOA performance at infinite size.

Ultimately, the choice of driver in QAOA is arbitrary. In theory, the criteria for a good driver, are that it does not commute with a problem Hamiltonian, therefore generating non-trivial dynamics. While many potential options could drive transitions between computational basis states when alternated with a problem Hamiltonian, a stronger condition for a driver being *good* is that the driver exploits structure in the problem to which it is applied. In the context of adiabatic quantum computing, the question of exploiting problem structure is clearer than in the context of QAOA. It is well known that the annealing time needs to be proportional to the minimum gap between the ground and the first excited state. Choice of the driver can affect this gap, with existing works considering the addition of specifically tuned high-order terms to increase this minimal gap [63]. QAOA, on the other hand, does not depend so specifically on one particular parameter, though work in Chapters 3 and 4 work towards elucidating the impact of driver choice on the problem structure that is resolved by a given driver.

Work in this thesis concerns the standard X-driver used in QAOA and the Grover driver seen in Chapter 3, and, also, an additional line mixing operator in Chapter 4. One could also consider some range of drivers of increasing weight interpolating between these two extremes, but, preliminary numerical tests, have demonstrated that for reasons of poor synergy, or a difficult-to-optimize expected value landscape, such intermediate weighted drivers yield poor results on the problems tested.

2.4 The Algorithm

The QAOA ansatz state produced by the algorithm can be written as:

$$\left|\vec{\beta},\vec{\gamma}\right\rangle = \prod_{p'=1}^{p} \mathrm{e}^{i\beta_{p'}\hat{H}_{\mathrm{D}}}\mathrm{e}^{i\gamma_{p'}\hat{H}_{\mathrm{P}}} \left|+\right\rangle^{\otimes n}, \tag{2.9}$$

which can be efficiently prepared on a quantum computer given the driver and problem contain a polynomial number of terms, and, from which the resulting measurement outcome probability distribution can be sampled. Such a probability distribution is hypothesized not to be possible to sample from using resources scaling polynomially in the number of qubits on a classical computer. So, for sufficiently large instances, it is possible that QAOA can provide a method of solving combinatorial optimization problems that is intractable to simulate classically. The performance of a set of QAOA angles can be measured by the expectation value of the problem Hamiltonian under the QAOA state, given by:

$$\langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\gamma},\vec{\beta}} = \left\langle \vec{\beta},\vec{\gamma} \right| \hat{H}_{\mathrm{P}} \left| \vec{\beta},\vec{\gamma} \right\rangle. \tag{2.10}$$

Alternatively, and used in some works, but experimentally inaccessible on a real device implementation of QAOA would be the probability of measuring the best possible solution to a problem. Then, an objective function that could be considered would be described by:

$$\sum_{z\in\{z^*\}} \left| \left\langle z^* \middle| \vec{\gamma},\vec{\beta} \right\rangle \right|^2 \tag{2.11}$$

and by optimizing this one might obtain angles that more efficiently find the best possible solution to a combinatorial optimization problem. Such an objective function is considered by Akshay et al. [64] in work in which it is demonstrated that minimum circuit depths are required depending on problem characteristics.

2.5 Strategies for Finding Optimal Parameters

The choice of domain for the parameters $\vec{\gamma}$ and $\vec{\beta}$ for the two QAOA operators is often arbitrary but is typically taken such that

$$\vec{\gamma} \in [0, 2\pi]^p, \quad \vec{\beta} \in [0, \pi]^p. \tag{2.12}$$

This choice was used in the literature mainly due to the initial incarnation of QAOA focusing exclusively on solutions to MAX-CUT problems utilizing single-qubit-X drivers. For this context, the aforementioned range of angles produces the maximum span of possible operators, since the problem and driver operators are periodic outside these respective ranges. For problems that do not exhibit this periodicity, one might consider the use of a greater range of potential values. In the case of the work in Chapter 3, for example, the range of values for producing unique operators are not constrained, with $\vec{\gamma} \in \mathbb{R}^p$. Though there are strong arguments that values of these parameters optimizing the QAOA expectation value should be small since for large values of γ, problem Hamiltonians become unstructured and resemble operators simply applying random phases to each possible state.

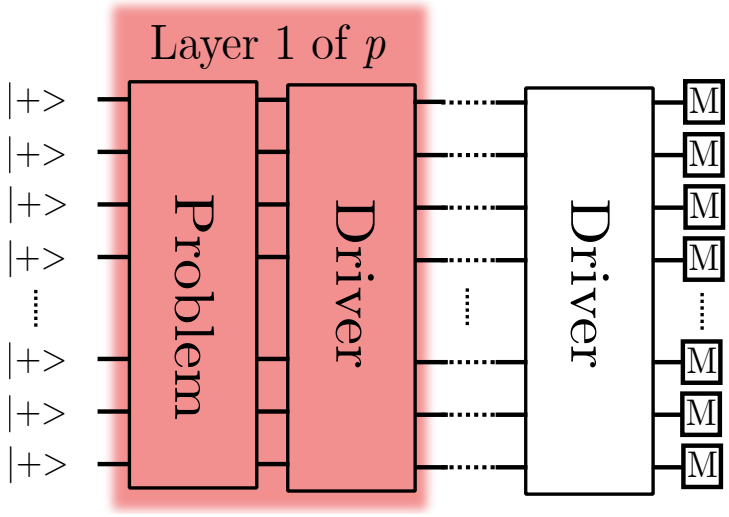

Figure 2.1: A typical QAOA state preparation circuit. A problem and driver Hamiltonian are alternated p times, applied to the $|+\rangle^n$ state, followed by measurement on all qubits.

2.5.1 Optimizers

There are numerous strategies for optimizing the high-dimensional expected output value of the QAOA circuit. The choice of optimizer depends, also, on whether one optimizes a noise-free, simulated version of QAOA, or whether one optimizes a noisy estimation of the objective function. In general, there are two choices one can make in picking an optimizer for QAOA. The first, the use of global optimizers, tries to explore the full search space, with, of course, the associated long runtime. Attempting to globally optimize a QAOA instance may, in some cases, itself, be an **NP-Hard** problem [65], so the use of well-motivated ansatz states and strategies is important for QAOA to form a viable approach.

Early algorithms discovered for extremizing continuous functions include the Newton-Raphson method [66] in which the local gradient is calculated and updates made assuming the cost function is linear. The Newton-Raphson method requires that one calculates the inverse of the Hessian of the objective function (The matrix second derivative), which is expensive to compute, or impossible to compute exactly for an unknown black box objective function. Therefore, methods to approximate such quantities based on a limited number of observations of the objective function are necessary and form the basis for quasi-Newton methods. Work has been carried out investigating the best practice for the use of optimizers for NISQ algorithms [67, 68] and some of the best-known optimizers are presented here.

Nelder-Mead

Perhaps the most widely used method for solving an unconstrained optimization problem [69, 70]. This downhill simplex method uses a simplex—essentially a generalization of a triangle—of $d + 1$ points, where d is the dimension of the search space, in the QAOA case, $2p$. The algorithm progresses by evaluating the objective function for each point in the simplex and replacing the worst point with a new

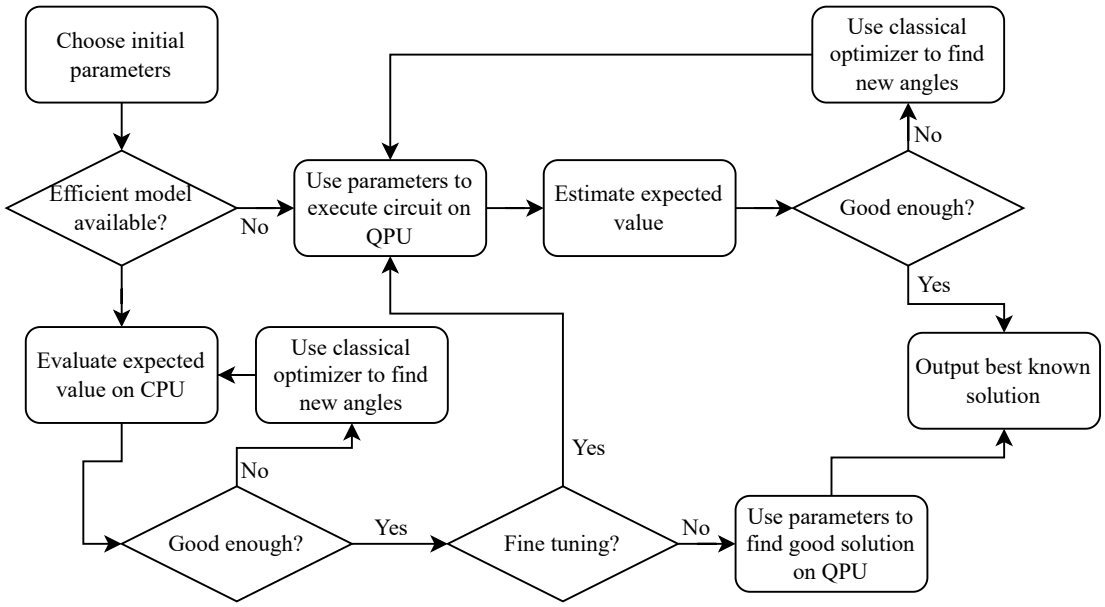

Figure 2.2: The optimization process followed in QAOA in the case where an efficient model of the expectation value is or is not available. Initial parameters are chosen, then the expected value thereof is calculated either with a CPU or via a quantum processing unit. A classical optimizer loop is followed until a sufficiently good solution is found. In the case where an efficient model is used, fine-tuning to account for instance-model differences can be performed on the QPU, starting from the CPU-pre-trained parameters.

point derived through reflection, expansion, or contraction along the direction of the center of mass of the other points.

The algorithm works well for low-dimensional optimization problems, such as QAOA with a small number of qubits or low circuit depth. It has historically enjoyed much use due to its lightweight properties—relatively little must be stored in memory and no gradients must be calculated—but fails to perform well on high-dimensional data, giving the algorithm poor performance on high-depth QAOA. On noisy data, as would be the case for QAOA in the presence of noise, the algorithm's performance further deteriorates. Where the derivative can be computed, other algorithms can outperform Nielder-Mead. For use in QAOA, one should use Nelder-Mead only in the case of $p = 1$ or other low-depth settings. For larger depths the method is inappropriate.

Cobyla

Constrained Optimization BY Linear Approximation (COBYLA) [71] is a derivative-free optimization algorithm that is used to solve constrained optimization problems. The algorithm works by approximating the objective function, and constraints, though in the case of QAOA there typically would not be constraints, as linear.

COBYLA operates via iterative refinement of the original problem using linear programming problems. In each iteration, a linear programming problem is solved which generates a candidate optimal solution. This solution can then be assessed

using the original objective function providing a new data point that can be used to improve the linear programming approximation for the next iteration. The method gradually reduces the step size and terminates when the step size becomes sufficiently small.

BFGS

The Broyden Fletcher, Goldfarb, Shanno—BFGS—method [72] is a quasi-Newton method in which the Hessian is approximated and iteratively updated at each step. For problems in which the number of variables is large, for example, high-depth QAOA, a limited memory version of BFGS is available, named L-BFGS [73]. This version is simply achieved by simply keeping track of a limited number of previous evaluation points. BFGS requires the objective function to be non-noisy to perform well.

Basin Hopping

A global optimizer, basin hopping [74] is a stochastic algorithm that searches the space of possible coordinates by applying cycles of random perturbations followed by the application of a local optimizer. The SciPy default minimizer for basin hopping is L-BFGS, though any local optimizer could be used in its place. Naturally, as a global optimizer, runtimes for this method become prohibitively large for relatively high-depth QAOA. Ultimately, one should have a good ansatz starting point for which a local optimizer can find a high-value QAOA state for QAOA to be a useful heuristic.

CMA-ES

An evolution strategy for optimization, Covariance Matrix Adaption Evolution Strategy (CMA-ES) [75] is a derivative-free, stochastic method well suited to noisy and high-dimensional objective functions. The strategy samples candidate solutions from a multi-dimensional normal distribution with a given mean and covariance matrix. Based on the quality of the generated candidate solutions, the algorithm then updates the mean and covariance matrix for the following round. Such a method would be useful in the context in which one performs QAOA with an open loop optimization resulting in a noisy objective function, though, the method is significantly more computationally intensive than others.

2.5.2 Start Point Strategies

The choice of starting location for the variational parameters in QAOA likely holds the greatest importance for the algorithm's usefulness. In some works, random initialization strategies are used, picking initial parameters at random and then repeating them for a number of trials. Anecdotally, it is observed that a strong initialization strategy is simply that of taking small, non-zero parameters. The use of parameters initialized at zero typically results in optimizer failure with this point being a zero gradient inflection point. A mixture of such strategies is also found to be successful, in which one randomly samples start points randomly, on

a range that only includes small values of the parameters. It is observed in Chapter 3 that for greater qubit numbers, objective function landscapes become less, not more, complex, with small-value initialization strategies working well in the infinite problem-size limit.

Given sufficiently a sufficiently high-depth QAOA circuit, one has a guarantee that the probability of finding the optimal state is extremized via the connection of QAOA to adiabatic quantum computing. Considering the general adiabatic quantum computing Hamiltonian given by

$$\hat{H}(t) = \left(1 - \frac{t}{T}\right)\hat{H}_{\mathrm{D}} + \frac{t}{T}\hat{H}_{\mathrm{P}} \tag{2.13}$$

applied to the ground state of $\hat{H}_{\mathrm{D}}$, where T is the total time over which the adiabatic algorithm occurs. Adiabatic quantum computing finds the ground state with probability 1 assuming that the time T is sufficiently large, with QAOA circuits being, for certain parameter regimes, a discretization, or Trotterization of this time evolution. An ansatz for QAOA parameters explored in work by Sack et al. [76], presumably especially useful at high depth, where QAOA serves as a good approximation for AQC would be:

$$\gamma_i = \frac{i}{p}\delta t,\ \beta_i = \frac{p-i}{p}\delta t. \tag{2.14}$$

For sufficiently large p, one would set $\delta t = 1$ for equivalence with adiabatic quantum computing. While adiabatic parameters may be one choice for optimal parameters for QAOA in the limit of large depth, for more realistic depths, such parameters may simply serve as a good starting point for an optimization process.

2.5.3 Barren Plateaus

Barren plateaus are a phenomenon found to be a general feature of variational quantum circuits. It was discovered that for parameterized random quantum circuits, typically used in quantum machine learning tasks, the magnitude of the gradient of the circuit at randomly selected points in the parameter space, tends to become exponentially small as the number of parameters becomes large [77]. Another consequence of this is that, since the probability that the gradient becomes exponentially small at a given point becomes large, and choice of start point becomes important for large parameterized random circuits. Work has linked the phenomenon of barren plateaus, also, to the expressibility of the variational ansatz used [78]. That is, the closer an ansatz comes to forming a 2-design[1], which is to say that the ansatz is highly expressive, the lower the variance in the cost function gradient. It is also expected that the presence of noise, depolarizing or otherwise[2] [79].

The discovery of barren plateaus as a generic feature of variational quantum algorithms becomes less relevant in the context of QAOA. While the algorithm is variational, several factors reduce the impact of such features. Firstly, start-point ansatz choices are known for QAOA. The link between adiabatic quantum computing

[1]Roughly, to be a 2-design is to be such that randomly sampling from the uniformly distributed parameters in the ansatz results in an equivalent distribution as that of sampling from a uniform distribution over unitaries.

[2]QAOA circuits should be sufficiently *twirling* that any biased noise should result in identical outcomes to depolarizing noise, given the circuit is sufficiently deep.

and QAOA gives rise to an obvious choice of this, from which a local optimizer can proceed [76]. For high-depth QAOA in which the number of variational parameters can be sufficiently large that the ansatz state is sufficiently expressive to suffer from vanishing gradients, assuming that parameters vary smoothly from layer to layer, with the schedule optimized via a Fourier coefficients can collapse the total number of parameters while maintaining performance [80]. The main reason, however, that QAOA might not suffer from barren plateaus as consequently as other variational quantum algorithms, is the relationship between the QAOA variational form and the objective function one extremizes. In the quantum machine learning context, there is no relationship between these two elements, whereas, in QAOA, the two are intimately related, with the problem Hamiltonian taking its values directly from the objective function by which the quality of the QAOA state is evaluated. For this reason, the logic of barren plateaus does not necessarily apply in this context.

Chapter 3

Problem-Size-Independent Angles for a Grover-Driven Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm (QAOA) requires that circuit parameters are determined that allow one to sample from high-quality solutions to combinatorial optimization problems. Such parameters can be obtained using either costly outer-loop optimization procedures and repeated calls to a quantum computer or, alternatively, via analytical means. In this work, we consider a context in which one knows a probability density function describing how the objective function of a combinatorial optimization problem is distributed. We show that, if one knows this distribution, then the expected value of strings, sampled by measuring a Grover-driven, QAOA-prepared state, can be calculated independently of the size of the problem in question. By optimizing this quantity, optimal circuit parameters for average-case problems can be obtained on a classical computer. Such calculations can help deliver insights into the performance of and predictability of angles in QAOA in the limit of large problem sizes, in particular, for the number partitioning problem.

3.1 Attribution & Copyright

This work was carried out by myself, with proofreading and editorial input from Prof. Wilhelm-Mauch. Some useful conversations with Dima Bagrets were important to this work, in which the suggestion of the number partitioning problem was given. Parts of this chapter are reprinted (excerpt, abstract, figures) with permission from "David Headley and Frank K. Wilhelm, Phys. Rev. A 107, 012412", Copyright (2023) by the American Physical Society.

3.2 Introduction

Variational Quantum algorithms aim to exploit the power of Noisy, Intermediate-Scale Quantum (NISQ) computers [81] through the use of parameterized quantum

circuits. The Quantum Approximate Optimization Algorithm (QAOA) [34] provides a universal [60, 82] ansatz state, efficiently preparable on a quantum computer, measurements from which might provide a useful heuristic method for the approximate solution of problems in combinatorial optimization.

QAOA circuits utilizing standard—single-qubit-X—drivers applied to Grover-like marked-state problems, recover the asymptotic scaling of Grover's algorithm [83]. Furthermore, it has been shown in the same unstructured search problem context, that parameters in QAOA concentrate [84]. That is, for increasing problem size, the variance in the optimal angles for QAOA circuits tends to zero and optimal angles concentrate at terminal values [85]. When considering QAOA in its original incarnation of single-qubit-X drivers and problem Hamiltonians encoding combinatorial optimization problems such as MAX-CUT, MAX-3-LIN-2, MAX-k-SAT, numerical studies have provided evidence that parameters also tend to concentrate, and, as such, can be determined for classes of similar problems without access to a quantum processing unit using classically tractable tensor network approaches [86].

That optimal angles in QAOA concentrate, for certain classes of problem, is useful due to the exponential cost of using classical optimizers in the number of parameters over which they operate. Should QAOA provide a good heuristic method for a number of layers greater than ≈ 10, it is likely that naive global classical optimization with repeated calls to the quantum computer to estimate and extremize the expectation value of the problem Hamiltonian is not the method used, due to the excessively large number of calls to the quantum computer required. Alternative methods to calculate optimal angles might use numerical methods on smaller or simplified problem sizes, as aforementioned, or use the averaged behavior of problems at a limiting size. Using this second approach, Farhi et. al. demonstrated that for Sherrington-Kirkpatrick type spin glasses and similar models, using single-qubit-X drivers, one can average over model instances to find optimal angles in a calculation that is independent of the number of spins in a model [87]. This provides a formula that has been numerically optimized to a depth of $p = 10$, a result later generalized [88] to show that the same technique extends to problems involving clauses incorporating more than two variables, providing an expression numerically tractable up to a depth $p = 20$ with the specific hardware used. This work was later widened to the context of MAX-CUT problems on varying random graph families [89], with Claes et. al. considering QAOA on infinite-size mixed-spin SK models [90] and showing that the expected performance of QAOA also concentrates in this context, at depth 1.

This chapter takes an alternative approach to finding angles in the large problem-size limit. By using a Grover-style driver that bestows maximal permutation symmetry to the expectation value of a problem Hamiltonian under the QAOA state produced, we simplify the calculation of optimal angles at large n. This procedure requires that the probability density function (pdf), or density of states of the problem to be solved is known. In particular, we provide a method to find optimal angles and their corresponding expectation values on a classical computer, such that high-quality solutions can be sampled from a quantum computer without the need for outer-loop optimization of the on-device QAOA state. We find that this method can be applied to the Number Partitioning Problem (NPP) and other problems where the probability density function is known or can be approximated, and, also, problems with a distribution of costs taking the form of or well approximated by a

Gaussian distribution.

The Ansätze used in variational quantum algorithms, in general, must have problem specificity to avoid the fate of barren plateaus [77, 91]. In QAOA, this problem specificity enters into the ansatz twice. Firstly, the problem Hamiltonian takes its eigenvalues directly from the objective function of the problem to be solved. Secondly, and more subtly, the driver Hamiltonian biases certain transitions between computational basis states over others. The use of Grover drivers removes the second of these and allows one to quantify where the performance in QAOA lies, between that of algorithms that use amplitude amplification-like structure agnostic speedups to solve problems, to those which derive performance from classically exploitable problem structure with respect to the single-qubit basis.

Other works have used Grover drivers in the context of QAOA. It has been shown that combinatorial optimization problems can be approximately solved using Grover's algorithm via the compilation of their objective function to a threshold function. Grover's algorithm is asymptotically optimal for the Grover problem [92], with the performance of the threshold-based algorithm numerically observed to outperform the non-threshold version, however, incurring additional compilation overhead [93, 94, 95]. Though one might not *a priori* know which threshold to use for such a compiled version, knowledge of the pdf of the problem, as used in this work would certainly be sufficient knowledge for the selection thereof.

While it is not expected that Grover-driven QAOA should provide superior performance to that of well-suited problem-specific drivers, it is expected that Grover-driven QAOA will not suffer from the issue of exponentially diminishing gradients—barren plateaus—as the angles can be obtained independently of problem size on a classical computer, provided the problem's pdf is known. As such, Grover-driven QAOA provides a point of reference to the performance of a QAOA-type algorithm in the very average case of driver-problem synergy.

3.3 Theory

QAOA, as previously introduced, is performed via p applications of driver and problem Hamiltonians applied for parameterized times $\vec{\beta}, \vec{\gamma}$. Giving:

$$\left|\vec{\beta}, \vec{\gamma}\right\rangle = \prod_{p'=1}^{p} \mathrm{e}^{i\beta_{p'}\hat{H}_{\mathrm{D}}} \mathrm{e}^{i\gamma_{p'}\hat{H}_{\mathrm{P}}} \left|+\right\rangle^{\otimes n}, \tag{3.1}$$

which is the problem-dependent QAOA ansatz state, depending on $2p$ free parameters. One is then required to find angles such that when sampling from the computational basis measurement outcome distribution of the ansatz state, one finds configurations corresponding to small or large eigenvalues of $\hat{H}_{\mathrm{P}}$ depending on whether a problem is one of maximization or minimization. To this end, the expectation:

$$E_p(\vec{\beta}, \vec{\gamma}) = \langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\beta},\vec{\gamma}} = \left\langle \vec{\beta}, \vec{\gamma} \right| \hat{H}_{\mathrm{P}} \left| \vec{\beta}, \vec{\gamma} \right\rangle, \tag{3.2}$$

should be extremized with respect to the variational parameters. Measurements of the optimized ansatz state then correspond to high-quality solutions to a binary-variable combinatorial optimization problem specifying $\hat{H}_{\mathrm{P}}$. This work concerns the calculation of this property in the case that Grover drivers are used, as introduced

in the following. A challenge in evaluating the performance of QAOA on a classical computer is the exponential resource requirements to simulate an n-qubit state, a dependence this work removes.

3.3.1 Properties of Grover Drivers

The driver Hamiltonian $\hat{H}_{\mathrm{D}}$ typically consists of single-qubit X operations,

$$\hat{H}_{\mathrm{D}} = \sum_{i=1}^{n} X_i. \tag{3.3}$$

Various works have investigated the use of drivers that differ from this standard single-qubit driver. For example, a version of QAOA that samples from a differing selection of higher order drivers for each layer has been shown to improve QAOA convergence [96, 97]. Drivers can also be used to shift complexity from problem to driver Hamiltonian in the case of problems of hard and soft constraints [98].

In this work, an alternate driver, which generates the Grover mixing operator, is used. The Grover driver can be written as a projector onto a Hadamard basis product state $|+\rangle^{\otimes n}$:

$$\hat{H}_{\mathrm{G}} = |+\rangle^{\otimes n} \langle +|^{\otimes n} = \left[\frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} |z\rangle\right] \left[\frac{1}{\sqrt{2^n}} \sum_{z=0}^{2^n-1} \langle z|\right]. \tag{3.4}$$

Importantly, this Hamiltonian is invariant under the re-labeling of any pair of states, i.e., under the permutation operator $\hat{U}_{j\leftrightarrow k}$ with

$$\hat{U}_{j\leftrightarrow k} = \mathbb{I} - |j\rangle \langle j| - |k\rangle \langle k| + |k\rangle \langle j| + |k\rangle \langle j| \tag{3.5}$$

and so it holds that

$$\hat{H}_{\mathrm{G}} = \hat{U}_{j\leftrightarrow k} \hat{H}_{\mathrm{G}} \hat{U}_{j\leftrightarrow k} \tag{3.6}$$

for any permutation. Moreover, the initial state of QAOA bears the same property, as an eigenstate of any permutation operator one can write:

$$|+\rangle^{\otimes n} = \hat{U}_{j\leftrightarrow k} |+\rangle^{\otimes n}. \tag{3.7}$$

Of course, a problem Hamiltonian is, in general, not invariant under any permutation. However, alongside a Grover-QAOA state (defined as in equation (3.1) using with $H_D = H_G$) its expectation is, since permutation operators may commute through the driver to annihilate or act as the identity on the initial state. So, for any permutation of a problem Hamiltonian $\hat{H}_{\mathrm{P}}$, given by $\hat{P}^\dagger \hat{H}_{\mathrm{P}} \hat{P}$ where

$$\hat{P} = \hat{U}_{j_m\leftrightarrow k_m} \cdots \hat{U}_{j_2\leftrightarrow k_2} \hat{U}_{j_1\leftrightarrow k_1} \tag{3.8}$$

is a sequence of permutations, the QAOA problem expectation produced by such a permuted problem Hamiltonian can be shown to be equivalent to that of the unpermuted Hamiltonian,

$$\langle \hat{P}^\dagger \hat{H}_{\mathrm{P}} \hat{P} \rangle_{\vec{\beta},\vec{\gamma}} = \langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\beta},\vec{\gamma}}, \tag{3.9}$$

as shown in Appendix 3.A. The consequence of this symmetry is that any two problems which have the same distribution of costs will have identical expectation values under Grover-driven QAOA states.

3.3.2 Problem Objective Functions as Random Variables

The only property of the problem Hamiltonian that therefore affects the expectation is its spectrum. The spectrum of a problem Hamiltonian can be seen as sampling from some random distribution with probability density function $f(c)$—describing the relative likelihood that the random variable C, modeling a problem objective function, takes real value c—analogous to the density of states function of solid state physics. Our key result is that as this pdf can be, in general, independent of the size of a problem, angles can be calculated directly from the pdf, rather than for any model instance with a given size. By computing angles for the large-n limit of Grover-driven QAOA, one can determine the asymptotic performance of the algorithm for large system sizes without suffering from unfavorable scaling in the number of qubits. The computed angles are for "typical" instances. However, as the problem size increases, the law of large numbers rapidly reduces the sample-to-sample fluctuations.

Consider a problem's objective function that has been sorted from low to high, denoted by $C_{\text{sort}}(z)$, such that

$$C_{\text{sort}}(x) \geq C_{\text{sort}}(y) \text{ if } x > y. \tag{3.10}$$

To this quantity, we can associate a quantile distribution function (qdf)

$$F^{-1}(p) = C_{\text{sort}}(z) \text{ for } p = \frac{z}{N} \tag{3.11}$$

denoting the value of C at its pth quantile. The inverse of the quantile distribution function is the cumulative distribution function (cdf) $F(c)$ with an associated pdf $f(c)$ in the continuum limit of large n

$$F(c) = \mathrm{P}(C < c) = \sum_{c'=-\infty}^{c} \mathrm{P}(C = c') \approx \int_{-\infty}^{c} f(c')\,\mathrm{d}c'. \tag{3.12}$$

The quantity on which the Grover-driven QAOA expectation value depends is the Fourier transform of this pdf, known as the characteristic function Γ which for a random variable X is defined as the expected value of $\mathrm{e}^{i\gamma X}$ and so for the random variable C modeling an ensemble of objective functions for a given problem:

$$\Gamma(\gamma) = \mathrm{E}\left[\mathrm{e}^{i\gamma C}\right]. \tag{3.13}$$

This quantity, associated with the random variable C can be expressed in terms of the pdf $f(c)$, the cdf $F(c)$ and the qdf $F^{-1}(p)$ as:

$$\Gamma(\gamma) = \int_{-\infty}^{\infty} f(c)\mathrm{e}^{i\gamma c}\,\mathrm{d}c = \int_{-\infty}^{\infty} \mathrm{e}^{i\gamma c}\,\mathrm{d}F(c) = \int_{0}^{1} \mathrm{e}^{i\gamma F^{-1}(p)}\mathrm{d}p \tag{3.14}$$

and is related via a Wick rotation and scaling factor of the system size to the partition function of the model.

A finite-size, n-qubit problem considered for QAOA is modeled here as taking $2^n = N$ samples from the ensemble random variable C, and as such the approximation made in this work is that the mean exponentiated objective function value can be replaced by the ensemble characteristic function Γ. That is, that

$$\Gamma(\gamma) \approx \sum_{z} \frac{\mathrm{e}^{i\gamma C(z)}}{N} \tag{3.15}$$

with equality in the infinite-N limit. Due to the exponential growth of N with the number of qubits, such an approximation rapidly improves with increasing system sizes. As such, it suffices to replace the finite sum in equation (3.15) by the characteristic function of the continuous pdf—from which the problem is modeled to sample from—in calculations.

3.4 Depth 1

In this section, we wish to obtain an expression for the expectation value of the depth $p = 1$ Grover-QAOA problem Hamiltonian $E_1(\gamma, \beta)$ in which the problem is expressed only via a characteristic function Γ. We denote the mean value of a random variable C as

$$\bar{C} = \int_{-\infty}^{\infty} cf(c)\,\mathrm{d}c \overset{n\to\infty}{=} \frac{1}{N}\sum_z C(z) = -i\Gamma'(0) \tag{3.16}$$

which can in general be set to zero by adding a constant to the objective function. Such a shift corresponds only to a global phase of the effected problem Hamiltonian unitary $\mathrm{e}^{-i\bar{C}\gamma}\mathbb{I}$. The expression derived, in full detail in Appendix 3.B consists of factors for which Equation 3.15 can immediately be substituted, with, however, one instance of $C(z)$ as a multiplicative factor rather than in an exponent. To substitute this factor, one may differentiate under the sum/integral in the characteristic function as:

$$\frac{1}{N}\sum_z C(z)\mathrm{e}^{i\gamma C(z)} = -\frac{i}{N}\frac{\mathrm{d}}{\mathrm{d}\gamma}\sum_z C(z)\mathrm{e}^{-i\gamma C(z)} = -i\frac{\mathrm{d}}{\mathrm{d}\gamma'}\Gamma(\gamma')\bigg|_\gamma = -i\Gamma'(\gamma). \tag{3.17}$$

Substituting the characteristic function Γ and its derivative above, one obtains the expression

$$E_1(\gamma,\beta) = \bar{C}(1 + BB^*\Gamma\Gamma^*) + 2\,\mathrm{Im}(B^*\Gamma^*\Gamma') \tag{3.18}$$

in which the first term can be set to zero by zeroing the mean value of the problem.

3.4.1 Invariance Under Shifts

Equation 3.18 implies explicit dependence upon the mean value of the distribution of a problem. We can however exploit some properties of Fourier transforms to show that besides a constant term, the general expression depends only trivially upon the absolute position of the distribution modeling the problem Hamiltonian.

For a cost function modeling random variable C, there is a pdf $f(c)$ and associated characteristic function

$$\Gamma(\gamma) = \mathcal{F}[f(c)](\gamma) = \int_{-\infty}^{\infty} f(c)\mathrm{e}^{i\gamma c}\ \mathrm{d}c. \tag{3.19}$$

One can use the shift property of Fourier transforms to determine that

$$\mathcal{F}[f(c-\bar{c})](\gamma) = \mathrm{e}^{i\bar{c}\gamma}\mathcal{F}[f(c)](\gamma) = \mathrm{e}^{i\gamma\bar{c}}\Gamma_{\text{Central}}(\gamma), \tag{3.20}$$

where Γ_{Central} is the characteristic function of the centered random variable with mean zero. Expressing the $p = 1$ expression in terms of this quantity, with

$$\Gamma \to \Gamma_{\text{Central}}\mathrm{e}^{i\gamma\bar{c}} \tag{3.21}$$

one finds the first term unchanged. For the primed term, one obtains the expression

$$\Gamma'(\gamma) = \frac{\partial}{\partial\gamma}\left[\Gamma_{\text{Central}}(\gamma)\mathrm{e}^{i\gamma\bar{c}}\right] = \Gamma'(\gamma)\mathrm{e}^{i\bar{c}\gamma} + i\bar{c}\Gamma(\gamma)\mathrm{e}^{i\bar{c}\gamma}. \tag{3.22}$$

Substituting these values one finds the expression

$$\begin{aligned}\langle\hat{H}_{\mathrm{P}}\rangle_{\gamma,\beta} &= \bar{c}(1 + BB^*\Gamma\Gamma^*) + 2\,\mathrm{Im}(B^*\Gamma^*\mathrm{e}^{-i\gamma\bar{c}}\left(\Gamma'(\gamma)\mathrm{e}^{i\bar{c}\gamma} + i\bar{c}\Gamma(\gamma)\mathrm{e}^{i\bar{c}\gamma}\right)) \\ &= \bar{c}(1 + BB^*\Gamma\Gamma^*) + 2\,\mathrm{Im}\left(B^*\Gamma^*\Gamma'\right) + 2\,\mathrm{Im}(i\bar{c}B^*\Gamma^*\Gamma) \\ &= \bar{c} + \bar{c}\Gamma\Gamma^*(BB^* - 2\,\mathrm{Re}(B)) + 2\,\mathrm{Im}(B^*\Gamma^*\Gamma') \\ &= \bar{c} + 2\,\mathrm{Im}(B^*\Gamma^*\Gamma'). \end{aligned} \tag{3.23}$$

As the expanded middle terms evaluate to and cancel to zero since $BB^* = 2\,\mathrm{Re}(B) = 2-2\cos(\beta)$. Thus, the expectation value of the Grover-QAOA state at $p = 1$ depends on the mean value of the cost function trivially through a constant term.

3.5 Depth 2

The expectation value of the problem Hamiltonian for a $p = 2$ Grover-driven QAOA state depends on four parameters and can be expressed as:

$$E_2(\gamma_1, \gamma_2, \beta_1, \beta_2) = \langle\gamma_1, \gamma_2, \beta_1, \beta_2|\, \hat{H}_{\mathrm{P}}\, |\gamma_1, \gamma_2, \beta_1, \beta_2\rangle \tag{3.24}$$

which, if one assumes that the mean of the distribution is zero, can be expanded in 10 terms seen in Appendix 3.6 with the help of computer algebra [99]. Further simplification and substitution of the depth-1 expression yields:

$$\begin{aligned}E_2(\gamma_1, \gamma_2, \beta_1, \beta_2) = \\ &E_1(\gamma_1, \beta_1) + E_1(\gamma_1 + \gamma_2, \beta_2) + |B(\beta_1)|^2|\Gamma(\gamma_1)|^2 E_1(\gamma_2, \beta_2) \\ &\quad + 2\,\mathrm{Im}\{B(\beta_1)^* B(\beta_2)^* \Gamma(\gamma_1)^* \Gamma(\gamma_2)^* \Gamma'(\gamma_1 + \gamma_2)\} \\ &\quad\quad + 2\,\mathrm{Im}\{B(\beta_1)B(\beta_2)^* \Gamma(\gamma_1)\Gamma'(\gamma_2)\Gamma(\gamma_1 + \gamma_2)^*\} \end{aligned} \tag{3.25}$$

which can be numerically extremized on a classical computer to find optimal parameters given a characteristic function Γ.

3.6 Arbitrary Depth

For arbitrary depth p, we have the problem Hamiltonian expectation value of:

$$E_p(\vec{\gamma}, \vec{\beta}) = \langle +|\prod_{j=-1}^{-p} \hat{U}_{\mathrm{P}}(\gamma_j)\hat{U}_{\mathrm{D}}(\beta_j)\hat{H}_{\mathrm{P}} \prod_{i=1}^{p} \hat{U}_{\mathrm{D}}(\beta_i)\hat{U}_{\mathrm{P}}(\gamma_i)\,|+\rangle \tag{3.26}$$

where we have introduced the convention that negative indices simply add a negative sign to the value, as:

$$\gamma_{-i} = -\gamma_i,\ \ \beta_{-i} = -\beta_i. \tag{3.27}$$

This can be expanded and expressed as a formula summing over 2^{2p} terms expressed in factors of the characteristic function Γ and the driver-dependent function B. The

full derivation and explanation of the notation used for which can be found in Appendix 3.D with the expression for the arbitrary depth expected value of the problem Hamiltonian taking the form:

$$E_p(\vec{\gamma},\vec{\beta}) = 2\,\mathrm{Im}\left[\sum_{k_{\mathrm{bra}}<k_{\mathrm{ket}}=0}^{2^p-1}\prod_{P\in P_{\mathrm{bra}}}\Gamma\left(\sum_{i\in P}\gamma_i\right)\Gamma'\left(\sum_{i\in P_{\mathrm{central}}}\gamma_i\right)\prod_{P\in P_{\mathrm{ket}}}\Gamma\left(\sum_{i\in P}\gamma_i\right)\prod_{-j|k^j_{\mathrm{bra}}=1\text{ or }j|k^j_{\mathrm{ket}}=1}B_j\right]. \quad (3.28)$$

Which can be evaluated independently of n on a classical computer.

3.7 Problem Distributions

To find suitable problems for this algorithm, one requires problems for which the probability density function is known, but for which no efficient classical algorithms for finding states of extreme energy exist. The standard QAOA problems such MAX-CUT, k-SAT and SK models approximately follow binomial and Gaussian distributions, and, therefore could also be considered as Gaussian problems under Grover-driven QAOA, but, a demonstration that they do so analytically is less obvious than in the case of the problems we consider here due to the presence of frustration. The random cost model and number partitioning problem do not feature the same frustration, with the random variables by which they are defined entering into the problem via single-qubit terms, with the number partitioning problem made non-trivial via a global constraint enforcing positivity of the cost function. As such, the distributions for the latter problems are readily derivable.

3.7.1 Gaussian Problems & The Random Cost Model

The Random Cost Model (RCM) is a toy model used as a testing ground for techniques in the study of disordered systems as is the simplest model exhibiting a phase transition [100]. The model is not a description of any physical system, nor does it describe an interesting hard problem. However, it does provide an example for which our procedure generates a concise analytical expression at low depth. The RCM is defined as one in which each energy level or objective function value samples from a normal distribution, so

$$C \sim \mathrm{Normal}(\mu=0,\sigma^2=1),\ f(c) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{c^2}{2}}. \quad (3.29)$$

When using the Grover driver, a problem with this pdf can also be obtained via n independent spins with Hamiltonian

$$\sum_i g_i Z_i = \sum_z \sum_i (-1)^{z_i} g_i \left|z\right\rangle \left\langle z\right| \quad (3.30)$$

in which each Pauli spin weight is set as $g_i \sim \mathrm{Normal}(\mu=0,\sigma^2=1/n)$ such that the resulting angles and energy expectations are independent of problem size. This Hamiltonian is, of course, structured and for another choice of driver could not be

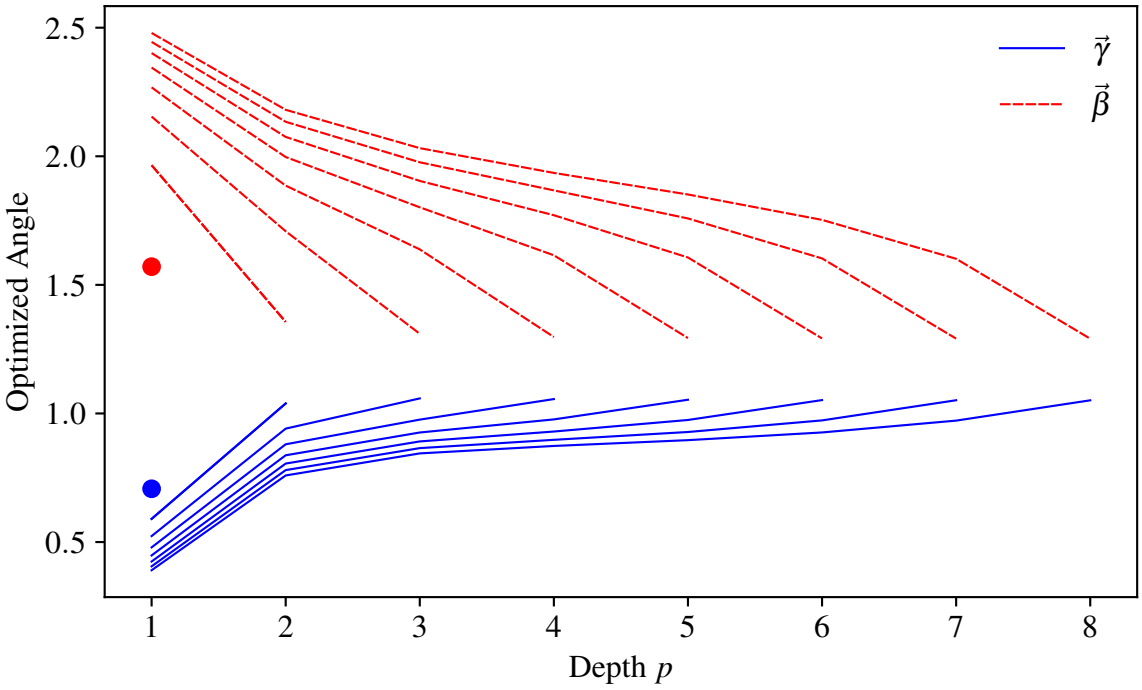

Figure 3.1: Angles for the optimized infinite-size RCM are seen for QAOA depths 1 to 8. As the depth p increases, a similar pattern of optimal parameters for the limiting distribution of the random cost model emerges with the problem-Hamiltonian angles monotonically increasing and the driver Hamiltonian angles monotonically decreasing. Such behavior invites comparison to adiabatic quantum computation, overlooking the non-zero start point of γ and end point of β.

usefully modeled as a Gaussian pdf alone, the Grover driver, however, does not see this additional structure. This problem yields the desired pdf as a normalized Gaussian with which is associated a characteristic function via Fourier transform with

$$\Gamma(\gamma) = \mathrm{e}^{-\frac{\gamma^2}{2}}. \tag{3.31}$$

When substituted into Equation 3.18, this characteristic function produces a $p = 1$ Grover-QAOA problem expectation value of

$$E_1(\gamma, \beta) = 2\gamma \mathrm{e}^{-\gamma^2} \sin(\beta) \tag{3.32}$$

for which maximization yields optimal angles of

$$\gamma = \frac{\sqrt{2}}{2}, \ \beta = \frac{\pi}{2}. \tag{3.33}$$

Angles for higher depth QAOA states can be found in Figure 3.1 and the associated expectation values in Figure 3.2. The best-known angles are determined via numerical optimization with BFGS [101] on a wide range of start points, in which the consistent behavior of the angles with increasing depth implies that with increasing depth one tends towards a continuous terminal schedule.

