A Study of the Quantum Approximate Optimisation Algorithm Applied to the Maximum Weighted Independent Set Problem

Charlie Hetherington
Level 4 MPhys Report
Supervisors: Dr. Viv Kendon
Department of Physics, Durham University
Submitted: April 26, 2021

Abstract

The Quantum Approximate Optimisation Algorithm (QAOA) is a variational hybrid quantum-classical algorithm well-suited for combinatorial optimisation tasks. A scheme for solving the Maximum Weighted Independent Set problem is devised. The similarity of optimal parameters is demonstrated at the $p = 1$ level. The feasibility of reusing one set of parameters across different problem instances is explored and patterns in optimal parameters for $p > 1$ are searched for.

Contents

1. Introduction

Quantum computation offers the tantalising prospect of using nature's own operating system for calculation. Quantum computing was first established as a theoretical possibility in the 1980's [1], saw its first experimental realisation in the 1998 [2] and is now being pursued in universities, national laboratories and commercial labs across the world. Last year, Google claimed to have broken the benchmark of 'quantum supremacy': carrying out a calculation that would take a classical computer thousands of years in just three minutes [3]. Nevertheless, realising the full potential of quantum computing still requires enormous progress to engineer computers greater in scale and more resistant to noise. In the meantime, there are still potentially useful quantum algorithms that can be run on present-day hardware. One such is the Quantum Approximate Optimisation Algorithm (QAOA), whose performance is still largely unknown. This study applies the QAOA to the Maximum Weighted Independent Set (MWIS) problem in simulation.

1.1 The Qubit

The the core of quantum computing is the quantum bit or *qubit*, the basic unit of quantum information. In classical computing, information is encoded in bits: a logical state that takes one of two distinct values, typically labelled 0 and 1. As the quantum analogue to this, the qubit describes a state in the simplest non-trivial quantum system. This is a two-dimensional Hilbert space whose orthonormal basis is denoted $\{|0\rangle, |1\rangle\}$, in deference to the classical world. The state of the qubit takes values of the form:

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

where α and β are complex numbers. As quantum objects, qubits do not conform to classical intuition. The state of a classical bit can be measured without disrupting the bit; for a qubit, this act of measurement irreversibly alters the state. In accordance with the Born rule, the chance of either eigenvalue being selected is equal to the probability-amplitude squared: the probability of measuring the state $|0\rangle$ is $|\langle 0|\psi\rangle|^2 = \alpha^2$. As total probabilities must sum to 1, this imposes the condition that $|\alpha|^2 + |\beta|^2 = 1$.

This constraint allows for the vector to be interpreted geometrically in spherical coordinates, where $\alpha = e^{i\psi} \cos\frac{\theta}{2}$ and $\beta = e^{i(\psi+\phi)} \sin\frac{\theta}{2}$. The term $e^{i\psi}$ is known as a global phase and has no physical relevance for a single qubit. The states of pure qubits can be visualised as a points on a unit sphere, known as the Bloch sphere. The Bloch sphere can be used to picture transformations to a qubit's state as rotations upon it.

1.2 Combinations of Qubits

Further difference between classical and quantum systems comes from the interactions of systems of multiple qubits. Systems of multiple qubits are combined through use of

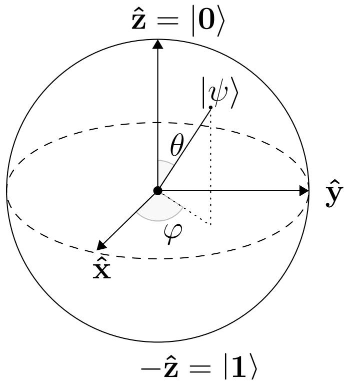

$\hat{\mathbf{z}} = |\mathbf{0}\rangle$
$|\psi\rangle$
θ
$\hat{\mathbf{y}}$
φ
$\hat{\mathbf{x}}$
$-\hat{\mathbf{z}} = |\mathbf{1}\rangle$

Figure 1.1: A Bloch sphere, with the computational basis states labelled. Reproduced from [4].

the tensor product: for a system of two qubits in the states $|\psi_1\rangle$ and $|\psi_2\rangle$ respectively, the state of the combined system is $|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$. The Hilbert space of the new system correspondingly expands with a basis defined by the four possible states of the two qubits: $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ [1]. As for the 2-D Hilbert space, these basis vectors can be written in Dirac notation with the object $|00\rangle$ known as a bitstring.

This generalises such that the state-space of an n qubit system has dimension 2^n, with each basis state represented as a bitstring of length n:

$$\underbrace{|\,00\ldots00}_{n}\rangle, |00\ldots01\rangle, \ldots, |11\ldots10\rangle, |11\ldots11\rangle.$$

Any arbitary state in the composite vector space can be written as a weighted sum of these basis vectors. Not all states can be cleanly decomposed into the tensor product of single product states. This can be seen by considering the complex coefficients of the two qubit system known as a Bell state: $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

As a product of two states, $|\Phi^+\rangle$ would be expressed as so:

$$\begin{aligned} \left|\Phi^+\right\rangle &= (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) \\ &= a_1a_2|00\rangle + a_1b_2|01\rangle + a_2b_1|10\rangle + b_1b_2|11\rangle \end{aligned}$$

This would imply that $a_1b_2 = a_2b_1 = 0$, meaning that either $a_1a_2 = 0$ or $b_1b_2 = 0$. As neither $a_1a_2 \neq 0$ nor $b_1b_2 \neq 0$, it arises that the state of the first qubit can only described with reference to the second: the two qubits are correlated [5]. This phenomenon is innate to quantum mechanics and is known as 'entanglement'. This makes it possible to create a complete 2^n complex vector space to perform calculations using just n qubits. For $n = 500$, this creates a quantum system of 2^{500} probability amplitudes available for computation, a number greater than the number of atoms in the known universe [6]. The immense dimensionality of the state space can be used to store a vast number of probability amplitudes. These can be fed into algorithms and processed simultaneously in parallel, with desired results extracted to gain a potentially huge speed up over classical

[1] Typically the $\otimes$ symbol is omitted: $|\psi_1\rangle \otimes |\psi_2\rangle$ becoming $|\psi_1\psi_2\rangle$.

devices.

Nevertheless, from these 2^n bits of information, measurement collapses the superposition such that only n bits of information can ever be obtained from a n qubit system. To properly exploit the high dimensionality of the Hilbert space, further quantum mechanical phenomena like interference can be used. Combined, these effects can be leveraged to efficiently solve certain problems that are intractable on machines operating through solely classical principles.

1.3 The Gate Model of Quantum Computation

The basic method of computation is that data is encoded, calculations are performed and results are extracted. There are multiple models of quantum computing that break down this task in different ways. The method of quantum computing that most closely resembles its classical counterpart is the quantum gate model.