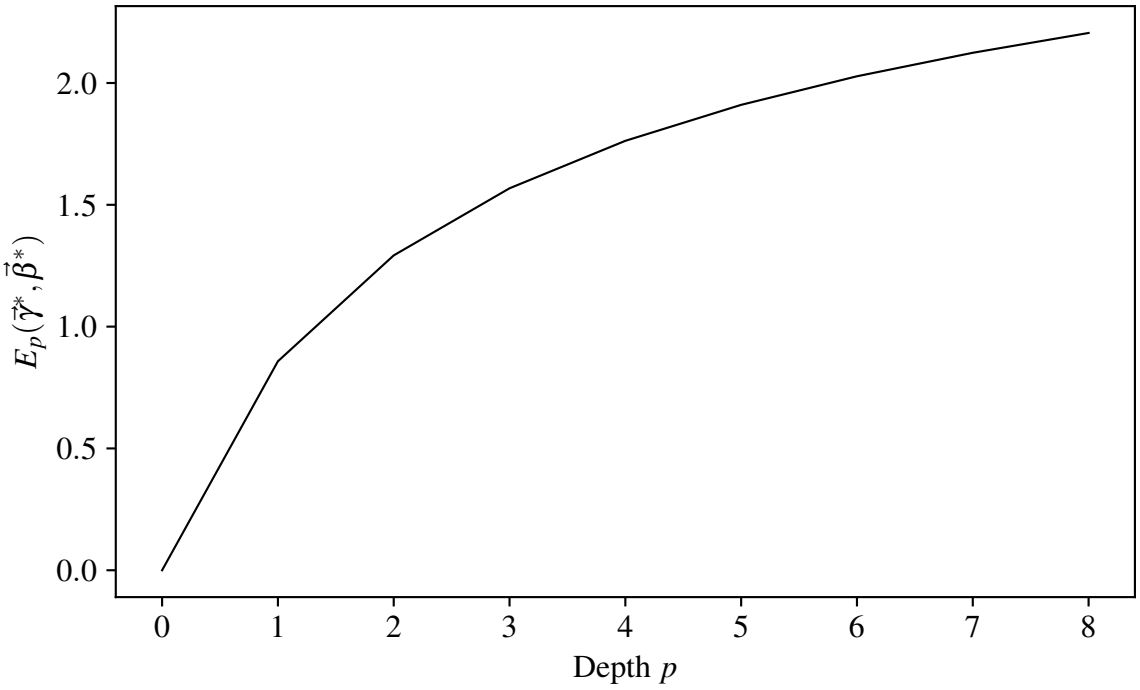

Figure 3.2: The expectation value of the Gaussian-distributed random cost problem at optimal angles exhibits an increasing expectation value at a decreasing rate.

3.7.2 The Number Partitioning Problem

The Number Partitioning Problem (NPP) is that of finding a partition, described by a number, $z \in \{0, \ldots, N-1\}$, of a set of n positive real numbers $\{x_0, \ldots, x_{n-1}\}$ indexed by the set $\{0, \ldots, n-1\}$ into two subsets with indices $\{i \mid z_i = 0\}$ and $\{i \mid z_i = 1\}$ such that the sum of the numbers indexed by each subset is equal. NPP has applications ranging from task scheduling in multi-core processing to choosing equally matched teams from a selection of players. It is an example of a problem for which classical heuristics such as simulated annealing do not perform well, notably, due to the similarity of the problem to totally unstructured cost problems [102]. The problem has seen study in the context of adiabatic quantum computing [103] due to the relative ease of computing-related properties in the large problem size limit, thus, allowing the calculation of the minimum energy gap and asymptotic time complexity in this context. In the large problem-size limit, adiabatic quantum computing returns a Grover-like quadratic speed up in run-time over random guessing for this problem.

One may define the decision version of the NPP as to satisfy the equality:

$$\sum_{i|z_i=0} x_i = \sum_{i|z_i=1} x_i. \tag{3.34}$$

Which, for an approximate minimization version of the problem, is characterized by an objective function:

$$C(z) = \left| \sum_{i|z_i=0} x_i - \sum_{i|z_i=1} x_i \right| = \left| \sum_i (-1)^{z_i} x_i \right|. \tag{3.35}$$

When considering this problem in the context of QAOA, or other quantum optimization contexts in which many-body interactions are nontrivial to implement, one can consider an equivalent minimization problem defined by the squared problem $C^2(z)$. To express this as an Ising spin glass Hamiltonian, one can first consider the signed single qubit problem Hamiltonian:

$$\sum_i x_i Z_i = \sum_z \left(\sum_i (-1)^{z_i} x_i \right) |z\rangle \langle z| , \tag{3.36}$$

of which the square results in the problem Hamiltonian:

$$\begin{aligned} \sum_i x_i Z_i \sum_j x_j Z_j &= \sum_i x_i^2 I + 2 \sum_{i<j} x_i x_j Z_i Z_j \\ &= \sum_{z,z'} \left(\sum_i (-1)^{z_i} x_i \right) |z\rangle \langle z| \left(\sum_i (-1)^{z'_i} x_i \right) |z'\rangle \langle z'| \\ &= \sum_z \sum_{i,j} (-1)^{z_i + z_j} x_i x_j |z\rangle \langle z| = \sum_z C^2(z) |z\rangle \langle z| , \end{aligned} \tag{3.37}$$

which is a problem Hamiltonian encoding for the squared residue of the number partitioning problem, as:

$$C^2(z) = \left| \sum_i (-1)^{z_i} x_i \right| \cdot \left| \sum_j (-1)^{z_j} x_j \right| = \sum_{i,j} (-1)^{z_i + z_j} x_i x_j . \tag{3.38}$$

Consequently, the NPP Hamiltonian is an all-to-all coupled Mattis-type Ising spin glass in which couplings consist of the products of random variables. Now we must determine the pdf $f(c)$ and the associated partition function Γ for this problem. For convenience, we consider the numbers to be partitioned to be sampled from distributions such that the resulting cost function distribution has a unit mean. As such the numbers x_i are allowed to take values between 0 and $x_{\max} = 1/\sqrt{3n}$ uniformly, i.e. $x_i \sim U(0, x_{\max}) \; \forall \; i \in [1 \dots n]$. The probability density function for the individual variables is then defined as:

$$f(c) = \frac{1}{x_{\max}} \left(\Theta(c) - \Theta(c - x_{\max}) \right) \tag{3.39}$$

where Θ is the Heaviside step function. As such, the pdf for the distribution of residues defined as C_{SQ}, prior to squaring the cost function, is distributed as:

$$C_{\mathrm{SQ}} \sim \sum_i (-1)^{z_i} x_i = \sum_i (-1)^{z_i} U(0, x_{\max}) = \sum_i U(-x_{\max}, x_{\max}), \tag{3.40}$$

where the factor $(-1)^{z_i}$ imparts a minus sign with probability $\frac{1}{2}$ and results in a sum over uniform distributions over the symmetric interval $[-x_{\max}, x_{\max}]$. This sum of independent, identically distributed random variables, by the central limit theorem, approaches a Gaussian distribution in the large n limit. So, in this large n limit

$$\begin{aligned} C_{\mathrm{SQ}} \sim \mathrm{Normal} \left(\sum_i \mathrm{E}(U(-x_{\max}, x_{\max})), \sum_i \mathrm{Var}(U(-x_{\max}, x_{\max})) \right) \\ = \mathrm{Normal} \left(\mu = 0, \sigma^2 = 1 \right) . \end{aligned} \tag{3.41}$$

For the squared single-qubit cost function $C^2_{\mathrm{SQ}}(z) = C^2(z)$, one can use the standard result that the square of a Gaussian distributed random variable is the χ^2 distribution with a single degree of freedom with pdf:

$$f(c) = \begin{cases} (2\pi c \mathrm{e}^c)^{-\frac{1}{2}}, & c > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3.42}$$

The characteristic function of this pdf can readily be calculated as

$$\Gamma(\gamma) = \sqrt{\frac{1}{1 - 2i\gamma}}. \tag{3.43}$$

For $p = 1$, one can numerically optimize the expectation value $E_1(\gamma, \beta)$ to find minimizing angles of

$$\gamma^* = 0.241, \ \beta^* = 5.162 \tag{3.44}$$

to three decimal places. These angles can be seen in Figure 3.3 in which the value of β has been shifted by 2π for convenience. Such angles yield an expected value of $E_1(\vec{\gamma}^*, \vec{\beta}^*) = 0.557$. The expectation value of the problem Hamiltonian for this characteristic function can be minimized to obtain angles that are thought to be optimal. The best-known angles can be seen in Figure 3.3. The associated qualities of the solutions attained by these angles are seen in Figure 3.5. To demonstrate that these angles are appropriate for finite-size instances of the number partitioning problem, we plot in Figures 3.3 and 3.4 the average result when optimizing finite-size instances at increasing size for depth 1, 2 respectively. This demonstrates that with increasing problem size, the optimal angles for finite-size instances rapidly approach those of the analytically derived, ensemble average optimal angles. Finally, Figure 3.7 demonstrates the convergence of the landscape defined by the problem Hamiltonian expectation value of the QAOA state at a given point in the QAOA parameter space. As the number of qubits in the problem becomes larger, the QAOA solution quality landscape becomes indistinguishable from the analytically derived version.

3.8 Compiling the Grover Driver

Compilation of the Grover mixing operator can be attained with or without ancillary qubits. Without ancillary qubits, the operation can be performed using an $n-1$-controlled-Z rotation. Such a gate can be compiled into two n-qubit Toffoli gates alongside three two-qubit gates [104]. These Toffoli gates can in turn be compiled using a number of two-qubit gates and depth quadratic and linear, respectively, in the number of qubits [105, 106].

3.9 Connection to Grover's Algorithm

The Grover search oracle problem Hamiltonian searching for string z^* takes the form

$$\hat{H}_{\mathrm{Grover}} = |z^*\rangle \langle z^*| . \tag{3.45}$$

As a random variable, this corresponds to sampling a state with objective function value zero with probability $\frac{N-1}{N}$ and a solution with quality 1 with probability $\frac{1}{N}$.

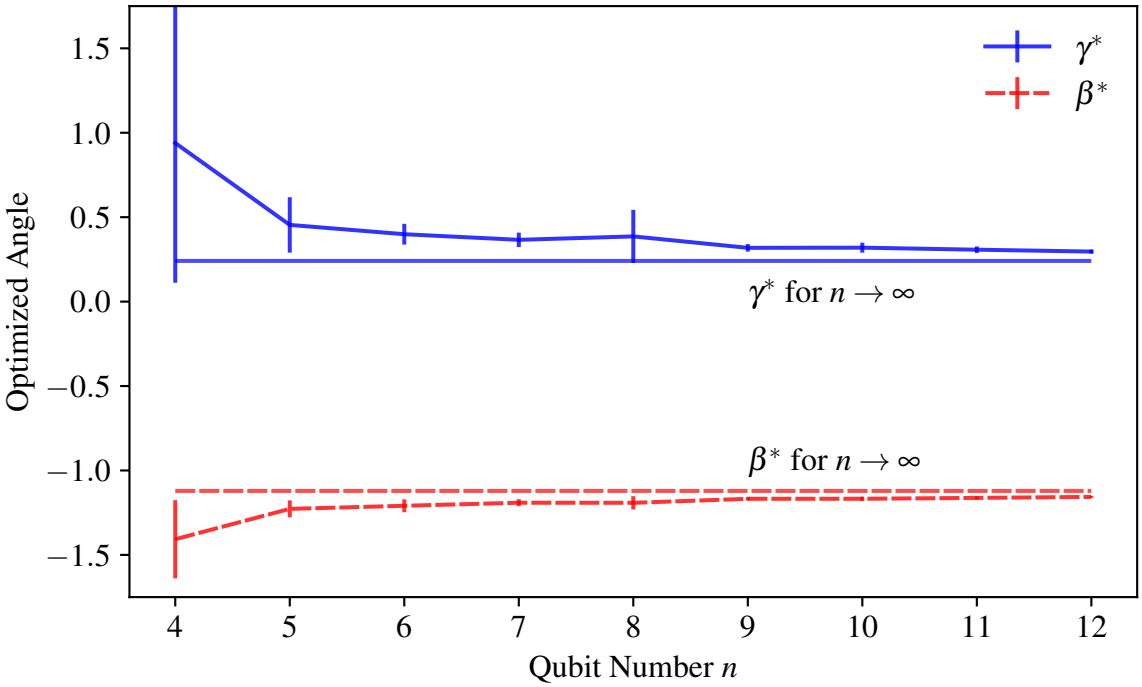

Figure 3.3: For $p = 1$, optimal parameters of the NPP converge to the infinite size angle for increasing n. QAOA states for 30 problem instances are optimized and angles averaged for each point.

The associated pdf, or probability mass function, given this is a discrete distribution is:

$$f(c) = \frac{N-1}{N}\delta(c) + \frac{1}{N}\delta(c-1). \tag{3.46}$$

The characteristic function of this random variable would then be:

$$\Gamma_{\text{Grover}}(\gamma) = \int_{-\infty}^{\infty} \mathrm{d}c\ \mathrm{e}^{i\gamma c}\left[\frac{N-1}{N}\delta(c) + \frac{1}{N}\delta(c-1)\right] = \frac{N-1}{N} + \frac{1}{N}\mathrm{e}^{i\gamma}, \tag{3.47}$$

which could be utilized in the formula derived for Grover-QAOA. The probability mass function, here, is dependent on the number of qubits. In a way that is not typical of problems studied in this chapter.

3.10 Performance And Distribution Properties

In this section, we consider the performance of Grover-QAOA as it relates to the shape of the distribution on which it acts. First one should note that the scale of the distribution is inconsequential, simply corresponding to a re-scaling of the problem parameter γ. That is, for some distribution $f(c)$, one can re-scale this by a real parameter a to obtain a pdf given by $af(ac)$. If γ maximizes performance on the former, then $\frac{\gamma}{a}$ maximizes performance on the latter.

The usual metric for QAOA performance is that of the approximation ratio, given by

$$r = \frac{E_p(\vec{\gamma}, \vec{\beta})}{C(z^*)} \tag{3.48}$$

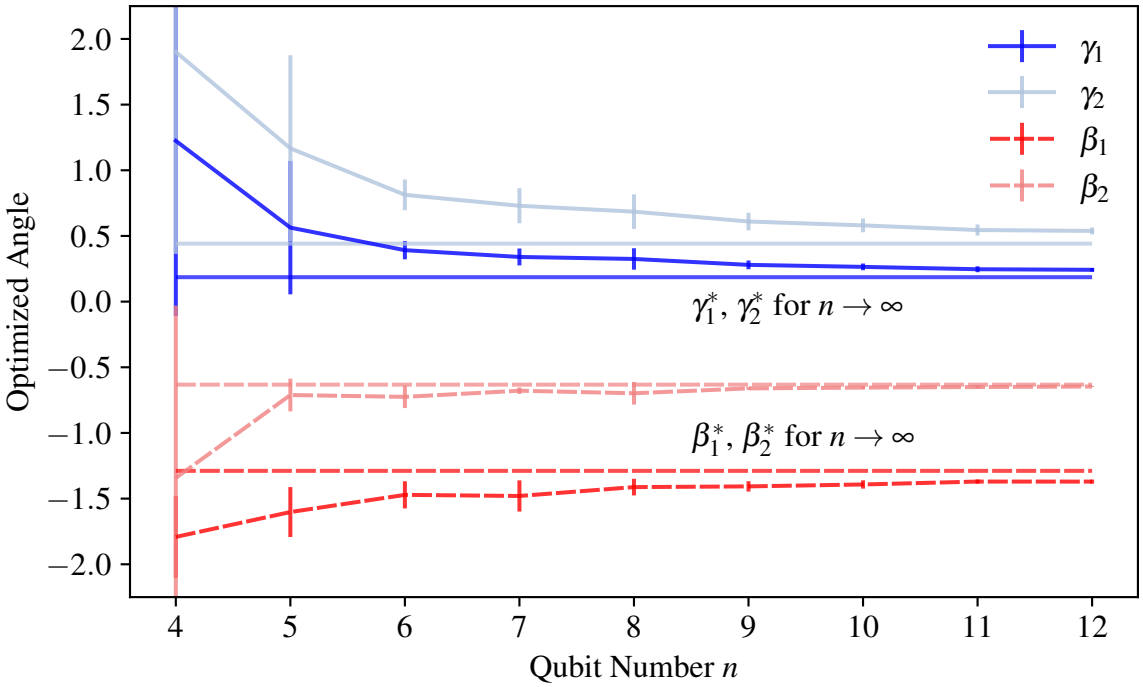

Figure 3.4: For $p = 2$, optimal parameters of the NPP converge to the infinite size angle for increasing n. QAOA states for 30 problem instances are optimized and angles averaged for each point.

where $C(z^*)$ is the optimal attainable cost function value. This metric measures the sum of the values of clauses that are satisfied when sampling from a given QAOA state, divided by the number satisfied by the optimal configuration. While this metric is good for comparing the performance of different algorithms on the same class of problems, it is not useful for comparing different problems. For example, earlier in this chapter, the number partitioning problem was considered alongside a random cost problem. A random cost problem could produce arbitrarily large negative costs, but the number partitioning problem can only ever have positive values. As such, methods for more meaningful comparisons are needed. In the context of this work, for example, one might have cost functions that are non-zero on their entire domain (a Gaussian, for example) and therefore for which there exists no maximal cost function value in the large-n limit.

A better metric, therefore, is the quantile of the mean solution. To obtain this quantity, one can take the QAOA solution quality $E_p(\vec{\gamma}, \vec{\beta})$ and can compute the quantile with which this average quality is associated, using the cumulative distribution function of the problem in question $F(c)$. By using this metric, it becomes possible to compare the performance across problems described by different probability distributions. This measure of success tells you how good your solution would be compared to the whole spectrum of possible solutions, rather than just comparing to the optimal value.

As a test of dependence on the performance of Grover QAOA on the distribution of the problem it solves, we compare the performance of three distributions. The first, is a rectangular function in which each value of a cost function on a symmetric range is equally likely to be selected, a triangular cost function, and, finally, the

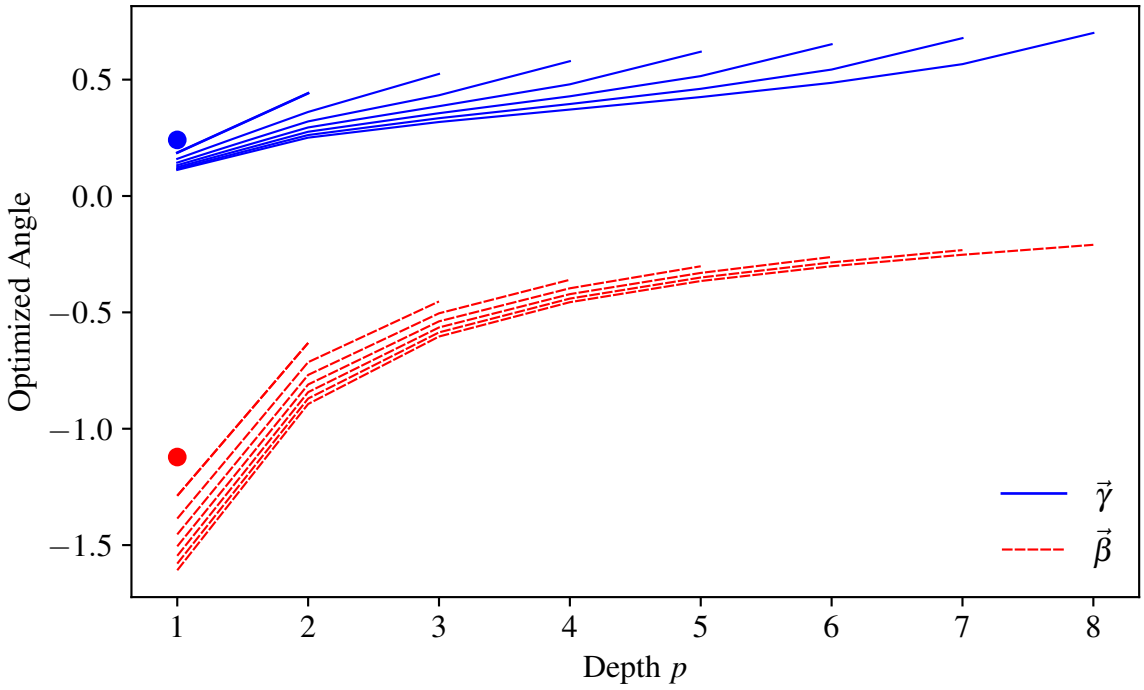

Figure 3.5: Angles for the optimized infinite-size NPP model are seen for QAOA depths 1 to 8. Optimal parameters follow annealing-reminiscent schedules and vary smoothly. Values are obtained here through the optimization of the Formula 3.28 via the BFGS optimizer and a random start point strategy.

Gaussian distribution as treated previously, as the random cost model. These three distributions are displayed in Figure 3.8.

The uniform distribution is one in which values are picked at random in some range, with identical likelihood, that is, it is distributed with pdf:

$$f_{\text{Uniform}}(c) = \begin{cases} 1/2 \text{ if } -1 < c < 1, \\ 0, \text{ otherwise.} \end{cases} \tag{3.49}$$

The triangular distribution is related to the uniform distribution as it can be obtained via the sum of two identical, independently distributed uniform distributions. The pdf for this random variable takes the form:

$$f_{\text{Triangle}}(c) = \begin{cases} \frac{c-2}{4} \text{ if } -2 < c < 0, \\ \frac{2-c}{4} \text{ if } 0 < c < 2, \\ 0 \text{ otherwise.} \end{cases} \tag{3.50}$$

To link these two distributions to the Gaussian distribution, we can use the central limit theorem, which states that for some sum of independent, identically distributed random variables, as their number tends to infinity, the resulting distribution tends to be that of a Gaussian distribution, assuming the moments of the summed distributions are well defined.

When comparing the performance of the Grover problem to differently shaped problems in Figure 3.9 (top), a value of $N = 64$ is used. In this context, a fair comparison to the other distributions would be the mean solution percentile being

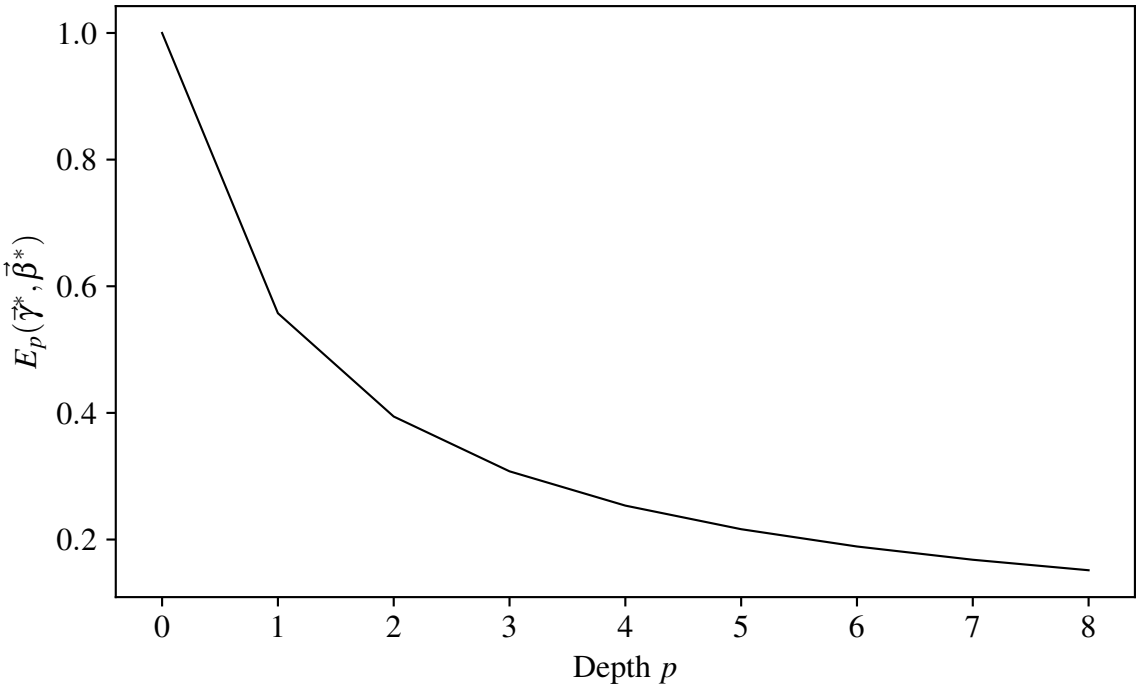

Figure 3.6: Minimizing the NPP for increasing depth p sees monotonically decreasing expectation. The depth 0 value of 1 is added as a reference point attained by random sampling from the computational basis.

greater than or equal to $\frac{63}{64}$. As can be seen, the Gaussian pdf returns the highest value solutions of the distributions tried. In Figure 3.9 (bottom), the solution quantile, rather than simply the expected value is used as a metric for success. One sees that the Gaussian problem still performs best out of the continuous distributions, with the square pdf performing worst. The Grover problem for $n = 6$ outperforms the Gaussian pdf, requiring depth 6 to attain a percentile of solution for which the Gaussian pdf requires a depth in excess of 7 for a comparable percentile.

3.11 Conclusion

The power of QAOA as a heuristic for solving problems in combinatorial optimization is not yet certain and derives from some combination of Grover-like speedup alongside performance attributable to the synergy of the problem and driver used. In this work, we demonstrate a method to determine likely optimal QAOA angles in the case of a driver with average-case synergy, in which the problem structure cannot be exploited. For such problem-driver combinations, concentrations of optimal angles occur for increasing problem sizes and tend to a limit calculable via the pdf of the problem. It is expected that barren plateaus, limiting the ability of an optimizer to find good angles for QAOA, will affect problem-driver combinations in which synergy is not present (for example, problems with high-order terms and single-qubit drivers). The problem of barren plateaus—in essence the no free lunch theorem of quantum optimization [107]—necessitates that one finds heuristics for which drivers and angles to use for a given problem instance. For which, the performance of a driver in comparison to the Grover version is instructive as to whether a strategy

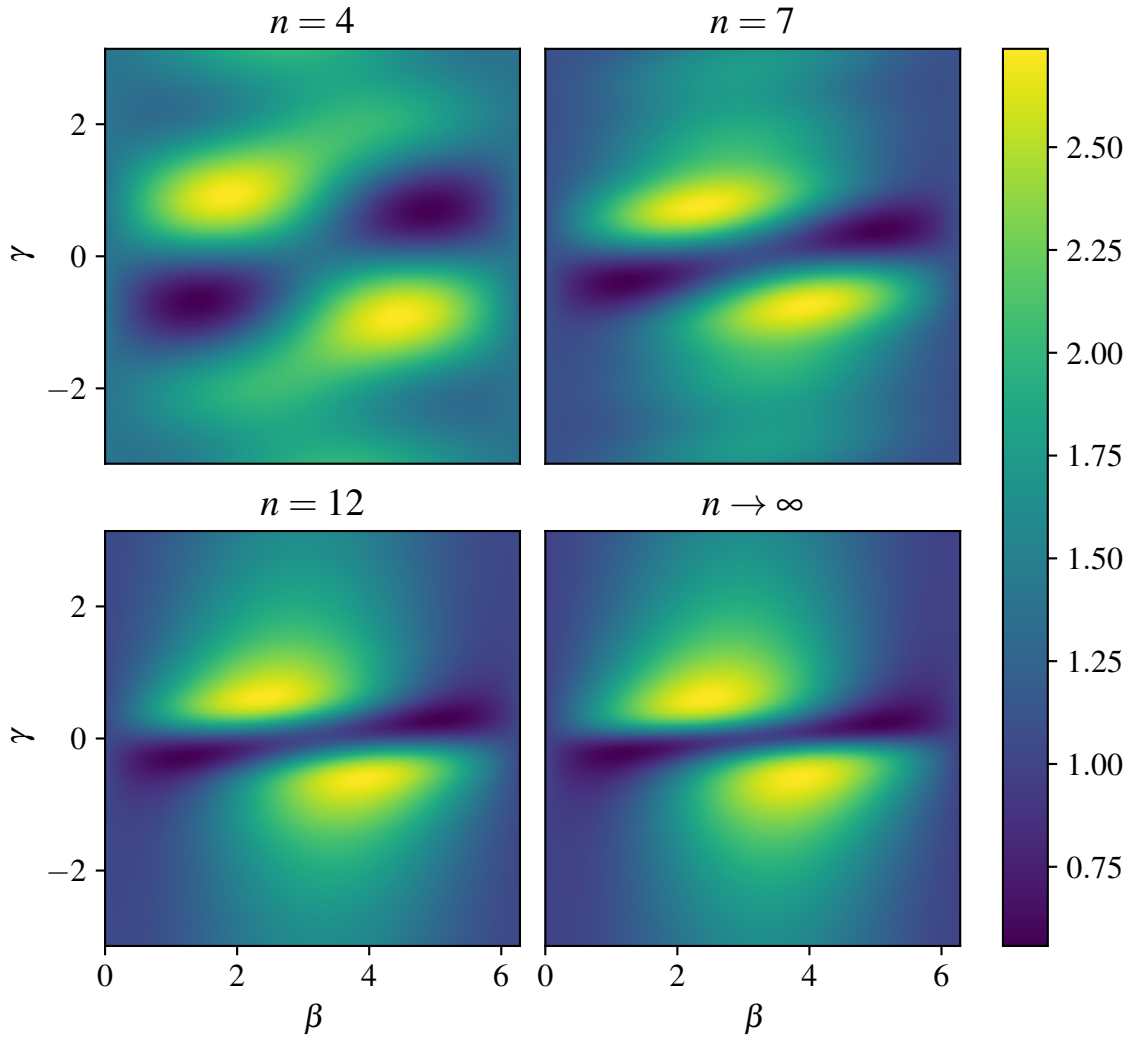

Figure 3.7: For $p = 1$ the landscape can be seen to converge to that of the large-n limit for $n = 4, 7, 12$. Periodicity is always observed in β, but as n increases, for low n, local maxima and minima resurge in γ due to common multiples in problem energy eigenvalues. For large n, increasing γ past a region near zero results in the expectation monotonically returning to the mean value of 1.

exploits problem structure and to what degree. The strategy of optimizing QAOA angles for statistically defined ensembles of problems, rather than instances thereof, comes with the added advantage of smoother QAOA objective function landscapes with fewer local extrema, being the average over instance landscapes, as evidenced by Figure 3.7.

This work originated from asking the question as to whether a single-qubit-X driver should outperform the Grover driver as an ansatz for the infinite range models in the work of Farhi et al. [87]. While this work has not directly compared the performance of Grover QAOA on the SK model, a future direction could compare this, via the application of methods applied to infinite-range SK models, to the number partitioning problem, or by finding or approximating an expression for

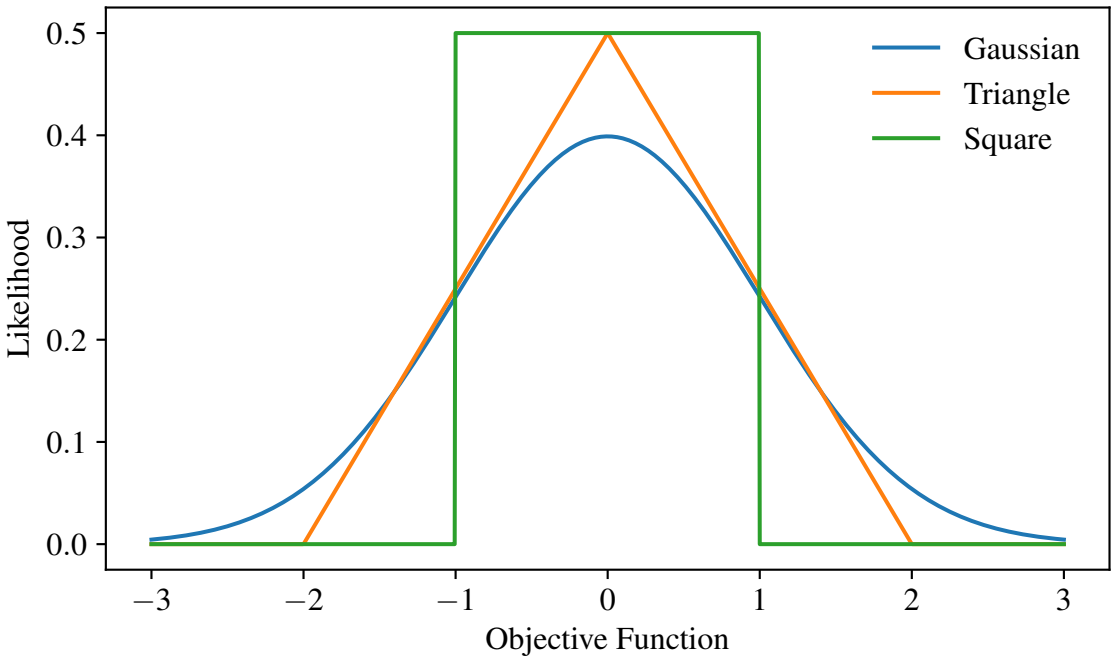

Figure 3.8: Three probability density functions are compared for performance under the Grover driver. As the mean and variance of the distributions do not affect the Grover-QAOA performance, simply scaling the problem variable γ, the performance depends on higher order moments, which differ for the distributions considered.

the limiting-size pdf of an SK-type model. Future work could also investigate the performance of subspace-type drivers e.g. XY type in problems of hard and soft constraints of limiting size [98], in which the form of the drivers on the valid-solution subspace differs from that of Grover drivers or standard single-qubit mixers. Work could also determine the dependence of QAOA performance on the shape of the distribution of the unstructured random variable on which it acts, as has been considered in the context of non-adiabatic annealing [108]. Various works have provided analytical results for the use of Grover-type problems or mixing operators to solve search problems in the context of QAOA, quantum random walks, and adiabatic quantum computing [109, 58, 110, 84], with this work demonstrating similar analytical procedures, can be applied to find optimal angles for Grover-driven QAOA with non-oracle problems. The work in this paper could also be applied to other contexts, such as that of quantum random walks, in which fully connected graphs have been previously considered for numerical studies on finding ground states of spin glasses [111].

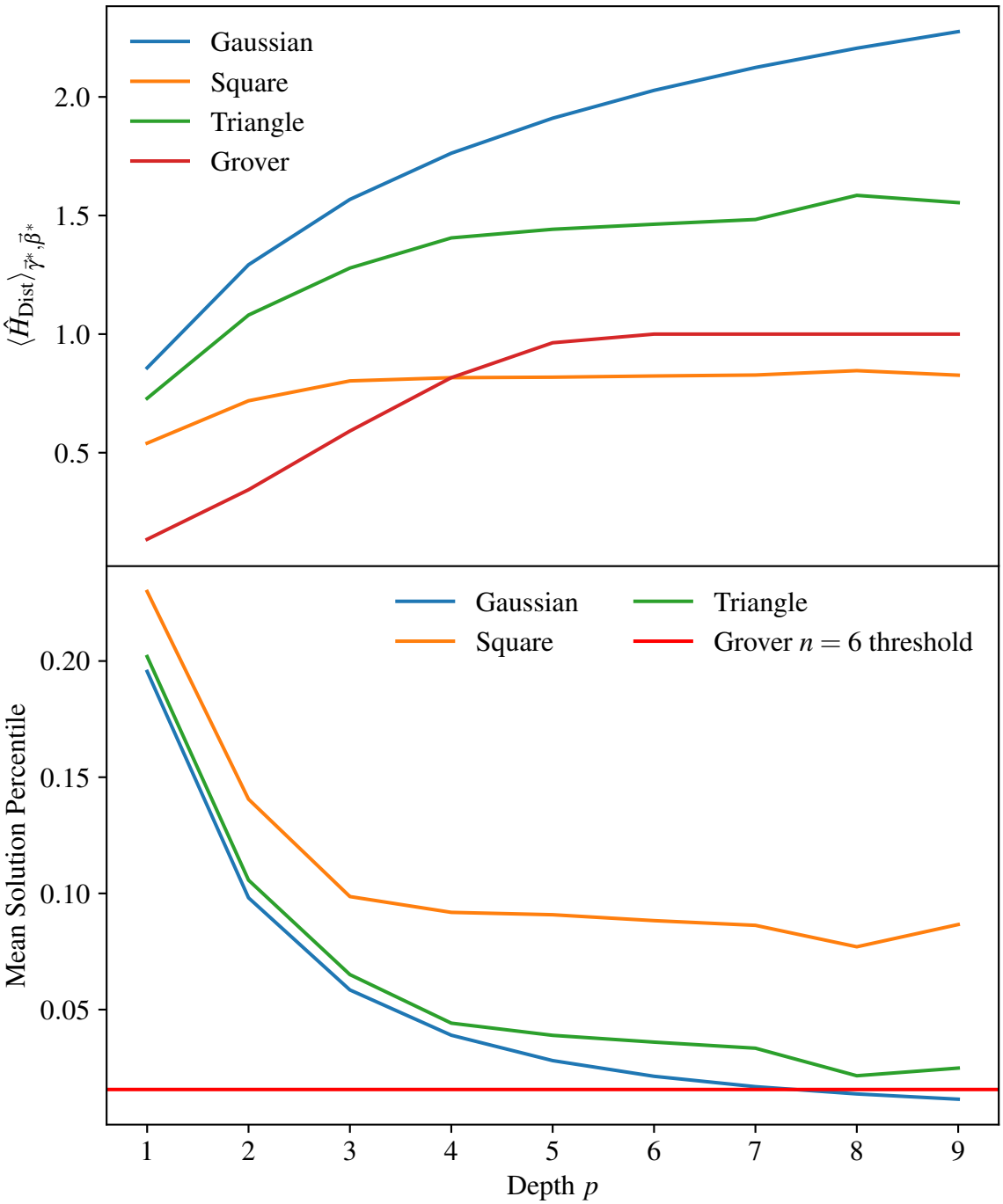

Figure 3.9: The expected solution quality (top) of the Grover-driven QAOA state is greatest for the Gaussian-distributed variable, this is, however, an unfair comparison since the greatest attainable solution quality for the square distribution considered is 1. The Gaussian problem has no limit to the solution quality that can be theoretically attained at an infinite size. When considering the performance of Grover-QAOA for differently distributed problems, the Gaussian problem still outperforms when using the metric that allows fair comparisons, the percentile of the mean solution (bottom). While the Grover problem pdf saturates at around $p = 5$ for the choice of $n = 6$, the Gaussian pdf requires a depth of around 8 to reach a comparable percentile of solution quality.

Appendix

3.A Permutation Symmetry of Problem Hamiltonians Under Grover-QAOA States

The invariance of a QAOA problem Hamiltonian under the permutation operator

$$\hat{P} = \hat{U}_{j_m \leftrightarrow k_m} \dots \hat{U}_{j_2 \leftrightarrow k_2} \hat{U}_{j_1 \leftrightarrow k_1} \tag{3.51}$$

can be demonstrated using the identities

$$\hat{P}\hat{P}^\dagger = \hat{I},\ \mathrm{e}^{\hat{P}^\dagger \hat{O} \hat{P}} = \hat{P}^\dagger \mathrm{e}^{\hat{O}} \hat{P}. \tag{3.52}$$

Therefore:

$$\begin{aligned}
\langle \hat{P}^\dagger \hat{H}_\mathrm{P} \hat{P} \rangle_{\vec{\beta},\vec{\gamma}} &= \langle + |^{\otimes n} \prod_{p'=1}^{p} \left[\mathrm{e}^{i\gamma_{p'} \hat{P}^\dagger \hat{H}_\mathrm{P} \hat{P}} \mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \right] \hat{P}^\dagger \hat{H}_\mathrm{P} \hat{P} \prod_{p'=1}^{p} \left[\mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \mathrm{e}^{i\gamma_{p'} \hat{P}^\dagger \hat{H}_\mathrm{P} \hat{P}} \right] |+\rangle^{\otimes n} \\
&= \langle + |^{\otimes n} \prod_{p'=1}^{p} \left[\hat{P}^\dagger \mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \hat{P} \mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \right] P^\dagger \hat{H}_\mathrm{P} \hat{P} \prod_{p'=1}^{p} \left[\mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \hat{P}^\dagger \mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \hat{P} \right] |+\rangle^{\otimes n} \\
&= \langle + |^{\otimes n} \hat{P}^\dagger \prod_{p'=1}^{p} \left[\mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \hat{P} \mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \hat{P}^\dagger \right] \hat{H}_\mathrm{P} \prod_{p'=1}^{p} \left[\hat{P} \mathrm{e}^{i\beta_{p'} \hat{H}_\mathrm{D}} \hat{P}^\dagger \mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \right] \hat{P} |+\rangle^{\otimes n} \\
&= \langle + |^{\otimes n} \hat{P}^\dagger \prod_{p'=1}^{p} \left[\mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \mathrm{e}^{i\beta_{p'} \hat{P} \hat{H}_\mathrm{D} \hat{P}^\dagger} \right] \hat{H}_\mathrm{P} \prod_{p'=1}^{p} \left[\mathrm{e}^{i\beta_{p'} \hat{P} \hat{H}_\mathrm{D} \hat{P}^\dagger} \mathrm{e}^{i\gamma_{p'} \hat{H}_\mathrm{P}} \right] \hat{P} |+\rangle^{\otimes n} \\
&= \langle \hat{H}_\mathrm{P} \rangle_{\vec{\beta},\vec{\gamma}}
\end{aligned} \tag{3.53}$$

demonstrating that the expectation value is invariant under any permutation.