In conventional computers, circuits consist of wires which shuttle information around to be processed by logic gates; these are idealised devices which manipulate bits in accordance with Boolean algebra. The simplest case for a single bit (aside from a gate which just maps a value to itself) is the NOT gate, which swaps 0 for 1 and vice versa. This concept can be extended to gates that act on combinations of bits, such as the NOR gate which takes in the values of two bits and outputs a positive 1 outcome only when both inputs are 0. A series of gates forms a universal set if all possible Boolean functions - and so all possible bit manipulations - can be implemented solely though combinations of the set [6].

In the quantum arena, a similar NOT gate can be formulated, which interchanges the probability amplitudes of $|0\rangle$ and $|1\rangle$: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow |\psi\rangle = \beta|0\rangle + \alpha|1\rangle$. That the gate acts linearly is a trait inherent to quantum mechanics, and so is shared by all quantum gates. It also allows for matrix representation, as so:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The only constraint for a quantum gate is that its transformation is unitary. This constraint imposes the condition of reversibility: the inputs into any quantum gate can always be reconstructed from the gate's outputs. Nevertheless, this added restriction does not place any limitation on the power of quantum gates [7]. Some other single quantum gates of note include the Z gate and the H, known as a Hadamard gate. In a typical circuit, quantum states are channelled by 'wires' to gates, which then transform the state.

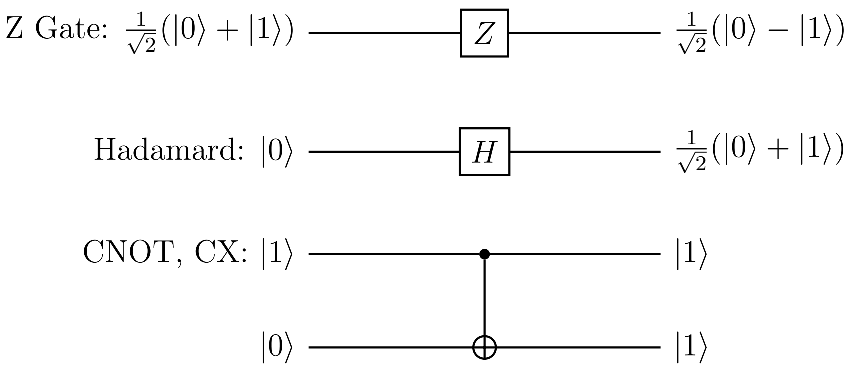

Z Gate: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
Z
$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
Hadamard: $|0\rangle$
H
$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
CNOT, CX: $|1\rangle$
$|1\rangle$
$|0\rangle$
$|1\rangle$

Figure 1.2: Some simple Quantum Gates.

All single qubit gates can act on qubits that are part of a multi-qubit system by combining the operations through the use of tensor products. The action of a X and H gate on a two-qubit system can be described as so:

$$X\,|\psi_1\rangle \otimes H\,|\psi_0\rangle = (X \otimes H)\,|\psi_1\psi_0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}.$$

Like their classical counterparts, quantum circuits that perform more than trivial operations require interactions between qubits: multi-qubit gates that act upon one qubit (or more) depending on the state of another. A typical example is a CX (also known as a CNOT) gate, which flips a target qubit depending on the state of a control bit.

Through combinations of quantum gates, any unitary transformations can be applied. The basic operation of a quantum circuit follows from this: first a set of qubits are initialised in a known pure state, ready for manipulation. Then, the set of quantum gates that implement a desired algorithm are performed. This operation can be thought of as a giant unitary operation which rotates the starting input state to the desired output state. Finally, the resulting states of the qubits are measured, with the results being placed in easily accessible classical bits. In this abstract sense, the creation of quantum algorithms amounts to the design of gate combinations to perform wanted tasks.

1.4 The Promise and Present Reality of Quantum Computing

A commonly cited starting point for the entrance into the minds of physicists of a quantum form of computation is an address titled *'Simulating Physics with Computers'*, chosen no doubt due to the renown of its deliverer, Richard Feynman. In the keynote, Feynman noted that entanglement made it exponentially costly to model all but the simplest physical systems on a classical computer; he instead raised the possibility of a computer that operated on quantum principles [1].

A number of foundational papers were published in the years after, such as the 1985 paper by the British physicist David Deutsch that showed a universal quantum computer could evolve a superposition of quantum states that could follow coherently distinct paths - the heralding of quantum parallelism [8]. The first significant quantum algorithm for a useful task followed with Shor's algorithm [9] for integer factoring in 1994. This gave an proven exponential speedup over the best known classical methods for integer factorisation, the first major sign of quantum advantage for a useful task. This was shortly followed by Grover's discovery of a search algorithm [10] which offered a quadratic speed-up over classical computation for finding a desired entry in an unsorted database .

Without the means to implement these methods, these achievements would be nothing more than academic curiosities. However, increasing mastery over small atomic systems led to the first practical demonstration of a quantum computation in 1998, when Jones and Mosca controlled the spin states of atomic nuclei to implement a simple algorithm [2].From these early beginnings, the scale and sophistication of quantum computers has only grown. Much of the impetus for this drive has come from the commercial sector where behemoths like Google and IBM compete alongside smaller, more specialised firms. At present, the largest quantum computers contain around 50 qubits; this is a point of particular significance as it represents the limits of what present day supercomputers can simulate. These implementations are yet to be perfected however, and are extremely susceptible to noise [11].

There are multiple sources of noise in quantum gate computers [12]. Over time, qubits increasingly become entangled with their environment. This leads to loss of coherence and a reduction in the quantum state's fidelity—a measure of how far away the experimental state is from the ideal state [13]. This acts as a time limit on the length of a quantum computation. Another source of error comes from physically implemented gates not precisely corresponding to gates specified by the user, a problem that scales with the number used. This sets a limit on the depth of quantum circuits before meaningful results are overwhelmed with noise.

A rich array of Quantum-Error Correction codes have been devised that can prevent such errors by spreading the information of a single logical qubit over multiple physical qubits. When combined with gates that can perform transformations to a certain standard of accuracy, the quantum threshold theorem means that logical error rates can be reduced to an arbitrarily low level [14]. To properly implement these safeguarding techniques requires a overhead in the number of gates used for a computation: to use Shor's algorithm to gain an improvement over the best classical computers requires a device with 10^5 fully-corrected qubits [15], orders of magnitude more than the largest quantum computers of today.

As such, the field has entered what has been dubbed the 'Noisy Intermediate-Scale Quantum' era; quantum computers beyond the simulating capabilities of the best classical supercomputers, but lacking the scale and qubit quality needed to fully implement the most powerful capabilities of the field. [11].

Instead, many expect the first practical use of NISQ devices - and so the first tangible benefits of quantum computing - to come from the implementation of hybrid quantum-classical algorithms [11]. These algorithms employ parameterised quantum circuits that are optimised with a classical CPU, and various forms have found use in tasks such as machine learning and quantum chemistry simulations [16] [17].

One such algorithm is the Quantum Approximate Optimisation Algorithm which seeks the ground state energy of a Hamiltonian H_c that encodes the solution to a combinatorial optimisation problem.