3.B Calculation of Depth-1 Expectation Value

We can write down the unitaries for the problem Hamiltonian:

$$\hat{U}_\mathrm{P}(\gamma) = \sum_z \mathrm{e}^{i\gamma C(z)} |z\rangle \langle z| \tag{3.54}$$

and driver Hamiltonian:

$$\begin{aligned}
\hat{U}_\mathrm{D}(\beta) = \mathrm{e}^{i\beta |+\rangle\langle +|} &= \mathbb{I} + i\beta |+\rangle \langle +| + \frac{(i\beta)^2}{2} |+\rangle \langle +| + \dots \\
&= \mathbb{I} + (\mathrm{e}^{i\beta} - 1) |+\rangle \langle +| = \mathbb{I} + B(\beta) |+\rangle \langle +|
\end{aligned} \tag{3.55}$$

with $B(\beta) = \mathrm{e}^{i\beta} - 1$. For $p = 1$ the problem expectation can be written as:

$$\begin{aligned}\langle \hat{H}_\mathrm{P} \rangle_{\gamma,\beta} &= \langle + | \hat{U}_\mathrm{P}^\dagger(\gamma) \hat{U}_\mathrm{D}^\dagger(\beta) \hat{H}_\mathrm{P} \hat{U}_\mathrm{D}(\beta) \hat{U}_\mathrm{P}(\gamma) | + \rangle \\ &= \sum_{z_{-1},z_0,z_1} \langle + | \, \mathrm{e}^{-i\gamma C(z_{-1})} \, | z_{-1} \rangle \langle z_{-1} | \left[\mathcal{I} + B^* | + \rangle \langle + | \right] C(z_0) \\ &\qquad \cdot | z_0 \rangle \langle z_0 | \left[\mathbb{I} + B | + \rangle \langle + | \right] \mathrm{e}^{i\gamma C(z_1)} \, | z_1 \rangle \langle z_1 | + \rangle \\ &= \frac{1}{N} \sum_{z_{-1},z_0,z_1} \mathrm{e}^{-i\gamma C(z_{-1})} \langle z_{-1} | \left[\mathbb{I} + B^* | + \rangle \langle + | \right] \\ &\qquad\qquad C(z_0) | z_0 \rangle \langle z_0 | \left[\mathbb{I} + B | + \rangle \langle + | \right] \mathrm{e}^{i\gamma C(z_1)} \, | z_1 \rangle . \end{aligned} \tag{3.56}$$

Which yields four terms as follows:

$$\begin{aligned}\bar{C} + \frac{1}{N^2} BB^* \bar{C} \sum_z \mathrm{e}^{i\gamma C(z)} \sum_z \mathrm{e}^{-i\gamma C(z)} + \frac{1}{N^2} B \sum_z \mathrm{e}^{i\gamma C(z)} \sum_z C(z) \mathrm{e}^{-i\gamma C(z)} \\ + \frac{1}{N^2} B^* \sum_z \mathrm{e}^{-i\gamma C(z)} \sum_z C(z) \mathrm{e}^{i\gamma C(z)}. \end{aligned} \tag{3.57}$$

One then can substitute the characteristic function of the ensemble to obtain an expression for the expectation value of the QAOA state, with

$$\frac{1}{N} \sum_z C(z) \to \bar{C}, \; \sum_z \frac{\mathrm{e}^{i\gamma C(z)}}{N} \to \Gamma(\gamma). \tag{3.58}$$

To express the term containing $C(z)$ not in the exponent, one may differentiate under the sum or integral as

$$\sum_z C(z) \mathrm{e}^{i\gamma C(z)} = -i \frac{\mathrm{d}}{\mathrm{d}\gamma} \sum_z C(z) \mathrm{e}^{-i\gamma C(z)} \to -i \frac{\mathrm{d}}{\mathrm{d}\gamma'} \Gamma(\gamma') \Big|_\gamma = -i\Gamma'(\gamma). \tag{3.59}$$

Resulting in the expression:

$$\langle \hat{H}_\mathrm{P} \rangle_{\gamma,\beta} = \bar{C}(1 + BB^*\Gamma\Gamma^*) + 2 \operatorname{Im}(B^*\Gamma^*\Gamma'). \tag{3.60}$$

3.C Calculation of Depth-2 Expectation Value

The $p = 2$ expression can be calculated similarly to the $p = 1$ expression, however, with more terms and thus requiring the use of a computer algebra package [99]. The result of expanding the expression for the expectation value, setting the mean $\bar{C}$ to zero, and substituting the characteristic function and its derivative is the following expression of 10 terms:

$$\begin{aligned} E_2(\vec{\gamma}, \vec{\beta}) = {} & iB(\beta_1)B(\beta_2)\Gamma(\gamma_1)\Gamma(\gamma_2)B(\beta_1)^*\Gamma(\gamma_1)^*\Gamma'(\gamma_2)^* \\ & - iB(\beta_1)\Gamma(\gamma_1)B(\beta_1)^*B(\beta_2)^*\Gamma(\gamma_1)^*\Gamma(\gamma_2)^*\Gamma'(\gamma_2) \\ & - iB(\beta_1)^*\Gamma(\gamma_1)^*\Gamma'(\gamma_1) + iB(\beta_1)\Gamma(\gamma_1)\Gamma'(\gamma_1)^* \\ & + iB(\beta_1)B(\beta_2)\Gamma(\gamma_1)\Gamma(\gamma_2)\Gamma'(\gamma_1+\gamma_2)^* - iB(\beta_1)^*B(\beta_2)^*\Gamma(\gamma_1)^*\Gamma(\gamma_2)^*\Gamma'(\gamma_1+\gamma_2) \\ & - iB(\beta_1)\Gamma(\gamma_1)B(\beta_2)^*\Gamma'(\gamma_2)\Gamma(\gamma_1+\gamma_2)^* + iB(\beta_2)B(\beta_1)^*\Gamma(\gamma_1)^*\Gamma'(\gamma_2)^*\Gamma(\gamma_1+\gamma_2) \\ & - iB(\beta_2)^*\Gamma(\gamma_1+\gamma_2)^*\Gamma'(\gamma_1+\gamma_2) + iB(\beta_2)\Gamma(\gamma_1+\gamma_2)\Gamma'(\gamma_1+\gamma_2)^*. \end{aligned} \tag{3.61}$$

This expression can be further simplified via the substitution of the $p = 1$ expectation resulting in the final expression in Equation 3.25.

3.D Calculation of Arbitrary-Depth Expectation Value

Using the intuition from the $p = 1$ and $p = 2$ cases, we can write down the expression for any term in the arbitrary depth expectation value of the problem Hamiltonian. We can write the full expectation $E_p(\vec{\gamma}, \vec{\beta})$ as:

$$\left\langle \vec{\gamma}, \vec{\beta} \right| \hat{H}_{\mathrm{P}} \left| \vec{\gamma}, \vec{\beta} \right\rangle = \langle + | \prod_{j=-p}^{-1} \hat{U}_{\mathrm{P}}(\gamma_j) \hat{U}_{\mathrm{D}}(\beta_j) \hat{H}_{\mathrm{P}} \prod_{i=1}^{p} \hat{U}_{\mathrm{D}}(\beta_i) \hat{U}_{\mathrm{P}}(\gamma_i) | + \rangle \tag{3.62}$$

where we have introduced the convention that negative indices simply add a negative sign to the value, as:

$$\gamma_{-i} = -\gamma_i, \ \beta_{-i} = -\beta_i. \tag{3.63}$$

The expectation value can be expanded, substituting the expressions for the driver and problem Hamiltonians as:

$$\sum_{\vec{z}} \langle + | \prod_{j=-1}^{-p} \left[\mathrm{e}^{i\gamma_j C(z_j)} | z_j \rangle \langle z_j | \left(I + B_j | + \rangle \langle + | \right) \right] C(z_0) | z_0 \rangle \langle z_0 | \cdot \prod_{i=1}^{p} \left[\left(I + B_i | + \rangle \langle + | \right) \mathrm{e}^{i\gamma_i C(z_i)} | z_i \rangle \langle z_i | \right] | + \rangle . \tag{3.64}$$

Where the sum over $2p{+}1$ indices $\vec{z} = (z_{-p}, \ldots, z_0, \ldots, z_p)$ runs over the set $\{0, N{-}1\}$ for each index. Expanding the products in this expression will yield 2^{2p} terms that can be sorted in the number of occurrences of a B variable in the bra or ket.

For the one term with zero B incidences, one finds, for example:

$$\frac{1}{N} \sum_{\vec{z}} \mathrm{e}^{\sum_{j=-1}^{-p} i\gamma_j C(z_j)} C(z_0) \mathrm{e}^{\sum_{l}^{p} i\gamma_l C(z_l)} \cdot \langle z_{-1} | z_{-2} \rangle \langle z_{-2} | \ldots | z_{-p} \rangle \langle z_{-p} | z_0 \rangle \langle z_0 | z_p \rangle \langle z_p | \ldots | z_2 \rangle \langle z_2 | z_1 \rangle \tag{3.65}$$

in which the orthogonal bra-kets evaluate to zero leaving only one index yielding a non-zero term evaluating to the mean of the objective function $\bar{C}$. For weight-1 terms one has B in a position l so:

$$\sum_{\vec{z}} \langle + | \prod_{j=-1}^{-p} \left[\mathrm{e}^{-i\gamma C(z_j)} | z_j \rangle \langle z_j | \right] C(z) | z_0 \rangle \langle z_0 | \prod_{i=l}^{p} \left[| + \rangle \langle + | \, \mathrm{e}^{i\gamma C(z_i)} | z_i \rangle \langle z_i | \right] B_l \cdot \prod_{i=1}^{l-1} \left[| + \rangle \langle + | \, \mathrm{e}^{i\gamma C(z_i)} | z_i \rangle \langle z_i | \right] | + \rangle , \tag{3.66}$$

for which the orthogonal bra-kets can be evaluated as:

$$\frac{1}{N^2} \sum_{\vec{z}} \mathrm{e}^{\sum_{j=-1}^{-p} i\gamma_j C(z_j)} C(z_0) \mathrm{e}^{\sum_{i}^{p} -i\gamma_i C(z_i)} B_k \delta_{z_{-1}, z_{-2}} \ldots \delta_{z_{-p}, z_0} \delta_{z_0, z_p} \ldots \delta_{z_{k+2}, z_{k+1}} \delta_{z_k, z_{k-1}} \ldots \delta_{z_2, z_1}. \tag{3.67}$$

The sum is eliminated for all but two partitions giving:

$$\frac{1}{N^2}\sum_z \mathrm{e}^{\sum_{j=-1}^{-p} i\gamma_j C(z)} C(z) \mathrm{e}^{\sum_{i=k+1}^{p} i\gamma_i C(z)} \sum_z \mathrm{e}^{\sum_{i=1}^{k} i\gamma_i C(z)} B_k, \tag{3.68}$$

leaving two variables over which we sum and leaving the non-exponentiated $C(z)$ and, therefore, the differentiated characteristic function Γ' in the factor corresponding to the partition containing the central variable z_0 as

$$-iB_k\Gamma'\left(\sum_{i=-1}^{-p}\gamma_i + \sum_{i=k+1}^{p}\gamma_i\right)\Gamma\left(\sum_{i=1}^{k}\gamma_i\right). \tag{3.69}$$

To expand the expression more generally, we can introduce indices

$$k_{\text{bra}} \in \{0,\ldots,2^p-1\} \text{ and } k_{\text{ket}} \in \{0,\ldots,2^p-1\}, \tag{3.70}$$

each of which are p-bit numbers such that the presence of a 1 in the binary representation of the index implies a factor of $B_i\left|+\right\rangle\left\langle+\right|$ at the bit position in the bra or ket for the relevant term. The formula can be rewritten as

$$\begin{aligned}\sum_{k_{\text{bra}},\,k_{\text{ket}}=0}^{2^p-1}\sum_{\vec{z}}\left\langle+\right|\prod_{j=-p}^{-1}\left[\mathrm{e}^{i\gamma_j C(z_j)}\left|z_j\right\rangle\left\langle z_j\right|\left(B_j\left|+\right\rangle\left\langle+\right|\right)^{k_{\text{bra}}^j}\right] \\ \cdot C(z_0)\left|z_0\right\rangle\left\langle z_0\right|\prod_{i=1}^{p}\left[\left(B_i\left|+\right\rangle\left\langle+\right|\right)^{k_{\text{ket}}^i}\mathrm{e}^{i\gamma_i C(z_i)}\left|z_i\right\rangle\left\langle z_i\right|\right]\left|+\right\rangle.\end{aligned} \tag{3.71}$$

When the index $k_{\text{ket/bra}}$ is zero, this expression $[B_i\left|+\right\rangle\left\langle+\right|]^{k_{\text{bra/ket}}}$ is the identity and adjacent non-identical z indices result in a Kronecker delta, with $\langle z_j|z_k\rangle = \delta_{z_j,z_k}$ collapsing a sum over one of the z indices. As such, these terms are collected into a single sum. When a bit in the index $k_{ket/bra}$ is one, the projector $\left|+\right\rangle\left\langle+\right|$ annihilates with a basis state vector $\langle z_i|$ and simply produces a factor of $1/N$, with $\langle+|z\rangle = 1/\sqrt{N}\ \forall z$, leaving the sums over z intact.

To substitute the characteristic function, we note that unbroken chains of 0's of length n^* in the indices $k_{ket/bra}$ will result in factors of the form:

$$\begin{aligned}&\sum_{z_1,z_2\ldots z_{n^*}}\mathrm{e}^{i\gamma_1 C(z_1)}\left|z_1\right\rangle\left\langle z_1\right|\mathrm{e}^{i\gamma_2 C(z_2)}\left|z_2\right\rangle\left\langle z_2\right|\ldots\mathrm{e}^{i\gamma_{n^*} C(z_{n^*})}\left|z_{n^*}\right\rangle\left\langle z_{n^*}\right| \\ &\quad=\sum_{z_1,z_2\ldots z_{n^*}}\delta_{z_1,z_2}\delta_{z_2,z_3}\ldots\delta_{z_{n^*-1},z_{n^*}}\mathrm{e}^{i\gamma_1 C(z_1)}\mathrm{e}^{i\gamma_2 C(z_2)}\ldots\mathrm{e}^{i\gamma_{n^*} C(z_{n^*})} \\ &\quad=\sum_z \mathrm{e}^{iC(z)\sum_{i\in[1,\ldots,n^*]}\gamma_i} = \Gamma\left(\sum_{i\in[1,\ldots,n^*]}\gamma_i\right).\end{aligned} \tag{3.72}$$

To substitute the characteristic function for all incidences of the objective function $C(z)$, it is convenient to define a set of boundary indices between those that are grouped as above. We can define these as

$$S_{\text{bra}} = \{0\} + \{i|k_{\text{bra}}^i = 1\},\ S_{\text{ket}} = \{0\} + \{i|k_{\text{ket}}^i = 1\} \tag{3.73}$$

determined by the locations of 1's in the binary representation of k_{bra}, k_{ket}. The additional boundary at 0 represents the boundary imposed by the initial state of

QAOA. In the partitions formed by these boundaries, we collect the indices contained in the ith partition of the bra/ket as P^i_{bra}, P^i_{ket} with

$$P^i_{\text{bra}} = \{-j | j > k^i_{\text{bra}}, j \leq k^{i+1}_{\text{bra}}\}, P^i_{\text{ket}} = \{j | j > k^i_{\text{ket}}, j \leq k^{i+1}_{\text{ket}}\}. \tag{3.74}$$

And, then, collect these into a set of sets

$$P_{\text{bra}} = \{P^i_{\text{bra}} | i \in [0, \text{Weight}(k_{\text{bra}})]\}, P_{\text{ket}} = \{P^i_{\text{ket}} | i \in [0, \text{Weight}(k_{\text{ket}})]\}. \tag{3.75}$$

Finally, one defines a central partition spanning the indices between the last occurrence 1 in the binary representation of k_{bra}, k_{ket} and containing the non-exponentiated factor of the objective function

$$P_{\text{central}} = \{-j | j > \max S_{\text{bra}}\} + \{j | j > \max S_{\text{ket}}\}. \tag{3.76}$$

The expectation value after the substitution of the characteristic function and its derivative can then be expressed as:

$$E_p(\vec{\gamma}, \vec{\beta}) = \sum_{k_{\text{bra}}, k_{\text{ket}}=0}^{2^p-1} -i \prod_{P \in P_{\text{bra}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \Gamma'\left(\sum_{i \in P_{\text{central}}} \gamma_i\right) \prod_{P \in P_{\text{ket}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \prod_{-j | k^j_{\text{bra}}=1} B_j \prod_{j | k^j_{\text{ket}}=1} B_j. \tag{3.77}$$

For which, exchanging k_{bra}, k_{ket} conjugates the term, allowing the expression to be split into two parts, one in which $k_{\text{bra}} = k_{\text{ket}}$ and one otherwise, so:

$$\begin{aligned} E_p(\vec{\gamma}, \vec{\beta}) = &\sum_{k=0}^{2^p-1} -i \prod_{P \in P_{\text{bra}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \Gamma\left(\sum_{i \in P} \gamma_i\right)^* \Gamma'(0) \prod_{j | k^j=1} B_j B^*_j \\ &+ \sum_{k_{\text{bra}} < k_{\text{ket}}=0}^{2^p-1} -i \prod_{P \in P_{\text{bra}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \Gamma'\left(\sum_{i \in P_{\text{central}}} \gamma_i\right) \\ &\prod_{P \in P_{\text{ket}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \prod_{-j | k^j_{\text{bra}}=1} B_j \prod_{j | k^j_{\text{ket}}=1} B_j + \text{Conjugate}. \end{aligned} \tag{3.78}$$

If the problem mean is zeroed by global phase, $\Gamma'(0) = 0$ and the first sum disappears, leaving:

$$E_p(\vec{\gamma}, \vec{\beta}) = 2 \,\text{Im}\left[\sum_{k_{\text{bra}} < k_{\text{ket}}=0}^{2^p-1} \prod_{P \in P_{\text{bra}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \Gamma'\left(\sum_{i \in P_{\text{central}}} \gamma_i\right) \prod_{P \in P_{\text{ket}}} \Gamma\left(\sum_{i \in P} \gamma_i\right) \prod_{-j | k^j_{\text{bra}}=1 \text{ or } j | k^j_{\text{ket}}=1} B_j\right]. \tag{3.79}$$

A Python script to evaluate the Grover-QAOA objective function $E_p(\vec{\gamma}, \vec{\beta})$ for a given characteristic function is available upon request to the author.

Chapter 4

Driver-Problem Synergy and QAOA Performance

In Chapter 3, we considered a situation in which a Grover driver was utilized for QAOA, allowing the calculation of optimal angles for problems in the infinite problem-size limit. In this chapter we extend this treatment to the typically used single-qubit-X driver and, in doing so, are required into account a greater amount of information about the problems that we solve. This work illustrates the mechanism through which the Quantum Approximate Optimization Algorithm can exploit structure through its drivers. This is the case, as the methods in the chapter make clear the dependence of the QAOA objective function on the conditional probability of obtaining a state of a given value when starting from another and taking a number of steps along a metric defined by a driver. We demonstrate this in the case of two specific drivers, the line driver, and, also, the typical hypercube, single-qubit-X driver typically used in QAOA.

4.1 Attribution

This chapter was written exclusively by myself. The work presented here may be published in the future with myself as the first author and Prof. Wilhelm-Mauch as an advisory figure.

4.2 Introduction

In a landmark result, Farhi et al. obtained an analytical formula for the performance of QAOA on SK models of limiting size, at fixed depth [87]. Later work generalized this result to k-spin models [90]. In a related piece of work, a parameter setting heuristic was developed [112], exploiting the conditional probabilities also explored in this chapter.

This chapter aims to take an alternative approach to this area of studying QAOA on ensembles of problems. Rather than expressing the QAOA expectation as a k-spin model, then computing an expected value of the QAOA state for this, we attempt to express the QAOA expected solution value generically in terms of conditional probability density functions, in the spirit of the previous chapter. To do this, we break the algorithm into two parts and study a problem-dependent part in isolation.

By isolating the dependence of this problem dependent part on the problems solved, one can study a key factor in the performance of QAOA, the exploitation of problem structure. This work, therefore represents a step towards the determination of good QAOA angles and QAOA performance, based on generic statistical properties of problems, which can be computed without access to a quantum computer, and at limiting problem size.

4.3 Theory

Driver Hamiltonians in QAOA consist of a sum over Hermitian operators. In the case of the Grover driver, addressed in the previous section, this sum has just one term regardless of the number of qubits in the QAOA instance. This low number of terms leads to favorable scaling in the results of the previous chapter, and results that are intrinsically decoupled from the size of the instance. Once exponentiated to a unitary operator by a variational parameter used, this Grover-generated unitary contains 2 terms—the identity and the many-body projector $|+\rangle\langle+|$—multiplied by $\vec{\beta}$-dependent pre-factors. With $2p$ layers in the expectation value of the problem Hamiltonian, this results in 2^{2p} terms in the expectation value. A way to view this decomposition is that the driver imposes a distance measure on the configuration space of the computational basis states of the problem solved. In the case of the Grover driver, there are just two possible distances, 0 and 1, that is, whether two states are the same or are different. These outcomes, then, correspond to the identity and projection operators of the exponentiated, expanded, Grover driver.

$$\hat{H}_{\mathrm{G}} = |+\rangle^{\otimes n}\langle+|^{\otimes n} = \left[\frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}|z\rangle\right]\left[\frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}\langle z|\right]. \tag{4.1}$$

$$\hat{U}_{\mathrm{D}}(\beta) = \mathrm{e}^{i\beta|+\rangle\langle+|} = \mathbb{I} + i\beta\,|+\rangle\langle+| + \frac{(i\beta)^2}{2}|+\rangle\langle+|\ldots = \mathbb{I} + (\mathrm{e}^{i\beta}-1)\,|+\rangle\langle+|\,. \tag{4.2}$$

In this chapter, we aim to generalize this. The Grover driver sees structure with respect to whether two states are the same or different. A direct consequence of this is that at limiting size, the Grover-driven algorithm simply depends on a pdf describing a distribution of possible cost function values. Then, in the case of more complex drivers of more terms, what does the algorithm *see* and on what properties, like that of the pdf describing the distribution of randomly sampled states, does the algorithm depend? The question this chapter aims to answer is that of which statistics of a problem QAOA resolves at a large size. While the Grover driver is chosen specifically such that only the distribution of a problem is resolved, other drivers do not see problems so simply, depending on the relationship between values of the cost function, not simply the distribution thereof.

For the Grover driver, the driver Hamiltonian is a sum of just 2 terms, and since there are $2p$ instances of such a driver in a QAOA expectation value, results in an expectation value summing 2^{2p} terms.

For other choices of drivers, as in the case of the more standard, typically used single-qubit-X drivers, such a sum will then consist of more terms. This number of terms will depend on the number of qubits in a QAOA instance, therefore encoding significantly more complexity than in the Grover case. When such a sum over driver

terms is exponentiated, we can subsequently group terms by order. The single-qubit-X driver is written:

$$\hat{H}_{\mathrm{D}} = \sum_i X_i \tag{4.3}$$

which exponentiates by parameter β to a driver unitary

$$\hat{U}_{\mathrm{D}}(\beta) = \mathrm{e}^{i\beta\sum_j X_j} = \prod_j [Ic + isX_j] \tag{4.4}$$

where the shorthand $s = \sin\beta$, $c = \cos\beta$ has been utilized. Importantly, we can expand these terms and collect their pre-factors by weight, so:

$$\hat{U}_{\mathrm{D}}(\beta) = \sum_k c^{n-w}(is)^w X_0^{k^0} X_1^{k^1} \dots X_n^{k^n}, \tag{4.5}$$

where k is an n-bit number that indexes each possible term, with $w = \mathrm{Weight}(k)$ as the number of bits in k equal to 1, the Hamming weight of k.

We can perform a similar expansion for a step, or line mixer. The line mixer occurs in the logical subspace in settings such as that of a one-hot encoding of variables. Typically, combinatorial optimization problems in this thesis utilize binary variables, which are naturally mapped immediately to qubits. However, if a problem consists of variables that may take more than two values, a qubit embedding of such variables must be selected. An efficient option for such a variable, taking L different possible values, would be a binary embedding on $\lceil \log_2(L) \rceil$ qubits. A challenge here is that clauses on such a variable then tend to be high-weight operations. An alternative used in a quantum annealing context is to use a unary encoding, using L qubits for such a variable. Typically, all qubits will be in the state 0, with the exception of one qubit in the state 1, with this qubit determining the state of the variable. An alternative has also been explored, using domain walls to denote the state of the variable, rather than a *hot* qubit [113]. While, depending on the size of the variables, this encoding is significantly more inefficient in terms of qubit count, one can gain the benefit of lower weight interaction terms, which are preferred in continuous-time contexts such as quantum annealing in which compiled gates are not possible. So, on a line of L qubits comprising a unary encoded variable, a driver Hamiltonian consisting of XY interactions can be used to keep a state—initially in a valid state with a single qubit in state $|1\rangle$—in a number-preserving subspace. Such Hamiltonians take the form:

$$\hat{H}_{\mathrm{Line}} = \sum_i X_i X_{i+1} + Y_i Y_{i+1}, \tag{4.6}$$

and simply move excitations between qubits, rather than changing the number of qubits in the $|1\rangle$ state. In the single excitation subspace, this Hamiltonian can be expressed as:

$$\hat{H}_{\mathrm{Line}} = \sum_{|i-j|=1} |i\rangle\langle j| + |j\rangle\langle i|, \tag{4.7}$$

and, when exponentiated, one obtains a unitary of a form:

$$\hat{U}(\vec{\beta}) = \sum_d f(d,\vec{\beta}) \sum_i \left[\, |i\rangle\langle i+d| + |i+d\rangle\langle i| \,\right], \tag{4.8}$$

which contains terms of differing distance d. The idea here is that one may calculate the performance of QAOA with consideration towards the properties of the problem with respect to this driver. In particular, in the line mixer case, the arithmetic distance between two states is the metric of importance. The work of this section provides progress towards elucidating the relationship between the choice of driver and the performance of QAOA. By evaluating the performance of drivers in a context in which a problem has no synergy or best-case synergy with the driver used, one can determine, approximately, the performance of QAOA on problems, given their size-independent properties. This approach, which simplifies problems to their asymptotic statistics, could be important for determining how large-scale QAOA will perform, compared to classical algorithms, and may provide a good method for optimizing the variational parameters of QAOA circuits. This could allow one to determine which drivers should be used for a given problem type.

4.4 Single-Qubit Drive

Single-qubit-X drivers are the most practical mixers for QAOA, since they present a choice with minimal cost-to-compile on a near-term quantum computer, in the context of which, such gates are practically free. To examine how they see the problems for which we use them, we can express the expectation value of the problem Hamiltonian under the QAOA state as expressed in Chapter 3, in terms of the weight of different operators. We first carry out these calculations at depth 1 and then generalize this to depth 2 and up to arbitrarily large circuit depth.

4.4.1 Depth 1

At depth 1 we can write the average solution quality generated by QAOA with single-qubit-X drivers as:

$$E_1(\gamma,\beta) = \langle \hat{H}_{\mathrm{P}} \rangle_{\gamma,\beta} = \langle + | \, \hat{U}_{\mathrm{P}}^{\dagger}(\gamma) \hat{U}_{\mathrm{D}}^{\dagger}(\beta) \hat{H}_{\mathrm{P}} \hat{U}_{\mathrm{D}}(\beta) \hat{U}_{\mathrm{P}}(\gamma) \, | + \rangle \tag{4.9}$$

and into this expression, we can substitute the expression for the single-qubit-X driver to produce:

$$E_1(\gamma,\beta) = \sum_{z_{-1},z_0,z_1} \langle + | \, \mathrm{e}^{-i\gamma C(z_{-1})} \, | z_{-1} \rangle \, \langle z_{-1} | \left[Ic - isX \right]^n \cdot C(z_0) \, | z_0 \rangle \, \langle z_0 | \left[Ic + isX \right]^n \mathrm{e}^{i\gamma C(z_1)} \, | z_1 \rangle \, \langle z_1 | + \rangle \, . \tag{4.10}$$

Simplifying this with the identity $\langle z | + \rangle = \frac{1}{\sqrt{N}}$, one obtains:

$$\frac{1}{N} \sum_{z_{-1},z_0,z_1} \mathrm{e}^{-i\gamma C(z_{-1})} \, \langle z_{-1} | \left[Ic - isX \right]^n C(z_0) \, | z_0 \rangle \, \langle z_0 | \left[Ic + isX \right]^n \mathrm{e}^{i\gamma C(z_1)} \, | z_1 \rangle \, . \tag{4.11}$$

Expanding the brackets, one obtains 2^{2n} terms. These can be indexed by two indices $k_{\mathrm{bra}}, k_{\mathrm{ket}}$. The indices are associated with the terms produced in the expansion, with

the jth bit of k_{bra}, denoted k^j_{bra}, representing whether the $\mathbb{I}$ or X operators feature in the jth qubit. Therefore:

$$E_1(\gamma,\beta) = \frac{1}{N}\sum_{k_{\text{bra}},k_{\text{ket}}} \left[c^{n-w_{\text{bra}}}(-is)^{w_{\text{bra}}}c^{n-w_{\text{ket}}}(is)^{w_{\text{ket}}}\right] \cdot \sum_{z_{-1},z_0,z_1} \mathrm{e}^{-i\gamma C(z_{-1})}\left\langle z_{-1}\right| X^{k^1_{\text{bra}}}X^{k^2_{\text{bra}}}\ldots X^{k^n_{\text{bra}}}C(z_0) \cdot \left|z_0\right\rangle\left\langle z_0\right|\mathrm{e}^{i\gamma C(z_1)}X^{k^1_{\text{ket}}}X^{k^2_{\text{ket}}}\ldots X^{k^n_{\text{ket}}}\left|z_1\right\rangle . \quad (4.12)$$

The X operators in each term can be applied to the outer and inner bra and ket respectively, flipping a corresponding bit in the index z_1 or z_{-1}. We use the notation $\oplus$ to denote the bitwise XOR of the indices, that is, for each bit, $a \oplus b$ is 0 if $a = b$ and 1 otherwise.

$$E_1(\gamma,\beta) = \frac{1}{N}\sum_{k_{\text{bra}},k_{\text{ket}}} \left[c^{n-w_{\text{bra}}}(-is)^{w_{\text{bra}}}c^{n-w_{\text{ket}}}(is)^{w_{\text{ket}}}\right] \cdot \sum_{z_{-1},z_0,z_1} \mathrm{e}^{-i\gamma C(z_{-1})}\left\langle z_{-1}\oplus k_{\text{bra}}\right| C(z_0)\left|z_0\right\rangle\left\langle z_0\right|\mathrm{e}^{i\gamma C(z_1)}\left|z_1\oplus k_{\text{ket}}\right\rangle \quad (4.13)$$

one can then perform an arbitrary change of variable, with $z_1 \to z_1 \oplus k_{\text{ket}}$ and $z_{-1} \to z_{-1} \oplus k_{\text{bra}}$. This change of variable moves the shift k from the index $z_{\text{bra/ket}}$ to inside the respective cost function occurring in the exponential. This results in:

$$= \frac{1}{N}\sum_{k_{\text{bra}},k_{\text{ket}}} \left[c^{n-w_{\text{bra}}}(-is)^{w_{\text{bra}}}c^{n-w_{\text{ket}}}(is)^{w_{\text{ket}}}\right] \cdot \sum_{z_{-1},z_0,z_1} \mathrm{e}^{-i\gamma C(z_{-1}\oplus k_{\text{bra}})}C(z_0)\mathrm{e}^{i\gamma C(z_1\oplus k_{\text{ket}})}\left\langle z_{-1}|z_0\right\rangle\left\langle z_0|z_1\right\rangle$$

$$= \frac{1}{N}\sum_{k_{\text{bra}},k_{\text{ket}}} \left[c^{n-w_{\text{bra}}}(-is)^{w_{\text{bra}}}c^{n-w_{\text{ket}}}(is)^{w_{\text{ket}}}\right] \cdot \sum_{z} C(z)\mathrm{e}^{i\gamma\left[C(z\oplus k_{\text{ket}})-C(z\oplus k_{\text{bra}})\right]}. \quad (4.14)$$

Which is a sum over 2^{2n} terms, each of which contains, also, a sum over 2^n possible strings, expressing the problem Hamiltonian's expected value. In this way, one obtains terms consisting of first a factor depending on the driving variational parameter β. We refer to this factor as the *driver factor*. The second factor, containing the sum over strings and depending on the cost function's variational parameter γ, we name as the *problem factor*. So, the problem factor is denoted $P = P(\gamma, k_{\text{bra}}, k_{\text{ket}})$, so

$$P(\gamma,k_{\text{bra}},k_{\text{ket}}) = \frac{1}{N}\sum_{z=0}^{N-1} C(z)\mathrm{e}^{i\gamma\left[C(z\oplus k_{\text{ket}})-C(z\oplus k_{\text{bra}})\right]} \quad (4.15)$$

and a driver factor is denoted by $B = B(\beta, k)$ with

$$B(\beta,k) = c^{n-w_{\text{k}}}(is)^{w_{\text{k}}}. \quad (4.16)$$

Where the full expectation value is a sum over the product of these two quantities for different $k_{\text{bra}}, k_{\text{ket}}$. Consolidating the driver factors for each shift to a single term,

one can write $D = B(\beta, k_{\text{bra}})^\dagger B(\beta, k_{\text{ket}})$

$$E_1(\gamma, \beta) = \sum_{k_{\text{bra}}, k_{\text{ket}}} D(\beta, k_{\text{bra}}, k_{\text{ket}}) \cdot P(\gamma, k_{\text{bra}}, k_{\text{ket}}). \tag{4.17}$$

In Chapter 3, the same decomposition was made, albeit with far fewer terms (2 rather than 2^{2n}). The problem factor in the Grover-driven case was also simpler, as could only depend on the characteristic function of the problem's pdf by symmetry. The procedure taken in the context of the Grover driver was to approximate the sum in the problem factor as an integral, resulting in an analytic expression independent of the size of the problem solved. In this chapter, we aim to do the same for the more complex problem factor derived above.

There are further simplifications we can make to the expression for E_1 to reduce the number of terms to be calculated in the sum. Firstly, if two shifts are zero, then the problem factor integral evaluates to the mean value $\bar{c}$, which can, itself, be arbitrarily set to zero. The same occurs in the case that k_{bra} and k_{ket} take the same value. This is evident since:

$$P(\gamma, k, k) = \sum_z \frac{C(z)}{N} \mathrm{e}^{i\gamma[C(z\oplus k) - C(z\oplus k)]} = \sum_z \frac{C(z)\mathrm{e}^{0i\gamma}}{N} = \bar{c} \to 0. \tag{4.18}$$

Secondly, by considering that $E_1^\dagger = E_1$, we can pair terms to find pairs for which imaginary parts cancel leaving the real contribution to the expectation value. Note, taking the complex conjugate of the expression is equivalent to swapping the values of k_{bra} and k_{ket} in a given term in the definitions of P and D. As such, terms and their complex conjugate pair can be collected in the sum, allowing us to recast the sum as fewer terms, so:

$$E_1(\gamma, \beta) = 2 \sum_{k_{\text{bra}} < k_{\text{ket}} = 0} \mathrm{Re}\{D(\beta, w_{\text{bra}}, w_{\text{ket}}) \cdot P(\gamma, k_{\text{bra}}, k_{\text{ket}})\}, \tag{4.19}$$

which has been reduced to contain $N(N-1)/2$ terms. Of course, the ultimate goal of an analytical procedure for calculating the expected value should be at worst to provide an algorithm scaling polynomially in the number of qubits. The expression above is still exponential in the number of qubits, so further processing must occur to reduce this.

It is reasonable to expect that such processing is possible, due to the symmetry inherent in the formula. All driver factors of identical weight shifts are identical, resulting in an averaging of corresponding problem factors. To collapse the exponential complexity here to one of polynomial complexity, one needs to find an expression for the expected value of the problem factor, averaging over all possible shifts corresponding to a given weight. For example, for n qubits, a term with weights $w_{\text{k}}, w_{\text{b}}$ has $\binom{n}{w_{\text{k}}} \cdot \binom{n}{w_{\text{b}}}$ possible shifts, over which one should average. The calculation of such an expectation value should then be efficient, scaling also at most polynomially in the number of qubits.

4.4.2 Depth 2

At depth 2 we can write down the expected value of the problem Hamiltonian as:

$$\langle \hat{H}_{\text{P}} \rangle_{\gamma_1, \gamma_2, \beta_1, \beta_2} = \langle \gamma_1, \gamma_2, \beta_1, \beta_2 | \, \hat{H}_{\text{P}} \, | \gamma_1, \gamma_2, \beta_1, \beta_2 \rangle \,, \tag{4.20}$$

which, inserting the expressions for the problem and driver Hamiltonians, results in:

$$\begin{aligned}
E_2(\gamma_1,\gamma_2,\beta_1,\beta_2) &= \langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\gamma},\vec{\beta}} \\
&= \langle + | \hat{U}_{\mathrm{P}}^{\dagger}(\gamma_1)\hat{U}_{\mathrm{D}}^{\dagger}(\beta_1)U_{\mathrm{P}}^{\dagger}(\gamma_2)\hat{U}_{\mathrm{D}}^{\dagger}(\beta_2)\hat{H}_{\mathrm{P}}\hat{U}_{\mathrm{D}}(\beta_2)\hat{U}_{\mathrm{P}}(\gamma_2)\hat{U}_{\mathrm{D}}(\beta_1)\hat{U}_{\mathrm{P}}(\gamma_1)|+\rangle \\
&= \sum_{z_{-2},z_{-1},z_0,z_1,z_2} \langle + | \, \mathrm{e}^{-i\gamma_1 C(z_{-1})} |z_{-1}\rangle \langle z_{-1}| \left[Ic_1 - is_1 X \right]^n \\
&\qquad \mathrm{e}^{-i\gamma_2 C(z_{-2})} |z_{-2}\rangle \langle z_{-2}| \left[Ic_2 - is_2 X \right]^n \\
&\quad C(z_0)|z_0\rangle\langle z_0| \left[Ic_2 + is_2 X \right]^n \mathrm{e}^{i\gamma_2 C(z_2)} |z_2\rangle \langle z_2| \left[Ic_1 + is_1 X \right]^n \mathrm{e}^{i\gamma_1 C(z_1)} |z_1\rangle \langle z_1 | + \rangle .
\end{aligned} \tag{4.21}$$

From here, one may substitute $\langle z|+\rangle = N^{-1/2}$ and expand the driver unitaries into sums over *shifts*, each of which takes a value between 0 and 2^n. Again, we introduce a driver-related function:

$$B(k_j) = \cos(\beta_j)^{n-w_{k,j}} (i \sin \beta_j)^{w_{k,j}}. \tag{4.22}$$

We also introduce the notation in which negative indices of k refer to those in the bra, with positive indices referring to those in the shift of the ket. So $k_{\mathrm{bra},i} \to k_{-i}$ and $\beta_{-i} = -\beta_i$. So

$$\begin{aligned}
&= \frac{1}{N} \sum_{k_{-1},k_1,k_{-2},k_2} \sum_{z_{-2},z_{-1},z_0,z_1,z_2} \mathrm{e}^{-i\gamma_1 C(z_{-1})} \langle z_{-1}| B(k_{-1}) X^{k_{-1}^1} 1 X^{k_{-1}^2} \dots X^{k_{-1}^n} \\
&\qquad \mathrm{e}^{-i\gamma_2 C(z_{-2})} |z_{-2}\rangle \langle z_{-2}| B(k_{-2}) X^{k_{-2}^1} X^{k_{-2}^2} \dots X^{k_{-2}^n} \\
&\qquad C(z_0)|z_0\rangle\langle z_0| B(k_2) X^{k_2^1} X^{k_2^2} \dots X^{k_2^n} \\
&\qquad \mathrm{e}^{i\gamma_2 C(z_2)} |z_2\rangle \langle z_2| B(k_1) X^{k_1^1} X^{k_1^2} \dots X^{k_1^n} \mathrm{e}^{i\gamma_1 C(z_1)} |z_1\rangle .
\end{aligned} \tag{4.23}$$

We can simplify this as in the depth-1 case by applying the X operators outwards from the center of the expression. The operators have the action of permuting the labels in their outermost bra or ket. Therefore the expression can be written as:

$$\begin{aligned}
&= \frac{1}{N} \sum_{k_{-1},k_1,k_{-2},k_2} \sum_{z_{-2},z_{-1},z_0,z_1,z_2} B_1(k_{-1}) B_2(k_{-2}) B_2^{\dagger}(k_2) B_1^{\dagger}(k_1) \\
&\qquad \mathrm{e}^{-i\gamma_1 C(z_{-1})} \langle z_{-1} \oplus k_{-1} | z_{-2}\rangle \, \mathrm{e}^{-i\gamma_2 C(z_{-2})} \langle z_{-2} \oplus k_{-2} | z_0 \rangle \, C(z_0) \\
&\qquad \mathrm{e}^{i\gamma_2 C(z_2)} \langle z_0 | z_2 \oplus k_2 \rangle \langle z_2 | \, \mathrm{e}^{i\gamma_1 C(z_1)} |z_1 \oplus k_1\rangle .
\end{aligned} \tag{4.24}$$

Where the orthogonal bra-ket terms here are equivalent to Kronecker delta functions between their arguments and can be written as:

$$\begin{aligned}
&= \frac{1}{N} \sum_{k_{-1},k_1,k_{-2},k_2} \sum_{z_{-2},z_{-1},z_0,z_1,z_2} B_1(k_{-1}) B_2(k_{-2}) B_2^{\dagger}(k_2) B_1^{\dagger}(k_1) \\
&\mathrm{e}^{-i\gamma_1 C(z_{-1})} \delta_{z_{-1}\oplus k_{-1}, z_{-2}} \mathrm{e}^{-i\gamma_2 C(z_{-2})} \delta_{z_{-2}\oplus k_{-2}, z_0} C(z_0) \mathrm{e}^{i\gamma_2 C(z_2)} \delta_{z_0, z_2 \oplus k_2} \mathrm{e}^{i\gamma_1 C(z_1)} \delta_{z_2, z_1 \oplus k_1}.
\end{aligned} \tag{4.25}$$

Now, to simplify the expression and evaluate the Kronecker delta functions, one can make changes of variable $z_0 \to z$, $z_{-1} \to z_{-1} \oplus k_{-1} \oplus k_{-2}$, $z_{-2} \to z_{-2} \oplus k_{-2}$, $z_1 \to z_1 \oplus k_1 \oplus k_2$, $z_2 \to z_2 \oplus k_2$. This change moves the shifts from the Kronecker delta functions to the instances of the objective function, allowing the sum to easily be collapsed to a single z variable as:

$$= \frac{1}{N} \sum_{\vec{k}} D(\vec{\beta}, \vec{k}) \cdot \sum_{z} C(z) \mathrm{e}^{-i\gamma_1 C(z \oplus k_{-1} \oplus k_{-2}) + i\gamma_1 C(z \oplus k_2 \oplus k_1) - i\gamma_2 C(z \oplus k_{-2}) + i\gamma_2 C(z \oplus k_2)}. \quad (4.26)$$

Where the driver factor D is a product of all factors B. Which has a problem factor with four terms in the exponent rather than the two seen in the $p = 1$ case. Exchanging the shifts in the bra and ket has the effect of conjugating the expression, and, therefore, the sum can be re-cast accordingly.

4.4.3 Arbitrary Depth

At this point the pattern extending the expression to arbitrary depth becomes clear, and this can be written as:

$$E_p(\vec{\gamma}, \vec{\beta}) = \sum_{\vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}} D(\vec{\beta}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}) \cdot P(\vec{\gamma}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}). \quad (4.27)$$

This can be simplified somewhat first by considering that, as in Chapter 3, if the shifts are all the same, the problem factor will simply evaluate to the mean value of the cost function, which can be arbitrarily set to zero. Secondly, we can also pair terms here such that we only consider a sum over real values. Exchanging the values of $\vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}$ has the same effect as taking the complex conjugate, so we can reduce the number of terms by a factor of two by grouping terms and their complex conjugate. As such, we can express the sum as:

$$E_p(\vec{\gamma}, \vec{\beta}) = \sum_{\vec{k}_{\mathrm{bra}} < \vec{k}_{\mathrm{ket}}} 2\,\mathrm{Re}\Big\{ D(\vec{\beta}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}) \cdot P(\vec{\gamma}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}) \Big\}. \quad (4.28)$$

In which the notation $\vec{k}_{\mathrm{bra}} < \vec{k}_{\mathrm{ket}}$ can be understood as considering $\vec{k}_{\mathrm{bra/ket}}$ as an pn bit number for which the comparator $>$ is defined. Alternatively, one can remove the factor of two in front of the real value and make the replacement $<\to\neq$.

This expression scales exponentially in n and p, but critically we can consolidate many of these terms such that the expression is only polynomial in n. Here, the driver factor can be expressed as:

$$D(\vec{\beta}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}) = \prod_{i=1}^{p} \big[\cos\beta_i{}^{n-w_{-i}} (-i \sin\beta_i)^{w_{-i}} \cos\beta_i{}^{n-w_i} (i \sin\beta_i)^{w_i} \big], \quad (4.29)$$

with the problem factor as:

$$P(\vec{\gamma}, \vec{k}_{\mathrm{bra}}, \vec{k}_{\mathrm{ket}}) = \sum_{z} C(z) \mathrm{e}^{i \sum_i \gamma_i [C(z \oplus s_i) - C(z \oplus s_{-i})]}, \quad (4.30)$$

where the substitutions:

$$s_i = k_i \oplus k_{i+1} \ldots \oplus k_p \tag{4.31}$$

and

$$s_{-i} = k_{-i} \oplus k_{-i-1} \ldots \oplus k_{-p} \tag{4.32}$$

have been made. So the problem factor depends on $2p$ shifts that accumulate. And, subsequently, the problem of approximating this sum can be reduced to taking the expected value of the quantity

$$P(\vec{\gamma}, \vec{k}) \approx \mathrm{E}\left[c\mathrm{e}^{i\sum_j \gamma_j \left[c_{s_j} - c_{s_{-j}}\right]}\right] \tag{4.33}$$

for which we need to know the pdf describing the distribution of the cost function C denoted by $f(C = c)$, alongside the conditional pdfs describing the likelihood of obtaining a cost c_{s_i} after starting at a string with cost function value c and making a move described by the permuting string s_i. Since we take the difference of variables described by such conditional pdfs, we must also determine the covariance matrix of the $2p$ shifts' associated random variables.