2. The Quantum Approximation Optimisation Algorithm

The QAOA was first proposed in 2014 by the theorist Edward Fahri [18] and can be applied to a number of NP-hard problems. NP-hard problems are related to the broader computational complexity class NP. These are decision problems which have solutions that can be checked in polynomial time [19]; NP-complete problems are the hardest class of problems in NP, which all NP problems can, in polynomial time, be reduced to. NP-hard problems are at least as hard as the NP-complete problem. NP-hard problems are not typically amenable to an exact solution and so are practically solved with approximation methods which do not obtain exact solutions, but rather approximate solutions that are 'good enough'. As its name suggests, QAOA is such an algorithm and so does not have performance guarantees in general. Despite this, it has been shown to have non-trivial performance on select problems in its crudest form and, as classical computers are unable to efficiently simulate it, it is seen as a strong candidate for finding quantum speedups on NISQ hardware [18] [20]. This study applies the QAOA to the Maximum Weighted Independent Set problem, which is used as a representative example of the broader class of combinatorial optimisation tasks.

2.1 Combinatorial Optimisation

Combinatorial optimisation seeks to find the optimal solution to a problem from a large but finite set of possibilities. They are ubiquitous through science and industry with famous instances including the Travelling Salesman Problem and the Knapsack problem [21]. Typically, the solution space is far too large to exhaustively search and so an algorithmic approach must be taken.

Formally, combinatorial optimisation problems can be posed in the form of a cost function (also known as an objective function) $C(z)$ whose minimum or maximum is sought:

$$\max C(z) = \sum_{\alpha=1}^{m} C_\alpha(z),$$

where variables $C_\alpha(z)$ are binary functions termed clauses into which an n length bit string z is inputted $z = z_1 z_2 \dots z_n$ where $z_i \in [0, 1]$. Each clause is a constraint that acts on some number - typically only a few - number of bits. If a bit string z satisfies clause α, $C_\alpha = 1$ and is 0 otherwise. Solving combinatorial optimisation problems thus entails finding the bit string z which satisfies the most clauses. Indeed, as the problem is NP-hard, typically an approximate solution ('good enough') is sought. A typical figure of merit used to classify the quality of solutions is the approximation ratio:

$$\frac{C(z)}{C_{\text{max}}} = r, \tag{2.1}$$

where $C_{\text{max}} = \max_z C(z)$.

To solve problems of this form on a quantum computer, this objective function is mapped onto a operator H_C, called the 'problem' Hamiltonian. Here H_C is defined as an operator on the 2^n dimensional Hilbert space where each bitstring z is a basis vector $|z\rangle$ such that the solution bitstring of the classical problem is the highest energy eigenstate of H_C. This Hamiltonian is diagonal, with values corresponding to values of the objective function. This means that the Hamiltonian acts upon a basis state as so:

$$H_C|z\rangle = C(z)|z\rangle.$$

The Hamiltonian is exponentially large and so is not constructed explicitly (this would be equivalent to calculating the objective value of each 2^n possibility - 'brute-forcing' the problem). Instead, it can be constructed by mapping binary variables onto the eigenvalues of Pauli Z operators.

2.2 The Quantum Approximate Optimisation Algorithm

QAOA can best be understood as a discretised approximation to adiabatic computing. Adiabatic quantum computing (AQC) is an entirely separate model of computation to the unitary circuit method which exploits the adiabatic principle for calculation. This concept, as first formulated by Born and Fock [22], states that:

> 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's spectrum.

In AQC, a problem Hamiltonian H_c is designed such that its groundstate encodes the desired solution of a problem. The system is prepared in a trivial state $|\psi_0\rangle$ and then is subjected to a driver Hamiltonian H_D whose ground state is $|\psi_0\rangle$. A time dependent Hamiltonian $H(t)$ is then defined on the interval $t \in [0, T]$:

$$H(t) = (1 - s(t))H_D + s(t)H_C,$$

where $s(t)$ is known as the schedule and is any smooth function such that $s(0) = 0$ and $s(T) = 1$. A simple choice is linear interpolation where $s(t) = \frac{t}{T}$. The system then evolves under this time-varying Hamiltonian.

If this evolution is gradual enough, and there is a large enough spectral gap at all points between the groundstate and the first excited state of $H(t)$, the adiabatic theorem holds and the system remains in the groundstate throughout. This means at T, the system is in the groundstate of H_C and the solution to the problem can be found through measurement.

In QAOA, the complex time evolution operator is decomposed into discrete steps of size Δt through use of the Trotter formula [?] as so:

$$\begin{aligned} U(T) &:= \mathcal{T}\exp\left[-i\int_0^T H(t)dt\right] \\ &\approx \prod_{a=0}^{p-1}\exp[-iH(a\Delta t)\Delta t]. \end{aligned} \tag{2.2}$$

Here $U(T)$ is the time evolution operator and p is an integer that specifies the 'chunk size' $\Delta t = T/p$ over which the Hamiltonian is held to be constant [23]. This determines the quality of approximation: as p increases, the approximation better resembles the adiabatic transition.

With the Lie-Trotter-Suzuki decomposition $e^{i(\hat{A}+\hat{B})x} \approx e^{i\hat{A}x}e^{i\hat{B}x} + \mathcal{O}(x^2)$ the problem and driver Hamiltonians can separated:

$$\begin{aligned} U(T,0) &\approx \prod_{a=0}^{p-1}\exp\left\{-i(1-s(a\Delta t))H_D + s(a\Delta t)H_C)\Delta t\right\} \\ &\approx \prod_{a=0}^{p-1}\exp\left\{-i(1-s(a\Delta t))H_D\Delta t\right\}\exp\left\{-is(a\Delta t)H_C\Delta t\right\}. \end{aligned} \tag{2.3}$$

In this way, AQC can be approximated by alternately evolving the system under the problem Hamiltonian H_C for some small interval $s(a\Delta t)\Delta t$ and then under the driver Hamiltonian H_D for the interval $(1-s(a\Delta t))\Delta t$. These truncated evolutions are applied through the use of unitaries $U_B(\beta) = e^{-i\beta H_D}$, $U_P(\gamma) = e^{-i\gamma H_C}$ where the time intervals are represented through the angles β and γ respectively.

While this connection to AQC guarantees that QAOA can succeed in the limit that $\Delta t \to 0$, it does not mean QAOA always operates via the adiabatic mechanism. When the instantaneous eigenvalues of $H(t)$ are close, maintaining the adiabaticity condition requires that the transition period to be so long that the algorithm becomes inefficient. It has been shown that QAOA can circumvent these critical points by accessing higher dimensions in the transistion [24] [25].