4.5 Evaluating Problem Factors Analytically

The problem factor sum can be approximated as an integral over continuous variables. As in Chapter 3, we can use the following substitutions and, therefore approximate the problem factor in the continuum limit. This continuum limit problem factor would be expected to be accurate for modestly large n due to the exponential scaling of the number of terms in the sum.

Treating the quantity $C(z)$ as a random variable c, we can first substitute into the sum, the quantile distribution function, and change the variable of integration, resulting in an expression in terms of the pdf describing the objective function.

$$\frac{1}{N} = \delta p \to dp,\ C(z) \to F^{-1}(p) \tag{4.34}$$

which when applied to the problem factor at $p = 1$ yields:

$$\begin{aligned} P = \sum_{z=0}^{N-1} F^{-1}(p)\mathrm{e}^{i\gamma\left[C(z\oplus k_{\mathrm{ket}}) - C(z\oplus k_{\mathrm{bra}})\right]}\delta p \xrightarrow{n\to\infty} \int_0^1 F^{-1}(p)\mathrm{e}^{i\gamma\left[c_{k_{\mathrm{ket}}} - c_{k_{\mathrm{bra}}}\right]}\,\mathrm{d}p \\ = \int_{-\infty}^{\infty} cf(c)\mathrm{e}^{i\gamma\left[c_{k_{\mathrm{ket}}} - c_{k_{\mathrm{bra}}}\right]}\,\mathrm{d}c = \mathrm{E}_c\left[c\mathrm{e}^{i\gamma\left(c_{k_{\mathrm{ket}}} - c_{k_{\mathrm{bra}}}\right)}\right]. \end{aligned} \tag{4.35}$$

This expression, compared to that derived in Chapter 3, has an extra factor, therefore, the problem factor is not easily replaced by a simple characteristic function as was the case when the Grover driver was used. The expression depends on two quantities $c_{k_{\mathrm{ket}}}$ and $c_{k_{\mathrm{bra}}}$, which represent the value of the cost function after the input string yielding the objective function value c is permuted by the shifts k_{bra} and k_{ket}.

We can, also, model the quantities $c_{k_{\mathrm{ket}}}$ and $c_{k_{\mathrm{bra}}}$ as continuous random variables. These are, however, not independent of the value of c. Therefore, these are conditional distributions, describing the likelihood of the problem's objective function taking the value c_k given it takes value c and has its index modified by string k.

Using the law of the unconscious statistician, we may take the expectation over the random variables to write the expected value of the integral as

$$P(k_{\text{ket}}, k_{\text{bra}}, f, f_{|c,k_{\text{ket}},k_{\text{bra}}}) = \mathrm{E}_{c_{k_{\text{ket}}},c_{k_{\text{bra}}}}\left[\int_{-\infty}^{\infty} cf(c)\mathrm{e}^{i\gamma\left[c_{k_{\text{ket}}}-c_{k_{\text{bra}}}\right]}\,\mathrm{d}c\right]$$
$$= \iiint_{\mathbb{R}^3} cf(c)\mathrm{e}^{i\gamma\left[c_{k_{\text{ket}}}-c_{k_{\text{bra}}}\right]} f_{k_{\text{bra}}}(c_{k_{\text{bra}}}, c_{k_{\text{ket}}}, c) f_{k_{\text{ket}}}(c_{k_{\text{ket}}}, c_{k_{\text{bra}}}, c)\,\mathrm{d}c\,\mathrm{d}c_{k_{\text{ket}}}\,\mathrm{d}c_{k_{\text{bra}}}. \tag{4.36}$$

A quantity that can be evaluated independently of the problem size n with, however, dependence on the size of the problem entering via the exponentially large number of potential shifts on which the problem depends.

Complicating the evaluation of this integral is the non-independence of the quantities $c_{k_{\text{ket}}}$ and $c_{k_{\text{bra}}}$. Therefore, rather than performing the integrals over both random variables, it will normally make sense to perform the sum of the random variables, consolidating them to a single, independent variable, before performing the integration. The extension of this to higher depth simply adds more terms to the exponent, depending on an increasing number of shifts.

Regardless of the depth, the problem factor for a set of shifts then takes the form of an expectation value of the quantity

$$P = \mathrm{E}\left[c\mathrm{e}^{iX}\right] \tag{4.37}$$

in which, the quantity X is a random variable constituted by the sum of a number $2p$ other, not necessarily independent, random variables. Assuming that the sum consists of Gaussian-distributed random variables, the variable X will, itself, be Gaussian. However, making no assumptions about the specific way that X is distributed, but that it is distributed with mean μ and variance σ^2, with pdf described by $\frac{f_X\left(\frac{x-c\mu}{\sigma}\right)}{\sigma}$. Then, the problem factor can be expressed as follows, with the dependence on the specific shifts, on which is dependent, contained within the mean and variance of the variable X. The expected problem factor can be written as

$$\iint_{\mathbb{R}^2} cf(c)f_X(x,c)\mathrm{e}^{ix}\,\mathrm{d}x\mathrm{d}c = \iint cf(c)\frac{f_X\left(\frac{x-c\mu}{\sigma}\right)}{\sigma}\mathrm{e}^{ix}\mathrm{d}x\mathrm{d}c \tag{4.38}$$

where f_X is normalized to have mean 1 and variance 1 and describes the form of the random variable X. We have defined $u = \frac{x-c\mu}{\sigma}$ such that $x = u\sigma + c\mu$. This assumes that the dependence of X on c is entirely mediated via the mean and variance of X, such that this change of variables decouples the dependence. Then, we can write

$$= \iint cf(c)f_X(u)\mathrm{e}^{i(u\sigma+c\mu)}\mathrm{d}u\mathrm{d}c = \int cf(c)\mathrm{e}^{ic\mu}\mathrm{d}c\int f_X(u)\mathrm{e}^{iu\sigma}\mathrm{d}u. \tag{4.39}$$

To account for the non-exponentiated instance of c one can differentiate under the integral with respect to the mean value of X (in units of c), μ, then, one obtains:

$$-i\partial_\mu \int f(c)\mathrm{e}^{ic\mu}\mathrm{d}c\int f_X(u)\mathrm{e}^{iu\sigma}\mathrm{d}u \tag{4.40}$$

for which one can recognize that the expression is that of two characteristic functions, though with differing frequency domain variables. one obtains:

$$-i\sigma\partial_\mu\Gamma(\mu)\Gamma_X(\sigma). \tag{4.41}$$

Assuming the two characteristic functions are both those of Gaussian distributed variables, one can then substitute the standard expression for the characteristic function of a Gaussian with

$$\Gamma(\gamma) = \mathrm{e}^{-\frac{1}{2}\gamma^2}, \tag{4.42}$$

and obtain:

$$i\mu \mathrm{e}^{-\frac{\mu^2+\sigma^2}{2}}. \tag{4.43}$$

For $\sigma = 0$ the Gaussian over x tends to a Dirac delta distribution, but, no special treatment is required and this evaluates unproblematically to the above. It might be expected that for high-depth QAOA, where all parameters are non-zero, thus, in a non-trivial region where one might expect optima to occur, the central limit theorem might result in any X taking Gaussian form even if the quantities from which it is derived are not. This would follow from the central limit theorem, given X would take its value from a large sum of random variables.

4.6 Monte Carlo Estimation of Expected Solution Quality

While other works compute the expectation value of QAOA states by manual computation of a multinomial sum [87, 90]. In this section, an alternative procedure based on sampling terms from the objective function is described. The overarching logic here is that while the number of terms in the QAOA expectation derived in the previous sections scales exponentially as N^{2p}, one would expect that the overall complexity of the formula is not increasing, since, properly normalized, increasing problem sizes will necessarily tend to a limiting value.

A Monte Carlo technique takes some integral, or sum, given by

$$A = \int_{\Omega} f(\vec{x})\ \mathrm{d}\vec{x} \approx \sum_{\vec{x}} f(\vec{x}_i)\delta x^d \tag{4.44}$$

and approximates the integral by sampling from M points $\vec{x}_i \in \Sigma$. Given uniform sampling, the value of the integral can then be approximated as

$$A \approx \frac{1}{M}\sum_{i=1}^{M} f(\vec{x}_i). \tag{4.45}$$

To avoid frequent sampling regions of the problem in which an integrand is very small, one can incorporate importance sampling into this procedure, in which case, a co-ordinate $\vec{x}_i$ is sampled with likelihood described by a pdf $g(\vec{x}_i)$. Then the integral can be approximated by:

$$A \approx \frac{1}{M}\sum_{i=1}^{M} \frac{f(\vec{x}_i)}{g_i(\vec{x}_i)}, \tag{4.46}$$

in which terms contribute less according to how frequently they would be sampled. All that remains for this method is the choice of function designating the importance of different terms. There exist algorithms for optimizing such prior distributions, such as the VEGAS algorithm [114], in which $g(\vec{x}_i)$ is assumed to be separable in

the coordinates. By adaptively improving the prior, the algorithm learns to sample more frequently in areas for which the integrand is concentrated.

Ultimately, in the application of this method to QAOA, we wish to isolate the dependence on the size of the problem in question from the complexity of evaluating the expectation value of the QAOA state for this problem. For problems in this section, in the context where the statistical properties of the problems are those which matter, not the finite-size fluctuations of an instance of a problem, this should be possible. Through some abuse of notation, one might imagine that the expectation value of the problem Hamiltonian at infinite size can be expressed in the form of an integral over continuous weights as:

$$E_p(\vec{\gamma},\vec{\beta}) = \int \prod_{i=1}^{n} \binom{n}{w_i}\binom{n}{w_{-i}} \mathrm{Re}\Big\{D(\vec{w}_{\mathrm{ket}},\vec{w}_{\mathrm{ket}},\vec{\beta})\cdot \mathrm{E}(P(\vec{w}_{\mathrm{ket}},\vec{w}_{\mathrm{ket}},\vec{\gamma}))\Big\}\mathrm{d}\vec{w}_{\mathrm{bra}}\mathrm{d}\vec{w}_{\mathrm{ket}}. \tag{4.47}$$

With some continuous generalization of the combinatorial factors, which, however, are not necessitated by a sampling approach. This quantity, which would be expected to be asymptotically static with n, therefore should require a number of samples independent of the size of a problem.

An alternative application of Monte Carlo methods would be to evaluate the expected value of the problem factor for a given set of weights. One can use the Monte Carlo approach to calculate expectation values of the problem factor given a set of shift weights, giving

$$\mathrm{E}_{\vec{w}_{\mathrm{bra}},\vec{w}_{\mathrm{ket}}}\left[P(\vec{k}_{\mathrm{bra}},\vec{k}_{\mathrm{ket}})\right]. \tag{4.48}$$

This would require a number of iterations per each possible allocation of $2p$ weights and therefore the evaluation would scale as $O(n^{2p})$, assuming that the required number of samples to obtain a good approximation of the problem factor is constant. Especially when assuming a well-informed prior of how the problem factor is distributed with respect to shifts, one would expect this constant scaling of required samples to be the case.

4.7 Applying the Method to Classes of Problem

Now we have a frame for expressing QAOA expectation values in the context of problems as random variables, we can find problems for which the problem factor is calculable. That is, assuming one can find an expression for the problem factor in the continuum limit, by determining expressions for the probability density functions describing the random variables for a given set of shifts, and then computing the integral over weights. As in the Grover case, a respectable application of this method would find problems for which the problem statistics are known, while still constituting a problem that is computationally interesting and *hard*.

4.7.1 The Random Cost Model

For the simplest possible application of this method, we can calculate the performance of X-driven QAOA on problems that do not have any structure with respect to the driver. In this context, no structure refers to the fact that the conditional

distribution describing the probability of finding a state of cost c', given a move from a state of cost c, is the same as the global pdf. It does not matter where you start from, if you make any non-zero move, this is equivalent to re-sampling a state randomly. As previously introduced in Chapter 3, this is named the random cost model. We define the pdf for the random cost model to be Gaussian, though any normalized distribution could be used. Using a Gaussian distributed variable, however, carries the advantage that sums of such independent Gaussian distributed variables are also Gaussian. So,

$$f(c) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\frac{-c^2}{2}} \text{ where } \int_{\infty}^{\infty} f(c)\,\mathrm{d}c = 1. \tag{4.49}$$

We can also define the conditional pdf defining c_k identically for non-zero shifts. In terms of clauses on n qubits, the random cost model can be attained via a sum over all possible Pauli-Z-based strings, so:

$$\hat{H}_{\mathrm{RCM}} = \sum_{i=0}^{2^n} g_i Z^{i_1} Z^{i_2} \dots Z^{i_n}, \tag{4.50}$$

where subscripts j on i_j denote the jth bit of the index i. The weights g_i, in this case, would need to be taken to be Gaussian distributed with mean 0 and variance $\frac{1}{N}$ such that the distribution of objective function values is distributed with the standard parameters. For a shift $k \neq 0$, the cost function has no dependence on the starting position and results in a pdf equivalent to that of randomly sampling a new state. So,

$$f_{k\neq 0}(c_k|c) = f(c_k) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\frac{-c_k^2}{2}} \text{ where } \int_{\infty}^{\infty} f(c_k)\,\mathrm{d}c_k = 1. \tag{4.51}$$

When a shift $k = 0$, the variable c_k is, of course, unchanged from c and would have a pdf described by a Dirac delta function centered on c, so

$$f_{k=0}(c_k|c) = \delta(c_k - c). \tag{4.52}$$

Given such a pdf and conditional pdf, the task remains to evaluate the expectation value of the problem factor.

Depth 1

At $p = 1$ there is one problem parameter $\gamma = \gamma_1$. The problem factor depends on two *shifts*, k_{bra} and k_{ket}. The random variable X for which we take an expectation value is therefore:

$$X = \gamma(c_{k_{\mathrm{ket}}} - c_{k_{\mathrm{bra}}}). \tag{4.53}$$

To understand this random variable, we can consider all possible cases of the shifts.

One can first note that if both of k_{bra}, k_{ket} are the same, the difference of the variables $c_{k_{\mathrm{ket}}} - c_{k_{\mathrm{bra}}}$ is zero or described by a Dirac delta distribution centered at zero. In this case, the variable describes the difference in the values of a cost function when applying two identical sets of flips. Of course, with the same flips, one arrives at the same location and the difference between the value of the cost function and itself at this location is zero. Via Equations 4.41 and 4.43, this is also apparent as

the mean of the variable X is zero, or the mean value of the cost function which can be trivially set to zero. So,

$$P(\gamma, \vec{k}_{\text{bra}} = k, \vec{k}_{\text{ket}} = k) = 0. \tag{4.54}$$

In the less trivial case in which both k_{bra}, k_{ket} are non-zero and non-equal, one notes that the variances and means of these Gaussian sums, since they are identically independently distributed, are equal. As the means are equal, the difference of these mean values is zero, and therefore via Equation 4.43 the problem factor evaluates to zero. Essentially, since the random variable X is completely independent of the value of the start point c, the integrals are separable from the odd integral over symmetric bounds in c, therefore evaluating to the mean value of 0.

This leaves the final case in which one of k_{bra}, k_{ket} is zero and one is non-zero. These terms will have mean $\mu = \pm\gamma$ depending on which term is non-zero and variance γ (in units of c) since the non-zero is a normalized Gaussian centered at zero, with the zero shift term being a Dirac delta distribution centered at c. So, Equation 4.43 dictates that the problem factor evaluates to:

$$\pm i\gamma \mathrm{e}^{-\gamma^2}. \tag{4.55}$$

Putting these possible cases together, one can write the full, case-wise expression for the problem factor as:

$$P(k_{\text{bra}}, k_{\text{ket}}, f_c, f_{|c,k}) = \begin{cases} 0 \text{ if } k_{\text{bra}} = k_{\text{ket}}, \\ 0 \text{ if } k_{\text{bra}} \neq 0 \text{ and } k_{\text{ket}} \neq 0, \\ i\gamma \mathrm{e}^{-\gamma^2} \text{ if } k_{\text{bra}} = 0 \text{ and } k_{\text{ket}} \neq 0, \\ -i\gamma \mathrm{e}^{-\gamma^2} \text{ if } k_{\text{ket}} = 0 \text{ and } k_{\text{bra}} \neq 0. \end{cases} \tag{4.56}$$

And, so, we can evaluate the structure factor for every possible pair of shifts.

One then needs to combine this with the driver factor and evaluate the full sum over shifts. For this specific case, we can re-factorize to evaluate all terms as a single algebraic expression. While in this case, it is possible to factorize the full expression, it is expected that this is not typical for more complicated problems and at higher depth. It would, in general, be worthwhile to obtain QAOA expectation via this method even in the case that we could simply reduce the size of the sum from an exponential number of terms to a number polynomial in the number of qubits, so such a compact result is not essential.

Substituting the analytical expression for the problem factor into the full expression, we obtain:

$$E_1(\gamma, \beta) = \sum_{k_{\text{bra}} k_{\text{ket}}} \left[c^{n-w_{\text{bra}}} (-is)^{w_{\text{bra}}} c^{n-w_{\text{ket}}} (is)^{w_{\text{ket}}} \right] \cdot P(k_{\text{bra}}, k_{\text{ket}}, f_c, f_{|c,k}) \tag{4.57}$$

in which we can remove terms that evaluate to zero as per Equation 4.56. This results in a set of $2^n - 1$ terms and their complex conjugates:

$$E_1(\gamma, \beta) = \sum_{k_{\text{ket}}=1}^{N-1} \left[c^{n-w_{\text{ket}}} (is)^{w_{\text{ket}}} \right] c^n i\gamma \mathrm{e}^{-\gamma^2} + c.c. \tag{4.58}$$

To evaluate all such terms in a single formula, we can note that the β dependent pre-factor can be expressed using Euler's formula as

$$[c+is]^n = \sum_{w=0}^{n} \binom{n}{w} [c^{n-w}(is)^w] = \mathrm{e}^{in\beta}, \tag{4.59}$$

and, therefore, subtracting the zero weight term to match the initial indices of the sums above and below, one obtains

$$\sum_{k_{\mathrm{ket}}=1}^{N} \left[c^{n-w_{\mathrm{ket}}}(is)^{w_{\mathrm{ket}}}\right] = \mathrm{e}^{in\beta} - \cos\beta^n. \tag{4.60}$$

When substituted, this results in

$$\left[i\sum_{k=1}^{n-1} c^{n-w}(is)^w - i\sum_{k=1}^{n-1} c^{n-w}(-is)^w\right] \cdot c^n\gamma\mathrm{e}^{-\gamma^2} = \left[i\mathrm{e}^{in\beta} - i\mathrm{e}^{-in\beta}\right] \cdot c^n\gamma\mathrm{e}^{-\gamma^2}$$
$$= 2\sin(n\beta)\cos(\beta)^n\gamma\mathrm{e}^{-\gamma^2}. \tag{4.61}$$

This formula is the analytically derived solution quality for a QAOA state for the random energy model at $p = 1$. The formula is plotted in Figure 4.2 alongside the convergence of finite-size instances to the analytically calculated landscape. As this expression is separable, to find optimal parameters for this formula, one may simply differentiate the expectation value with respect to the variational parameters and evaluate this to zero. For γ this optimal value is the same as the random cost model for Grover-driven QAOA in Chapter 3 with a maximizing value $\gamma = \frac{1}{\sqrt{2}}$. The maxima for β will occur where

$$\frac{\partial}{\partial\beta}\sin(n\beta)\cos(\beta)^n = 0 \tag{4.62}$$

which some algebra determines is satisfied at points:

$$\beta^* = \frac{2\pi m + \pi}{2n+2} \tag{4.63}$$

for integer m. For large n, with $m = 0$ corresponding to the first maxima, one determines that

$$\beta^* \approx \frac{\pi}{2n} \tag{4.64}$$

so

$$\sin(n\beta^*)\cos^n(\beta^*) \approx \sin\left(\frac{\pi}{2}\right)\cos\left(\frac{\pi}{2n}\right)^n \approx 1. \tag{4.65}$$

Therefore, QAOA at depth 1 on random cost models obtains a mean sampled solution quality of

$$E_1(\gamma^*, \beta^*) = \sqrt{\frac{2}{e}}. \tag{4.66}$$

This is identical to the performance of the Grover-driven version of the algorithm in Chapter 3 for an identical problem. The expectation-maximizing driver parameter, however, is smaller for large n in the case of single-qubit-X driven problems. This could potentially result in issues with the desired precision for the single-qubit gate, as increasing problem sizes would require increasingly shorter-duration X-gates. The convergence of finite-size instances to these analytically determined angles is seen in Figure 4.1.

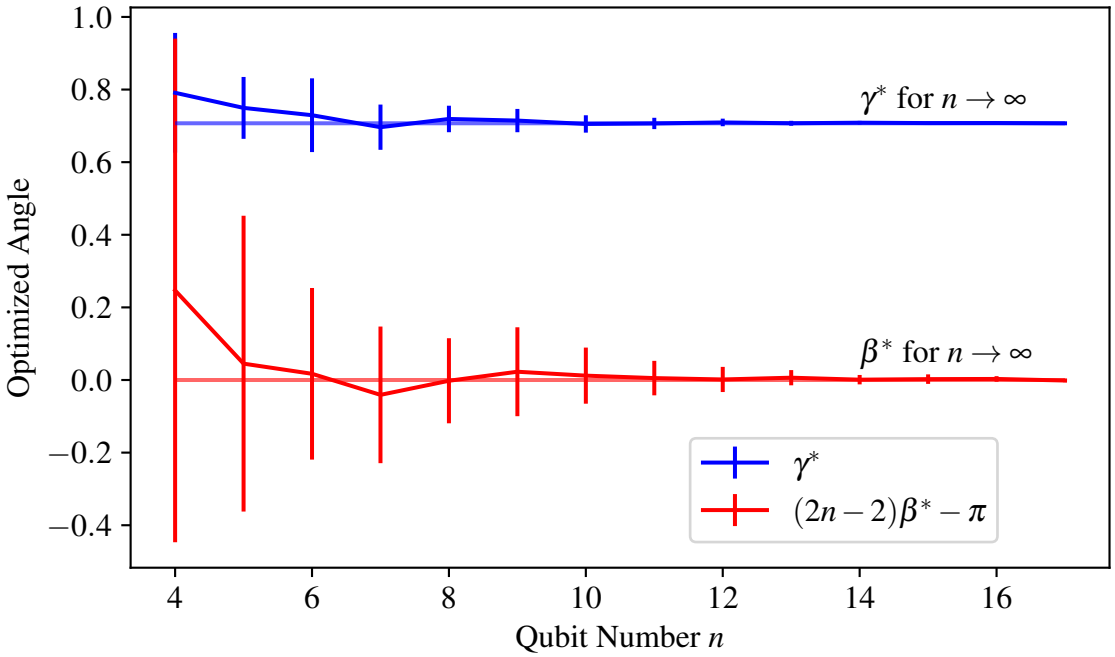

Figure 4.1: Generated instances of unstructured Gaussian problems are used to find optimal QAOA parameters at $p = 1$. The parameters concentrate on the theoretically predicted values for increasing problem sizes.

Depth 2

Computing the expectation value of QAOA applied to the random cost model at depth 2 results in the need to compute a problem factor in which the exponentiated random variable X contains four terms rather than the previous 2, alongside an additional γ parameter. And so, one is required to calculate the mean and variance of the sum of four terms X where:

$$X = \gamma_1 \left[c_{s_1} - c_{s_{-1}}\right] + \gamma_2 \left[c_{s_2} - c_{s_{-2}}\right]. \tag{4.67}$$

The calculation of the mean and variance of this property is simplified by the Gaussian nature of the variables which are summed, but complicated by the non-independence of these variables. For example, if the shifts associated with a pair of these c_k's are equal, they are co-dependent, with mean values that add, but variances that do not, whereas, in the case that the shifts are different, their distributions are independent.

The mean does not depend on the independence of the variables and can be evaluated as:

$$\langle X \rangle = c\gamma_1(\delta_{s_1,0} - \delta_{s_{-1},0}) + c\gamma_2(\delta_{s_2,0} - \delta_{s_{-2},0}) = c\mu. \tag{4.68}$$

To compute the variance of X, we can sum over the potential shifts k. For each k, we add together the factors that contribute to that shift, then square the contributions from each shift. This is because the variances for independent variables combine additively, with each shift composing an independent random variable. That is,

$$\mathrm{Var}(aX_1 + bX_2) = a^2\mathrm{Var}(X_1) + b^2\mathrm{Var}(X_2), \tag{4.69}$$

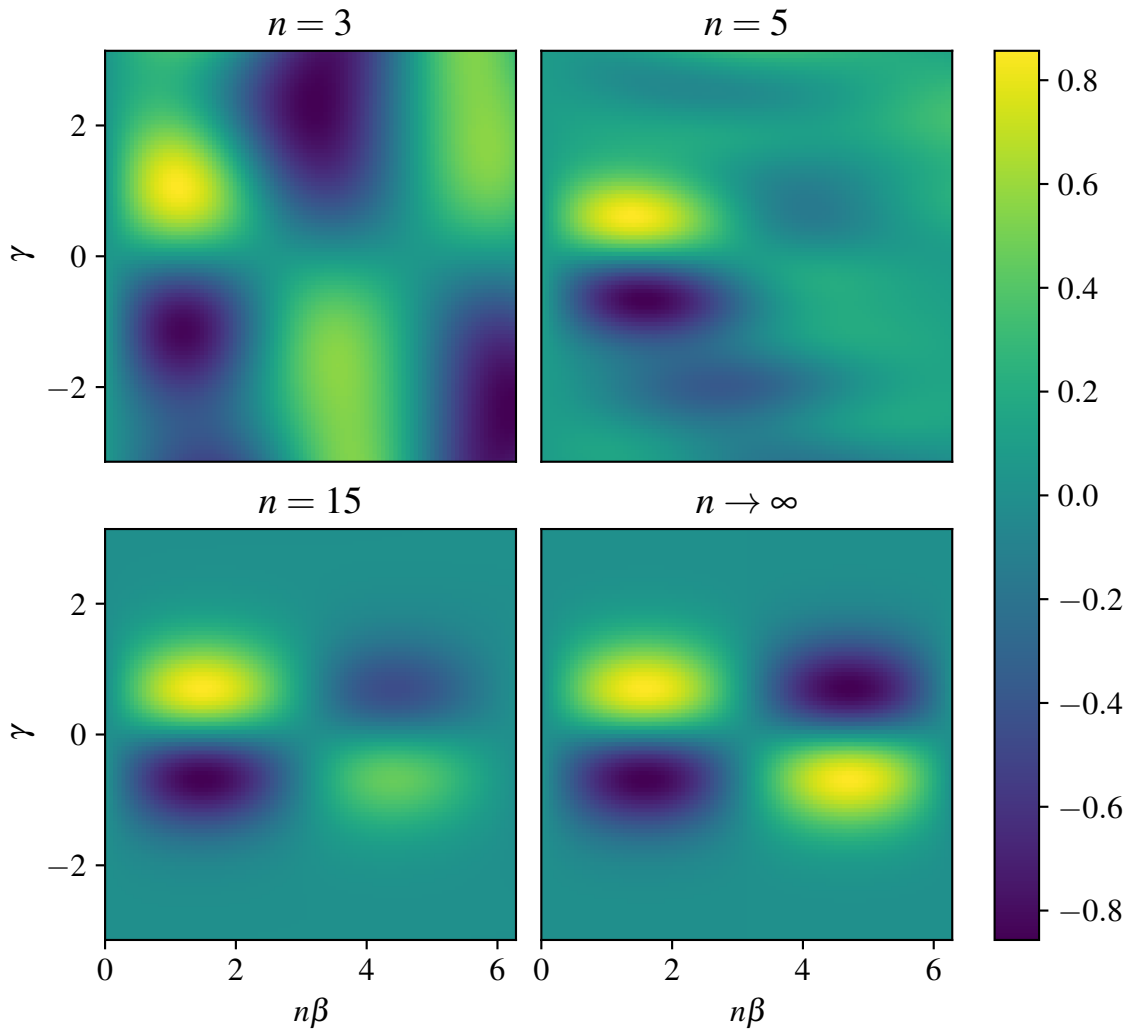

Figure 4.2: For $p = 1$ the landscape can be seen to converge to that of the large-n limit for $n = 3, 5, 15$ for random cost models. As the landscape compresses in the driver-parameter β as the number of qubits increases, the x-axis is scaled by the qubit number to keep the main features in the landscape visible. The landscape exhibits the same maxima as that of the Grover-driven version of Chapter 3.

and so

$$\mathrm{Var}(X) = \sigma^2 = \sum_{k \in \{k_1, k_{-1}, k_2, k_{-2}\}} \left[\gamma_1 \left(\delta_{k_1,k} - \delta_{k_{-1},k}\right) + \gamma_2 \left(\delta_{k_2,k} - \delta_{k_{-2},k}\right)\right]^2 . \quad (4.70)$$

So, for depth 2, there are at most 4 shifts to consider for each term (There is no contribution to the variance if the shift we are considering has no overlap with those of the term). Of course, we don't want to calculate each of these exponentially many terms independently. For $p = 1$ we were able to case-by-case consider the possibilities, however, for $p = 2$ the number of cases is significantly larger. Firstly, if no shift is non-zero, the integral evaluates to zero. If all shifts are non-zero then the integral evaluates, also, to zero. This means one, two, or three shifts must be non-zero to give non-zero contributions. There are, then, many cases that must be

categorized by their mean and variance. While not addressed in this thesis, this combinatorics problem is expected to be tractable in polynomial time on a classical computer.

Arbitrary depth

To evaluate the structure factor at arbitrary depth, a similar procedure applies, though, with n^{2p} different weight combinations, or N^{2p} combinations of shifts to consider. For a set of shifts, one has a random variable consisting of a sum over $2p$ shifts. As in the previous case, the problem factor would be zero in cases where all shifts are non-zero, and, therefore, one would expect the expectation value of the objective function to be constituted largely from a relatively small fraction of the space of possible shifts. This is, therefore a pathological case for the use of Monte Carlo techniques with random sampling, as one should know exactly where the non-zero expectation is concentrated in order to approximate the sum with a number of queries scaling sub-exponentially in the number of qubits of the problem.

To evaluate QAOA performance on random cost models, one could pick a prior distribution of where we expect the large elements of the expectation value to lie, given what we know about the problem. That is, we should sample from a combination of weights in the QAOA expectation value in Equation 4.47 where one expects the contribution to be non-zero. Taking intuition from the $p = 1$ case, this should be cases in which one or more of the weights are zero.

Implications

In Chapter 3, we demonstrated that we can calculate the performance of structure agnostic drivers known as Grover drivers when they are used for QAOA. In the previous section, we have moved this structurelessness to the problem, while still using the usual single-qubit-X driver of QAOA. We have shown explicitly that at $p = 1$, with Gaussian problems, the performance of Grover-driven QAOA is identical to that of X-driven QAOA, presuming that the problems have no structure relative to the driver, in the single-X case, and have any structure, but with identical global pdf, in the Grover case.

4.7.2 Generalized Gaussian Correlation Models

In early work by Mezzard and Gross [115], the properties of k-spin models were investigated. This followed from work by Derrida [116] in which the large-k limit of such models results in properties identical to those of random costs. For a Hamiltonian given by:

$$\hat{H} = \sum_{\text{Weight}(l)=k} g_l Z^{l_1} Z^{l_2} \dots Z^{l_n}, \tag{4.71}$$

where g_l takes non-zero values for terms l with k non-zero bits. Derrida sets the coupling values with pdf

$$f(g_l) = \sqrt{\frac{n^{k-1}}{\pi J^2 k!}} \exp\left[-\frac{g_l^2 n^{k-1}}{J^2 k!}\right]. \tag{4.72}$$

That is, the values are normally distributed with mean 0 and variance $\frac{2J^2k!}{n^{k-1}}$. The probability of sampling a random state with energy c, in the large n limit can then be written as:

$$f(c) = \frac{1}{\sqrt{n\pi J^2}} \exp\left[-\frac{c^2}{J^2 n}\right]. \tag{4.73}$$

Which is, in fact, independent of k[1]. The joint probability for finding two states with costs c_1, c_2 can be written in terms of a normalized distance between the two states

$$q = 1 - \frac{2w}{n} \tag{4.74}$$

where w is a number of bit flips separating strings with values c_1 and c_2, and so, is written as:

$$f(c_1, c_2, q) = \left[n\pi J^2(1+q^k)n\pi J^2(1-q^k)\right]^{-\frac{1}{2}} \cdot \exp\left[-\frac{(c_1+c_2)^2}{2n(1+q^k)J^2} - \frac{(c_1-c_2)^2}{2n(1-q^k)J^2}\right]. \tag{4.75}$$

This joint pdf does not depend explicitly on the configuration of a given spin, only the cost thereof, and the Hamming distance separating the two. Derrida showed that in the limit of large k, the joint probability distribution tends to separability. That is, for strings that are neither identical nor the negated version of the other, the probability that they will take given energy is independent of their distance, which is the key property defining the random cost model. Therefore, from this, one would expect the QAOA parameters obtained for the random cost model to be valid for k-spin problems with high k.

One observation one could make here is that in order to keep a constant random sampling pdf, with variance 1, as is used in this thesis[2] would be to set $J^2 = \frac{2}{n}$. This means that for a k-spin model, the pre-factor of a given term takes a variance of $\frac{2J^2k!}{n^{k-1}} = \frac{k!}{n^k}$.

$p = 1$

To determine the value of the problem factor associated with this joint pdf, one should determine the mean of the marginal distribution. So, for a random variable derived via that starting at a value c and moving by a shift of weight w, the conditional distribution for the value of the new string is given by Bayes theorem, with:

$$f(c^*|c) = \frac{f(c^*, c, q)}{f(c)}, \tag{4.76}$$

which gives:

$$f(c^*|c, d) = \left[2\pi(1+q^k)(1-q^k)\right]^{-\frac{1}{2}} \exp\left[-\frac{(c+c^*)^2}{4(1+q^k)} - \frac{(c-c^*)^2}{4(1-q^k)} + \frac{c^2}{2}\right], \tag{4.77}$$

[1]That is, there is a whole class of Gaussian correlated problems for which Grover-driven QAOA performs equally well, for which, we would expect there is a significant advantage to using X-drivers.

[2]In the literature, the variance of the overall pdf is not kept constant. Often, the important quantity to keep steady is the free energy per spin, and therefore a different convention is used.

for which the exponent can be factorized as:

$$f(c^*|c,d) = \left[2\pi(1+q^k)(1-q^k)\right]^{-\frac{1}{2}} \exp\left[-\frac{c^* - q^k c}{2(1-q^k)(1+q^k)}\right], \tag{4.78}$$

from which a mean and variance can readily be calculated. Determining the problem factor at depth 1 requires that one takes the difference of two such random variables determined by two such conditional distributions. One could mechanically calculate a pdf for such a difference of random variables, though a simpler method is to use the associated rules for adding the first and second moments of such distributions.

For the mean value of a difference between two variables, one simply takes the difference of the constituent random variables. Consequently:

$$\mathrm{E}(X - Y) = \mathrm{E}(X) - \mathrm{E}(Y). \tag{4.79}$$

Computing the expected value of the marginal distribution, one then obtains:

$$\mathrm{E}(c^*) = cq^k. \tag{4.80}$$

from which the mean of the difference of two such variables follows trivially. The variance of the difference between the two variances is complicated by the fact that the distributions are not independent and can co-vary. The variance of the difference of two, non-independent random variables, c_k, c_b is given by

$$\mathrm{Var}(c_\mathrm{k} - c_\mathrm{b}) = \mathrm{Var}(c_\mathrm{k}) + \mathrm{Var}(c_\mathrm{b}) - 2\mathrm{Cov}(c_\mathrm{b}, c_\mathrm{k}). \tag{4.81}$$

The variances can be computed from the marginal distribution in Equation 4.77 and evaluate to:

$$\mathrm{Var}(c^*) = 1 - q^{2k}. \tag{4.82}$$

Which leaves the covariance to be calculated. Clearly, if the two distributions are totally co-dependent, with the bit-distance between the two variables being zero, the total variance should go to zero. So with $q^k = 1$ and $q_\mathrm{b} = q_\mathrm{k}$, the covariance should then be equal to $1 - q_\mathrm{b}^{2k}$. This works for a covariance of $q^k - q_\mathrm{b}^k q_\mathrm{k}^k$, has the appropriate symmetry when interchanging q_b and q_k. This formula can also be verified numerically for finite but large n. Therefore, the variance of the difference is given by

$$\mathrm{Var}(c_\mathrm{b} - c_\mathrm{k}) = 2 - q_\mathrm{b}^{2k} - q_\mathrm{k}^{2k} - 2q^k + 2q_\mathrm{k}^k q_\mathrm{b}^k \tag{4.83}$$

and the problem factor can be simply written as per Equation 4.43.

Arbitrary depth

To calculate the problem factor at a higher depth, a more involved calculation of the variance of the resulting exponentiated random variable in the problem factor must be performed. While the calculation of the mean value is not complicated by higher depth, more covariances must be considered.

For a random variable:

$$X = \sum_i \gamma_i (c_{\mathrm{b},i} - c_{\mathrm{k},i}), \tag{4.84}$$

the mean is simple to calculate:

$$\mathrm{E}(X) = \sum_i \gamma_i (\mathrm{E}(c_{\mathrm{b},i}) - \mathrm{E}(c_{\mathrm{k},i})), \tag{4.85}$$

and the variance would be given by:

$$\begin{aligned}\mathrm{Var}(X) = &\sum_i \gamma_i^2 (\mathrm{Var}(c_{\mathrm{b},i}) - \mathrm{Var}(c_{\mathrm{k},i})) \\ &+ \sum_{i<j} \gamma_i \gamma_j \left[\mathrm{Cov}(c_{\mathrm{b},i}, c_{\mathrm{b},j}) + \mathrm{Cov}(c_{\mathrm{k},i}, c_{\mathrm{k},j})\right] - \sum_{i \leq k} \gamma_i \gamma_j \left[\mathrm{Cov}(c_{\mathrm{b},i}, c_{\mathrm{k},j})\right], \end{aligned} \tag{4.86}$$

where the covariances depend on the distances between the incident strings. The calculation of which grows as $O((2p)^2)$ and therefore can be efficiently computed regardless of the depth.

As every term in the mean and variance depends on a distance raised to the power k, the mean and variance of the k-spin model can be seen to tend to the values determined for the random cost model, for example, the mean value at depth 1, of cq^k, rapidly becomes zero for all non-zero values of q in the limit of large k, likewise, the mean value of the conditional pdf for the random cost model is a Dirac delta function centered at zero for non-unity values of q, that is, zero weight shifts. However, the random cost model retains the symmetry in that applying an n-bit shift in the random cost model results in a randomized state, whereas, in the high-k-spin model this $\mathbb{Z}^2$, all bit-flip symmetry is maintained regardless of k, with flipping all bits either resulting no change for even k or multiplication by -1 for the case of odd-k models.

4.7.3 Free Spins in a Random Magnetic Field

After considering Gaussian correlation models generally, we can focus on the simplest example, that of the $k = 1$ case, corresponding to free spins situated in a randomly varying magnetic field. This problem type is one which, of all k-spin models, has a maximal structure with respect to the single-qubit basis, while still taking energy levels that are randomly distributed[3]. Namely, if one flips a bit, the value of the cost function changes by a small amount, rather than the total randomization that occurs in the case of the random cost model.

This model is simple to solve analytically as the QAOA circuit does not induce any entanglement between qubits. Therefore, it does not represent an interesting problem to solve on a quantum computer but can be studied as a limiting case of strong, Gaussian correlations in the cost function. The fact that the model is analytically solvable makes it a good test case for the method of QAOA performance via statistics, as the QAOA expected value landscape can be efficiently computed on a classical computer.

The problem is defined with

$$\hat{H}_{\mathrm{P}} = \sum_i g_i Z_i, \tag{4.87}$$

[3]Of course, problems could exist that are extremely structured with respect to the X driver, in the example of spins in a uniform magnetic field, QAOA will always be able to find a ground state at depth 1. Interesting problems, however, should be less structured.