2.3 Implementing the Algorithm

QAOA acts then through the alternate application of the unitaries $U_B(\beta) = e^{-i\beta H_D}$ and $U_P(\gamma) = e^{-i\gamma H_C}$ with the result that the state is evolved towards the groundstate of H_C. The algorithm proceeds as follows: first qubits are prepared in a uniform superposition of all the basis states, through the application of Hadamard gates $H^{\otimes N}$ to $|0\rangle^{\otimes N}$ where the notation $H^{\otimes N}$ denotes the application of H to all N qubits. Then the unitary forms of H_C and H_D are alternately applied p times to create the variational wavefunction:

$$|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle = U_B\left(\beta_p\right)U_C\left(\gamma_p\right)\cdots U_B\left(\beta_1\right)U_C\left(\gamma_1\right)|+\rangle^{\otimes N},$$

where $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_p)$ and $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_p)$. These $2p$ angles parameterise the wavefunction and can be adjusted to change the 'time' for which the unitary transformations are applied.

The expectation value of H_C is then calculated:

$$F_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\left|H_C\right|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle,$$

In a classical simulation of QAOA, the full statevector is known and so the value of $F_p(\boldsymbol{\gamma},\boldsymbol{\beta})$ can be calculated directly. In a real quantum computer, the quantum circuit that implements $|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle$ is instead sampled repeatedly, with each measurement producing the bit string z with probability $|\langle z|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle|^2$. An estimate of $F_p(\boldsymbol{\gamma},\boldsymbol{\beta})$ can be then formed with the sum $\sum_z P(z)C(z)$ over the measured samples z with the probability $P(z)$ approximated by the relative frequency of occurrence [26].

This value is then optimised by a classical computer which searches for the optimal parameters $(\boldsymbol{\gamma}^\star,\boldsymbol{\beta}^\star)$ such that $F_p(\boldsymbol{\gamma},\boldsymbol{\beta})$ is maximised. This is carried out in a continual loop by feeding suggested parameters values into the quantum circuit and measuring until a preset conidtion is met. When these optimum parameters are found, the state $|\psi_p(\boldsymbol{\gamma}^\star,\boldsymbol{\beta}^\star)\rangle$ is then prepared. If $F_p(\boldsymbol{\gamma},\boldsymbol{\beta})$ has been optimised well, then it is likely that $|\psi_p(\boldsymbol{\gamma}^\star,\boldsymbol{\beta}^\star)\rangle$ contains the groundstate, or at least solutions z close to the optimum.

2.4 QAOA Circuit for the Maximum Weighted Independent Set Problem

In a graph $G = (V,E)$ with V vertices (also known as nodes) and E edges where each vertex $i \in \mathrm{V}(G) = \{1,\ldots,n\}$ is weighted by a positive rational number c_i, the maximum weighted independent set problem is the subset of vertices $S \subseteq \mathrm{V}(G)$ such that no two vertices in S are connected and the sum $\sum_{i\in S} c_i$ is maximised [27].

The problem can be characterised as a quadratic binary optimisation problem if the binary variable $x_i = 1$ when node i is a member of S and is 0 otherwise. If the penalty term $J_{ij} > \min\{c_i, c_j\}$, the maximum value of the objective function

$$y\left(x_1,\ldots,x_n\right) = \sum_{i\in\mathrm{V}} c_i x_i - \sum_{(i,j)\in\mathrm{E}} J_{ij}x_i x_j, \tag{2.4}$$

gives the maximum possible weight of a graph [28]. Here $y\left(x_1,\ldots,x_n\right)$ is the psuedo-boolean function. The first sum adds the weights of the node c_i if the binary variable x_i indicates $i \in S$. The independence of the set is assured by the sum over the edges, which adds the penalty J_{ij} if $(i,j) \in S$. The simpler problem of Maximum Independent Set (MIS), sometimes referred to as the largest stable set problem, is a special case of MWIS where each node weight is $c_i = 1$ [27]. MWIS naturally arises in a number of real world problems and has attracted recent interest due to its application to wireless networking [29].

To formulate this objective function as a problem Hamiltonian, and so that it can be prepared on a QAOA circuit, we can map the binary variable x_i onto the eigenvalues of the Pauli Z_i operator using the transformation $x_i \to (1 - Z_i)/2$ [18] [1]. Subsituting these transformations into (2.4) gives the Hamiltonian as:

$$\begin{aligned} H_c =& \sum_{i \in \mathrm{V}}^{n} \frac{c_i}{2}(I - Z_i) - \sum_{i,j\ \in E}^{m} \frac{J_{ij}}{4}\left(I - Z_i\right)\left(I - Z_j\right) \\ =& \sum_{i \in \mathrm{V}}^{n} \frac{c_i}{2}(I - Z_i) - \sum_{i,j\ \in E}^{m} \frac{J_{ij}}{4}(I - Z_i - Z_j + Z_i Z_j). \end{aligned} \tag{2.5}$$

Typically computational optimisation problems are posed in the form of minimisation problems and so the signs of the objective Hamiltonian are reversed. In addition, for the purposes of computation the constant terms can be removed [30] to give:

$$H_c = \sum_{i \in \mathrm{V}}^{n} \frac{c_i}{2} Z_i + \sum_{i,j\ \in E}^{m} \frac{J_{ij}}{4}(Z_i Z_j - Z_i - Zj).$$

For the QAOA, this H_C is put into a unitary of the form $e^{-i\gamma H_c}$. $[Z_i, Z_i Z_j] = 0$ and so the exponential can be split :

$$\exp\{-i\gamma\left(\sum_{i \in \mathrm{V}}^{n} \frac{c_i}{2} Z_i\right)\}\exp\{-i\gamma\left(\sum_{i,j\ \in E}^{m} \frac{J_{ij}}{4} Z_i Z_j - Z_i - Z_j\right)\}.$$

To implement the Hamiltonian in a quantum circuit, the $e^{-i\theta Z_j}$ and $e^{-i\theta Z_i Z_j}$ operators need to be translated into quantum gates. The single qubit operator can be simply implemented as a single R_z gate. An R_z gate acts to rotate the state of a qubit about the z axis. That this applies $e^{-i\theta Z_j}$ can be seen through its basic definition:

$$R_z(2\theta) \equiv e^{-i\theta Z} = \cos\theta I - i \sin\theta Z = \begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix}.$$

The $Z_i Z_j$ operator can be performed by recognising that its operation adds a phase factor to the two qubits it acts on depending on their parity:

$$e^{-iZ \otimes Zt}|ab\rangle = e^{-i(-1)^{a \oplus b} t}|ab\rangle.$$

[1] Here Z_i denotes the operator acting on the i-th qubit: $I \otimes \ldots \otimes Z_i \otimes \ldots \otimes I$

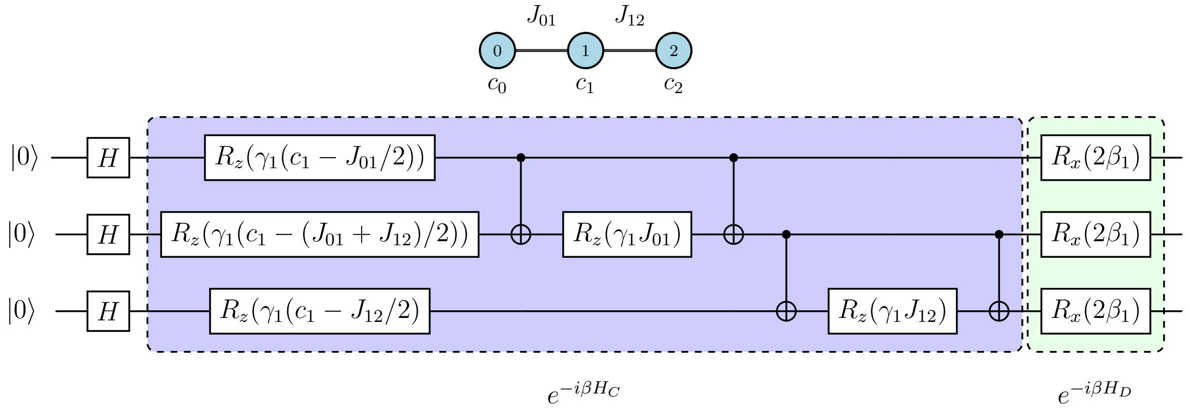


Figure 2.1: A Quantum Circuit Diagram for the first layer of QAOA applied to a simple path graph. Typically the first qubit of a circuit is labelled the '0-th'; this convention is reflected in the labelling of the graph nodes.

This can be realised by conjugating a single R_z gate with two $CNOT$ gates. As before, the terms of our problem Hamiltonian commute and so circuits that act on separate qubits can be concatenated.

Much as the $e^{-i\theta Z_j}$ unitary is implemented with an R_z gate, the driver operator $e^{-i\beta H_B} = e^{-i\beta \sum_j X_j}$ can be applied using an R_x gate, which acts to rotate the state of a qubit about the x axis. Combined, these parameterised gates allow us to apply our driver and problem Hamiltonians to the quantum system for selected chunks of 'time', so allowing a discrete approximation of AQC.

We note that the action of the single Z_i operators in the node weight term and the independence constraint term manifest as R_z gates that seek to rotate the qubit i in opposite directions about the z axis. By rearranging these alternate rotations, we can reduce the number of gates needed to implement H_C for any particular graph as so:

$$\exp\{-i\gamma\left(\sum_{i\in \mathrm{V}}^{n}\frac{c_i}{2}Z_i - \sum_{j:\{i,j\}\in E}^{m}\frac{J_{ij}}{4}Z_i\right)\}\exp\{-i\gamma\left(\sum_{i,j\ \in E}^{m}\frac{J_{ij}}{4}Z_iZ_j\right)\}. \tag{2.6}$$

These sums over values of c_i and J_{ij}, when multipled with the angles (γ_i, β_i), are used to parameterise the gates as can be seen in 2.1. With each layer of p the values of (γ_i, β_i) change, but the weights are constant for each problem case.

This means the total action of all the Z_i operators on a single qubit in one layer of QAOA can be combined in one single qubit gate. This could ostensibly be interpretted as a major improvement over other implementation found in [31]. If the number of gates needed to carry out a circuit were reduced then the impact of accruing gate error on computation could be mitigated. In reality, the basis gates of most physical implementations of quantum computers are different to those used in high-level circuit design: apparent differences between two circuits may disappear when transpiled to hardware.

Nodes were assigned random integer weights such that $c_i \in [1..10]$. It was found that scaling each value of c_i to be a fraction of the largest node weight for a particular graph instance (i.e $c_i \rightarrow c_i/\max\{c_i\}$) and setting $J_{ij} = 2$ produced circuits best capable of finding the groundstate of H_C. For the unweighted maximum independent set version of the problem, this meant the penalty term J_{ij} was twice as large as the corresponding node weight terms c_i.

2.5 Practical Considerations

2.5.1 Performance Metrics

The same performance measures were used to assess the quality of solutions found for a single problem case as those employed in [26]. The first, and most obvious, is the value of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ found. As the circuitry for the problem was configured to minimise the problem Hamiltonian, optimal values were as small as possible. The second measure is the ratio given by:

$$r = \frac{F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) - \max H_c}{\min H_c - \max H_c}, \tag{2.7}$$

which replicates the approximation ratio of 2.1. The third metric used was the success probabillity, which gives the solution's probability of giving the exact groundstate energy; this should be as high as possible.

A fourth metric was used to assess the quality of solutions between cases. Though it proved best for finding optimal solutions, a drawback of scaling the weights of c_i meant that the energy 'reward' for a node joining the independent set was far smaller than the penalty terms for violation of independence. This has the effect of creating a distribution of objective values that has a small negative 'head' and a large, positive 'tail'. By shunting these values to all be the same sign (i.e ensuring a value of $0 \leq r \leq 1$), it would in effect make comparions between graphs of different size misleading: graphs with more nodes would have much larger penalties and so all objective values would receive a disproportionate boost. Solution sets with no nodes present achieved r values of around 0.6 in some cases.

To make meaningful comparisons between solution qualities for graphs of varying size, a measure of the form:

$$k = \frac{\max H_C - F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})}{\max H_C}, \tag{2.8}$$

was devised. For this objective values were recalculated such that node weights added to the objective function and penalty terms subtracted from it (i.e $\max H_C$ was the energy of the best possible solution) meaning that for $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \to \max H_C$, $k \to 0$. While this method ultimately does not solve the root of the problem - that of the small node weights - it does give a clearer indicator of which solutions were 'useful'. In particular the value of k signifying an empty set still changes between instances. Despite this, it provides a linear scale with which to evaluate the performance of QAOA for this study in conjunction with r. All measures, aside from the first, require knowledge of optimum value of H_c and so only the value of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ can be of use in a real world setting. Despite this, for these simulated instances, they provide a reliable measure to compare the performance of QAOA between both different problem instances and separate problems of MWIS & MIS. The time taken for a simulation at each stage was not considered as it does not provide a meaningful comparison to QAOA implemented on a real quantum device [26].

2.5.2 Implementation

Simulated circuits for this investigation were created using the Python library Qiskit [32], and were executed with the IBM Aer simulator [32]. Auxillary functions were also implemented in Python with certain functions - in particular, evalution of the objective function $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ - optimised using the machine code compiler Numba. Unless otherwise specified, all graphs were implemented using the Erdős - Rényi method with edge probability 0.5 and varying size using the Networkx package; random weights assigned to each node for MWIS with Numpy. Exact energies for each problem instance were found using the IBM CPLEX library (as implemented in Qiskit) and independently verified using the KAMIS library for MWIS [33].

3. Applying QAOA

3.1 The Importance of Finding Optimal Parameters

Generally, the performance of QAOA improves with increasing layers of p: as $p \to \infty$, the approximation to AQC improves and $r \to 1$. This improvement with growing p is qualified by the increased difficulty of finding optimal parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ for the variational wavefunction. The quality of solutions QAOA finds is highly dependent on the quality of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$, as demonstrated in Figure3.1. That QAOA is so reliant on finding optimal angles presents a serious challenge to its usefulness as an algorithm. Analytical expressions for the value of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ can be found for certain, simple problem instances at p =1, but in general this is impractical. In their original paper Farhi *et al.* suggested a grid search approach where each parameter is discretised into $\mathcal{O}(\text{poly } n)$ points. At level p this requires calculating $N^{\mathcal{O}(p)}$ combinations, and so quickly becomes prohibitively expensive [18] [30]. This means that numerical optimisation over the landscape of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is the most practical option [26] [24]. Unfortunately, the terrain of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ is highly non-convex and is littered with sub-optimal local minima [24]. Moreover, it is highly-dimensional, with $2p$ parameters needing to be optimised for each layer p. As such, these efforts are non-trivial. Without efficient means to find good parameters for problem instances, any potential advantage from QAOA is lost [24].