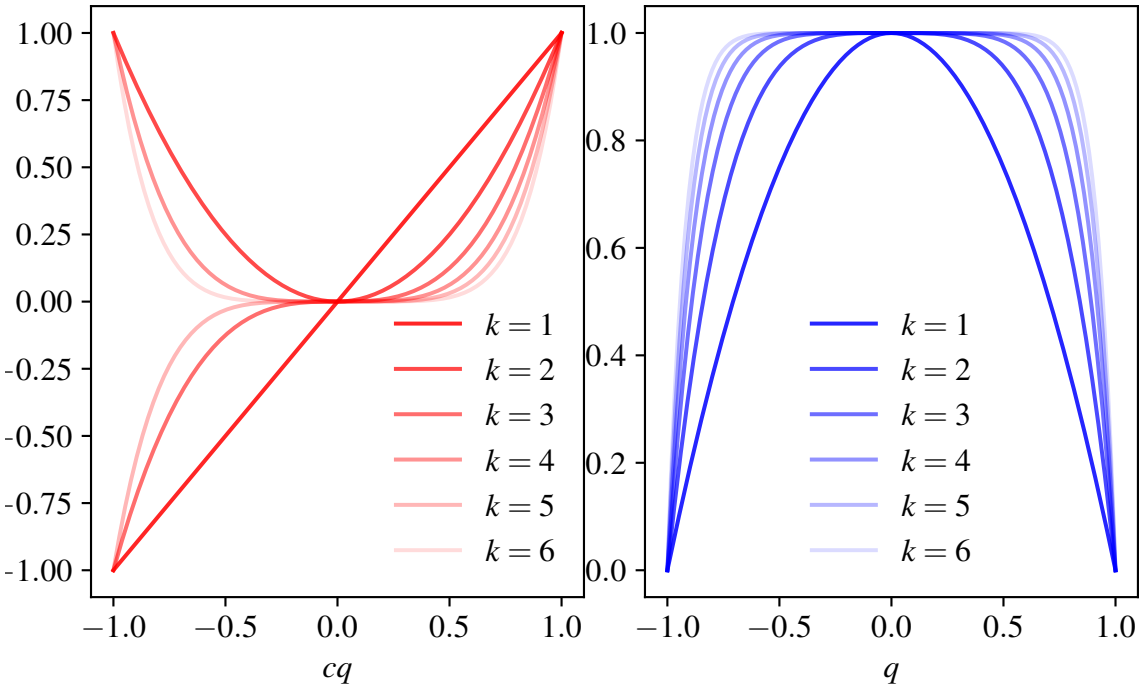

Figure 4.3: The mean (red) and variance (blue) of the cost function value of a configuration having started at cost c and undergone a number of bit flips determined by the variable q. As the number of bit flips tends to $n/2$, the variance of the resulting cost increases to a maximum, and the mean value of the resulting cost tends to zero. In the limit of large k, the mean is non-zero only when $|q| = 1$ and the variance tends to 1 for all values $|q| \neq 1$.

where $g_i \sim \mathcal{N}(\mu = 0, \sigma^2 = 1/n)$ which results in an overall pdf that is centered on zero and with unit variance. Random sampling from this problem can therefore sample solutions with value c with likelihood given by:

$$f(c) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\frac{-c^2}{2}} \text{ where } \int_{-\infty}^{\infty} f(c)\,\mathrm{d}c = 1. \tag{4.88}$$

Which is identical to that of the random cost model. The QAOA solution quality expectation value is possible to compute for this problem since it is a separable problem with no interactions between qubits. Simply considering the single-qubit Hamiltonian

$$\hat{H} = gZ, \tag{4.89}$$

we can write down the single qubit, $p = 1$ QAOA problem Hamiltonian expectation as:

$$\langle +| \mathrm{e}^{-i\gamma Z} \mathrm{e}^{-i\beta X} gZ \mathrm{e}^{i\beta X} \mathrm{e}^{i\gamma Z} |+\rangle = g \sin 2\beta \sin 2g\gamma \tag{4.90}$$

Which is the cost for a single spin. To compute the expected value of the QAOA problem Hamiltonian for n spins, we can sum many of such terms for differing values of g. Take g, as in Equation 4.87 to be normally distributed with variance $1/n$. We can find an expected value of the cost per spin. Therefore, we can obtain a value of the per-spin objective function and then multiply this by the number of spins to

obtain the total expected QAOA solution quality. As such,

$$\mathrm{E}[g \sin 2\beta \sin 2g\gamma] = \sin(2\beta) \int_{-\infty}^{\infty} g f(g) \sin(2g\gamma)\, \mathrm{d}g, \tag{4.91}$$

of which the factor of $\sin(2\beta)$ is separable, with the remainder, depending on γ, is a sine transform of $gf(g)$, otherwise expressible as the a sum of two Fourier transforms

$$\int_{-\infty}^{\infty} g f(g) \sin(2g\gamma)\, \mathrm{d}g = \frac{-\partial_\gamma}{4} \left[\int_{-\infty}^{\infty} f(g) \mathrm{e}^{2ig\gamma} \mathrm{d}g + f(g) \mathrm{e}^{-2ig\gamma}\, \mathrm{d}g \right]$$
$$\frac{1}{2} \frac{\mathrm{d}}{\mathrm{d}\gamma} \mathrm{e}^{-2\sigma^2\gamma^2} = 2\sigma^2 \gamma \mathrm{e}^{-2\sigma^2\gamma^2}. \tag{4.92}$$

Differentiating this to find the maximal value, one obtains:

$$\gamma^* = \pm \frac{1}{2\sigma} \tag{4.93}$$

from which one multiplies by n and substitutes the optimal parameters to obtain an objective function value of:

$$E_1(\gamma^*, \beta^*) = \sqrt{\frac{n}{\mathrm{e}}}. \tag{4.94}$$

This means that for this problem, as the number of qubits increases, the performance increases as $\sqrt{n}$, even though the distribution of the overall objective function remains static. This demonstrates a case in which QAOA can exploit problem structure. With a Grover driver that is not specific to the problem Hamiltonian, the best one would hope to attain would be an expected value that is static and approaching a terminal value with increasing n, since in Chapter 3 we have demonstrated that there is no dependence on n possible.

This single-qubit model is also treated in [117], however, in which the scaling is considered differently. Importantly, a situation in which each qubit has uniform Gaussian coupling is used, to calculate the average energy per spin. In this work, to force comparison to Grover drivers and unstructured problems, we consider the case in which the overall pdf is static in n.

Problem factors at $p = 1$

For this model, to compute the relevant moments, we simply need to take the formulae obtained in Equations 4.80 and 4.83. As such, the value of a cost function c^* starting at a state with value c and applying a shift with weight described by q, the mean value c^* that results is cq, with variance $1 - q^2$.

To evaluate the problem factor for this problem at $p = 1$, we must compute the difference of two random variables derived via separate sets of bit flips as in the previous section. Considering a start point c and two sets of shifts s_b, s_k. These shifts have weights w_b, w_k. From these weights come three weight, or distance parameters, q_b, q_k, q. Where the first and second encode the weights of the permuting strings on the bra- and ket-related terms. The third represents the bit-distance between the shifts k_b, k_k, with $w_\mathrm{ind} = k_\mathrm{b} \oplus k_\mathrm{k}$ and therefore:

$$q = 1 - \frac{2w_\mathrm{ind}}{n}, \tag{4.95}$$

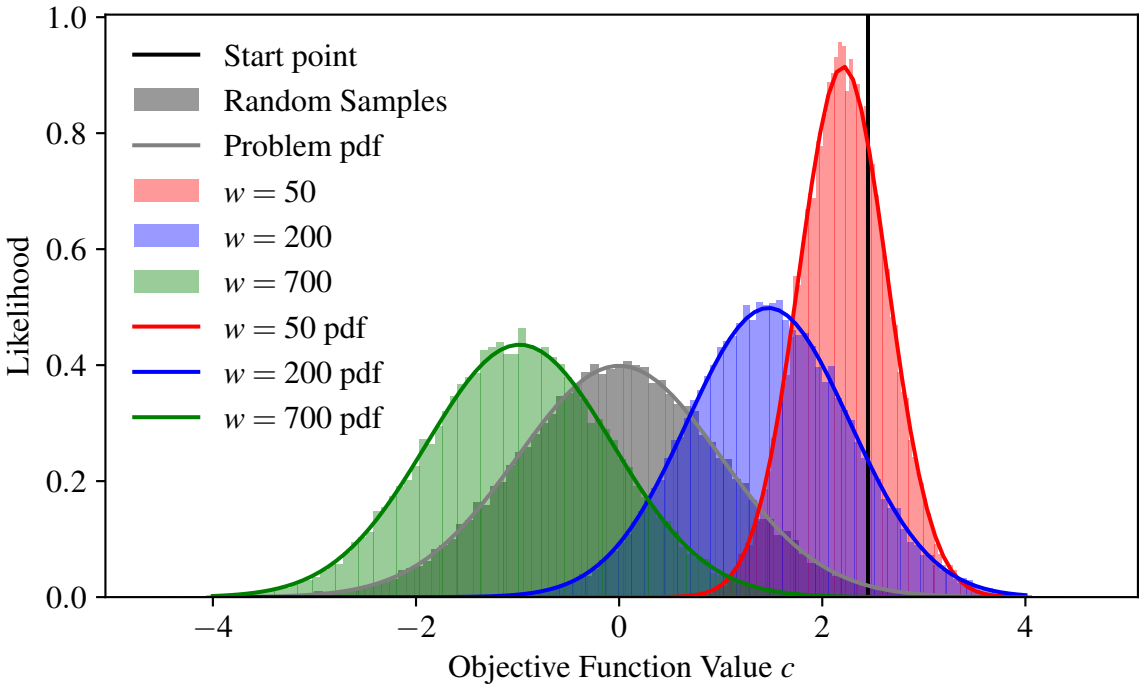

Figure 4.4: The statistics of a 1000-bit instance of the Gaussian-distributed single-Z model and its conditional statistics under random 50-bit 200-bit, and 700-bit flips. The black line denotes the value of a starting string z with histograms demonstrating the probability of finding a string of a given value after a number of random bit-flips are applied. The mean and variance of the analytical distributions (solid lines) match the numerical observations for mean and variance (transparent histograms) of a cost function starting at a string with a high value of around 2.5 and having 50, 200, or 700 bits flipped out of a possible 1000 bits. One observes that as more bits are flipped, the distribution tends closer to, and reaches the distribution attained through random sampling, before decreasing in variance as the fraction of flipped increases past 50%.

where $\oplus$ denotes the bit-wise XOR of the shifts, essentially the sum of weights of the two shifts, minus the weight of shifts that are common to both shifts. One considers two random variables, denoted c_b, c_k. From which the expected value and variance of the difference of such variables, multiplied by the problem parameter γ must be evaluated. This is, again, the procedure in Section 4.7.2, with k set to 1.

$$\mathrm{E}\left[c_\mathrm{k} - c_\mathrm{b}\right] = \mathrm{E}\left[c_\mathrm{k}\right] - \mathrm{E}\left[c_\mathrm{k}\right] = (q_\mathrm{b} - q_\mathrm{k})c = \frac{2\Delta w}{n}c \tag{4.96}$$

where Δw is the difference in weight of the two permuting strings. For the variances, one obtains:

$$\mathrm{Var}(c_\mathrm{b} - c_\mathrm{k}) = 2 - q_\mathrm{b}^2 - q_\mathrm{k}^2 - 2q + 2q_\mathrm{k}q_\mathrm{b} = \frac{4w_\mathrm{ind}}{n} - \frac{4\Delta w}{n}. \tag{4.97}$$

A numerical simulation of the resulting distribution is plotted in Figure 4.4 alongside the analytically determined Gaussian distributions, demonstrating that the two are in agreement for large system sizes of 1000 bits. For these computed values, the

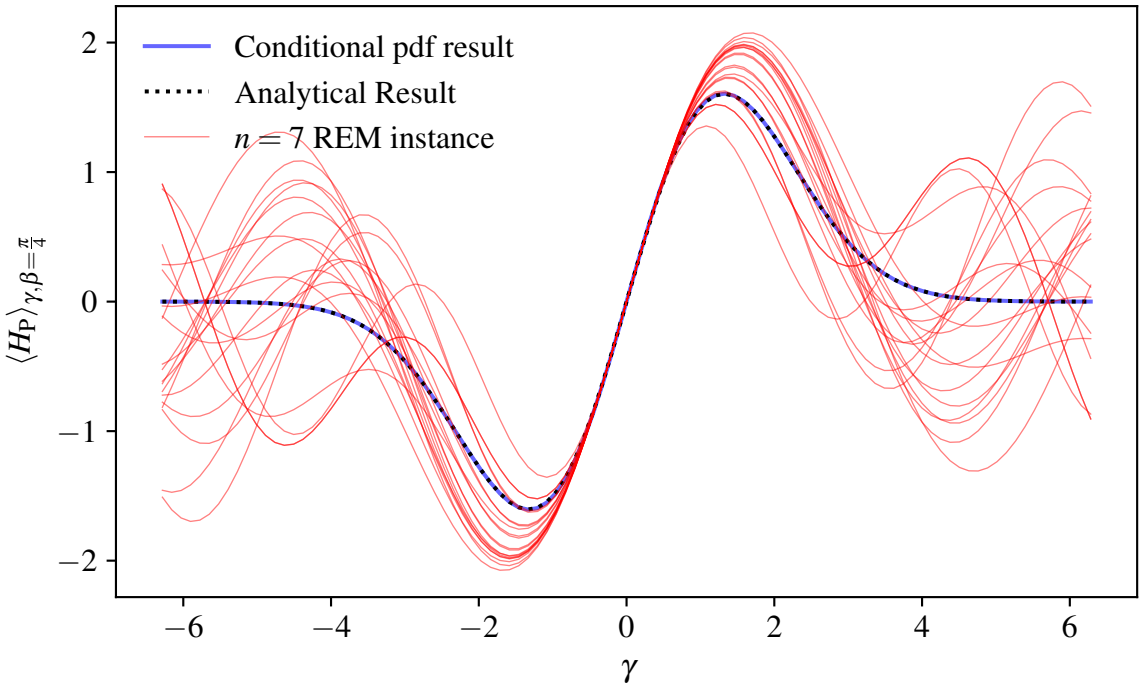

Figure 4.5: The energy landscape of QAOA states prepared for the average case random single-qubit-energy problem alongside 20 random 10-bit instances and the analytical average case.

problem factor can then be written using Equation 4.43 as:

$$P(k_\mathrm{b}, k_\mathrm{k}) = i\frac{2\Delta w}{n}\gamma \exp\left\{-\frac{1}{2}\left(\frac{2\Delta w}{n}^2 + \left(4\frac{w_\mathrm{ind}}{n} - 4\frac{\Delta w^2}{n^2}\right)^2\right)\gamma^2\right\}. \tag{4.98}$$

Comparing this problem factor to an analytical formula for the expectation shows that this analysis holds and the QAOA landscape, as a function of the variational parameters, is identical using this formula to that determined via the use of Equation 4.92. That this statistical treatment results in identical values can be seen in Figure 4.5, in which the result using the asymptotic statistics of the model, is compared to the landscapes obtained via an analytical shortcut.

This case of statistically-defined, non-interacting spins, is, of course not a hard problem for QAOA to approach. It does, however, verify that the method of using problem statistics, calculating problem factors via conditional probabilities, rather than being an approach determined by individual instances, provides the correct answer in a test case, and gives a succinct analytical optimal solution.

Problem factors at higher depth

The problem factors for higher depth can be evaluated in a similar fashion, however, their evaluation is complicated by the exponential number of different shifts to consider in p and n. The analytically determined average QAOA performance is simple to calculate for greater depth, with, however, an increasing number of sinusoidal terms to evaluate and integrate over. While not tested in this thesis, this test case provides a good method for verification of the higher-depth performance of QAOA, in the problem ensemble average frame.

4.8 Line Mixers

A second obvious context in which structure might exist, and be exploited by QAOA, is that structure with respect to the L_1 norm on the labels of a combinatorial optimization problem. Under such a norm the driver-metric induced distance between two states is simply the absolute value of the difference of their labels. Such a norm corresponds to what we describe as the line mixer, which occurs in the niche context of one-hot encoded variables. Alternatively, ignoring a need to motivate the investigation of the mixer with a real-world use case, the line mixer provides an alternative context to probe QAOA performance, in which the metric for the distance between states is simplified and perhaps more intuitive. Line mixers induce the arithmetic distance metric on the space of possible solutions, favoring structure that would be uncommon in typical QAOA problems expressed in qubit operators, but that which would be easier to identify visually, by simply plotting values of a cost function against their label z. Structure in this case can be thought of as states with similar labels, as measured by their absolute difference, having values of the cost function that are similar.

4.8.1 Driver Factors in Line Mixers

In this section, we will analyze the performance of a restricted QAOA driver, the line mixer, on statistically defined problems. In this case, we use the driver Hamiltonian:

$$\hat{H}_{\mathrm{D}} = \sum_{|i-j|=1} |i\rangle \langle j| + |j\rangle \langle i| \tag{4.99}$$

Exponentiating this, one obtains the driver unitary:

$$\begin{aligned} \hat{U}_{\mathrm{D}}(\beta) = \mathrm{e}^{i\beta\hat{H}_{\mathrm{D}}} = \exp\left(i\beta \sum_{z} |z+1\rangle \langle z| + |z\rangle \langle z+1| \right) \\ = \sum_{d\in[-N,N]} f(d) \sum_{z} \left[\, |z\rangle \langle z+d| \, \right], \end{aligned} \tag{4.100}$$

where we are assuming that addition on the state labels applies periodic boundary conditions. That is, for symmetry, the driver is connected at the ends, resulting in a driver symmetry in which permuting all values of the cost function by adding a number to their index, the QAOA expected solution quality is preserved. We can calculate the pre-factor $f(d)$ via:

$$f(d) = \langle z| \, \hat{U}_{\mathrm{D}}(\beta) \, |z+d\rangle \tag{4.101}$$

where the pre-factor must take terms from either an odd or even number of steps. To obtain $f(d)$, one must add together all of the paths in the expansion that result

in a term corresponding to a distance of d. So, for $f(0)$ we have

$$\begin{aligned} f(0) &= 1 + \langle x| \left[i\beta \sum_z |z+1\rangle \langle z| + |z\rangle \langle z+1| \right] |x\rangle \\ &+ \langle x| \left[i\beta \sum_z |z+1\rangle \langle z| + |z\rangle \langle z+1| \right] \cdot \left[i\beta \sum_{z'} |z'+1\rangle \langle z'| + |z'\rangle \langle z'+1| \right] |x\rangle + \dots \\ &= 1 + 0 - 2\frac{\beta^2}{2!} + 0 + \binom{4}{2}\frac{\beta^4}{4!} + \cdots \approx \sum_{k \text{ is even}} \binom{k}{\frac{k}{2}} \frac{i^k \beta^k}{k!}. \end{aligned} \tag{4.102}$$

Which, if the sum is taken to infinity, is a zeroth order Bessel function of the first kind, denoted J_0. We can write down the general expression for even values (at low β) as:

$$f(d) \approx \sum_{k \text{ is even} \leq d} \binom{k}{\frac{k+d}{2}} \frac{i^k \beta^k}{k!}, \tag{4.103}$$

and odd values as:

$$f(d) \approx \sum_{k \text{ is odd} \leq d} \binom{k}{\frac{k+d+1}{2}} \frac{i^k \beta^k}{k!}. \tag{4.104}$$

To evaluate the binomial coefficients at large k, we can use Sterling's formula, or a more accurate approximation, since the values for which we take combinations could grow large, complicating their calculation.

4.8.2 Expected QAOA Solution Quality for Line Mixers

The treatment of the problem factor for line mixers is very similar to the case of the X-driven mixer, however with a different structure of shifts. First, we consider a random cost model, then suggest the forms that models with synergistic structure might take.

Depth 1

At depth 1, the expectation value of the QAOA problem Hamiltonian is:

$$\begin{aligned} E_1(\gamma,\beta) = \langle \hat{H}_\mathrm{P} \rangle_{\gamma,\beta} &= \langle +| \hat{U}_\mathrm{P}^\dagger(\gamma) \hat{U}_\mathrm{D}^\dagger(\beta) \hat{H}_\mathrm{P} \hat{U}_\mathrm{D}(\beta) \hat{U}_\mathrm{P}(\gamma) |+\rangle \\ &= \frac{1}{N} \sum_{z_{-1},z_0,z_1} C(z_0) \mathrm{e}^{-i\gamma C(z_{-1})} \mathrm{e}^{i\gamma C(z_1)} \langle z_{-1}| \, \mathrm{e}^{i\beta \sum_{z'} |z'+1\rangle\langle z'|+|z'\rangle\langle z'+1|} \\ &\qquad \cdot |z_0\rangle \langle z_0| \, \mathrm{e}^{i\beta \sum_{z''} |z''+1\rangle\langle z''|+|z''\rangle\langle z''+1|} \, |z_1\rangle \\ &= \sum_{d_\mathrm{bra}, d_\mathrm{ket}} f(d_\mathrm{bra})^* f(d_\mathrm{ket}) \cdot \sum_z C(z) \mathrm{e}^{-i\gamma[C(z+d_\mathrm{ket}) - C(z+d_\mathrm{bra})]} \end{aligned} \tag{4.105}$$

to which one can apply the treatment of instances of $C(z)$ as random variables, allowing one to transform the sum over z to an integral in the limit of large n.

Arbirtary depth

At arbitrary depth, the expectation value can be calculated as follows, firstly, one writes:

$$\begin{aligned}
E_p(\vec{\gamma},\vec{\beta}) &= \langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\gamma},\vec{\beta}} \\
&= \langle + | \hat{U}_{\mathrm{P}}^{\dagger}(\gamma_1)\hat{U}_{\mathrm{D}}^{\dagger}(\beta_1)U_{\mathrm{P}}^{\dagger}(\gamma_p)\ldots\hat{U}_{\mathrm{D}}^{\dagger}(\beta_p)\hat{H}_{\mathrm{P}}\hat{U}_{\mathrm{D}}(\beta_p)\hat{U}_{\mathrm{P}}(\gamma_p)\ldots\hat{U}_{\mathrm{D}}(\beta_1)\hat{U}_{\mathrm{P}}(\gamma_1)|+\rangle \\
&= \frac{1}{N}\sum_{\vec{d}}\prod_{p'=1}^{p}[f(d_{p'})]\sum_{\vec{z}}\sum_{\vec{x}} C(z_0)\mathrm{e}^{i\sum_{i\in[-p,p]}\gamma_p C(z_p)} \\
&\cdot \langle z_{-1}|x_{-1}\rangle\langle x_{-1}+d_{-1}|z_{-2}\rangle\langle z_{-2}|x_{-2}\rangle\langle x_{-2}+d_{-2}|\ldots|z_{-p}\rangle\langle z_{-p}|x_{-p}\rangle\langle x_{-p}+d_{-p}| \\
&\cdot |z_0\rangle\langle z_0|z_{-p}\rangle\langle z_{-p}|x_{p-1}+d_{p-1}\rangle\langle x_{p-1}|\ldots|z_2\rangle\langle z_2|x_1+d_1\rangle\langle x_1|z_1\rangle
\end{aligned} \tag{4.106}$$

in which the bra-kets evaluate to Dirac deltas and collapse all sums with the exception of a single index z. Each time we obtain a bra-ket of the form $\langle z+d|z'\rangle$, one makes the substitution $z \to z-d$ to obtain a simple Kronecker delta. This results in:

$$E_p(\vec{\gamma},\vec{\beta}) = \sum_{\vec{d}}\left[\prod_{p'=1}^{p} f(d_{p'})\right]\frac{1}{N}\sum_{z} C(z_0)\mathrm{e}^{i\sum_{i=1}^{p}\gamma_p[C(z_p-s_i)-C(z_p-s_{-i})]} \tag{4.107}$$

in which the shifts s_i are defined such that

$$s_i = d_i + d_{i-1} + \ldots + d_1,\ \ s_{-i} = d_{-i} + d_{-i+1} + \ldots + d_{-1}. \tag{4.108}$$

This is similar to the expression derived for X-driven QAOA albeit following the structure of additive distance between state labels. This formula can be treated with the substitution of random variables in the place of values of $C(z)$ similarly to the other cases.

4.8.3 Problems for Line Mixers

Taking intuition from the treatment of the X driver, problems for line mixers might be expected to perform well on should have a structure that results in the value of the objective function decaying to its average value slowly as steps increasing the distance are applied. An extreme example in the case of this is the case of the RCM in which, on average, the value of the objective function decays to the mean after a single, non-zero step.

Random Cost Model

The random cost model, used in the cases of the Grover and X-driven QAOA can also be considered in the context of the line mixer. Since the $p = 1$ performance of the Grover and X driver have been found to be equivalent in this problem, it is informative to determine whether this is a general feature, or whether line mixers perform differently on this problem.

Depth 1

for $p = 1$, one must compute the random variable X which is the difference between two random variables depending on two shifts. The calculation in Equation 4.43 can also be used here, as the quantities are Gaussian. which means the problem factor is

$$i\gamma \mathrm{e}^{-\gamma^2}, \tag{4.109}$$

if one shift is zero and the other is non-zero and zero otherwise. This means the full $p = 1$ expectation can be expressed as:

$$E_1(\gamma, \beta) = \sum_{d>0} f(0)^* f(d) i\gamma \mathrm{e}^{-\gamma^2} + c.c. \tag{4.110}$$

In which the γ-dependent part is separable and identical to the single-X case. Therefore, one must evaluate the sum:

$$\sum_{d>0} i f(0)^* f(d) + c.c. \tag{4.111}$$

In the previous case, it sufficed to calculate the sum over all distances, which gave a simple substitutable expression. In the line mixer case, we can write a similar expression with:

$$\begin{aligned}\sum_d f(d) &= \sum_d \langle x| \hat{U}_{\mathrm{D}}(\beta) |x + d\rangle \\ &= \sum_d \langle x| \exp\left(i\beta \sum_z |z\rangle \langle z+1| + |z+1\rangle \langle z| \right) |x + d\rangle = \mathrm{e}^{2i\beta}.\end{aligned} \tag{4.112}$$

Therefore, by subtracting the zero distance value, and taking the sum to infinity for the limit of large problem sizes, one obtains a $p = 1$ expected value of:

$$i f(0)^* \sum_{d>0} f(d) + c.c = J_0(2\beta) \cdot \left[\mathrm{e}^{2i\beta} - J_0(2\beta)\right] + c.c = 2\sin(2\beta) J_0(2\beta). \tag{4.113}$$

We can numerically optimize this formula to determine an optimal β parameter and expected performance in the limit of large problem sizes. One finds, that, as in the case of the Grover driver, but in contrast to the case of X-driven QAOA, the performance and also the optimal angle for line mixers is asymptotically independent of the number of qubits. This independence occurs since the maxima occur at a value of β which corresponds to low order terms, and therefore, the structure cannot depend on the total size. We find an optimal driving angle of

$$\beta^* \approx 0.516, \tag{4.114}$$

with the expected value of samples from the QAOA circuit being

$$E_1(\gamma, \beta) \approx \frac{1.289}{\sqrt{2e}}. \tag{4.115}$$

Which under-performs the Grover and X-driven versions, attaining a mean solution quality on average $\approx 64\%$ as high as in the aforementioned cases.

Higher-depth solution qualities could be investigated for this problem and driver, however, for this driver and problem combination, properties approach the infinite-size limit rapidly. The utility of calculating the performance of arbitrary depth QAOA circuits at infinite size is then questionable, when modest system sizes, with results averaged over many instances, obtain near-identical values.

4.9 Conclusion

To conclude the chapter, we have applied the procedure of Chapter 3 to two, more general contexts. The first of these contexts is that of the standard driver used in QAOA, the single-qubit-X driver. The second, a niche driver that serves to elucidate structure that can be observed simply by plotting the energies of a problem by their label, is that of the line mixer. By applying the method of expressing QAOA expectation values in terms of asymptotic statistical properties of a problem, we determine that, while Grover QAOA of Chapter 3 depends only on how the energies are distributed, other drivers require more information. In particular, the expected QAOA solution quality requires the conditional probability of measuring a state of a given cost, supposing it is a certain distance from a state of initial cost, where distance is defined by a driver-determined metric. For X-driven QAOA, such a metric is the Hamming distance, while for line mixers, the relevant metric is the arithmetic distance between state labels. This work opens up the possibilities for a workflow for QAOA in which statistical properties are determined from a problem via an efficient sampling procedure, then, expected QAOA solution quality calculated, with no use of a quantum processor. Such capabilities would allow for few-shot use of a large quantum computer to find good solutions to combinatorial optimization problems without the need for on-device optimization procedures, and, benchmarking of QAOA performance without access to physical devices.

To test this method of determining performance from conditional cost function values, we used the test case of the random cost model. As the asymptotic performance of QAOA on this model could be calculated for the Grover, X, and line driver, we could evaluate that Grover and X drivers perform identically at $p = 1$, with line drivers underperforming these. This suggests, that for problems with vanishing structure, higher-weight terms might improve performance for line mixers, but are unlikely to help QAOA performance for X-driven QAOA. This conclusion is apparent, since adding higher distance terms to a driver takes one closer to the Grover driver, with equality if all distances are used.

We demonstrated how the method could be used for k-spin models, which contain Gaussian correlations in the cost function that decay as a function of the distance from a reference state. One might expect, that given combinatorial optimization problem, to solve, one could determine problem statistics and match such a problem to a value of k. This could provide either angles for finding good solutions with QAOA, or at a minimum, angles to start an on-device optimization of the expected value.

The answer to the question of whether QAOA-type algorithms can outperform other, potentially classical, heuristic methods for approximately solving combinatorial optimization problems might be expected to come in the form of a combination of Grover-type speedup, and exploitation of problems structure, via the driving Hamiltonian. Therefore, the question of how QAOA resolves this problem structure, as this chapter attempts to explain, takes us some way to understanding how and where QAOA might be useful.

Chapter 5

Approximating QAOA with Digital-Analog Interactions

Quantum algorithms that may run on quantum computers in the near term (5 to 10 years) must run on imperfect devices subject to environmental noise, imperfect gates, and a lack of correction mechanisms for errors. As such, greater performance may potentially be attained from co-designed, algorithm-specific devices. This chapter, alongside the chapter that follows, is an attempt to imagine what such a device might look like for performing a wide variety of QAOA problems, simplifying device design at the expense of additional sources or errors.

5.1 Paradigms of Quantum Computing

The usual approach in quantum computing is to create a device with a universal gate set, for which, many possibilities are known. Universality is achieved by the capacity to implement a basic gate set from which any gate can be implemented. The most commonly considered choice for a device level gate set would be an interacting gate such as the CNOT, CZ or iSWAP gates alongside the full spectrum of single-qubit gates offered by single-qubit rotations on two axes. For encoded logic gates in error correcting codes such as the surface code, a discrete gate set of Clifford gates—the set of gates effecting permutations of Pauli operators—plus T-gates—the $\pi/4$ Z-rotation gate—can be used to achieve universality on qubits encoded within a topological error correcting code. The digital-analog scheme provides a way to make use of an alternative gate set, one in which single-qubit gates are readily available, but interacting gates are consolidated to a single resource interaction, itself shaped by single-qubit gates. As opposed to the gate-based paradigm of quantum computing, this is named the digital-analog paradigm.

5.2 Spin Echo and Dynamical Decoupling

The digital-analog paradigm, or scheme, traces its roots to quantum computing with nuclear magnetic resonance, and spin echo processes. Spin echo is a technique to refocus a large number of spins using a resonant driving field when there is uncertainty as to the procession frequency of the spins. This technique exploits the effective time reversal caused by a single-qubit-X π pulse. Considering a one-qubit

system with the Hamiltonian

$$\hat{H}(t) = \alpha Z + \beta(t) X, \tag{5.1}$$

where α is unknown and $\beta(t)$ is a controlled parameter. If $\beta(t) = 0$ time evolution effected by $\hat{H}(t)$ is

$$\hat{U}(t) = \mathcal{T} \exp\left\{ i \int_{t=0}^{t} \hat{H}(t) \right\} = \mathcal{T} \exp\left\{ i \int_{t=0}^{t} \alpha Z + \beta(t) X \right\}. \tag{5.2}$$

The idea in spin echo sequences is that $\beta(t)$ can be set such that the result of the time evolution is insensitive to the value of α. Increasing the time between spin-echo pulses allows one to calculate the T_2 decoherence time of a system, which is the time taken for a system to dephase. Such procedures are the simplest use of average Hamiltonian theory, in which one designs pulse sequences, such that, when one time averages the applied Hamiltonian, the result is a simpler, desired Hamiltonian [118]. Another use of average Hamiltonian theory is that of dynamical decoupling [119], in which a similar procedure aims to average interactions with the environment to zero.

The digital-analog scheme introduced in this chapter is a generalization of this to a many-qubit case, setting control pulses such that unwanted interactions are averaged to zero. Though the scheme in this work requires significantly more complex sequences of pulses to effect the average Hamiltonian required, the result is a large number of possible average Hamiltonians that can be effected, by one, generic resource.

5.3 Attribution & Copyright

The idea for this project occurred when Ana Martin gave a talk in Saarland and the idea to apply the ideas presented there to QAOA was immediately obvious to Thorge Mueller and myself. The document was written by myself, with the exception of Section 5.12 which was written entirely by Prof. Wilhelm-Mauch. Software development and the execution of large-scale simulations for the paper were performed by Thorge Mueller, in the thesis of whom this work also appears. Parts of this chapter are reprinted (excerpt, abstract, figures) with permission from "David Headley, Thorge Müller, Ana Martin, Enrique Solano, Mikel Sanz, and Frank K. Wilhelm, Phys. Rev. A 106, 042446", Copyright (2022) by the American Physical Society.

5.4 Introduction

Quantum computing is entering an era in which classical computers cannot simulate the behavior of programmable quantum computers [54]. In this new era of quantum information processing, it is likely that the first algorithms that will be useful for solving computational problems will be *heuristic* in nature. These algorithms come without provable performance guarantees provided by the likes of Shor's factoring algorithm [18] or the Grover search algorithm [20] but are encouraged by strong motivation from classical algorithm research, in that the most effective algorithms for

solving certain problems classically are often not provably so. At present, two such algorithms are most likely to prove useful in the near term [120]—The Variational Quantum Eigensolver (VQE) [121] and the Quantum Approximate Optimization Algorithm (QAOA) [34] otherwise known as the Quantum Alternating Operator Ansatz [59]. These are *variational* algorithms, using classical optimizers and parameterized quantum circuits to mitigate the effects that errors may introduce on quantum devices making no use of quantum error correction. This work concerns the latter of the two.

The development of QAOA was motivated by a need for algorithms that can run on noisy, pre-error correction devices. Algorithms used on devices of this era will necessarily have a degree of co-design between architecture and algorithm. Work by Rigetti, for example, used a noisy, programmable quantum device to solve a combinatorial optimization problem inspired by the on-device layout of qubits [122]. Following this approach, we extend the work of Parra-Rodriguez et al. on the Digital-analog (DA) paradigm of quantum computation [123, 124, 125], in which a device is designed and operated in the style of a *quantum simulator* with always-on multi-qubit interactions. We show that QAOA is a natural algorithm for this setting. This paradigm, leveraged to minimize errors associated with turning on and off gates on a quantum device, could allow for a simpler design in which only the timing of single-qubit gates must be considered, reducing the control complexity and, therefore, the mechanisms through which environmental noise can corrupt the computing system. The problem of negative interaction times, introduced by the aforementioned scheme, is resolved in this work by exploiting periodicity in time applied to the resource interaction, or problem solved, allowing any two-local problem to be solved with a homogeneous resource interaction and problems exhibiting periodicity (MAX-CUT, MAX-2-SAT) to be solved with heterogeneous resource interactions.

The chapter begins with a mathematical description of the digital-analog computational paradigm and the combination of this scheme with QAOA. The costs of compilation for embedding QAOA within the digital-analog paradigm are examined and the potential hardware platforms on which digital-analog quantum computing for this purpose could be performed are discussed. The performance of DA-QAOA is examined using computational simulations and the errors of the method are analytically bounded. It is finally concluded that while using the digital-analog scheme for QAOA results in additional errors, these errors are not as damaging as in other potential uses of the digital-analog paradigm due to the variational nature of QAOA. The novel result of this chapter is that the combination of DA computation with QAOA is synergistic. For realistic device parameters, errors introduced by DA computation do not adversely affect the performance of QAOA.

5.5 The Digital-Analog Quantum Computational Paradigm

Quantum algorithms can, in general, be separated into two classes: continuous and discrete. At the extreme end of continuous quantum algorithms lie those of quantum simulators, devices fabricated to follow dynamics of interest with, however, no capacity for complicated programmed time evolution [126]. Likewise, continuous algorithms such as the quantum adiabatic algorithm [23, 127] and quantum random

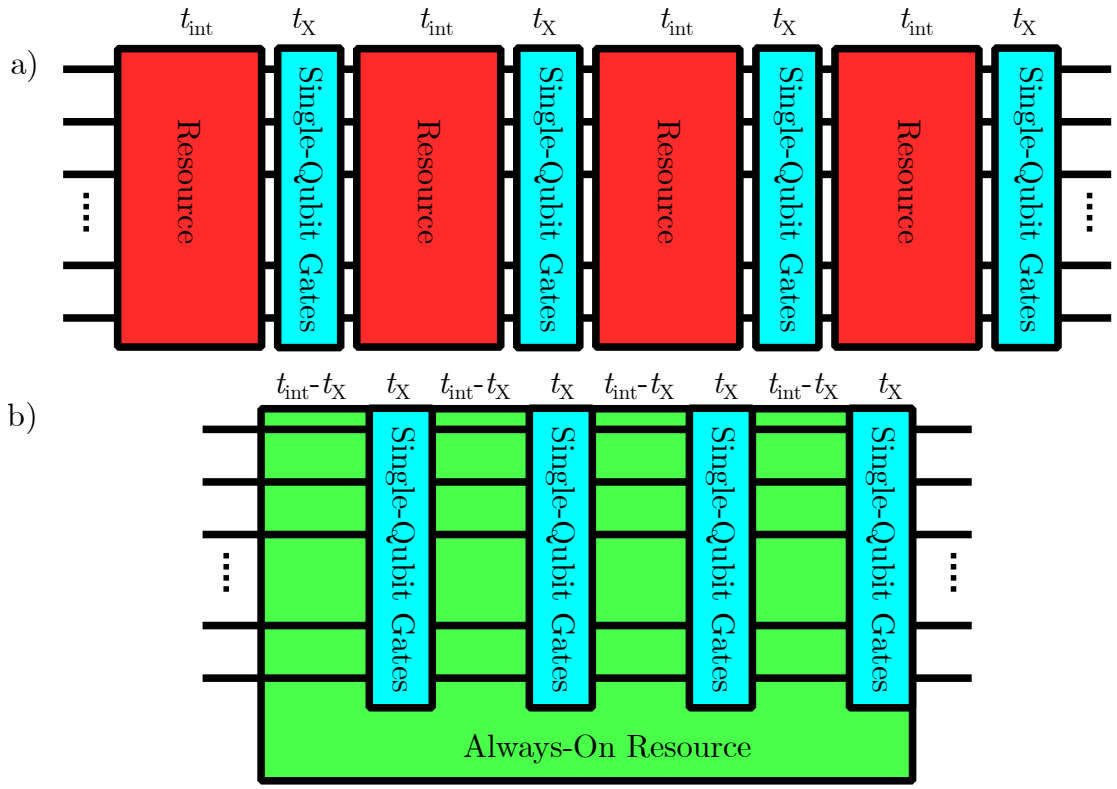

Figure 5.1: The two schemes for digital-analog computation. (a) The stepwise or sDAQC scheme in which a series of programmable *digital* single-qubit gates are applied in alternation with *analog* resource interactions. (b) The always-on or bDAQC scheme in which the resource interaction is never turned off and single-qubit operations are applied in parallel with the resource interactions. Performing the single-qubit operations simultaneously with the resource interaction introduces coherent errors but reduces device control requirements. The first interaction block denoted with the time interval t_0 corresponds to the *idle* block.

walk algorithms [128, 129] make use of a predefined *analog* Hamiltonian with generally limited programmability to solve computational problems. Discrete algorithms, on the other hand, are defined in terms of sequences of *digital* unitary gates. A gate-based quantum computer can perform any discrete, gate-based algorithm, including Trotterized versions of continuous algorithms [130]. Whereas a device designed for continuous quantum algorithms, though may be theoretically universal, will generally only be able to run the restricted set of time evolutions for which they are built—it is, in theory, possible to express any discrete-time quantum algorithm (e.g. Shor's) as a continuous time quantum algorithm, but generally not practical [131]. Devices of analog quantum computation such as quantum simulators can benefit from superior noise resilience characteristics stemming from a reduced requirement to completely control the full dynamics of every qubit [132], as is required in a fully gate-based, digital model.

The DA paradigm [123, 124] is designed to take the best features of both digital and analog quantum computing and has been shown to yield an implementation

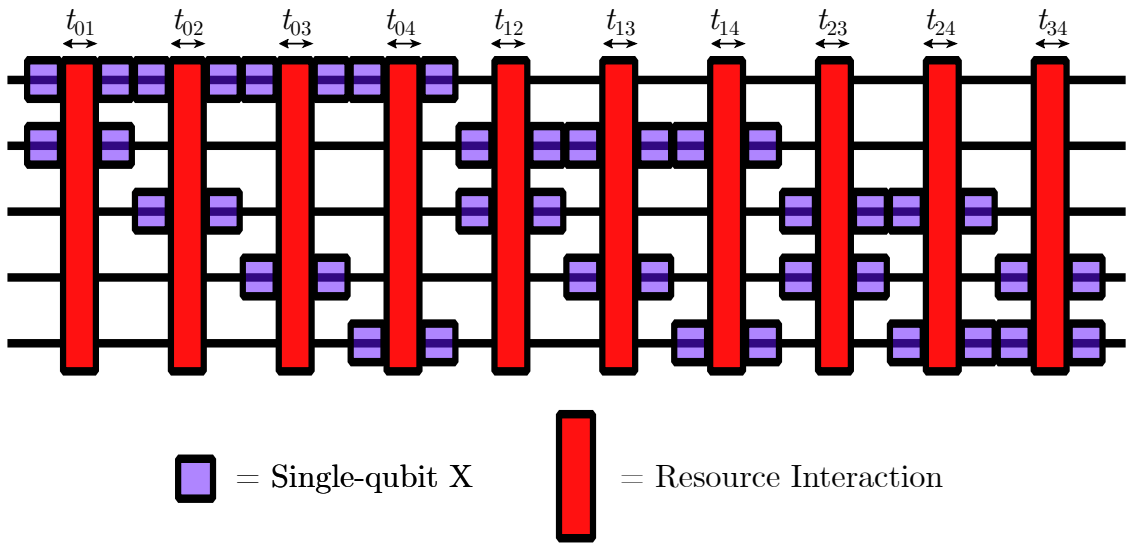

Figure 5.2: A quantum circuit depicting the digital-analog time evolution required to simulate an arbitrary Ising Hamiltonian on 5 qubits. 10 uses of a resource Hamiltonian are required, each surrounded by a unique combination of single-qubit-X operations.

for the quantum Fourier transform and the Harrow-Hassidim-Lloyd algorithm, in which significant advantages over regular digital schemes are demonstrated for reasonable coherent-control error assumptions [125, 133]. The basic premise of the DA scheme is that blocks resembling the time evolution of an analog simulator are performed, punctuated by digital single-qubit operations. This premise is depicted in quantum circuit form in Figure 5.1. The DA paradigm yields two options for its implementation, the *step-wise* scheme (sDAQC) and the *always-on* or *banged* scheme (bDAQC), in which error is introduced to computations due to the non-commutativity of single-qubit operations used simultaneously with an entangling resource interaction. This work proposes the use of an always-on-resource (bDAQC-QAOA) on near-term quantum devices for discrete optimization. To this aim, the step-wise scheme provides a guarantee that if single-qubit gates can be applied sufficiently fast, no errors are introduced and the always-on scheme produces the same state as a standard gate-based implementation.

We consider a context in which we have access to a *resource* Hamiltonian consisting of a sum of all possible interaction ZZ terms between connected qubits j, k on an n-qubit device, each with relative strength r_{jk},

$$\hat{H}_{\text{Resource}} = \sum_{j<k}^{n} r_{jk} Z_j Z_k. \tag{5.3}$$

Where sums in this work over qubit indices start from 1. We use this resource as an interacting resource to simulate problem Hamiltonians required for QAOA. In sDAQC, we assume that our device has the capability for this resource Hamiltonian to be turned on and off, alongside the ability to perform arbitrary single-qubit gates. We can, therefore, alternate our resource interaction with single-qubit gates. In bDAQC we do not assume that the resource Hamiltonian can be turned off

and on, but retain the ability to perform arbitrary single-qubit gates at any time, simultaneously applied alongside the resource interaction.

Let us assume that the resource Hamiltonian available is an all-to-all (ATA) Ising Hamiltonian: where in the *homogeneous* case, $r_{jk} = 1 \; \forall \; j, k$ (the units of r are set such that 1 represents a relevant energy scale for the device). In the simulations and resource estimates performed in this work, we consider only resource Hamiltonians in which no element of r is zero. However, the following procedure does not require an all-to-all connected resource interaction to succeed. Here, we require only that the interactions present in the arbitrary Hamiltonian that we intend to simulate are also non-zero in the resource Hamiltonian. Although this is not necessary in general [134], we assume it for the sake of simplicity. Experimental settings likely to provide such resource Hamiltonians are discussed in Section 5.9, and this discussion is extended by work in Chapter 6.

To use a fixed interaction resource Hamiltonian to simulate the time evolution of an arbitrary spin-glass Hamiltonian, older techniques developed for quantum computing with nuclear magnetic resonance spin systems can be deployed. Named average Hamiltonian theory [135, 136], one can design a sequence of interactions and single-qubit operations such that the time average of such evolutions is identical to that of a Hamiltonian of interest.

There are $n(n-1)/2$ individual degrees of freedom in an arbitrary Hamiltonian to be simulated by our resource. We will, therefore, require $n(n-1)/2$ time intervals over which the resource is applied, each with surrounding single-qubit operations such that each block is linearly independent of the others. In order to select the single-qubit operations with which to surround uses of the resource interaction, one can pick the $n(n-1)/2$ ways one can select two of n qubits, applying X gates to these qubits before and after an application of the resource Hamiltonian [124]. This choice of block-surrounding operations can be seen in Figure 5.2 for $n = 5$ qubits. Between adjacent blocks, single-qubit-X operators will sometimes cancel, reducing the total number of single-qubit gates required substantially.

The unitary evolution we wish to implement is that of an Ising Hamiltonian with arbitrary couplings:

$$\hat{U}_{\mathrm{Arb}}(t) = \mathrm{e}^{i\hat{H}_{\mathrm{Arb}}t} \quad \text{and} \quad \hat{H}_{\mathrm{Arb}} = \sum_{j<k}^{n} g_{jk} Z_j Z_k, \tag{5.4}$$

which we wish to express as a sequence of digital-analog blocks. One notes that the time evolution implemented by a digital-analog block with X operators on qubits a, b is equivalent to that of a constant effective Hamiltonian $X_a X_b \hat{H}_{\mathrm{Res}} X_a X_b$ as

$$(X_a \otimes X_b)\mathrm{e}^{t_{ab}\hat{H}_{\mathrm{Res}}}(X_a \otimes X_b) = \mathrm{e}^{t_{ab}(X_a \otimes X_b)\hat{H}_{\mathrm{Res}}(X_a \otimes X_b)}. \tag{5.5}$$

This immediately follows from the identity:

$$\mathrm{e}^{it\hat{U}\hat{V}\hat{U}^\dagger} = \sum_{k=0}^{\infty} \frac{(it)^k (\hat{U}\hat{V}\hat{U}^\dagger)^k}{k!} = \hat{U}\mathrm{e}^{it\hat{V}}\hat{U}^\dagger, \tag{5.6}$$

valid for any unitary operator U. Using the above, we may write an arbitrary Ising Hamiltonian as a sum of digital-analog effective Hamiltonians, one for each of

$n(n-1)/2$ blocks. Therefore:

$$\hat{H}_{\text{Arb}} = \sum_{j<k}^{n} \sum_{l<m}^{n} t_{lm} r_{jk} X_l X_m Z_j Z_k X_l X_m, \tag{5.7}$$

for some vector of times $\vec{t}$ to be computed. An illustration of this Hamiltonian applied on a 5-qubit device can be seen in Figure 5.2. Using the identity $X_i Z_i \equiv -Z_i X_i$ to commute Pauli-X operators to cancellation one obtains:

$$\sum_{j<k}^{n} \sum_{l<m}^{n} t_{lm} r_{jk} (-1)^{\delta_{lj}+\delta_{lk}+\delta_{mj}+\delta_{mk}} Z_j Z_k. \tag{5.8}$$

Through this expression we replace $n(n-1)/2$ possible interaction strengths g_{jk} between qubits j, k with $n(n-1)/2$ resource interaction times t_{lm} sandwiched by single-qubit-X operators on qubits l, m. Using the linear independence of Pauli strings, we can write

$$\frac{g_{jk}}{r_{jk}} = \sum_{l<m}^{n} t_{lm} (-1)^{\delta_{lj}+\delta_{lk}+\delta_{mj}+\delta_{mk}} \tag{5.9}$$

in which finding g_{jk} is a matrix inversion problem made apparent by consolidating the parameter pairs l, m and j, k each to one parameter

$$\kappa = n(l-1) - \frac{l(l+1)}{2} + m, \tag{5.10}$$

$$\mu = n(j-1) - \frac{j(j+1)}{2} + k. \tag{5.11}$$

We arrive at a solution time vector in time scaling at most $O(n^6)$ using Gaussian elimination on a classical computer of a matrix with dimension $n(n-1)/2 \times n(n-1)/2$ of

$$t_\kappa = M^{-1}_{\kappa\mu} (g/r)_\mu \quad \text{for} \quad M_{\kappa\mu} = (-1)^{\delta_{lj}+\delta_{lk}+\delta_{mj}+\delta_{mk}}. \tag{5.12}$$

For the case of $n = 4$, M is singular as, for example, $X_1 X_2 \hat{H}_{\text{Resource}} X_1 X_2 = X_3 X_4 \hat{H}_{\text{Resource}} X_3 X_4$ and the condition of linear independence of the effective Hamiltonians from different blocks is not met. This, however, is a special case and the obtained Hamiltonians are linearly independent for all $n > 4$. QAOA problems of interest, however, far exceed this value in size.

An obstacle to the usage of this scheme is that any of the times calculated in this procedure may be negative. Following the computation of a time vector t_κ providing a DA circuit to simulate a desired Ising Hamiltonian, negative times must be eliminated as it is experimentally impossible to run an always-on interaction—the nature of which we can't temporarily change—for a negative time.

5.6 Negative Digital-Analog Block Times

In this section, a procedure is presented that exploits the case in which the resource Hamiltonian is homogeneous, or one of $\vec{g}, \vec{r}$ takes only values with some high least common multiple, resulting in periodic time-evolution. This condition holds for MAX-CUT and SAT problems considered in this work (all ZZ Hamiltonian terms in

Section 5.7 have at least half-integer pre-factors or integer multiples thereof). In the case that the resource Hamiltonian is homogeneous, any negative time-block can simply be run for a positive time $2\pi + t$, exploiting the periodicity of the unitary effected as $e^{i\hat{H}t} = e^{i\hat{H}(2\pi+t)}$, for $r_{jk} = 1 \; \forall \; j, k$. As such, with homogeneous resource Hamiltonians, we can always replace t_{lm} with $t_{lm} \mod 2\pi$. This technique, involving a homogeneous resource, is unfortunately undesirable as a method to rectify all negative time blocks as we will take a time interval that is typically small and replace it with a larger time $2\pi - |t_{lm}|$. This will result in a DA schedule of single-qubit gates and analog block times requiring a longer total time to run on hardware, incurring greater error rates.

For an inhomogeneous resource Hamiltonian we need to consider one additional time block surrounded by no single-qubit operations. To determine the size of this *idle* block required, consider

$$\mathbf{M}\vec{t} = \mathbf{M}(\vec{t} - t_{\min}\vec{1} + t_{\min}\vec{1}). \tag{5.13}$$

$\mathbf{M}$ admits $\vec{1}$ as an eigenvector with eigenvalue λ. Intuitively, $\vec{1}$ is an eigenvector of $\mathbf{M}$ because when applying all possible two-X-surrounded DA blocks for an equal time, the time evolutions mostly cancel out leaving a smaller but homogeneous effective interaction. This produces

$$\mathbf{M}\vec{t} = \mathbf{M}(\vec{t} - t_{\min}\vec{1}) + \lambda t_{\min}\vec{1} \tag{5.14}$$

and considering a new, non-negative time vector $\vec{t}^{*} = \vec{t} - t_{\min}\vec{1}$

$$\mathbf{M}\vec{t} = \mathbf{M}\vec{t}^{*} + \lambda t_{\min}\vec{1}. \tag{5.15}$$

Applying all possible two-qubit-X DA blocks does not, however, result in a similar homogeneous contribution to the simulated Hamiltonian to resource Hamiltonian ratio for all system sizes. We wish to have an eigenvalue λ that is negative, such that when multiplied by negative $t_{\min}$ we produce a positive idle time. Unfortunately, the contributions to the ratio for NISQ-relevant cases with $n > 6$ are themselves, positive. The relation between $n(n+1)/2 + 1$ time-intervals and the Hamiltonian simulated can be written as

$$g_{\kappa} = M_{\kappa\mu} t^{*}_{\mu} r_{\kappa} + t_{\text{idle}} r_{\kappa} \tag{5.16}$$

with $t_{\text{idle}} = \lambda t_{\min}$. We solve the negativity problem by letting the always-on resource Hamiltonian run for time $\lambda t_{\min}$ surrounded by no single-qubit gates if $t_{\min}$ is negative. Since t_{idle} is negative for relevant cases of $n > 6$, we must use one of two methods to change the sign of this time, depending on whether a homogeneous or inhomogeneous resource Hamiltonian is available. If the resource is homogeneous we can evolve for time $t_{\text{idle}} \mod 2\pi$, as before. This cost of running for this positive time will only add a small contribution to the total algorithm run-time since it only occurs once per set of DA blocks. In realistic experimental cases, however, we expect only non-homogeneous resource Hamiltonians to be available. Even with non-homogeneous resource Hamiltonians, non-negative idle time is still possible through exploiting properties of the simulated problem Hamiltonian. By setting $\hat{H}_{\text{Arb}} \rightarrow -\hat{H}_{\text{Arb}}$ in Equation 5.4 and using the fact that all ZZ coupling constants in $\hat{H}_{\text{Arb}}$ will be integer multiples of $1/2$ or zero for MAX-CUT and MAX-2-SAT problems, we can

simulate the Hamiltonian of the correct sign by exploiting the periodicity of the unitary eff ected, as $e^{it\hat{H}_{\text{Problem}}} = e^{i(-t)(-\hat{H}_{\text{Problem}})} = e^{i(-t \mod 2\pi)(-\hat{H}_{\text{Problem}})}$. This factor of -1 in front of the problem Hamiltonian can then be absorbed into the matrix $\mathbf{M}$ in Equation 5.16 causing the eigenvalue λ to become negative. This allows a positive idle-time correction, resolving the negative sign issue for inhomogeneous cases.

The method presented here provides a convenient decomposition of an arbitrary Ising Hamiltonian into time blocks of our resource interaction surrounded by two pairs of single-qubit rotations. The problem of negative times is resolved for the case of resource or target Hamiltonians satisfying certain constraints. For Hamiltonians not satisfying the aforementioned constraints, approaches including the decomposition into multiple DA sequences satisfying these constraints, or a strategy involving a higher number of analog blocks could still be pursued. To incorporate a greater number of time blocks one could solve the under-determined linear system $\vec{g} = \mathbf{M}\vec{t}$ with $\vec{t} \geq 0$ where the time-vector $\vec{t}$ is of higher dimension than $\vec{g}$ and, therefore, $\mathbf{M}$ is no longer square (and invertible). Approaches to solving such a problem are complicated by the time non-negativity constraints and require a quadratic programming approach. Recent work by Galicia et al. [134] extends the digital-analog paradigm to the scenario in which only interactions available on a device with linear, nearest-neighbor connectivity can be used to systematically produce an all-to-all connected arbitrary Hamiltonian. Strategies for architectures that are more connected than linear, yet not fully connected, can therefore also produce arbitrary Ising Hamiltonians, by restriction to a linear chain, or by manually inserting the SWAP operations of a SWAP network [137], themselves compiled to digital-analog sequences.

5.7 Digital-analog QAOA

In Digital-Analog QAOA (DA-QAOA), we use the DA paradigm to perform an algorithm that approximates the QAOA. We take access to the device Hamiltonian utilized in the previous sections with:

$$\hat{H}_{\text{Device}}(t) = f(t)\hat{H}_{\text{resource}} + \alpha \sum_{i=1}^{n} (x_i(t)X_i + z_i(t)Z_i) \tag{5.17}$$

where

$$\hat{H}_{\text{Resource}} = \sum_{j<k}^{n} r_{jk} Z_j Z_k. \tag{5.18}$$

We use this device interaction to apply or approximately apply a QAOA circuit with a problem Hamiltonian comprising of a subset of the non-zero entries in the resource Hamiltonian r_{jk}.

Problems we might consider would primarily be those of MAX-CUT and MAX-2-SAT. As mentioned in the discussion of negative times, these are problems in which periodicity can be exploited to remove the problem of negative interaction times for the scheme. However, a resource Hamiltonian that also exhibits periodicity, any two-local problem could be addressed, for example, finding low energy states for SK-models (Gaussian distributed couplings), or number partitioning-type problems (Mattis type squared single-qubit Gaussian terms addressed in Chapter 3). Other problems that exhibit the same periodicity would be weighted MAX-CUT or weighted

MAX-2-SAT, presuming, that the weights have a common increment that can be expressed with relatively few bits of precision[1].

In the stepwise scheme (sDA-QAOA), we assume control over the parameters f, x_i, z_i each taking values from $\{0, 1\}$. In this case, it is possible to express a QAOA circuit exactly, given the problem or Hamiltonian has properties that allow for the elimination of negative times. In the banged scheme (bDA-QAOA), f is assumed to be fixed at unity by hardware constraints, with only the single-qubit parameters permitted to be altered. The single-qubit terms are assumed to be stronger than the resource Hamiltonian by a factor $\alpha \geq 1$ and in typical applications, α is expected to fall between $10 - 1000$ depending on architecture [138, 139]. In near-term architectures, it is almost always the case that single-qubit operations are orders of magnitude faster than interactions, due to inherently lower complexity. While current devices tend to exhibit a ratio of single-qubit rotation speed to interaction strength at the lower end of this range, around $\alpha = 100$, current engineers have had little to gain from making these operations faster, since they are typically limited by two-qubit interaction times and fidelity. Single-qubit operations are often considered as zero cost since the error budget is dominated by the two-qubit interactions. We, therefore, expect that a device optimizing for DA applications could be engineered for greatly higher ratios α, resulting in higher fidelity for DA-QAOA than would be implied by current devices. Also, it would typically be possible to use slower multi-qubit interactions relative to the single-qubit terms, to further reduce error.

During driving in bDAQC, all single-qubit-X operations are set to 1, Z terms to 0, giving a driver Hamiltonian of

$$\hat{H}_{\text{bDA-Driver}} = \sum_{j<k} r_{jk} Z_j Z_k + \alpha \sum_{i=0}^{n} X_i \tag{5.19}$$

applied for device time given by the variational parameter β divided by the driver strength α with $\beta \in [0, \pi]$. One notes, that this range of possible values for β is picked due to the periodicity of the QAOA driving operator. However, in banged QAOA, this periodicity is no longer present. Regardless, observations of where the optimal parameters land in practice demonstrate that the original range is still the best choice. This can be argued for from an entropic perspective. The single-qubit-X driver sees bit-wise structure in the problems it solves. When one applies the bDA-QAOA Hamiltonian for more than π, one has an operator further from that of the typical QAOA driver, and, therefore, one might expect it to exploit the bit-wise structure less adeptly.

During the DA resource Hamiltonian steering operations, in which we apply single-qubit gates to transform the resource Hamiltonian to something best approximating a problem Hamiltonian of interest, we use a similar Hamiltonian in which only a specific set of single-qubit-X terms are active. As described in Section 5.5, we wish to implement a full X gate before and after each resource block. The time to apply this gate will be $\Delta t = \frac{\pi}{\alpha}$. Applying the DA-QAOA device Hamiltonian for a single QAOA layer thus effects the following unitary

$$\hat{U}_{\text{DA}-\text{QAOA}} = \mathcal{T} \exp\left(-i \int_0^{t_{\text{total}}} \hat{H}_{\text{Device}}(t)\ \mathrm{d}t\right), \tag{5.20}$$

[1]The size of the idle block introduced in Section 5.5 will be directly proportional to the inverse of the minimum spacing in the couplings in the problem Hamiltonian.

with $\vec{x}(t)$ defined by the aforementioned matrix inversion procedure, $\mathcal{T}$ is the time-ordering meta-operator and $\vec{z}(t)$ used in the case that we are solving a 2-SAT problem or otherwise a problem with single-qubit terms. t_{total} is the sum of all times in the non-negative DA time vector multiplied by the variational parameter γ in addition to the driving time β/α. A depiction of this device Hamiltonian used to apply a MAX-CUT problem Hamiltonian is presented in Figure 5.3.

5.8 Compilation Costs of DA-QAOA

In this section, the cost in on-device time to perform QAOA using DAQC and different resource Hamiltonians is evaluated. We include in our comparison the time taken by a completely digital quantum computer under reasonable assumptions. We emphasize here that the time taken to perform an algorithm is only a good indication of the fidelity of the algorithm's experimental implementation (or quality of solution) if the device running the experiment is *coherence limited*. In contemporary quantum processors, the limitation is typically not coherence time but the error incurred during the use of two-qubit operations, per operation. An evaluation of whether a device using a DAQC or DQC paradigm performs better would require in-depth knowledge of the error mechanisms of a device operating in the respective paradigm. Such an analysis is expected to favor DAQC given the reduction of errors from turning couplings on and off.

When comparing the performance of a device making use of the DAQC paradigm to a device running completely digital computations, we must make fair assumptions concerning the capability of each device. We compare the case in which both DAQC and DQC can perform interactions between any pair of qubits. In DQC, the key limitation we apply—besides the differing error models that are expected to comprise the main advantage of DAQC—is the inability to perform simultaneous two-qubit gates on a single qubit. A given QAOA problem Hamiltonian in DQC must therefore be decomposed into a number of time steps. This number of time steps for a graph-based problem can be shown to, at most, equal the maximum vertex degree of the problem graph plus one. DAQC in comparison applies all operations at once but must utilize many time blocks to time-average the device resource interaction to the problem Hamiltonian of interest. An example of a QAOA MAX-CUT problem compiled to both the DQC and DAQC is found in Figure 5.3 for a 5-regular MAX-CUT problem on an 8 qubit device. One notes that for this particular problem, the DA circuit can be performed faster than the digital for sufficiently fast X gates.

Comparisons of the time taken to implement a problem Hamiltonian are presented in Figure 5.4. In this plot, we compare Hamiltonians from Section 5.9, with homogeneous and inhomogeneous resource interactions. For the inhomogeneous resource interactions, we use couplings $r_{jk} \sim \mathcal{N}(1, \delta^2)$ where δ is the fractional standard deviation of the coupling strength. Values of 5% and 10% are used for this inhomogeneity. For the $|r|^{-6}$ and $|r|^{-3}$ power law behavior, we assume that qubits are placed in a linear array. For a fair comparison between this resource Hamiltonian and the others, we scale the couplings such that the average coupling between two qubits is the same for all resource Hamiltonians used. Disregarding the speed advantage from exploiting the periodicity of a homogeneous resource, small deviations in the couplings do not greatly affect the compiled time. If, however, any individual coupling becomes especially small, the compilation time grows correspondingly

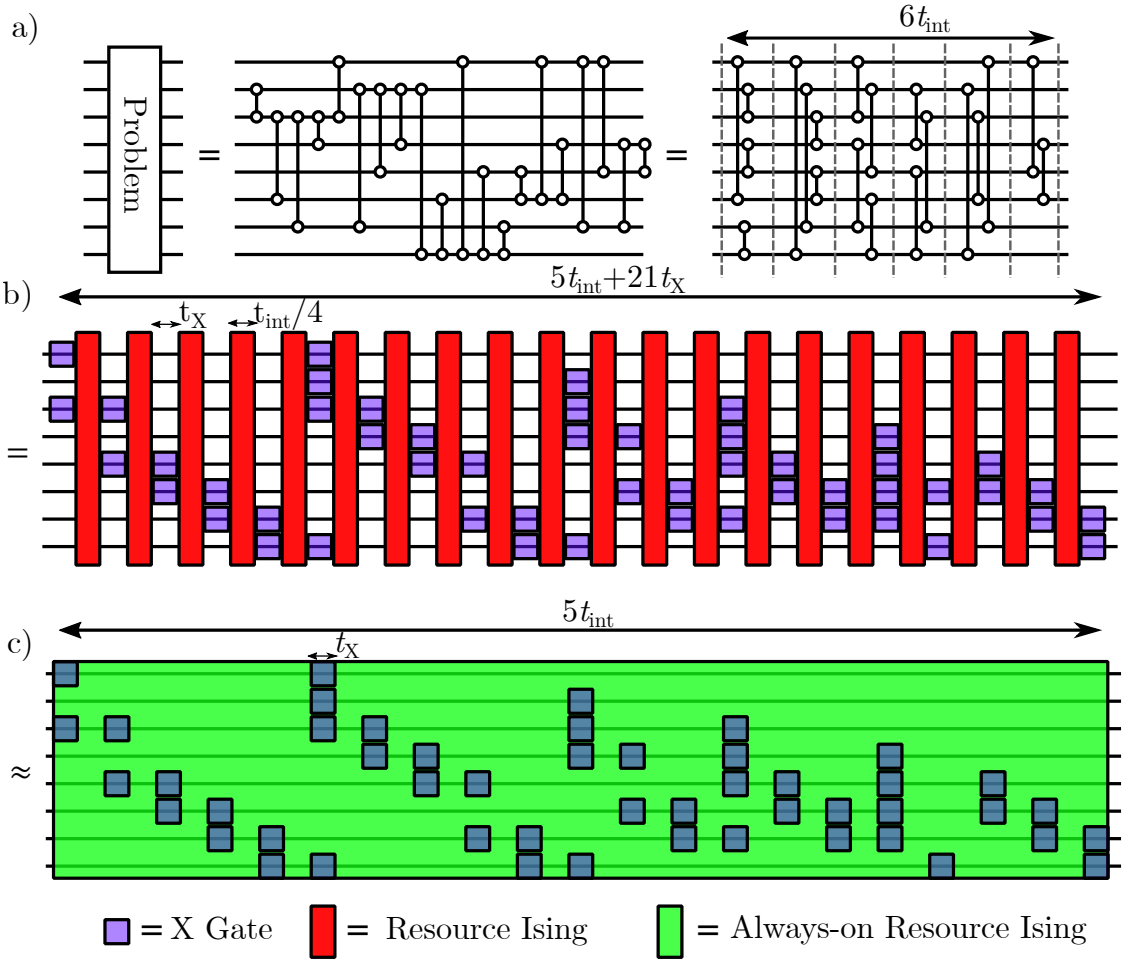

Figure 5.3: Circuit compilations for an 8-qubit MAX-CUT problem on a 5-regular graph. (a) A decomposition of the MAX-CUT problem Hamiltonian into ZZ interactions. The lines connected to two open circles represent ZZ interactions applied for time t_{int}. On the right-hand side, we see that the circuit can be parallelized into six time steps, each of which sees one qubit interact with only one other qubit at a time. Six time steps corresponds to the maximum degree of a vertex plus one. (b) The same circuit can be compiled into the scheme of sDAQC in which a resource all-to-all homogeneous Ising Hamiltonian is turned on and off, punctuated by single-qubit gates. This decomposition requires 20 uses of the resource Hamiltonian and 50 single-qubit-X operations. The time taken to apply the problem Hamiltonian is $5t_{\text{int}} + 21t_X$. (c) Finally, we compile the problem Hamiltonian in the bDAQC scheme, in which the resource Hamiltonian remains on throughout the procedure. This circuit only approximates the time evolution invoked by the QAOA problem Hamiltonian but can be carried out in time $5t_{\text{int}}$ and also with 49 single-qubit-X operations. In the limit of infinitely fast X gates, (c) is equivalent to (a) and (b).

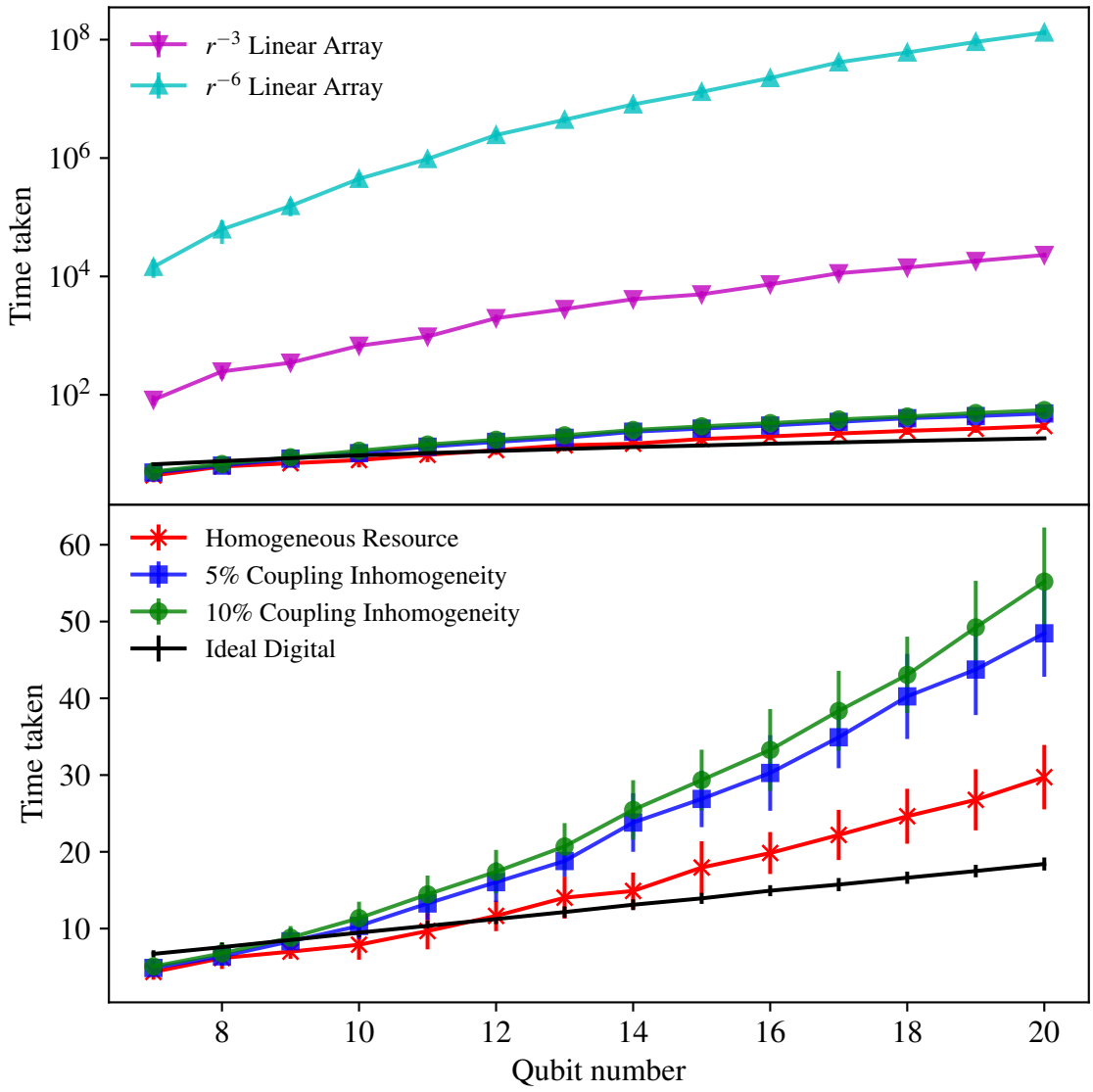

Figure 5.4: Plots showing the on-device required time for implementation of a QAOA problem Hamiltonian. In this case, random Erdős Rényi MAX-CUT problems are used with a filling factor of 0.75. Units of time are defined relative to the native device resource interaction strength. In these plots, we show the time taken when using various possible resource Hamiltonians. The left-hand side demonstrates that if the resource Hamiltonian available varies across orders of magnitude in coupling strength, as happens in the case of an inverse power law coupling between qubits in an array, the time taken in the digital-analog scheme becomes extremely large (and likely impractical). The right plot shows the series on the left excluding the inverse power law couplings. A homogeneous resource Hamiltonian is competitive with an idealized digital compiling scheme, though higher for qubit numbers higher than 10. The upper blue (square marker) and green (circle marker) lines demonstrate that if we use a resource Hamiltonian with normally distributed couplings close to 1 (standard deviations 0.05 and 0.1), the time taken is longer. The significant gap between the Homogeneous and non-homogeneous series occurs as the periodicity of the homogeneous resource Hamiltonian's effected time evolution can be exploited to reduce the idle time. The units for the time displayed in the plot are dimensionless multiples of a timescale defined by the resource Hamiltonian.

large.

The asymptotic scaling of the algorithm is at worst $O(n^2)$ as $n(n-1)/2$ interaction windows are needed to simulate arbitrary Hamiltonians.

5.9 High-Connectivity Hardware Platforms

In this section, we consider potential hardware realizations of DA-QAOA. A NISQ device able to solve a wide variety of combinatorial optimization problems running QAOA would require a highly connected quantum device to avoid the need for swapping operations. One could then utilize platforms in which non-local interactions occur natively while benefiting from the reduced control overhead provided by the digital-analog scheme. It has been proposed to use the digital-analog scheme to compile SWAP gates themselves to sequences of digital and analog blocks [134]. For realistic near-term hardware, however, we expect any algorithm utilizing swap operations to be out of reach, whether compiled to digital gates or to digital-analog time blocks, due to their excessive contribution to circuit depth and, therefore, decoherence.

Generally considered to be the most mature platform for quantum computing, superconducting solid-state qubit architectures tend to have low connectivity due to their 2d-designed nature and are, in current manifestations, not an ideal candidate for performing DA-QAOA [55], the potential use of this platform was explored for DAQC in transmon qubits utilizing the cross resonance effect in work by Gonzalez-Raya et al. [140]. Other systems, for example, Rydberg neutral atom arrays or cold, trapped-ion architectures allow for native interactions between all qubits in a device.

Rydberg neutral atoms are atoms in which one or more electrons are in a highly excited state. Excited states of these atoms have high lifetimes owing to their large spatial extent and, therefore, small spatial overlap with the ground state of the atom [141]. Optical lattices of Rydberg atoms can feature non-local, all-to-all Vander-Waals interactions scaling with $|1/r|^6$ for distances r greater than the optical lattice spacing. Such an interaction is highly non-homogeneous but could be utilized for the digital-analog scheme.

Trapped ion systems have demonstrated the highest fidelity two-qubit operations [138, 142] and the highest coherence time of any existing platform [143]. These systems, however, fail to achieve high fidelity when many qubits are loaded into a trap. This limitation occurs due to the frequency crowding of the energy levels used to address the coupling of each ion to the collective motional state of the trapped ions. When the requirement for control over interactions between individual ions in a trap is relaxed, trapped ion platforms perform exceptionally as simulators [144] and qubit numbers competitive with the best superconducting processors can be used to explore physics outside the reach of classical simulation. Interactions between trapped ions scale on the order of $|1/r|^\delta$ with δ typically varying between $0-3$ [145], with r as the distance between two trapped ions. A system utilizing a value of $\delta = 0$—in which the interaction is mediated by the joint vibrational modes—would have a homogeneous coupling if no other non-homogeneous behavior is present between pairs of ions. The case of $\delta = 3$ occurs when the interaction is mediated purely via spin-spin interactions, incurring a dipolar decay law.

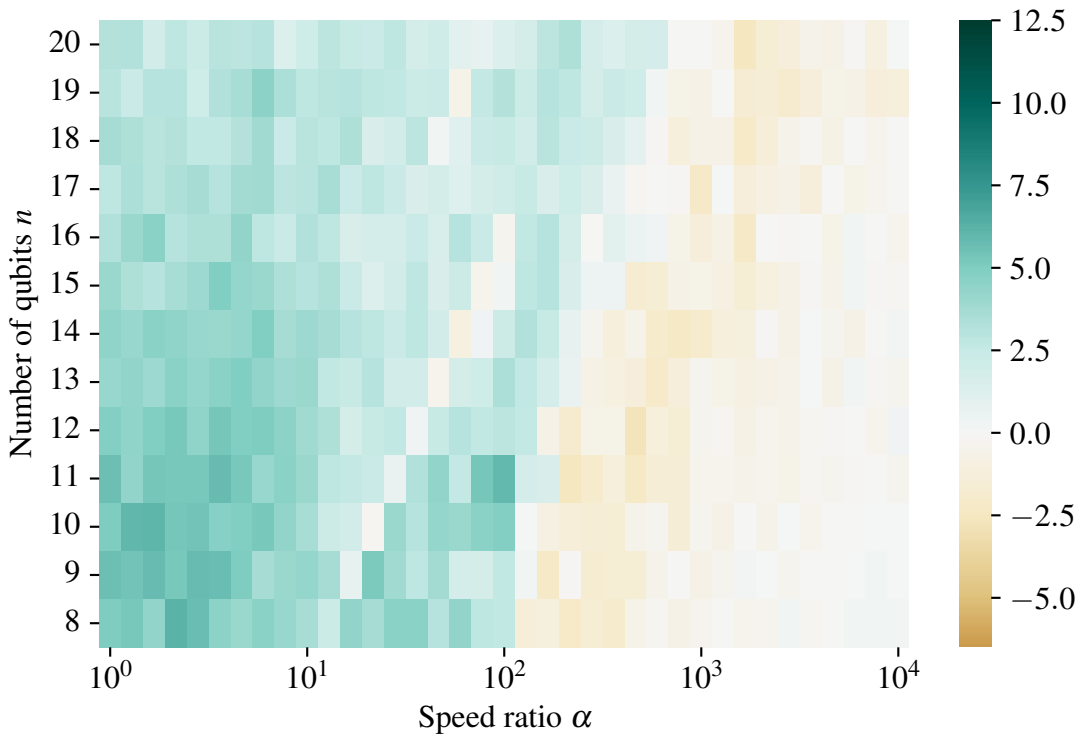

Figure 5.5: The percentage difference between the mean approximation ratio attained by bDA-QAOA and error-free QAOA, averaged over 50 randomly generated MAX-CUT problems with constant filling factor $p_{\text{clause}} = 0.7$. Blue (dark region on the far left side of the figure) colors indicate that the bDA-QAOA ansatz state is worse than that provided by error-free QAOA, whereas, shades of brown (the light gray region immediately left of the white region of zero error) indicate an improvement of the bDAQC over error-free QAOA. On the x-axis, the ratio α of single-qubit to problem Hamiltonian term strength is seen where error-free QAOA exists in the limit as this ratio becomes infinite.

5.10 Results

In this section, the main computational results of our work are presented. In the first subsection, we present the performance of bDA-QAOA in comparison to a standard QAOA circuit for a set of chosen randomly generated problems. In the second subsection, we analyze this further to demonstrate that the algorithm is performing better than one might expect for a banged DA algorithm and that this boost in performance results from the variational freedom of DA-QAOA.

5.10.1 Performance of bDA-QAOA

In bDA-QAOA, we perform QAOA using the ansatz state prepared by applying QAOA layers of the form described in Equation 5.20 as

$$\left|\vec{\beta},\vec{\gamma}\right\rangle^{\alpha,\text{DA}} = \hat{U}_{\alpha\text{-DA-QAOA}} \left|+\right\rangle^{\otimes n}. \tag{5.21}$$

bDA-QAOA introduces errors in the form of the misspecification of the problem and driver Hamiltonians, between which, due to relative times taken on the device

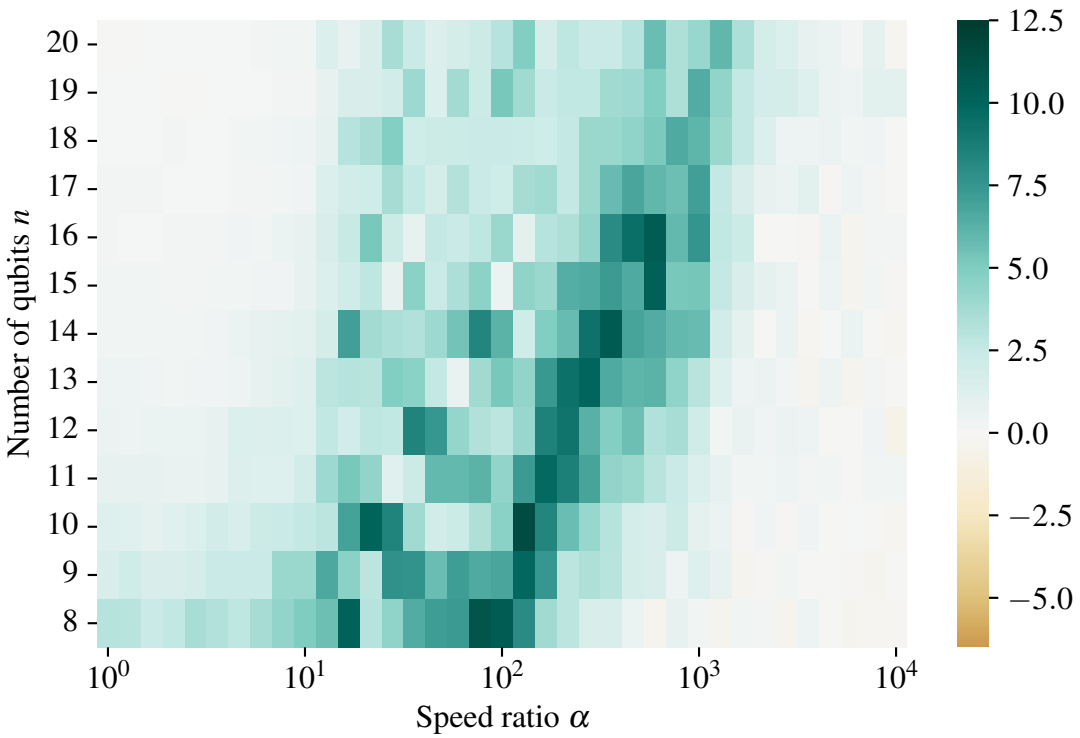

Figure 5.6: The percentage difference in mean approximation ratio attained by bDA-QAOA at parameters maximizing error-free QAOA and bDA-QAOA with optimized parameters, averaged over 50 randomly generated MAX-CUT problems with constant filling factor $p_{\text{clause}} = 0.7$. On the x-axis, the ratio α of single-qubit to problem Hamiltonian term strength is seen where error-free QAOA exists in the limit as this ratio becomes infinite. Darker colors imply that for the concerned speed ratio and qubit number, the variational nature of QAOA can account for differences between bDA-QAOA and the ideal algorithm. This plot shows the benefit of using a variational algorithm such as QAOA over a non-variational algorithm in the DA context where the coherent error is introduced.

to perform and that the driver is generic to all problems, the misspecification of problem Hamiltonian will introduce more error. This problem of misspecification is not new to the field of quantum optimization and is known in quantum annealing literature as J-chaos, in which critical characteristics of a problem to be solved are not correctly incorporated into the dynamics of an annealing device. Such issues can be fatal to the performance of AQC if error mitigation strategies are not utilized [146]. bDA-QAOA finds a connection to the algorithms of quantum random walks [111] and AQC in that the problem Hamiltonian and single-qubit driving operators are performed simultaneously. One might therefore expect that simply running a problem Hamiltonian at the same time as a driver in QAOA should not be fatal, in fact, scheduled quantum random walks and diabatic AQC are active fields themselves [109, 147]. Simulations performed of bDA-QAOA in which the resource Hamiltonian is identical to the problem Hamiltonian to be solved, such that no DA steering single-qubit operations are required, indeed, showed no discernible net-negative impacts when compared to error-free QAOA.

Coherent errors occurring in bDA-QAOA with a non-problem-specific resource Hamiltonian, however, are expected to be more damaging than the errors in QAOA with an always-on problem Hamiltonian. This is because in bDA-QAOA the always-on resource is not specific to the problem we wish to solve. These errors should therefore result in a less problem-specific QAOA ansatz state which results in a worse expected approximation ratio at a given depth. Figure 5.5 displays the mean approximation ratio attained by the bDA-QAOA ansatz state. For high α we see a regime in which, as expected, bDA-QAOA performs identically to error-free QAOA. Secondly, we see an intermediate regime where minor increases in the mean approximation ratio are observed. Finally, in the case of low α we observe consistently worse results for bDA-QAOA, due to problem misspecification induced by coherent DAQC errors.

5.10.2 Variational Resilience of DA-QAOA to DA Errors

QAOA is a variational algorithm. It is expected that variational algorithms have better error tolerance properties due to the fact that a classical optimizer can account for systematic coherent over- or under-rotations and other systematic coherent errors [148], making variational quantum algorithms appealing candidates for NISQ quantum computing. QAOA works by finding a parameter set $\vec{\beta}^*, \vec{\gamma}^*$ maximizing $\langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\beta},\vec{\gamma}}$. However, when we change the QAOA ansatz operators to those of bDAQC, there is no clear reason why the parameters $\vec{\beta}^*, \vec{\gamma}^*$ maximizing $\langle \hat{H}_{\mathrm{P}} \rangle_{\vec{\beta},\vec{\gamma}}$ also maximize $\langle \hat{H}_{\mathrm{P}} \rangle^{\alpha,\mathrm{DA}}_{\vec{\beta},\vec{\gamma}}$ where

$$\langle \hat{H}_{\mathrm{P}} \rangle^{\alpha,\mathrm{DA}}_{\vec{\beta},\vec{\gamma}} = \left\langle \vec{\beta},\vec{\gamma} \right|^{\alpha,\mathrm{DA}} \hat{H}_{\mathrm{P}} \left| \vec{\beta},\vec{\gamma} \right\rangle^{\alpha,\mathrm{DA}}. \tag{5.22}$$

Figure 5.6 suggests that this is not the case and shows that a significant increase in the success probability of QAOA results from the variational freedom of the algorithm. Figure 5.6 should be understood to demonstrate the parameter regimes for which it makes more sense to perform a variational algorithm such as QAOA rather than a fixed gate sequence algorithm such as the quantum Fourier transform. For high α, the error introduced by the scheme is negligible, and both variational algorithms and fixed sequence algorithms will perform similarly. In the middle of the plot, a dark turquoise band can be seen implying that while a non-variational algorithm will have low fidelity due to the presence of DA-induced coherent errors, the variational algorithm still functions. For low enough α, we enter a regime in which even the variational algorithm fails to recover any performance through altering parameters. We interpret that this lack of ability of DA-QAOA to absorb error in the low α regime is a manifestation of barren plateaus in the objective function [77]. Barren plateaus are a feature discovered to occur in the optimization landscapes of quantum neural networks. When parameterized random circuits are used as ansätze in variational algorithms, the gradient of the objective function with respect to the variational parameters is observed to become exponentially small in the number of qubits used. When α reduces to a certain value, we interpret that the DA-QAOA ansatz loses specificity to the problem Hamiltonian of interest. The variational form used for optimization no longer bears similarity to the objective function used and is, consequently, no better an ansatz than a random quantum circuit. At this point of low α, we observe that the gradient of our objective function with respect to the

variational parameters $\vec{\gamma}, \vec{\beta}$ tends to become prohibitively small and the approximation ratio attained therefore varies little with differing parameters.