This places great reward on finding strategies that can aid in the search for good parameters [24]. Equally, when such parameters have been found for one instance, it is of interest to see whether they can be resused on similar problem cases. To begin to understand what strategies could be employed to intelligently search for optimal parameters, we will begin by looking at the first layer of QAOA.

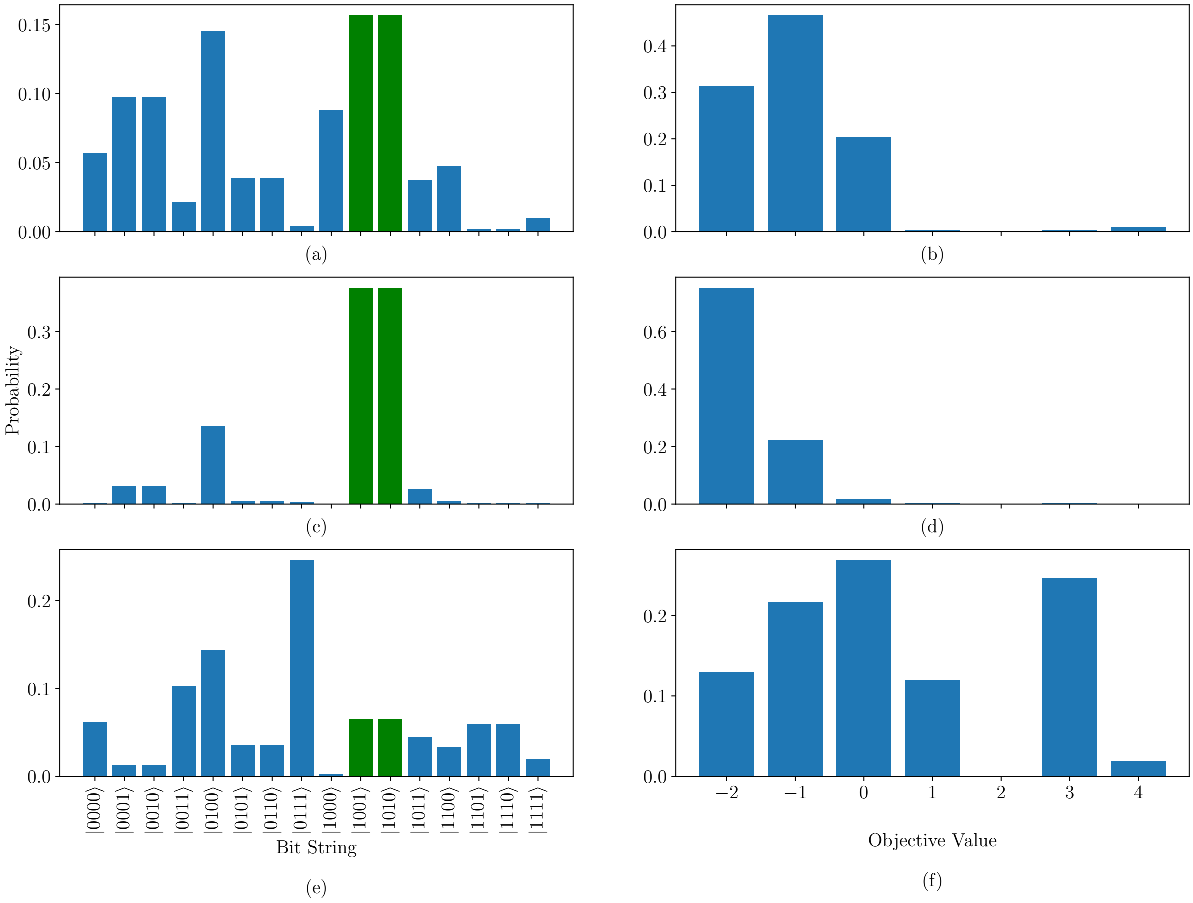
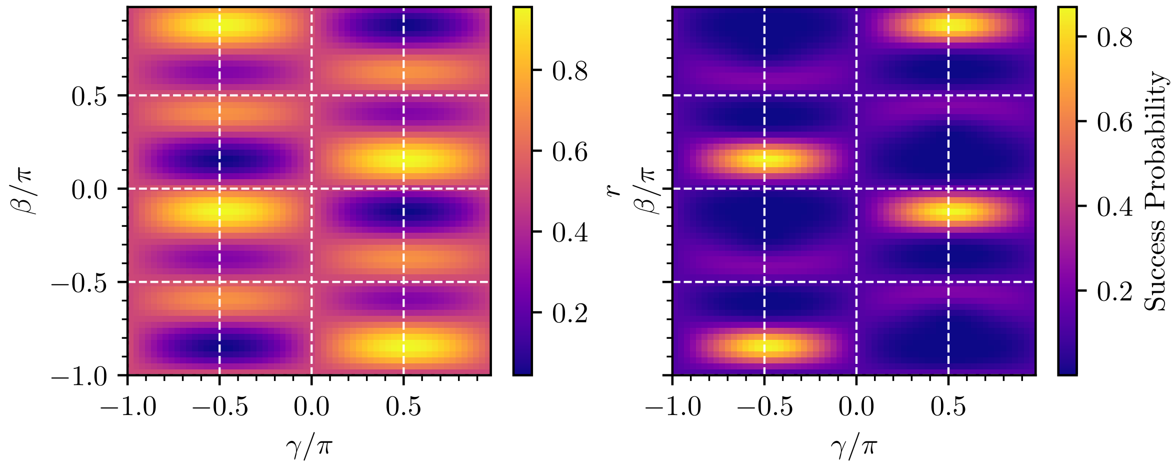


Figure 3.1: QAOA is used to solve the Maximum Independent Set problem for a simple four node graph instance. (a), (c) and (e) show the probability of different solutions being selected for $p = 1, p = 5$ with optimised angles and $p = 5$ with random angles respectively; the optimum solutions are highlighted. (b), (d) and (e) show the corresponding objective function distributions. As can be seen, the quality of solutions drastically increases with as $p : 1 \to 5$. (Note different scales for each subplot). Nevertheless, when QAOA is run at same level of p but with random angles, its performance drops drastically, far below that of $p = 1$. The value of r for each instance is 0.83, 0.95 and 0.59 respectively.



(*a*) (*b*)

Figure 3.2: (a) The r ratio and (b) success probability landscapes as a function of γ & β at $p = 1$ for a five node, unweighted graph. β is has a icity of π whereas γ has a period of 2π.

3.2 QAOA at the $p = 1$ level.

Figure 3.2 shows the value of $F_1(\boldsymbol{\gamma}, \boldsymbol{\beta})$ ($F_i(\boldsymbol{\gamma}, \boldsymbol{\beta})$ denotes $p = i$), the value of r and the success probability for the maximum independent set problem (i.e unweighted) for a simple five node Erdős - Rényi graph evaluated with (γ, β) on a 2-D grid. That QAOA gives a 80% chance of finding the groundstate reflects more on the simplicity of the problem instance than the effectiveness of the algorithm. However, it does show some promising features to be exploited at higher p. First of all, there is an abundance of symmetry within the terrain. In general, QAOA has time-reversal symmetry - i.e $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = F_p(-\boldsymbol{\gamma}, -\boldsymbol{\beta})$ - as both H_C and H_D are real-valued [24]. This can be seen on the 2-D grid as the symmetry between diagonal quadrants. In addition to this, β is periodic on the interval $[0, \pi]$ while γ is periodic on the interval $[0, 2\pi]$. Helpfully, for this unweighted case, this periodicity means each local minimum corresponds to the global optimum value.

In general, regions with high success probability correspond with areas of high r, as predicted. For some cases however, there is a divergence between the values (γ, β) that give the highest success probability and those that give optimal value of $F_1(\boldsymbol{\gamma}, \boldsymbol{\beta})$, a phenomenon also observed in [26] for the similar MAXCUT problem.

Figure 3.3 replicates 3.2 for the MWIS problem. QAOA performs much worse with the maximum success probability dropping to just 14%. That the algorithm does so badly at $p = 1$ is not overly surprising, as this is the worst possible approximation for the full adiabatic transistion. Of more concern is the loss of periodicity in γ; fortunately, optimal values were found be $\mathcal{O}(1)$ for the majority of cases when searches of up to $[-15\pi, 15\pi]$ were carried out. β fortunately maintains its periodicity upon $[0, \pi]$. Curiously, there is a second region of high success probability (γ, β) which does not correspond to optimal values of $F_1(\boldsymbol{\gamma}, \boldsymbol{\beta})$.

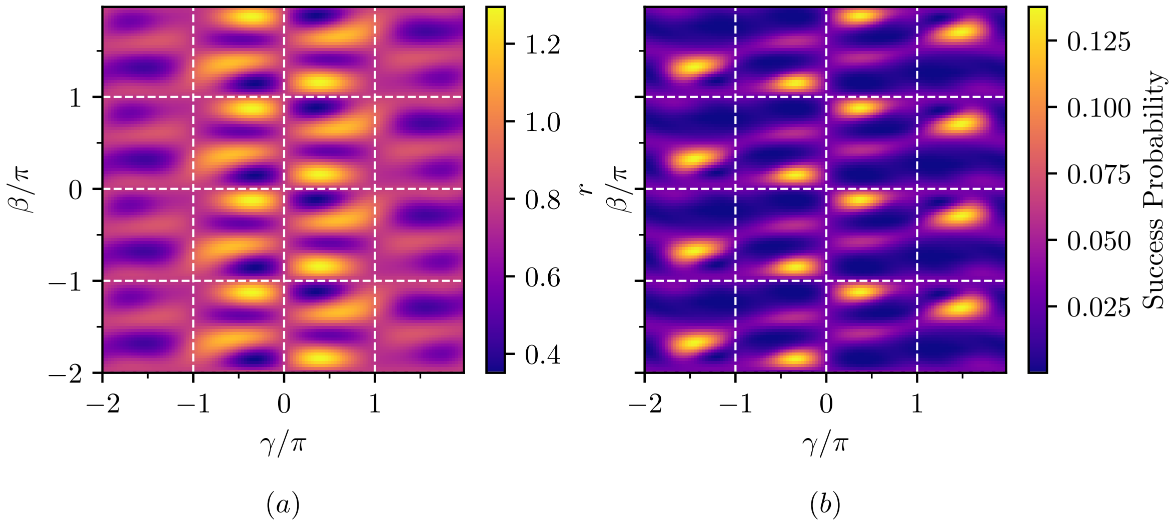


Figure 3.3: (a) The r ratio and (b) success probability landscapes as a function of γ & β at $p = 1$ for the five node, graph used in 3.2 with each node assigned a random weight C_i. Note that different range used to illustrate the loss of periodicity in γ.

3.2.1 Congregation of Optimal Parameters

Recent research into QAOA has suggested that for particular problem cases, optimal parameters tend to congregate [34] [30]. By reusing parameters between different problems the cost of searching for optimal $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ could be significantly reduced, with fewer executions of the quantum circuit and less outer-loop classical optimisation needed. To explore whether this could be applied to independent set problems, first a method of finding optimal parameters was needed.

The approach for optimal parameter finding was adopted from [24]. For each problem instance and given level p, a random set of $2p$ angles were generated in the parameter space using Numpy. From these intialisation points the gradient-based optimisation algorithm BFGS was used to minimise the value of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$. This was implemented using Scipy [35]. This random initialisation was repeated 10^3 times for each problem instance and the set of parameters that gave the best value of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ taken to be the global optimum. Degeneracies due to time-reversal symmetry and the periodicity of β (and γ in the unweighted case) were then reduced.

Fig 3.4 shows the distribution of optimal (γ, β) that gave the best value of $F_1(\boldsymbol{\gamma}, \boldsymbol{\beta})$ for 30 randomly generated graphs with eight nodes. For unweighted problem instances, random intialisation points where chosen from the interval $\gamma, \beta \in [0, \pi]$. As is immediately apparent, there is significant amounts of clustering for optimal parameters. The same graphs were then weighted and a similar parameter search carried out on the interval $(\gamma, \beta) \in [0, 2\pi]([0, \pi])$; despite the extended range of γ all optimum values were found to be between $0 \leq \gamma \leq \pi$. There is an obvious similarity between the two distributions, suggesting the possibility of similarity between the optimal solutions for different graphs instances at higher p. Equally, it also suggests that the optimal parameters for unweighted