5.11 Analytical Fidelity Bounds for bDA-QAOA

In this section, we demonstrate that the error introduced by performing QAOA in the banged digital-analog paradigm in comparison to regular QAOA can be analytically bounded. In particular, we place a lower bound on the fidelity of a state that arises from a bDA-QAOA circuit in comparison to a state prepared by error-free QAOA. This error consists of multiple steps, each of which is of the same nature as that occurring when Trotterizing a Hamiltonian with non-commuting terms for simulation. In the case of bDAQC-induced error, there are, however, two complications. Firstly, there is only a single Trotterization time-step which cannot be made arbitrarily small with the use of higher numbers of Trotter blocks. Secondly, we use Trotterization in reverse in this fidelity bound. In a usual Trotterization procedure, the simultaneous case is 'correct' and the digitized version introduces error. In the digital-analog scheme, however, the opposite is true. The sequential Hamiltonian is ideal and the simultaneous Hamiltonian introduces error. These differences do not affect the validity of the bound, since the bound used in this work is valid for arbitrarily large time steps. The bound used, to our knowledge, represents the current lowest bound on Trotter error [149] and limits the size of the greatest eigenvalue of an operator derived via the difference of two unitaries. The first unitary is that generated by the sequential application of two Hamiltonians $\hat{A}, \hat{B}$, with the second generated by the simultaneous application of such Hamiltonians:

$$\begin{aligned} \| \exp\Big(i\hat{A}/2\Big) \exp\Big(i\hat{B}\Big) \exp\Big(i\hat{A}/2\Big) - \exp\Big(i(\hat{A}+\hat{B})\Big) \| \\ \leq \frac{1}{12} \| \left[\left[\hat{A}, \hat{B} \right], \hat{B} \right] \| + \frac{1}{24} \| \left[\left[\hat{A}, \hat{B} \right], \hat{A} \right] \|. \end{aligned} \tag{5.23}$$

The system Hamiltonian during time periods in which both the single-qubit operations and the resource Hamiltonian are active is

$$\hat{H}_{\text{Steering, }\mu} = \alpha \sum_{m \in S_\mu} X_m + \hat{H}_{\text{Resource}} \tag{5.24}$$

where S_μ is the index set of X operators applied following DA time-block μ. There are $n(n+1)/2+2$ periods of time for which we apply this bound, $n(n+1)/2$ sets of single-qubit operations following interaction blocks, one idle block, and the driving block of the QAOA algorithm. We wish to compute the error resulting from using this, rather than its single-step Trotterization. Of these error effecting blocks, $n-3$ will have four full single-X rotations, $n(n-1)/2-(n-3)+1$ will have two full single-X rotations and one, the driver, will have n single-X rotations of duration $\beta \leq \pi$. We can allocate $\hat{A} = t\hat{H}_{\text{R}}$ and $\hat{B} = t\alpha \sum_{m \in S_\mu} X_m$. Where every term in $\hat{H}_{\text{R}}$

consists only of Pauli-Z strings. We can thus write

$$\begin{aligned}\left\| \mathrm{e}^{\frac{it\hat{H}_{\mathrm{R}}}{2}} \mathrm{e}^{it\alpha \sum_{m\in S_\mu} X_m} \mathrm{e}^{\frac{it\hat{H}_{\mathrm{R}}}{2}} - \mathrm{e}^{it\left(\alpha \sum_{m\in S_\mu} X_m + \hat{H}_{\mathrm{R}}\right)} \right\| \\ \leq \frac{\alpha^2 t^3}{12} \left\| \left[\left[\hat{H}_{\mathrm{R}}, \sum_{m\in S_\mu} X_m \right], \sum_{m\in S_\mu} X_m \right] \right\| \\ + \frac{\alpha t^3}{24} \left\| \left[\left[\hat{H}_{\mathrm{R}}, \sum_{m\in S_\mu} X_m \right], \hat{H}_{\mathrm{R}} \right] \right\|, \quad (5.25)\end{aligned}$$

because every block in the DA-QAOA setting will be surrounded by resource blocks, it does not matter whether the Trotterization is symmetric or asymmetric. To calculate the first commutator, we can expand the sums and compute each individual term

$$\begin{aligned}&\sum_{m\in S_\mu} \sum_{m'\in S_\mu} \sum_{j<k} \left[[Z_j Z_k, X_m], X_{m'} \right] \\ &= \sum_{m\in S_\mu} \sum_{m'\in S_\mu} \sum_{k>m} 2\left[Z_m Z_k X_m, X_{m'} \right] + \sum_{m\in S_\mu} \sum_{m'\in S_\mu} \sum_{j<m} 2\left[Z_j Z_m X_m, X_{m'} \right] \\ &= \sum_{m\in S_\mu} \sum_{m'\in S_\mu} \sum_{j\neq m} 2\left[Z_j Z_m X_m, X_{m'} \right] \\ &= \sum_{m\in S_\mu} \left(\sum_{j\neq m} 4 Z_m Z_j + \sum_{m'\in S_\mu | m'\neq m} 4i Y_m Y_{m'} \right). \quad (5.26)\end{aligned}$$

We have here assumed that the resource is homogeneous. In a given single-qubit block there will be s possible indices in the corresponding set S_μ. As such we will have $s(n-1)$ Z strings and $s(s-1)$ Y strings. The eigenvalues of sets of terms that can be simultaneously diagonalized will add linearly. We have two orthogonal bases in which eigenvalues are added as such, the sums of which will add in quadrature. The greatest eigenvalue of the entire sum in Equation 5.26 can then be written $s\sqrt{(s-1)^2 + (n-1)^2}$. The contribution to the bound in Equation 5.25 from this commutator can therefore be written

$$\frac{\alpha^2 t^3 s \sqrt{(s-1)^2 + (n-1)^2}}{3}. \quad (5.27)$$

For a homogeneous resource, the commutator in the second term can be written as

$$\begin{aligned}
&\sum_{m\in S_\mu}\sum_{j,j'<k,k'}\left[\left[Z_jZ_k,X_m\right],Z_{j'}Z_{j'}\right] \\
&\qquad=\sum_{m\in S_\mu}\sum_{j,j'<k,k'}2(\delta_{jm}+\delta_{km})\left[Z_jZ_kX_m,Z_{j'}Z_{k'}\right] \\
&\qquad\quad=\sum_{j'<k'}\sum_{m\in S_\mu}\sum_{j\neq m}2Z_j\left[Z_mX_m,Z_{j'}Z_{k'}\right] \\
&\qquad\quad\quad=\sum_{m\in S_\mu}\sum_{j\neq m}\sum_{j'\neq m}4Z_jZ_{j'}Z_mX_mZ_m \\
&\qquad\quad\quad\quad=\sum_{m\in S_\mu}\sum_{j\neq m}\sum_{j'\neq m}-4Z_jZ_{j'}X_m, \quad (5.28)
\end{aligned}$$

where the negative sign is of no consequence. We have $(n-1)^2$ terms per single-qubit-X operator and s X-operators giving $s(n-1)^2$ strings. At worst the greatest eigenvalue of this operator sum will be equal to the number of Pauli strings. We, therefore, obtain a full bound of

$$\Delta_\mu=\frac{\alpha st^3}{3}\left(\frac{(n-1)^2}{2}+\alpha\sqrt{(s-1)^2+(n-1)^2}\right). \quad (5.29)$$

We wish to bound the minimum fidelity of a coherent erroneous operation caused by using bDA-QAOA

$$f_{\alpha-\mathrm{DA}-\mathrm{QAOA}}=\min_{\psi}\left\|\langle\psi|\,\hat{U}^\dagger_{\mathrm{QAOA}}\hat{U}_{\alpha\text{-DA-QAOA}}\,|\psi\rangle\right\|^2. \quad (5.30)$$

If the magnitude of the greatest eigenvalue of the difference between two unitaries operators is bounded as in the Trotterization bound

$$\|\hat{U}-\hat{U}_\alpha\|\leq\Delta \quad (5.31)$$

then

$$\|\hat{U}^\dagger\hat{U}_\alpha-I\|\leq\Delta \quad (5.32)$$

which yields a bound of

$$\|\mathrm{e}^{i|\theta|_{\max}}-1\|\leq\Delta \quad (5.33)$$

where $\mathrm{e}^{i|\theta|_{\max}}$ is the greatest eigenvalue of $\hat{U}^\dagger\hat{U}_\alpha$, assuming the eigenvalues are small such that all angles lie on the interval $[-\pi/2,\pi/2]$. The greatest phase acquired under the erroneous evolution can then be related to the greatest eigenvalue bound as

$$2\sin\left(\frac{|\theta|_{\max}}{2}\right)=\Delta,\quad \theta=2\sin^{-1}\left(\frac{\Delta}{2}\right) \quad (5.34)$$

where $0<\Delta<1$. Consider the fidelity of a state under an erroneous operator $\hat{O}=\hat{U}^\dagger\hat{U}_{\mathrm{imperfect}}$ corresponding to the time-step μ during which a round of single-qubit-operations are performed:

$$\left\|\langle\psi|\,\hat{O}\,|\psi\rangle\right\|^2. \quad (5.35)$$

We can write this in the basis diagonalizing $\hat{O}$,

$$\left\| \langle \psi' | \operatorname{diag}\left(\mathrm{e}^{i\vec{\theta}}\right) |\psi'\rangle \right\|^2 \tag{5.36}$$

and this fidelity is minimized when the state is an equal superposition of the most positive and most negative argument eigenstates of $\hat{O}$.

$$\begin{aligned}
\min_{\psi'} \left\| \langle \psi' | \operatorname{diag}\left(\mathrm{e}^{i\vec{\theta}}\right) |\psi'\rangle \right\|^2 & \\
&= \frac{1}{4} \left\| \left(\langle \theta_{\max}| + \langle \theta_{\min}| \right) \hat{O} \left(|\theta_{\max}\rangle + |\theta_{\min}\rangle \right) \right\|^2 \\
&= \frac{1}{4} \left\| \langle \theta_{\max}| \, \mathrm{e}^{i\theta_{\max}} |\theta_{\max}\rangle + \langle \theta_{\min}| \, \mathrm{e}^{i\theta_{\min}} |\theta_{\min}\rangle \right\|^2 \\
&\geq \frac{1}{4} \left| \mathrm{e}^{i|\theta|_{\max}} + \mathrm{e}^{-i|\theta|_{\max}} \right|^2 \\
&= \cos\left(|\theta|_{\max}\right)^2.
\end{aligned} \tag{5.37}$$

So the fidelity of a single-qubit block μ is

$$f_\mu \geq \cos\left(|\theta|_{\max}\right)^2, \tag{5.38}$$

or in terms of the bound of the greatest eigenvalue of this set of single-qubit-operations Δ_μ:

$$f_\mu \geq \cos\left(2\sin^{-1}\left(\frac{\Delta_\mu}{2}\right)\right)^2 \geq 1 - \Delta_\mu^2, \tag{5.39}$$

with equality in the limit of small Δ. Using the sub-additivity of infidelity [150] we can finally express

$$f_{\alpha-\mathrm{DA}-\mathrm{QAOA}} \geq 1 - \sum_{\mu=1}^{n(n-1)/2+2} (1 - f_\mu) = 1 - \sum_{\mu=1}^{n(n-1)/2+2} \Delta_\mu^2. \tag{5.40}$$

For one set of single-qubit-rotations, the driver, all X terms are active, giving $s = n$. The remaining time blocks either have 2 or 4 single-qubit gates active determined by whether cancellations occur. No cancellations occur for $n - 3$ blocks resulting in $s = 4$, with the remaining $n(n-1)/2 - (n-2)$ blocks taking $s = 2$. The time taken to perform each block is $t = \pi/\alpha$. For large n we find that the speed of single-qubit gates must increase approximately as the number of qubits squared for high-fidelity with the ideal QAOA state. While this discussion has considered only the use of a homogeneous resource Hamiltonian in the interest of brevity, this is not critical to the calculation of the bound. A heterogeneous resource would result in a sum of Pauli strings with non-unit pre-factors, which could be subsequently summed straightforwardly, as in the homogeneous case.

5.12 Sensitivity to Other Errors

Next to the errors discussed in the previous sections that are imminent to the hardware simplification provided by our digital-analog approach, the algorithm is exposed to other sources of errors common to NISQ computing. As detailed error

budgets of concrete hardware are currently hard to determine, we would like to qualitatively evaluate their impact on our technique. A detailed evaluation of the effects of different noise sources applicable to digital-analog quantum computing in comparison to a gate-based approach was performed in [151], with results in favor of the digital-analog paradigm.

On the one hand, single-qubit gate errors induced by decoherence measured by $T_{1/2}$ will have a full impact on this algorithm as these are repeatedly executed. Small errors in the rotation axis will also have a full impact as they can be mistaken for a modified problem Hamiltonian. Errors in the rotation angle can be expected to be less critical as some of them can be accommodated in the classical optimization process. So all in all, single-qubit errors have the same if somewhat smaller impact than in a compiled gate model QAOA.

Two-qubit gates do not appear directly in our scheme thus avoiding two-qubit gate control errors as well as the additional entry points for noise through fast two-qubit control ports. However, the interaction mediated by the problem Hamiltonian can still create entangled states, which decay faster than non-entangled states. Notably, an n-qubit GHZ state de-phases in a time T_2/n [152]. The precise degree of entanglement needed for a specific problem instance is currently unknown for any quantum optimization algorithms. Yet, we can summarize that the sensitivity of digital-analog QAOA to two-qubit errors is lower than the compiled version. Given a single-qubit error rate, alongside the total execution time of the algorithm relative to T_2, the depth at which this algorithm can be faithfully executed could be inferred.

In this estimate, we need to keep in mind whether coherent over-rotation errors have an effect different from incoherent errors. This case could occur if they interfered in a structured way. Given the randomization effect of the problem Hamiltonian to any state, this is unlikely and we expect that their impact is faithfully represented by the measured fidelity.

5.13 Conclusion

The possibilities of using models of quantum computation less conventional than the standard gate-based approach have not been fully considered. In this work, we show that while an alternative approach—the digital-analog paradigm—might introduce errors of its own, the device complexity required to control the time evolution of the system can be reduced and errors introduced are of a nature that can be non-fatal to variational algorithms such as QAOA in certain regimes. We demonstrate that the digital-analog paradigm is an ideal setting in which to do QAOA, as each problem Hamiltonian operator can be performed in a single DAQC block, that resource Hamiltonians expected from hardware can be utilized to implement QAOA Hamiltonians mitigating swapping overhead associated with mainstream approaches and that QAOA displays error resilience beyond that of pre-programmed algorithms in the digital-analog paradigm. This work presents new possibilities for the design of NISQ devices for combinatorial optimization, bridging the gap between current devices and full, fault-tolerant quantum computers, and bringing hardware closer to the point of demonstrating a quantum advantage for real-world problems.

Chapter 6

High-Connectivity Devices for DA-QAOA

In Chapter 5, we devised a scheme for performing QAOA on a device with non-standard native interactions. In this chapter, we discuss how a device designed for such a purpose could look in practice. The digital-analog scheme was originally considered for the context of trapped ion quantum computing, however, the paradigm of superconducting qubits is arguably a more mature technology with recent state-of-the-art results occurring on such devices. Therefore, we demonstrate in this chapter how a superconducting device can be purpose-built for Digital-Analog QAOA and explore the feasibility of such a design.

6.1 Attribution

The work in this chapter was carried out by myself, with the general idea for the design of such devices coming from Prof. Wilhelm-Mauch. The work was helped by useful discussions with Dr. M. Ansari and J. Pazem.

6.2 Introduction

Most key implementations of QAOA to date are those of gate-based architectures. For example [153], the most impressive example of a QAOA implementation on a device is that in which the Google Sycamore chip, a device with 53 qubits of which 23 were used, ran the QAOA algorithm to find solutions to non-planar MAX-CUT problems and SK instances. Non-planar problems have the drawback of requiring more advanced compilation methods on any device in which the qubits find themselves embedded on a 2d plane, without crossing coupling links, as superconducting devices typically do. Swap gates are therefore needed, and are expensive, requiring three two-qubit gates to apply a single swap between two qubits. The need for swap gates also introduces a classical routing problem in which logical qubits must be routed efficiently such that interactions can occur between all required pairs of qubits [154, 155, 156].

A device for QAOA, therefore, would stand to gain from the capability to apply arbitrary interactions without the need for routing and swaps, though such capabilities would always come at the cost of additional engineering challenges. This chapter

is an attempt to imagine such a device, using the digital-analog scheme to enable a device that is simplified in control requirements, but complicated in fabrication and characterization.

There have been attempts to consider the digital-analog scheme for superconducting devices, as is attempted in this chapter. For example, Gonzalez et al. [140] considered the use of the cross-resonance effect to generate interactions between transmon qubits, the most commonly utilized interaction for transmon devices [157]. A second work utilizes charge qubits [158] arranged in linear chains, with tunable SQUID devices mediating interactions. This approach, alongside relevant digital-analog single-qubit gate schedules, allows the simulation of various spin latices, such as Fermionic Hubbard models. These two implementations are engineered to emphasize scalability. Scalability to large qubit numbers is likely a sensible goal for the use of a quantum device for simulating large models and observing phenomena thereof. As an example, work on Rydberg arrays in a 2d optical trap [159] observed large-scale entanglement and quantum phase transition phenomena on 256 qubits, far more than contemporary gate-based quantum computers. Such large-scale investigations are made possible by the reduced requirement to control devices than is required for the typical gate sets for universal quantum computation.

6.3 General Timescales and Design Parameters

When building a device for digital-analog QAOA, we are constrained by certain timescales. On the upper end, we have the coherence time of a device. The coherence time of a device can be compared to the T_2 phase relaxation time in nuclear magnetic quantum computing. This is the time it takes for the transverse magnetization of the system to decrease to a value of e^{-1} times its initial magnitude. In a more general sense, coherence time represents the duration for which a digital-analog computation can maintain a certain level of fidelity, such that meaningful results can still be obtained. Depending on the realization of a qubit, there exists some characteristic timescale over which the information encoded in a qubit decoheres, via interaction with the environment. Such a timescale will typically represent an upper limit on the time in which a calculation should be complete before the results of such a calculation are scrambled by noise.

On the lower end of constraining timescales, we have the shortest time in which single-qubit operations can be performed. Single-qubit operation times are determined by the specific qubit design, with the factors that result in fast or slow single-qubit gates being beyond the scope of this thesis. Another important factor is the quality of readout schemes for the device's state, allowing one to obtain the results of a computation.

In between these upper and lower timescales is, typically, the two-qubit interaction time. For performing QAOA with digital-analog quantum computation, with QAOA at depth p, with m digital-analog blocks required per layer (where in the best case we simply have $m = 1$), and assuming that the problem parameters $\gamma \approx 1$, one would therefore choose to engineer a resource interaction strength such that the circuit would be finished in a time on the order of the coherence time, so:

$$g_{\text{int}} t_{\text{Coherence}} \approx mp. \tag{6.1}$$

However, work in Chapter 5 demonstrates that the ratio of single-qubit term strength

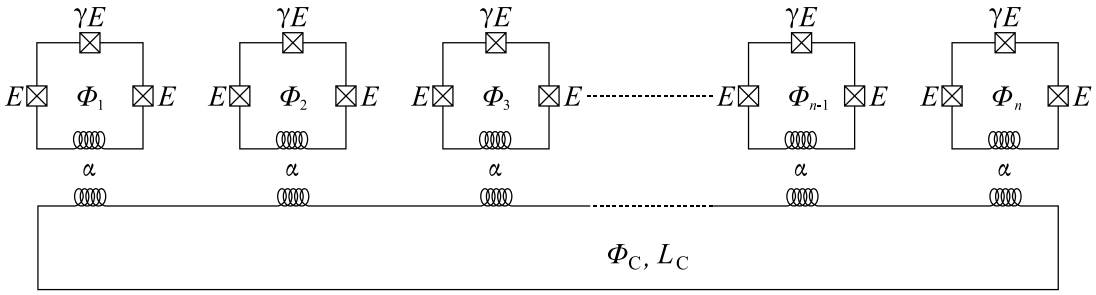

Figure 6.1: A design schematic for a digital-analog QAOA device using inductively coupled flux qubits. Flux qubits are coupled inductively to a central loop such that a homogeneous all-to-all Ising interaction is realized.

to interaction term strength, the speed ratio used in Figure 5.5, should take a value of order ≈ 100 for Digital-Analog QAOA to perform well and retain a problem-specific variational ansatz. This places a limit on circuit depths attainable with a given coherence time. For depth $p = 1$ QAOA, the interaction strength can, therefore, be $m/t_{\text{Coherence}}$ and so the ratio of the single-qubit gate time to the interaction strength will be $t_{\text{Coherence}}/mt_{\text{sq}}$.

Typical single-qubit gates can be performed in $\approx 10\,\text{ns}-20\,\text{ns}$ [160] and coherence times for transmon and flux qubits are approximately $30\,\mu\text{s}$ to $70\,\mu\text{s}$ [161, 162], giving a ratio between the single-qubit and interaction times of $1500 - 7000$. Taking into account a factor of 100 for the minimum speed ratio for a relevant QAOA variational ansatz, this allows for QAOA depths of up to $15 - 70$ with problem-specific device connectivity, or a product of DA-interaction time, and p, equaling such a value. In a device with coupling inhomogeneity, as would be found in some inductively interacting flux qubits, one might only attain a QAOA depth of $p = 1$, given the high, worst-case interaction times needed. Information as to required DA interaction times can be found in Figure 5.4, where the cost displayed in this figure, multiplied by the QAOA depth p is the constrained quantity.

6.4 Device Proposals

The most basic designs one could construct that produces the desired all-to-all resource interaction involve engineering a device such that transverse interactions produce the resource Hamiltonian of the digital-analog scheme. For flux qubits, transverse interactions are commonly used, as such interactions naturally arise when considering such qubits in the persistent current basis. In the context of charge and transmon qubits, such interactions, named parasitic interactions, arise from couplings with higher energy levels [163], and are typically unwanted, therefore engineered to be as small as possible, or removed. While we focus on the case of the flux qubits, one could also explore the use of parasitic interactions on transmon qubits, to perform the digital-analog scheme for QAOA of Chapter 5.

6.4.1 A Flux-Qubit Device

Figure 6.1 presents a proposed circuit diagram for a device in which flux qubits are inductively coupled together via a global resonator. Because flux qubits have, as their basis states, persistently flowing macroscopic currents, we can mediate interactions between them by coupling each, inductively, to a shared central resonator. The author made use of the resources of [164, 165] in the development of this section, with the first [164] providing a general overview and the second providing a specific use-case [165].

To determine the form of a resulting Hamiltonian from such a device, including the interaction Hamiltonian, we can begin with the classical Kirchhoff circuit laws alongside the Josephson relation and the loop flux quantization condition. Kirchhoff's law conserves current flowing into and out of any element of an electrical circuit and ensures that the sum of changes in electric potential around any loop is zero. Therefore,

$$\sum_{\text{node edges}} I_{\text{edge}} = 0, \quad \sum_{\text{loop edges}} V_{\text{edge}} = 0. \tag{6.2}$$

The central superconducting loop of the device[1] is characterized by a current I_{C} induced by the inductively coupled qubits. This current is associated with a superconducting phase ϕ_{C} which is locked via the flux quantization condition to match the external flux through the loop, which can be set to zero. For the flux qubits surrounding, one has that the current flowing through the Josephson qubit elements is equal to that of the qubit inductor, via application of the current law around each loop, so

$$I_{\text{Qubit}} = I_{\text{Qubit inductor}}. \tag{6.3}$$

Where we always take the convention that current is defined as positive in a clockwise direction, and denote the jth qubit current as I_j There is a per-qubit loop quantization requirement written as

$$\Phi_{\text{Inductor}} + \Phi_{\text{Qubit, } j} = \Phi_{\text{ext, } j} + 2k\pi \tag{6.4}$$

in which the external parameter is typically utilized to perform single-qubit gates, but is set about a value of a half-integer number of flux quanta[2] to result in the desired double well potential. For the low energy states considered, we use $k = 0$ always here. Optionally, a Josephson junction can be replaced by a SQUID to allow for additional tunability. Using the flux quantization condition on the mediating loop, one can write that

$$L_{\text{C}} I_{\text{C}} - \sum_j M_j I_j + \Phi_{\text{C,ext}} = 2k\pi, \tag{6.5}$$

in which the negative signs occur due to the oppositely induced current to that of the qubits. The external flux is set to zero, and we assume that for small currents, $k = 0$ also. M_j is the mutual inductance between qubit loop j and the coupler loop,

[1] The electrical resistance of any circuit element used for a qubit-based application must have zero or negligible electrical resistance.

[2] Other designs, namely those utilizing π junctions [166, 167], allow for the degeneracy point to be moved to correspond to an external flux of zero, reducing control complexity when this operating point is used.

given by $\alpha\sqrt{L_j L_{\mathrm{C}}}$, where $0 < \alpha < 1$ is the coupling coefficient between the two inductors, which can be made small and large depending on the spatial geometries of the inductor loops. Assuming that zero external flux is applied to the coupler loop, the current conservation condition becomes

$$I_{\mathrm{C}} = \sum_j \frac{M_j}{L_{\mathrm{C}}} I_j \tag{6.6}$$

as the coupler loop's state is entirely determined by that of the surrounding qubits. For each qubit, the total flux through the inductor can be written as:

$$I_j L_j + M_{\mathrm{J}} I_{\mathrm{C}} = \Phi_j, \tag{6.7}$$

where I_j is the qubit current. We can isolate I_j to be eliminated, so:

$$I_j = \Phi_j / L_j - M_j / L_j I_{\mathrm{C}}, \tag{6.8}$$

and substitute this into Equation 6.6 to obtain a formula for the coupler current in terms of the qubit inductor fluxes alone. This gives:

$$I_{\mathrm{C}} = \frac{\Phi_{\mathrm{C}} - \sum \alpha_j \Phi_j}{(L_{\mathrm{C}} - \sum_j \alpha_j M_j)}. \tag{6.9}$$

This suggests the re-scaling of the coupler inductance to account for the qubit coupler interactions as $L_{\mathrm{C}} \to \tilde{L}_{\mathrm{C}}$ with

$$\tilde{L}_{\mathrm{C}} = L_{\mathrm{C}} - \sum_j \alpha_j M_j, \tag{6.10}$$

and so we can simply write the current as

$$I_{\mathrm{C}} = \frac{-\sum \alpha_j \Phi_j}{\tilde{L}_{\mathrm{C}}}. \tag{6.11}$$

Equating the qubit current I_j^* to the current flowing through the qubit inductor, one obtains:

$$I_j^* = \Phi_j / L_j - \frac{\alpha_j \sum_{j'} \alpha_{j'} \Phi_{j'}}{\tilde{L}_{\mathrm{C}}}, \tag{6.12}$$

where, note, $\alpha_j = M_j / L_j$. That equation can be then identified as an equation of motion, written now in the phases alone, as would be determined via the Euler-Lagrange equation from the system Lagrangian as:

$$\frac{\partial}{\partial \Phi_j} \mathcal{L}(\Phi, \dot{\Phi}) = \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial}{\partial \dot{\Phi}_j} \mathcal{L}(\Phi, \dot{\Phi}). \tag{6.13}$$

We can then write the system Lagrangian as

$$\mathcal{L}_j(\Phi, \dot{\Phi}) = \frac{\alpha_j}{\tilde{L}_{\mathrm{C}}} \sum_{j'} \alpha_{j'} \Phi_{j'} \Phi_j + \sum_j \mathcal{L}_j(\dot{\Phi}_j, \Phi_j), \tag{6.14}$$

where one can verify that for a simple single-junction flux qubit at a typical operating point, one has

$$\mathcal{L}_j(Q_j, \Phi_j) = \frac{\dot{\Phi}_j}{2C} + \frac{\Phi_j^2}{2L_j} + E_j \cos(\Phi_j), \tag{6.15}$$

for which the Euler-Lagrange equation produces the correct equation of motion in the absence of the coupling term as:

$$\dot{Q} = \frac{\Phi_j}{L_j} - E_j \sin(\Phi_j) = \frac{\Phi_j}{L_j} - I_j^* = 0. \tag{6.16}$$

As a Hamiltonian this returns simply the same expression with, however, a positive contribution from the potential (phase) energy:

$$\hat{H} = \left(\sum_j \dot{\Phi}_j \partial_{\dot{\Phi}_j} - 1 \right) \mathcal{L}(\Phi, \dot{\Phi}) = \sum_j \hat{H}_j(\Phi, \dot{\Phi}) + \frac{\alpha_j}{\tilde{L}_\mathrm{C}} \sum_{j'} \alpha_{j'} \Phi_{j'} \Phi_j, \tag{6.17}$$

which is a transverse, all-to-all, ZZ-type interaction, when projected in the persistent current basis with a sufficiently large barrier.

Choice of flux qubit and operating regime

What remains is to determine parameter regimes for the design of the device and the desired interaction strengths. The obvious operating regime for a flux qubit for the purpose of digital-analog computing is to be far from the degeneracy point. Away from the degeneracy point, the qubit eigenbasis is made close to that of the persistent current basis, and so, the approximation of interactions as longitudinal becomes closer to exact in the qubit basis. To design to this specification, one can either increase the difference in energy of the two-qubit states, localized to the wells of the double well potential, or, increase the size of the barrier separating the qubits. The downside to this configuration, and the reason why many modern iterations of flux qubits do not operate at this point, is the sensitivity of the qubit to external flux noise, resulting in shorter coherence times. However, recent advances in qubit fabrication have proved successful in reducing this with absorbed O_2 molecules being a major cause of such errors [168].

The ideal form of a flux qubit for this application would be the three-junction flux qubit, where one junction is split to form a SQUID. Such a design allows for external control of the barrier height—the parameter γ on Figure 6.1—resulting in the capacity for strong transverse drive required for fast single-qubit gates.

The desired ratio between the many two-body terms and the maximum attainable single-qubit-X strength should be greater, and perhaps significantly greater, than 100, as determined in Chapter 5. With 10 ns single-qubit gate times, at a frequency of 200 MHz, this means two-body terms with a strength of around or less than 2 MHz. The strength of the interactions in Equation 6.17, assuming parameters for each qubit are set equally, are

$$E_\mathrm{int} = \frac{\alpha^2}{L_c - n\alpha^2 L_j} L_j^2 I^{*2}, \tag{6.18}$$

where the current I^* is that of the persistent current in the flux qubits. Due to the speculative nature of this work, determining exact figures for system parameters is not useful, but order-of-magnitude estimates for where such parameters might lie, and to which regimes they could be engineered to operate in, can instruct as to how a device might be used.

Early work showing that flux qubits, around 10 µm in size could be entangled with inductive interactions [169] used qubit inductances of 39 pH and critical currents of 400 nA and persistent currents of 320 nA. In this setting, the mutual inductance between the two qubits was estimated to be 2.7 pH. This means a value for α of this device around the 1% level, resulting in an interaction at 410 MHz. This interaction was presumably engineered to be particularly strong due to the constraints on coherence time in such an early experiment, being an early demonstration of entanglement with flux qubits.

Work proposing more coherent quantum annealing proposes qubits operating with a lower persistent current magnitude of 180 nA [50] compared to the value of 3 µA used by the D-Wave systems [170], with qubit inductances of 265 PH. Work involving resonators used to couple qubits in the literature involve inductances on the order of 1.8 nH [171], 1 nH [172]. Assuming persistent currents at the lower 100 nA and qubit inductance of 50 pH, with $\alpha = 0.01$ to target an interaction strength of 2 MHz, and, a 10 qubit device, this fixes the reduced central resonator inductance at a value of 2 pH which is smaller than typically utilized by two orders of magnitude. Greater coupler inductances could be obtained via the usage of smaller persistent currents, qubit inductances, or coupling coefficients. Due to the non-homogeneous contribution of qubit-qubit direct inductive coupling, it may well be favorable to utilize a large loop with small mutual inductance with individual qubits, but such that the qubits are spatially separated such that the direct interactions are negligible.

6.4.2 Charge Qubits and Transverse Interactions

In charge and transmon qubits, longitudinal interactions are not so readily available as in flux qubits away from the degeneracy point. While such interactions exist, it is rare to find an interaction that is purely longitudinal as is the case for persistent current basis flux qubits. Charge and transmon qubits typically use capacitor-mediated interactions, which result in an XX-type coupling in the qubit basis. XY-type interactions, generating iSWAP gates are also available, being considered more expressive than the pure XX or YY interaction types [173]. Considering a charge, or transmon qubit analog of the longitudinally coupled flux qubit device, one could construct a device with all qubits capacitively coupled to a central lumped charge, as seen in Figure 6.2. Alternatively, a similar type interaction could be obtained by using the flux qubit device of Figure 6.1 with the qubits operating at the first order flux insensitive degeneracy point. The challenge of using such transverse interactions, while fine for typical uses in quantum computing with tunable interactions, is problematic when one considers always-on interactions since the qubits are not stationary in the basis on which the interaction acts. It is not, then, obvious how to extract digital-analog-like dynamics in this context, as is in the context of transverse interactions. While work has considered the use of transmon qubits for digital-analog computation with the cross resonance effect [140], such work utilizes a restricted connectivity setting in which capacitively coupled qubits can be selectively brought into resonance with their neighbors, compared to the high connectivity envisaged for the implementation of digital-analog QAOA.

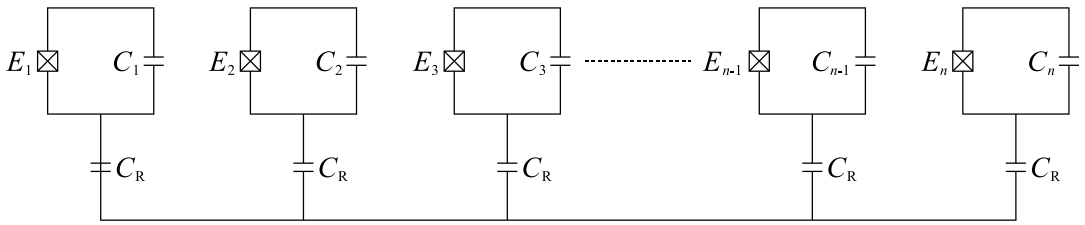

Figure 6.2: A DA-QAOA device design schematic for use with charge qubits. Charge qubits are coupled capacitively to a central charge island such that a close-to-homogeneous all-to-all interaction is realized. As a capacitive interaction, the interaction is transverse to the qubit eigenbasis and would require more complicated control schemes than in the case of persistent current qubits.

6.4.3 Galvanic Coupling of Flux Qubits

Galvanic coupling might be a favorable method for use in a device for digital-analog QAOA, primarily due to the interaction deriving from kinetic, rather than geometric inductance. Rather than placing qubits and coupler loops in proximity to each other, such that mutual inductance leads to interactions, galvanic coupling physically wires circuits together, resulting in potentially stronger interactions. First considered by Levitov et al. [174] in 2001, and developed by Massen van den Brink in 2006 [175], the topic still sees attention as a viable scheme for coupling flux qubits [176]. When using coupling of this type, one is less bound to the constraints of interactions that typically scale as the cube of the distance between loops, embedded onto a 2d plane. Galvanic interactions do, however, typically result in particularly strong interactions, which is often cited as an advantage, but, in the context of always-on digital-analog interactions, might be disadvantageous and induce greater error.

The galvanic analog of the flux qubit design presented in Figure 6.1 is presented in Figure 6.3. In the following treatment, we simply minimize the classical potential induced by the loop, which gives an approximation of the form of the interaction potential. A more full treatment could consider the coupler loop with quantum degrees of freedom and apply a Schrieffer–Wolff transform as is done in works where more exotic regimes are required, for example, when one wants greater than two-body interactions to be present [177]. Note, the following assumes inductances are vanishingly small, such properties could be attained by manufacturing qubits to have a small size, with the double well potential deriving only from contributions from Josephson junctions. To start the analysis of the potential landscape in Figure 6.3, we may write down a Hamiltonian with cosine terms for each Josephson junction in the device. The indexing scheme for the phases ϕ_{ij} is such that the first index refers to the qubit, with the second referring to one of the three-qubit junctions, where the indices $1, 2$ refer to the identical junctions running vertically. Indices 0 and 3 refer to the junctions associated with α and γ respectively. ϕ_{C} refers to the phase across the single Josephson junction associated with the parameter β. Uppercase Φ refers to an externally applied flux, which is expressed in dimensionless units of the

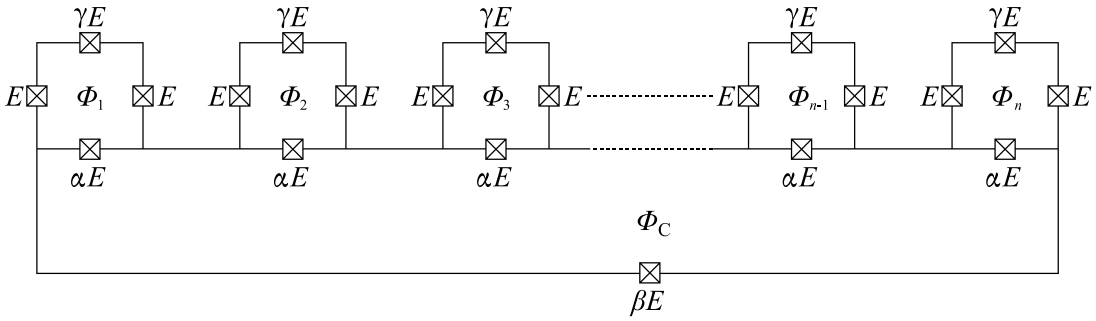

Figure 6.3: A device proposal in which three-junction flux qubits are wired directly to a coupler loop, such that a homogeneous all-to-all Ising interaction is realized via kinetic inductance.

flux quantum.

$$\mathcal{U}\left(\vec{\phi}\right) = -\sum_{i\in[0,n],j\in\{1,2\}} E\cos(\phi_{ij}) - \sum_{i\in[0,n]} \gamma E\cos(\phi_{i3}) - \sum_{i\in[0,n]} \alpha E\cos(\phi_{i0}) - \beta E\cos(\phi_{\mathrm{C}}). \quad (6.19)$$

Eliminating one degree of freedom per loop using the flux quantization condition

$$\Phi_i = \sum_{j\in\{0,1,2,3\}} \phi_{ij},\ \phi_{\mathrm{C}} = \sum_i \phi_{i0} + \Phi_{\mathrm{C}}, \quad (6.20)$$

results in the formula:

$$\mathcal{U}\left(\vec{\phi}\right) = -\sum_{i\in[0,n],j\in\{1,2\}} E\cos(\phi_{ij}) - \beta E\cos\left(\Phi_{\mathrm{C}} + \sum_i \phi_{i0}\right) - \sum_{i\in[0,n]} \left(\gamma E\cos(\Phi_i - \phi_{i0} - \phi_{i1} - \phi_{i2}) + \alpha E\cos(\phi_{i0})\right). \quad (6.21)$$

Minimizing this potential using $\partial_{\phi_{ij}}\mathcal{U} = 0$ for the indices $j = 1, 2$, one finds the qubit current conservation condition:

$$E\sin(\phi_{i1}) = \gamma E\sin(\Phi_i - \phi_{i0} - \phi_{i1} - \phi_{i2}) = E\sin(\phi_{i2}) \quad (6.22)$$

for each qubit, showing that for low-lying states, the remaining loop phases remain identical and $\phi_{i1} = \phi_{i2}$. A second set of equations can be derived by minimizing with respect to the junction phases linking the flux qubits to the coupler loop, ϕ_{i0}, as:

$$\gamma\sin(\Phi_i - \phi_{i0} - \phi_{i1} - \phi_{i2}) = \alpha\sin(\phi_{i0}) + \beta\sin\left(\Phi_{\mathrm{C}} + \sum_i \phi_{i0}\right) = \sin(\phi_i). \quad (6.23)$$

Which is a current conservation condition dictating that the current around the coupler loop should be equal to the current flowing around the top of a qubit loop plus the current flowing around the bottom of a qubit loop. By setting $\alpha >> 1$ one

ensures that the inner loop phase is small relative to the qubit phases. Using this approximation, we can solve the first equation to obtain the usual phase minima of flux qubits of $\cos(\phi) = \frac{1}{2\gamma}$. Applying zero flux bias to the coupling loop and using the small angle approximation for the inner junction phases, one obtains the current condition

$$\alpha\phi_{i0} + \beta\sum_i \phi_{i0} = \sin(\phi_i) \tag{6.24}$$

by the symmetry of the Hamiltonian, assume that the internal ϕ_{i0} phases can take one of two values ϕ_+, ϕ_- depending on whether the associated qubit state has a left- or right-circulating current. For positive, and negative phase qubits, this equation will then read

$$\alpha\phi_+ + \beta\sum_i \phi_{i0} = \sin(\phi),\ \ \alpha\phi_- + \beta\sum_i \phi_{i0} = -\sin(\phi). \tag{6.25}$$

Assuming k qubits have positive phase and $n-k$ have negative phase, combining these equations gives:

$$\alpha\sum_i \phi_{i0} = (2k-n)\sin(\phi) - n\beta\sum_i \phi_{i0}, \tag{6.26}$$

and can be rearranged to:

$$\beta\sum_i \phi_{i0} = \beta\frac{2k-n}{\alpha+n\beta}\sin(\phi) = w\sin(\phi). \tag{6.27}$$

Now define w, which can be interpreted as the magnetization of the system,

$$w = \frac{2k-n}{\alpha+n\beta}\beta, \tag{6.28}$$

and so Equations 6.25 can be combined to eliminate the sum of internal phases to produce:

$$\phi_\pm = \frac{-\sin(\phi)}{\alpha}(w \pm 1). \tag{6.29}$$

Plugging these values into the relevant terms in the potential energy and using a small angle approximation, one obtains the usual single-qubit potential, but, also, an interaction term given by

$$\mathcal{U}_{\rm int}(\vec{\phi}_{i0}) = -\sum_{i\in[0,n]}(\alpha E\cos(\phi_{i0})) - \beta E\cos\left(\Phi_{\rm C} + \sum_i \phi_{i0}\right), \tag{6.30}$$

which can be expressed in terms of the magnetization w as:

$$\mathcal{U}_{\rm int}(k)/E_{\rm J} = -\alpha\left[k\cos\left(\frac{\sin(\phi)}{\alpha}(w+1)\right) + (n-k)\cos\left(\frac{\sin(\phi)}{\alpha}(w-1)\right)\right] - \beta\cos\left(\frac{w\sin(\phi)}{\beta}\right). \tag{6.31}$$

With a small angle approximation, for large α, this gives:

$$\mathcal{U}_{\rm int}(k)/E_{\rm J} = \frac{\sin(\phi)^2}{2\alpha}\left(k\left((w+1)\right)^2 + (n-k)\left((w-1)\right)^2\right) + \frac{\sin(\phi)^2 w^2}{2\beta}. \tag{6.32}$$

This can then be expressed as:

$$\mathcal{U}_{\text{int}}(k) \approx \frac{3nw(k)^2 \sin(\phi)^2 E_{\text{J}}}{2\alpha} = E^*(n-2k)^2. \tag{6.33}$$

and, finally, this, as demonstrated in Appendix 6.5, can be finally expressed in terms of ZZ interactions as:

$$\hat{H}_{\text{int}} = \frac{3n\beta^2 \sin(\phi)^2 E_{\text{J}}}{\alpha(\alpha+n\beta)} \sum_{i<j}^{n} Z_i Z_j, \tag{6.34}$$

which is the desired form. While such an interaction can be made strong in the case that β is made large and α small, breaking the assumptions of the section, for the low interaction strengths needed for digital-analog QAOA, one can fabricate α as large as needed and β as small as necessary, to result in the appropriate strength.

6.5 Conclusion

To conclude the section, flux qubit devices for digital-analog QAOA could be viable and could be a promising area for experimental attention. Regardless, scalability could prove challenging due to the quadratic circuit depth incurred in the interaction time required for a computation. Such a device for digital-analog QAOA should also be extremely well characterized in its interaction, as uncertainty in the form of a resource interaction formed would result in a QAOA ansatz state not resembling the ideal version attained by a noiseless gate-based implementation. Building highly symmetric devices, as proposed, could help with issues of heterogeneous interaction, as symmetry in the design manifests in symmetry in the interactions formed.

In the near term of noisy quantum computation at an intermediate scale of $10-100$ qubits, devices such as the D-Wave annealer have been seen as a promising step and a potential way to leverage quantum advantage, even if theoretical proof of quantum advantage cannot be attained. Devices and schemes such as the one presented in this and the previous chapter, therefore, might offer an alternative route to the scalability of useful quantum computers, albeit, with more coherent qubits.

Appendix

6.A Weight-Dependent Potential to ZZ Interactions

To calculate the effective interactions yielded by a potential term, quadratic in the weight of the state of a set of qubits, we can use a n-bit binary Fourier transform of the energy spectrum resulting from the multi-qubit interaction. The device interaction potential in Equation 6.33 can be expressed as:

$$\hat{U} = \sum_{\vec{s}\in[0,1]^n} g(\vec{s}) Z^{s_0} Z^{s_1} \dots Z^{s_n}, \tag{6.35}$$

where

$$g(\vec{s}) = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n} \mathcal{U}(k)(-1)^{\vec{x}\cdot\vec{s}}. \tag{6.36}$$

Substituting in our potential, where k is the weight of a string $\vec{x}$ gives

$$g(\vec{s}) = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n} (n-2k)^2 (-1)^{\vec{x}\cdot\vec{s}}. \tag{6.37}$$

Consider a value of g corresponding to a single-qubit Z_j interaction. For such a term, $s_i = \delta_{i,j}$ and so

$$g(\delta_{i,j}) = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n} (n-2k)^2 (-1)^{x_j} \tag{6.38}$$

$$= \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n,\not\ni_j} (n-2k)^2 - (n-2-2k)^2 \tag{6.39}$$

$$= \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n,\not\ni_j} -4 + 4n - 8k. \tag{6.40}$$

Evaluating the average over the weight of all remaining bit strings gives a value of $k = (n-1)/2$ which, substituted into the expression gives:

$$= \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n,\not\ni_j} 4(n-1) - 8(n-1)/2 = 0, \tag{6.41}$$

meaning that the interaction Hamiltonian results in no single-qubit terms.

Considering possible two-qubit ZZ interactions, one takes $s_i = \delta_{i,j_1} + \delta_{i,j_2}$ and, substituting this into the binary Fourier transform, one obtains:

$$g(\vec{s}) = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n} (n-2k)^2 (-1)^{\vec{x}\cdot\vec{s}} = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n} (n-2k)^2 (-1)^{x_{j_1}+x_{j_2}}$$
$$= \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n, \not{x}_{j_1}, \not{x}_{j_2}} (n-2k)^2 - 2(n-2-2k)^2 + (n-4-2k)^2, \quad (6.42)$$

which simplifies to

$$= \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n, \not{x}_{j_1}, \not{x}_{j_2}} n^2 - 4nk + 4k^2 - 2(n^2 + 4 + 4k^2 - 4n - 4nk + 8k)$$
$$+ (n^2 + 16 + 4k^2 - 8n - 4nk + 16k) = \frac{1}{2^n} \sum_{\vec{x}\in[0,1]^n, \not{x}_{j_1}, \not{x}_{j_2}} 8 = 2. \quad (6.43)$$

This means that the interaction produces all ZZ interactions with a pre-factor of 2. Now we can show that the higher order terms evaluate to zero, and determine how the general case evaluates. Taking an n^*-local term, we can expand the Fourier coefficient as:

$$g(n^*) = \sum_{\vec{x}\in[0,1]^{n-n^*}} \sum_{p=0}^{n^*} \binom{n^*}{p} (n-2p-2k)^2 (-1)^p, \quad (6.44)$$

which can be expanded as a polynomial in p

$$\sum_{\mathrm{p}} \binom{n^*}{p} (n^2 + 4p^2 + 4k^2 - 4np - 4kn + 8pk)(-1)^p \quad (6.45)$$

$$\sum_{\mathrm{p}} (n^2 - 4kn + 4k^2)\binom{n^*}{p}(-1)^p - 4(n-2k)\binom{n^*}{p}p(-1)^p + 4\binom{n^*}{p}p^2(-1)^p. \quad (6.46)$$

To evaluate these sums we can use the binomial theorem,

$$(x+1)^n = \sum_{p}^{n} \binom{n^*}{p} x^p. \quad (6.47)$$

Setting $x = -1$ demonstrates that the first term in the sum is zero for all $n > 0$. To obtain the second term, one can operate with $x\partial_x$ and $x\partial_x x\partial_x$ under the sum, giving

$$nx(x+1)^{n-1} = \sum_{p}^{n} \binom{n^*}{p} p x^p \quad (6.48)$$

and

$$nx(x+1)^{n-1} + nx^2(x+1)^{n-2} = \sum_{p}^{n} \binom{n^*}{p} p^2 x^p. \quad (6.49)$$

Setting $x = -1$ in the first of these gives 0 if $n > 1$ and the second gives 0 for all $n > 2$. Therefore, our device interaction potential induces no terms of weight greater than two and can be expressed as required as:

$$\hat{H}_{\mathrm{int}} = \frac{3En\beta^2 \sin(\phi)^2}{\alpha(\alpha + n\beta)} \sum_{i<j}^{n} Z_i Z_j. \quad (6.50)$$

Conclusion

Quantum computing technologies are developing rapidly, with vast improvements to state of the art capabilities in the past five years. From the development of many-qubit devices sustaining error rates sufficient to exploit their size for algorithmic uses, to the first demonstration of quantum error correction codes that incur decreasing logical error rates with increasing code distance [178]. While such developments are impressive, there exists, still, a gulf between the current state of the technology and the capabilities expected to be required for algorithmic advantage with quantum computing. That is, running Shor's algorithm on a fault-tolerant quantum computer to break 1024-bit RSA encryption would require logical qubit counts in the tens of millions [179], a number that might be reduced by improvements to physical hardware. There is, therefore, a gap for intermediate applications, to make use of devices between those that might exist in the future for full-scale, fault tolerant, quantum computing, and the current era of small devices.

This thesis presents work in this area. In the first half we presented analytical results aimed at understanding the performance of QAOA, and how this performance relates to average properties of the combinatorial optimization problems it solves. Firstly we show in Chapter 3 that performance is calculable in the infinite problem size, or average problem context, when using a Grover driver. Building on this work, in Chapter 4, we expand the context to non-Grover drivers, and demonstrate that we can obtain a general form for QAOA expectation values that depend only on problem-specific conditional probabilities. Therefore, we can calculate the performance of the algorithm on entire ensembles based on their statistics, and structure in the problems with respect to the driver chosen. We use this method to compare the performance of QAOA with two different drivers, that of the standard single-qubit-X drivers and that of line mixers. We use the random cost model for this comparison, determining that the Grover driver and the X driver perform identically at depth 1, with the line mixer underperforming the two on random cost problems. The primary objective of this research, then, might be to identify statistical properties from a problem, allowing for a less computationally demanding calculation of QAOA performance and optimal QAOA angles, instead of training QAOA on a single problem instance. This could be especially important for problems of large scale, for which procedures that take time polynomial in the problem size would have high utility for calculating optimal QAOA angles and expected performance. Further work could investigate the difficulty of determining relevant problem statistics from a given problem, with sub-exponentially scaling resource requirements in problem size, and also, to the problem of efficiently evaluating QAOA expectation values at depths greater than 1, a capability that the methods Chapter 3 succeeds in, but Chapter 4 falls short of. Further development in this area could negate the need for training of QAOA circuits in the way initially conceived, in which a noisy estimate

of the expected value of the state output by the quantum computer is iteratively improved with the use of a classical optimizer.

In the second part of the thesis, in Chapter 5 and 6, a novel scheme for performing QAOA with always on interactions is investigated. Building on previous applications of the digital-analog paradigm of quantum computing, we found that variational algorithms, and QAOA in particular, yield natural synergies to the digital-analog context. Such synergies occur because QAOA is naturally resilient to systematic coherent error, which is the variety of error introduced by the banged digital-analog scheme, with a resource Hamiltonian remaining on for the duration of a calculation. We showed the regimes of interaction strength, compared to the speed at which single-qubit operations could be performed, at which QAOA, performed in the error-inducing digital-analog scheme, performed comparably to an ideal error-free implementation of QAOA. We solved issues with the implementation that arose, including those of negative interaction times resulting from the derivation of digital-analog block times. Finally, we presented a blueprint for a flux-qubit-based device that could take advantage of the digital-analog context for QAOA. By connecting a number of flux qubits to a resonator, a homogenous interaction could be produced, ideal for digital-analog QAOA. As in the case of quantum annealing and the D-Wave quantum annealer, such ideas could give way to new devices that are not designed for universal quantum computation, but might leverage the engineering effort levied on building fault tolerant devices, to provide potentially useful, coherent machines for the solution of combinatorial optimization problems in the near term.

Appendix A

QAOA Simulation With Fast Walsh-Hadamard Transforms

To study the properties of QAOA, it must be simulated. While there will always be an upper limit to the size of a system one can completely simulate on a classical computer, the structure can be exploited to allow the largest possible system sizes to be tractable. In this section we demonstrate a technique for simulating QAOA at a basic level, allowing full control over the driver and problem Hamiltonians. A comparison to the Google CIRQ package is made.

The simulation of QAOA or other quantum algorithms with n qubits and thus $N = 2^n$ basis states, requires, at worst, that at each time step, one performs multiplies a matrix of size $N \times N$ by a length N vector. In this work, a Python package has been developed for accelerated simulation while maintaining a high level of flexibility and ease of driver selection. A typical QAOA circuit simulated in a Python package takes an initial state $|+\rangle$ and applies driver and problem Hamiltonians until all operators are applied. This state is then:

$$\left|\vec{\gamma}, \vec{\beta}\right\rangle = \hat{U}_{\mathrm{D}}(\beta_p)\hat{U}_{\mathrm{P}}(\gamma_p)\ldots\hat{U}_{\mathrm{D}}(\beta_1)\hat{U}_{\mathrm{P}}(\gamma_1)\left|+\right\rangle, \tag{A.1}$$

where

$$\hat{U}_{\mathrm{P}} = \exp\Big(i\gamma\,\mathrm{diag}(\vec{E}_c)\Big) = \mathrm{diag}(\exp\Big(i\gamma\,\vec{E}_c\Big). \tag{A.2}$$

Likewise, one can express the driver unitary similarly as

$$\hat{U}_{\mathrm{D}} = H^{\otimes n}\exp\Big(i\beta\,\mathrm{diag}(\vec{E}_X)\Big)H^{\otimes n} = H^{\otimes n}\mathrm{diag}(\exp\Big(i\beta\,\vec{E}_X\Big)H^{\otimes n}, \tag{A.3}$$

which means that since the Hadamard transforms can be performed in $O(N\log(N))$, with the fast Walsh-Hadamard transform, rather than the worst case $O(N^2)$ scaling from naive multiplication, application of the driver and problem unitaries can be applied via element-wise multiplication by a complex number in time $O(N)$. To obtain $\vec{E}_X$, consider that surrounded by Hadamards, the single-qubit-X operators become single-qubit-Z operators as $HXH = Z$. Combining this with the n-bit binary Fourier transform method for converting between energy and interaction representations of our problem Hamiltonian, one can find the energy representation of $\hat{H}_{\mathrm{D}}$ as

$$\hat{H}_{\mathrm{D}} = \sum_k h_k X_{k_1}X_{k_2}...X_{k_n} = H^{\otimes n}\left[\sum_k h_k Z_{k_1}Z_{k_2}...Z_{k_n}\right]H^{\otimes n}, \tag{A.4}$$

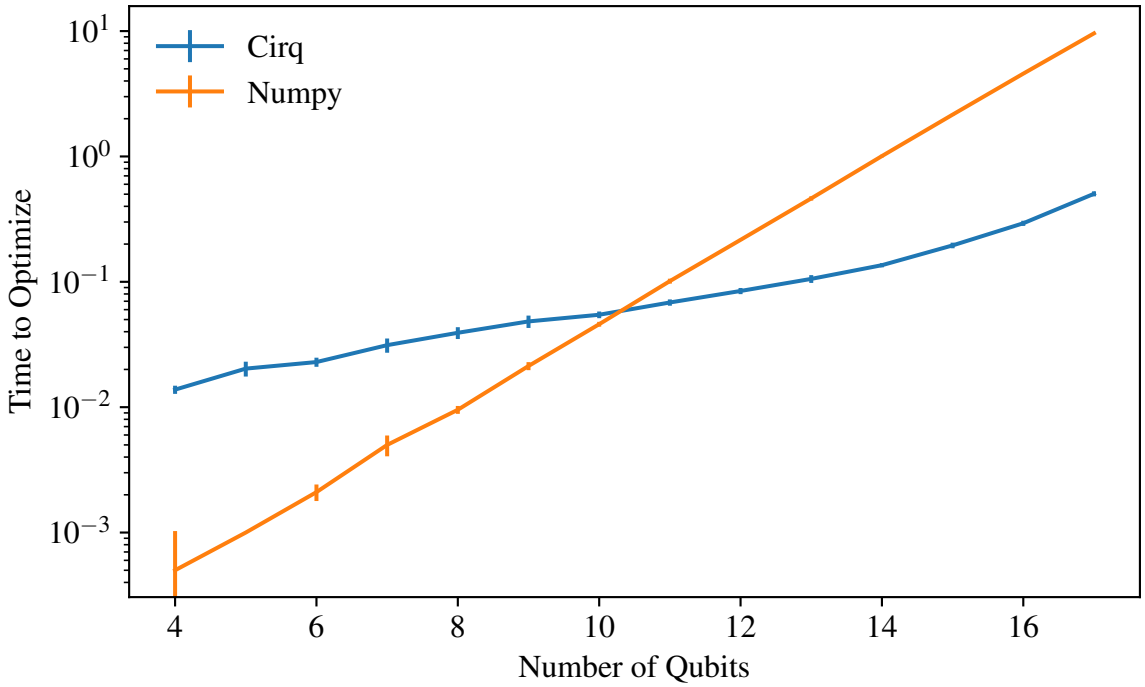

Figure A.1: A comparison between the time to optimize a depth-5 QAOA circuit with a problem Hamiltonian deriving from an instance of the number partitioning problem. The NumPy implementation described in this section is lightweight and results in a faster runtime for qubit counts below 10.

from which one can then take the Hadamard transform of $\vec{h}$ to obtain $\vec{E}_X = H^{\otimes n}\vec{h}$. The missing ingredient here is the ability to apply the n-qubit Hadamard gate without explicitly writing down its full $2^n \times 2^n$ matrix in memory to do this, one uses the fast Walsh-Hadamard transform.

A.1 Fast Walsh-Hadamard Transforms

The Fast Walsh-Hadamard Transform [180] of a vector is a method to compute the Walsh-Hadamard transform of a vector, using $N \log N$ additions or subtractions, in comparison to the N^2 operations required for a naive approach. One uses the recursive definition of the Walsh-Hadamard transform and uses a divide-and-conquer approach to perform the calculation.

$$H_n = \frac{1}{\sqrt{2}}\begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}. \tag{A.5}$$

Which is equivalent to applying the single-qubit Hadamard gates to an array from top to bottom, without expressing the single-qubit gates in the full n-qubit-spanned space.

A comparison of the aforementioned method with a state-of-the-art simulator written by Google is presented in figure A.1. It can be seen that for small instances the NumPy vector implementation with fast Hadamard-Walsh transforms performs better by an order of magnitude at low qubit counts but suffers from a poorer overall asymptotic scaling. The cause of this scaling is expected to be due to the

Google simulator exploiting the local nature of the problem solved in this plot. For differing driver choices, for example, the line mixers used in Chapter 4, encoding in Cirq becomes difficult and one might see better performance from the NumPy implementation.

Bibliography

[1] David Headley, Thorge Müller, Ana Martin, Enrique Solano, Mikel Sanz, and Frank K. Wilhelm. Approximating the quantum approximate optimization algorithm with digital-analog interactions. Phys. Rev. A, 106:042446, Oct 2022.

[2] David Headley and Frank K. Wilhelm. Problem-size-independent angles for a grover-driven quantum approximate optimization algorithm. Phys. Rev. A, 107:012412, Jan 2023.

[3] Thorge Mueller, Tobias Stollenwerk, David Headley, Michael Epping, and Frank K. Wilhelm. Coherent and non-unitary errors in zz-generated gates. arXiv preprint arXiv:2304.14212, 2023.

[4] Joschka Roffe, David Headley, Nicholas Chancellor, Dominic Horsman, and Viv Kendon. Protecting quantum memories using coherent parity check codes. Quantum Science and Technology, 3(3):035010, Jun 2018.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

[6] Walther Gerlach and Otto Stern. Der experimentelle nachweis der richtungsquantelung im magnetfeld. Zeitschrift für Physik, 9(1):349–352, 1922.

[7] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? Physical review, 47(10):777, 1935.

[8] John S Bell. On the einstein podolsky rosen paradox. Physics Physique Fizika, 1(3):195, 1964.

[9] Dominik Rauch, Johannes Handsteiner, Armin Hochrainer, Jason Gallicchio, Andrew S Friedman, Calvin Leung, Bo Liu, Lukas Bulla, Sebastian Ecker, Fabian Steinlechner, et al. Cosmic bell test using random measurement settings from high-redshift quasars. Physical review letters, 121(8):080403, 2018.

[10] Richard P Feynman. Simulating physics with computers. In Feynman and computation, pages 133–153. CRC Press, 2018.

[11] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818):97–117, 1985.

[12] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. Journal of statistical physics, 22(5):563–591, 1980.

[13] Richard P Feynman. Plenty of room at the bottom. In APS annual meeting, 1959.

[14] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439(1907):553–558, 1992.

[15] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. In Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, pages 11–20, 1993.

[16] Don Coppersmith. An approximate fourier transform useful in quantum factoring. arXiv preprint quant-ph/0201067, 2002.

[17] Alexander Semenovich Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. Problemy Peredachi Informatsii, 9(3):3–11, 1973.

[18] Peter 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.

[19] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. arXiv preprint quant-ph/9511026, 1995.

[20] Lov K Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996.

[21] Gilles Brassard and Peter Hoyer. An exact quantum polynomial-time algorithm for simon's problem. In Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems, pages 12–23. IEEE, 1997.

[22] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. Phys. Rev. A, 52:R2493–R2496, Oct 1995.

[23] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106, 2000.

[24] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. Physical Review A, 58(2):915–928, Aug 1998.

[25] A Yu Kitaev. Fault-tolerant quantum computation by anyons. Annals of physics, 303(1):2–30, 2003.

[26] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. Physical Review A, 86(3):032324, 2012.

[27] Emanuel Knill. Fault-tolerant postselected quantum computation: Schemes. arXiv preprint quant-ph/0402171, 2004.

[28] Stephen A. Cook. The complexity of theorem-proving procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.

[29] Richard M. Karp. Reducibility among Combinatorial Problems, pages 85–103. Springer US, Boston, MA, 1972.

[30] Katherine Lai and Michel X. Goemans. The knapsack problem and fully polynomial time approximation schemes (fptas). Seminar paper for 18.434: Seminar in Theoretical Computer Science at MIT, 2006.

[31] Michel X. Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM), 42(6):1115–1145, 1995.

[32] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. Commun. ACM, 5(7):394–397, Jul 1962.

[33] J.P. Marques-Silva and K.A. Sakallah. Grasp: a search algorithm for propositional satisfiability. IEEE Transactions on Computers, 48(5):506–521, 1999.

[34] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028, 2014.

[35] Mauro ES Morales, Timur Tlyachev, and Jacob Biamonte. Variational learning of grover's quantum search algorithm. Physical Review A, 98(6):062333, 2018.

[36] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions. Technical report, National Institute of Standards and Technology, Aug 2015.

[37] Max Born and Vladimir Fock. Beweis des adiabatensatzes. Zeitschrift für Physik, 51(3-4):165–180, 1928.

[38] Christof Zalka. Grover's quantum searching algorithm is optimal. Phys. Rev. A, 60:2746–2751, Oct 1999.

[39] T Schoning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039), pages 410–414. IEEE, 1999.

[40] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. Journal of the ACM (JACM), 51(4):671–697, 2004.

[41] Andrew M. Childs. Universal computation by quantum walk. Phys. Rev. Lett., 102:180501, May 2009.

[42] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03, page 59–68, New York, NY, USA, 2003. Association for Computing Machinery.

[43] Neil Shenvi, Julia Kempe, and K. Birgitta Whaley. Quantum random-walk search algorithm. Phys. Rev. A, 67:052307, May 2003.

[44] John M Martinis, Sae Nam, Joe Aumentado, and C Urbina. Rabi oscillations in a large josephson-junction qubit. Physical review letters, 89(11):117901, 2002.

[45] Matthias Steffen, Markus Ansmann, Radoslaw C Bialczak, Nadav Katz, Erik Lucero, Robert McDermott, Matthew Neeley, Eva Maria Weig, Andrew N Cleland, and John M Martinis. Measurement of the entanglement of two superconducting qubits via state tomography. Science, 313(5792):1423–1425, 2006.

[46] A Yu Dmitriev and OV Astafiev. A perspective on superconducting flux qubits. Applied Physics Letters, 119(8):080501, 2021.

[47] JE Mooij, TP Orlando, L Levitov, Lin Tian, Caspar H Van der Wal, and Seth Lloyd. Josephson persistent-current qubit. Science, 285(5430):1036–1039, 1999.

[48] Vladimir E Manucharyan, Jens Koch, Leonid I Glazman, and Michel H Devoret. Fluxonium: Single cooper-pair circuit free of charge offsets. Science, 326(5949):113–116, 2009.

[49] Feng Bao, Hao Deng, Dawei Ding, Ran Gao, Xun Gao, Cupjin Huang, Xun Jiang, Hsiang-Sheng Ku, Zhisheng Li, Xizheng Ma, et al. Fluxonium: an alternative qubit platform for high-fidelity operations. Physical Review Letters, 129(1):010502, 2022.

[50] Sergey Novikov, Robert Hinkey, Steven Disseler, James I Basham, Tameem Albash, Andrew Risinger, David Ferguson, Daniel A Lidar, and Kenneth M Zick. Exploring more-coherent quantum annealing. In 2018 IEEE International Conference on Rebooting Computing (ICRC), pages 1–7. IEEE, 2018.

[51] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. Physical Review A, 76(4):042319, 2007.

[52] Vincent Bouchiat, D Vion, Ph Joyez, D Esteve, and MH Devoret. Quantum coherence with a single cooper pair. Physica Scripta, 1998(T76):165, 1998.

[53] Alexander PM Place, Lila VH Rodgers, Pranav Mundada, Basil M Smitham, Mattias Fitzpatrick, Zhaoqi Leng, Anjali Premkumar, Jacob Bryon, Andrei Vrajitoarea, Sara Sussman, et al. New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds. Nature communications, 12(1):1779, 2021.

[54] Google Quantum AI. Quantum supremacy using a programmable superconducting processor. Nature, 574(7779):505–510, 2019.

[55] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. Applied Physics Reviews, 6(2):021318, 2019.

[56] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. arXiv preprint arXiv:1412.6062, 2014.

[57] Boaz Barak, Ankur Moitra, Ryan O'Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, and John Wright. Beating the random assignment on constraint satisfaction problems of bounded degree. arXiv preprint arXiv:1505.03424, 2015.

[58] Zhang Jiang, Eleanor G Rieffel, and Zhihui Wang. Near-optimal quantum circuit for grover's unstructured search using a transverse field. Physical Review A, 95(6):062317, 2017.

[59] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. Algorithms, 12(2):34, 2019.

[60] Seth Lloyd. Quantum approximate optimization is computationally universal. arXiv preprint arXiv:1812.11075, 2018.

[61] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. A quantum annealing architecture with all-to-all connectivity from local interactions. Science advances, 1(9):e1500838, 2015.

[62] Stuart Andrew Hadfield. Quantum algorithms for scientific computing and approximate optimization. PhD thesis, Columbia University, 2018.

[63] Natasha Feinstein, Louis Fry-Bouriaux, Sougato Bose, and PA Warburton. Effects of xx-catalysts on quantum annealing spectra with perturbative crossings. arXiv preprint arXiv:2203.06779, 2022.

[64] Vishwanathan Akshay, Hariphan Philathong, Mauro ES Morales, and Jacob D Biamonte. Reachability deficits in quantum approximate optimization. Physical review letters, 124(9):090504, 2020.

[65] Lennart Bittel and Martin Kliesch. Training variational quantum algorithms is np-hard. Physical review letters, 127(12):120502, 2021.

[66] John Wallis. A treatise of algebra, both historical and practical. Philosophical Transactions of the Royal Society of London, 15(173):1095–1106, 1685.

[67] Wim Lavrijsen, Ana Tudor, Juliane Müller, Costin Iancu, and Wibe De Jong. Classical optimizers for noisy intermediate-scale quantum devices. In 2020 IEEE international conference on quantum computing and engineering (QCE), pages 267–277. IEEE, 2020.

[68] Aidan Pellow-Jarman, Ilya Sinayskiy, Anban Pillay, and Francesco Petruccione. A comparison of various classical optimizers for a variational quantum linear solver. Quantum Information Processing, 20(6):202, 2021.

[69] John A Nelder and Roger Mead. A simplex method for function minimization. The computer journal, 7(4):308–313, 1965.

[70] Fuchang Gao and Lixing Han. Implementing the nelder-mead simplex algorithm with adaptive parameters. Computational Optimization and Applications, 51(1):259–277, 2012.

[71] M. J. D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation, pages 51–67. Springer Netherlands, Dordrecht, 1994.

[72] Jorge Nocedal and Stephen J Wright. Numerical optimization. Springer, 1999.

[73] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. Mathematical programming, 45(1-3):503–528, 1989.

[74] David J Wales and Jonathan PK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. The Journal of Physical Chemistry A, 101(28):5111–5116, 1997.

[75] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. Evolutionary computation, 9(2):159–195, 2001.

[76] Stefan H Sack and Maksym Serbyn. Quantum annealing initialization of the quantum approximate optimization algorithm. quantum, 5:491, 2021.

[77] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. Nature communications, 9(1):1–6, 2018.

[78] Zoë Holmes, Kunal Sharma, Marco Cerezo, and Patrick J Coles. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. PRX Quantum, 3(1):010313, 2022.

[79] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. Nature communications, 12(1):6961, 2021.

[80] Antonio A Mele, Glen B Mbeng, Giuseppe E Santoro, Mario Collura, and Pietro Torta. Avoiding barren plateaus via transferability of smooth solutions in a hamiltonian variational ansatz. Physical Review A, 106(6):L060401, 2022.

[81] John Preskill. Quantum computing in the nisq era and beyond. Quantum, 2:79, 2018.

[82] Mauro ES Morales, Jacob D Biamonte, and Zoltán Zimborás. On the universality of the quantum approximate optimization algorithm. Quantum Information Processing, 19(9):1–26, 2020.

[83] Zhang Jiang, Eleanor G. Rieffel, and Zhihui Wang. Near-optimal quantum circuit for grover's unstructured search using a transverse field. Physical Review A, 95(6), Jun 2017.

[84] Vishwanathan Akshay, Daniil Rabinovich, Ernesto Campos, and Jacob Biamonte. Parameter concentrations in quantum approximate optimization. Physical Review A, 104(1):L010401, 2021.

[85] D Rabinovich, R Sengupta, E Campos, V Akshay, and J Biamonte. Progress towards analytically optimal angles in quantum approximate optimisation. arXiv preprint arXiv:2109.11566, 2021.

[86] Michael Streif and Martin Leib. Training the quantum approximate optimization algorithm without access to a quantum processing unit. Quantum Science and Technology, 5(3):034008, 2020.

[87] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. Quantum, 6:759, 2022.

[88] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. The quantum approximate optimization algorithm at high depth for maxcut on large-girth regular graphs and the sherrington-kirkpatrick model. arXiv preprint arXiv:2110.14206, 2021.

[89] Sami Boulebnane and Ashley Montanaro. Predicting parameters for the quantum approximate optimization algorithm for max-cut from the infinite-size limit. arXiv preprint arXiv:2110.10685, 2021.

[90] Jahan Claes and Wim van Dam. Instance independence of single layer quantum approximate optimization algorithm on mixed-spin models at infinite size. Quantum, 5:542, 2021.

[91] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature communications, 12(1):1–12, 2021.

[92] Christof Zalka. Grover's quantum searching algorithm is optimal. Physical Review A, 60(4):2746, 1999.

[93] Andreas Bärtschi and Stephan 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.

[94] John Golden, Andreas Bärtschi, Daniel O'Malley, and Stephan Eidenbenz. Threshold-based quantum optimization. In 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pages 137–147. IEEE, 2021.

[95] John Golden, Andreas Bärtschi, Stephan Eidenbenz, and Daniel O'Malley. Evidence for super-polynomial advantage of qaoa over unstructured search. arXiv preprint arXiv:2202.00648, 2022.

[96] Linghua Zhu, Ho Lun Tang, George S Barron, FA Calderon-Vargas, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. Physical Review Research, 4(3):033029, 2022.

[97] Yanzhu Chen, Linghua Zhu, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. How much entanglement do quantum optimization algorithms require? In Quantum 2.0, pages QM4A–2. Optica Publishing Group, 2022.

[98] Stuart Hadfield, Zhihui Wang, Eleanor G Rieffel, Bryan O'Gorman, Davide Venturelli, and Rupak Biswas. Quantum approximate optimization with hard and soft constraints. In Proceedings of the Second International Workshop on Post Moores Era Supercomputing, pages 15–21, 2017.

[99] Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, et al. Sympy: symbolic computing in python. PeerJ Computer Science, 3:e103, 2017.

[100] Marc Mezard and Andrea Montanari. Information, physics, and computation. Oxford University Press, 2009.

[101] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. Springer Science, 35(67-68):7, 1999.

[102] Stephan Mertens. Random costs in combinatorial optimization. Physical Review Letters, 84(6):1347, 2000.

[103] Vadim N Smelyanskiy, Udo v Toussaint, and Dogan A Timucin. Dynamics of quantum adiabatic evolution algorithm for number partitioning. arXiv preprint quant-ph/0202155, 2002.

[104] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. Physical review A, 52(5):3457, 1995.

[105] Mehdi Saeedi and Massoud Pedram. Linear-depth quantum circuits for n-qubit toffoli gates with no ancilla. Physical Review A, 87(6):062318, 2013.

[106] Yong He, Ming-Xing Luo, E Zhang, Hong-Ke Wang, and Xiao-Feng Wang. Decompositions of n-qubit toffoli gates with linear circuit complexity. International Journal of Theoretical Physics, 56(7):2350–2361, 2017.

[107] David H Wolpert and William G Macready. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1):67–82, 1997.

[108] Bin Yan and Nikolai A Sinitsyn. Analytical solution for nonadiabatic quantum annealing to arbitrary ising spin hamiltonian. Nature Communications, 13(1):1–12, 2022.

[109] James G Morley, Nicholas Chancellor, Sougato Bose, and Viv Kendon. Quantum search with hybrid adiabatic–quantum-walk algorithms and realistic noise. Physical review A, 99(2):022339, 2019.

[110] Jérémie Roland and Nicolas J Cerf. Quantum search by local adiabatic evolution. Physical Review A, 65(4):042308, 2002.

[111] Adam Callison, Nicholas Chancellor, Florian Mintert, and Viv Kendon. Finding spin glass ground states using quantum walks. New Journal of Physics, 21(12):123022, 2019.

[112] James Sud, Stuart Hadfield, Eleanor Rieffel, Norm Tubman, and Tad Hogg. A parameter setting heuristic for the quantum alternating operator ansatz. arXiv preprint arXiv:2211.09270, 2022.

[113] Nicholas Chancellor. Domain wall encoding of discrete variables for quantum annealing and qaoa. Quantum Science and Technology, 4(4):045004, 2019.

[114] G Peter Lepage. A new algorithm for adaptive multidimensional integration. Journal of Computational Physics, 27(2):192–203, 1978.

[115] David J Gross and Marc Mézard. The simplest spin glass. Nuclear Physics B, 240(4):431–452, 1984.

[116] B. Derrida. Random-energy model: Limit of a family of disordered models. Phys. Rev. Lett., 45:79–82, Jul 1980.

[117] Jarrod R McClean, Matthew P Harrigan, Masoud Mohseni, Nicholas C Rubin, Zhang Jiang, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Low-depth mechanisms for quantum optimization. PRX Quantum, 2(3):030312, 2021.

[118] Andreas Brinkmann. Introduction to average hamiltonian theory. i. basics. Concepts in Magnetic Resonance Part A, 45(6):e21414, 2016.

[119] Lorenza Viola and Seth Lloyd. Dynamical suppression of decoherence in two-state quantum systems. Physical Review A, 58(4):2733, 1998.

[120] John Preskill. Quantum computing in the nisq era and beyond. Quantum, 2:79, Aug 2018.

[121] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. Nature communications, 5:4213, 2014.

[122] Johannes S Otterbach, Riccardo Manenti, Nasser Alidoust, A Bestwick, M Block, B Bloom, S Caldwell, N Didier, E Schuyler Fried, S Hong, et al. Unsupervised machine learning on a hybrid quantum computer. arXiv preprint arXiv:1712.05771, 2017.

[123] Lucas Lamata, Adrian Parra-Rodriguez, Mikel Sanz, and Enrique Solano. Digital-analog quantum simulations with superconducting circuits. Advances in Physics: X, 3(1):1457981, 2018.

[124] Adrian Parra-Rodriguez, Pavel Lougovski, Lucas Lamata, Enrique Solano, and Mikel Sanz. Digital-analog quantum computation. Physical Review A, 101(2):022305, 2020.

[125] Ana Martin, Lucas Lamata, Enrique Solano, and Mikel Sanz. Digital-analog quantum algorithm for the quantum fourier transform. Physical Review Research, 2(1):013012, 2020.

[126] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. Rev. Mod. Phys., 86:153–185, Mar 2014.

[127] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. Reviews of Modern Physics, 90(1):015002, 2018.

[128] Julia Kempe. Quantum random walks: an introductory overview. Contemporary Physics, 44(4):307–327, 2003.

[129] Vivien M Kendon. A random walk approach to quantum algorithms. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 364(1849):3407–3422, 2006.

[130] Seth Lloyd. Universal quantum simulators. Science, pages 1073–1078, 1996.

[131] D Aharonov, W van Dam, J Kempe, Z Landau, S Lloyd, and O Regev. Adiabatic computation is equivalent to standard quantum computation. arXiv preprint quant-ph/0405098, 2004.

[132] Antonio Acín, Immanuel Bloch, Harry Buhrman, Tommaso Calarco, Christopher Eichler, Jens Eisert, Daniel Esteve, Nicolas Gisin, Steffen J Glaser, Fedor Jelezko, Stefan Kuhr, Maciej Lewenstein, Max F Riedel, Piet O Schmidt, Rob Thew, Andreas Wallraff, Ian Walmsley, and Frank K Wilhelm. The quantum technologies roadmap: a european community view. New Journal of Physics, 20(8):080201, Aug 2018.

[133] Ana Martin, Ruben Ibarrondo, and Mikel Sanz. Digital-analog co-design of the harrow-hassidim-lloyd algorithm. arXiv preprint arXiv:2207.13528, 2022.

[134] Asier Galicia, Borja Ramon, Enrique Solano, and Mikel Sanz. Enhanced connectivity of quantum hardware with digital-analog control. Physical Review Research, 2(3):033103, 2020.

[135] David G Cory, R Laflamme, E Knill, L Viola, TF Havel, N Boulant, G Boutis, E Fortunato, S Lloyd, R Martinez, et al. Nmr based quantum information processing: Achievements and prospects. Fortschritte der Physik: Progress of Physics, 48(9-11):875–907, 2000.

[136] Lorenza Viola, Seth Lloyd, and Emanuel Knill. Universal control of decoupled quantum systems. Physical Review Letters, 83(23):4888, 1999.

[137] Bryan O'Gorman, William J Huggins, Eleanor G Rieffel, and K Birgitta Whaley. Generalized swap networks for near-term quantum computing. arXiv preprint arXiv:1905.05118, 2019.

[138] CJ Ballance, TP Harty, NM Linke, MA Sepiol, and DM Lucas. High-fidelity quantum logic gates using trapped-ion hyperfine qubits. Physical review letters, 117(6):060504, 2016.

[139] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. Proceedings of the National Academy of Sciences, 114(13):3305–3310, 2017.

[140] Tasio Gonzalez-Raya, Rodrigo Asensio-Perea, Ana Martin, Lucas C Céleri, Mikel Sanz, Pavel Lougovski, and Eugene F Dumitrescu. Digital-analog quantum simulations using the cross-resonance effect. PRX Quantum, 2(2):020328, 2021.

[141] Peter Schauss. Finite-range interacting ising quantum magnets with rydberg atoms in optical lattices-from rydberg superatoms to crystallization. arXiv preprint arXiv:1706.09014, 2017.

[142] J. P. Gaebler, T. R. Tan, Y. Lin, Y. Wan, R. Bowler, A. C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried, and D. J. Wineland. High-fidelity universal gate set for $^{9}Be^{+}$ ion qubits. Phys. Rev. Lett., 117:060505, Aug 2016.

[143] Ye Wang, Mark Um, Junhua Zhang, Shuoming An, Ming Lyu, Jing-Ning Zhang, L-M Duan, Dahyun Yum, and Kihwan Kim. Single-qubit quantum memory exceeding ten-minute coherence time. Nature Photonics, 11(10):646–650, 2017.

[144] Jiehang Zhang, Guido Pagano, Paul W Hess, Antonis Kyprianidis, Patrick Becker, Harvey Kaplan, Alexey V Gorshkov, Z-X Gong, and Christopher Monroe. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. Nature, 551(7682):601–604, 2017.

[145] Diego Porras and J Ignacio Cirac. Effective quantum spin systems with trapped ions. Physical review letters, 92(20):207901, 2004.

[146] Adam Pearson, Anurag Mishra, Itay Hen, and Daniel A Lidar. Analog errors in quantum annealing: doom and hope. NPJ Quantum Information, 5:1–9, 2019.

[147] Siddharth Muthukrishnan, Tameem Albash, and Daniel A Lidar. When diabatic trumps adiabatic in quantum optimization. arXiv preprint arXiv:1505.01249, 2015.

[148] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. New Journal of Physics, 18(2):023023, 2016.

[149] Ian D. Kivlichan, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Wei Sun, Zhang Jiang, Nicholas Rubin, Austin Fowler, Alá n Aspuru-Guzik, Hartmut Neven, and Ryan Babbush. Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization. Quantum, 4:296, Jul 2020.

[150] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information. Cambridge University Press, 2000.

[151] Paula García-Molina, Ana Martin, Mikel Garcia de Andoin, and Mikel Sanz. Noise in digital and digital-analog quantum computation. arXiv preprint arXiv:2107.12969, 2021.

[152] W Dür and H.J Briegel. Stability of macroscopic entanglement under decoherence. Physical review letters, 92(18):180403, 2004.

[153] Google Quantum AI. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. Nature Physics, 17(3):332–336, 2021.

[154] Alexander Cowtan, Silas Dilkes, Ross Duncan, Alexandre Krajenbrink, Will Simmons, and Seyon Sivarajah. On the qubit routing problem. arXiv preprint arXiv:1902.08091, 2019.

[155] Stephen Brierley. Efficient implementation of quantum circuits with limited qubit interactions. arXiv preprint arXiv:1507.04263, 2015.

[156] Matteo G Pozzi, Steven J Herbert, Akash Sengupta, and Robert D Mullins. Using reinforcement learning to perform qubit routing in quantum compilers. arXiv preprint arXiv:2007.15957, 2020.

[157] Chad Rigetti and Michel Devoret. Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies. Physical Review B, 81(13):134507, 2010.

[158] Jing Yu, Juan Carlos Retamal, Mikel Sanz, Enrique Solano, and Francisco Albarrán-Arriagada. Superconducting circuit architecture for digital-analog quantum computing. EPJ Quantum Technology, 9(1):1–35, 2022.

[159] Sepehr Ebadi, Tout T Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, et al. Quantum phases of matter on a 256-atom programmable quantum simulator. Nature, 595(7866):227–232, 2021.

[160] Lin Tian. A superconducting flux QuBit: measurement, noise and control. PhD thesis, Massachusetts Institute of Technology, 2002.

[161] Rami Barends, Julian Kelly, Anthony Megrant, Daniel Sank, Evan Jeffrey, Yu Chen, Yi Yin, Ben Chiaro, Josh Mutus, Charles Neill, et al. Coherent josephson qubit suitable for scalable quantum integrated circuits. Physical review letters, 111(8):080502, 2013.

[162] Chad Rigetti, Jay M Gambetta, Stefano Poletto, Britton LT Plourde, Jerry M Chow, Antonio D Córcoles, John A Smolin, Seth T Merkel, Jim R Rozen, George A Keefe, et al. Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms. Physical Review B, 86(10):100506, 2012.

[163] Junling Long, Tongyu Zhao, Mustafa Bal, Ruichen Zhao, George S Barron, Hsiang-sheng Ku, Joel A Howard, Xian Wu, Corey Rae H McRae, Xiu-Hao Deng, et al. A universal quantum gate set for transmon qubits with strong zz interactions. arXiv preprint arXiv:2103.12305, 2021.

[164] Dvir Kafri, Chris Quintana, Yu Chen, Alireza Shabani, John M Martinis, and Hartmut Neven. Tunable inductive coupling of superconducting qubits in the strongly nonlinear regime. Physical Review A, 95(5):052333, 2017.

[165] Marius Schöndorf. Qubit measurement and coupling strategies and their applications. PhD thesis, Saarländische Universitäts-und Landesbibliothek, 2020.

[166] Sergey M Frolov, Micah JA Stoutimore, Trevis A Crane, Dale J Van Harlingen, Vladimir A Oboznov, Valery V Ryazanov, Adele Ruosi, Carmine Granata, and Maurizio Russo. Imaging spontaneous currents in superconducting arrays of π-junctions. Nature Physics, 4(1):32–36, 2008.

[167] MI Khabipov, DV Balashov, F Maibaum, AB Zorin, VA Oboznov, VV Bolginov, AN Rossolenko, and VV Ryazanov. A single flux quantum circuit with a ferromagnet-based josephson π-junction. Superconductor Science and Technology, 23(4):045032, 2010.

[168] P. Kumar, S. Sendelbach, M. A. Beck, J. W. Freeland, Zhe Wang, Hui Wang, Clare C. Yu, R. Q. Wu, D. P. Pappas, and R. McDermott. Origin and reduction of $1/f$ magnetic flux noise in superconducting devices. Phys. Rev. Appl., 6:041001, Oct 2016.

[169] Aetal Izmalkov, M Grajcar, E Il'ichev, Th Wagner, H-G Meyer, A Yu Smirnov, MHS Amin, Alec Maassen van den Brink, and AM Zagoskin. Evidence for entangled states of two coupled flux qubits. Physical review letters, 93(3):037003, 2004.

[170] R Harris, J Johansson, AJ Berkley, MW Johnson, T Lanting, Siyuan Han, P Bunyk, E Ladizinsky, T Oh, I Perminov, et al. Experimental demonstration of a robust and scalable flux qubit. Physical Review B, 81(13):134510, 2010.

[171] Michael Shane Allman, Jed D Whittaker, Manuel Castellanos-Beltran, Katarina Cicak, Fabio da Silva, MP DeFeo, Florent Lecocq, Adam Sirois, John D Teufel, Jose Aumentado, et al. Tunable resonant and nonresonant interactions between a phase qubit and l c resonator. Physical review letters, 112(12):123601, 2014.

[172] Michael S Allman, Fabio Altomare, Jed D Whittaker, Katarina Cicak, Dale Li, Adam Sirois, Joshua Strong, John D Teufel, and Raymond W Simmonds. rf-squid-mediated coherent tunable coupling between a superconducting phase qubit and a lumped-element resonator. Physical review letters, 104(17):177004, 2010.

[173] Deanna M Abrams, Nicolas Didier, Blake R Johnson, Marcus P da Silva, and Colm A Ryan. Implementation of xy entangling gates with a single calibrated pulse. Nature Electronics, 3(12):744–750, 2020.

[174] LS Levitov, TP Orlando, JB Majer, and JE Mooij. Quantum spin chains and majorana states in arrays of coupled qubits. arXiv preprint cond-mat/0108266, 2001.

[175] Alec Maassen van den Brink. Galvanic coupling of flux qubits: simple theory and tunability. arXiv preprint cond-mat/0605398, 2006.

[176] DK Weiss, Helin Zhang, Chunyang Ding, Yuwei Ma, David I Schuster, and Jens Koch. Fast high-fidelity gates for galvanically-coupled fluxonium qubits using strong flux modulation. PRX Quantum, 3(4):040336, 2022.

[177] M Schöndorf and FK Wilhelm. Nonpairwise interactions induced by virtual transitions in four coupled artificial atoms. Physical Review Applied, 12(6):064026, 2019.

[178] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature, 614(7949):676–681, 2023.

[179] Craig Gidney and Martin Ekerå. How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits. Quantum, 5:433, 2021.

[180] Bernard J. Fino and V. Ralph Algazi. Unified matrix treatment of the fast walsh-hadamard transform. IEEE Transactions on Computers, 25(11):1142–1146, 1976.