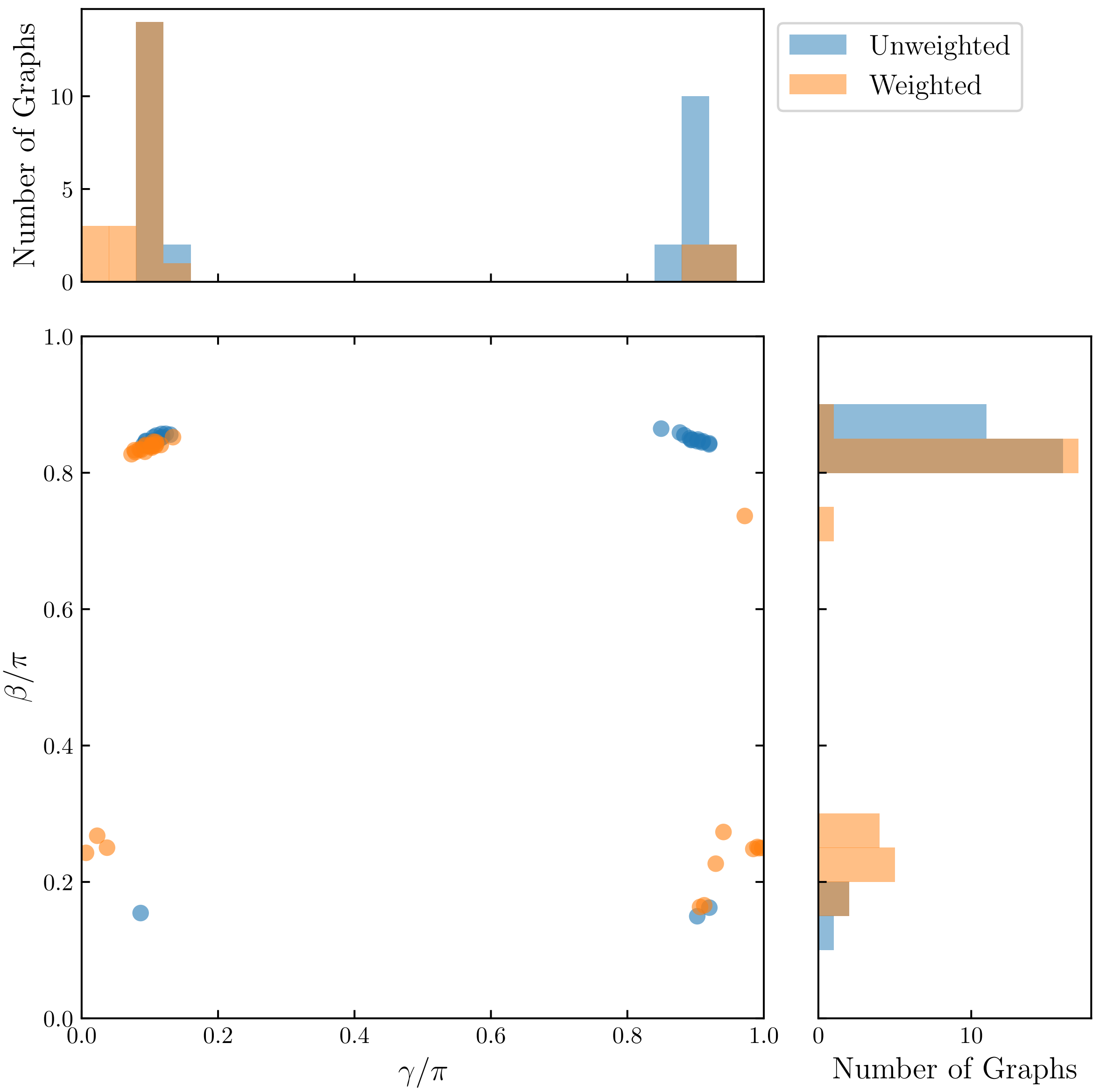


Figure 3.4: A Scatter Plot of values of (γ, β) that give the optimum value of $F_1(\boldsymbol{\gamma}, \boldsymbol{\beta})$ for 30 weighted and 30 unweighted graphs, each with eight nodes.

and weighted graph instances are related. We must be careful not to read too much into these patterns however: $p = 1$ represents the algorithm in its most primitive form and the resemblance of parameters between instances may disappear at the higher levels of p.

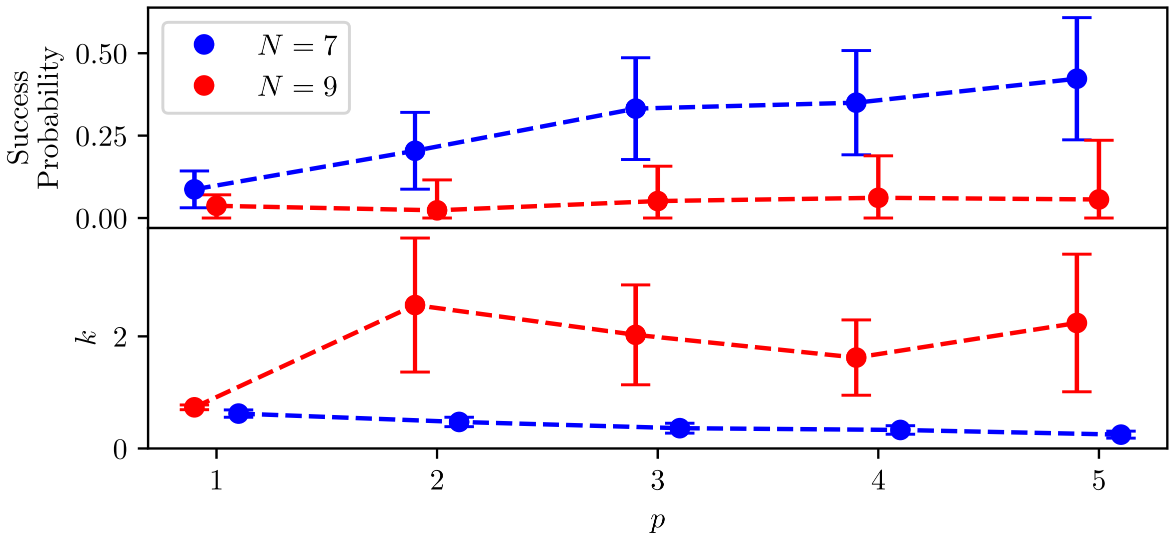


Figure 4.1: The success probability and value of k (as defined in 2.8) for 25 random unweighted graphs using the optimised angles of one single problem instance across different layers of p. The standard deviation of each metric is plotted except for where this would imply a negative probability. The two plots have been offset slightly only to aid clarity.

4. Parameter Optimisation Strategies

To test whether the reusing of optimal parameters is feasible, a single set of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ was found for each of the three cases studied here. This was done using the method outlined in 3.2.1. The effectiveness of these angles was tested on 25 separate problem instances of size $N = 7$ and $N = 9$ respectively at varying levels of p. At each stage, the success probability and the value of k was calculated.

4.1 Dissimilar Graphs

Figure 4.1 shows the effectiveness of parameters from one particular instance of MIS to 25 other unweighted cases. As would be expected, the solutions generally improve with increasing p, with the exception of the value of k for $N = 9, p = 5$. The technique performs better for $N = 7$ graphs than $N = 9$, a reflection of the smaller number of solution possibilities for each graph (2^7 to 2^9). In addition, the standard deviation of k does not grow with increased p again with the exception of $N = 9, p = 5$. It is however much larger for $N = 9$, implying that the increased variability for graph layout does indeed compromise the method. This would appear to limit its applicability for larger graph instances.

Figure 4.2 repeats the same plot for weighted graph instances. Immediately obvious is the poorer performance of the reused parameters across the different graphs. From this we see the apparent correlation of angles at $p = 1$ is more indicative of the ineffectiveness of the algorithm at low level p. That the parameters translate so unfavourably between

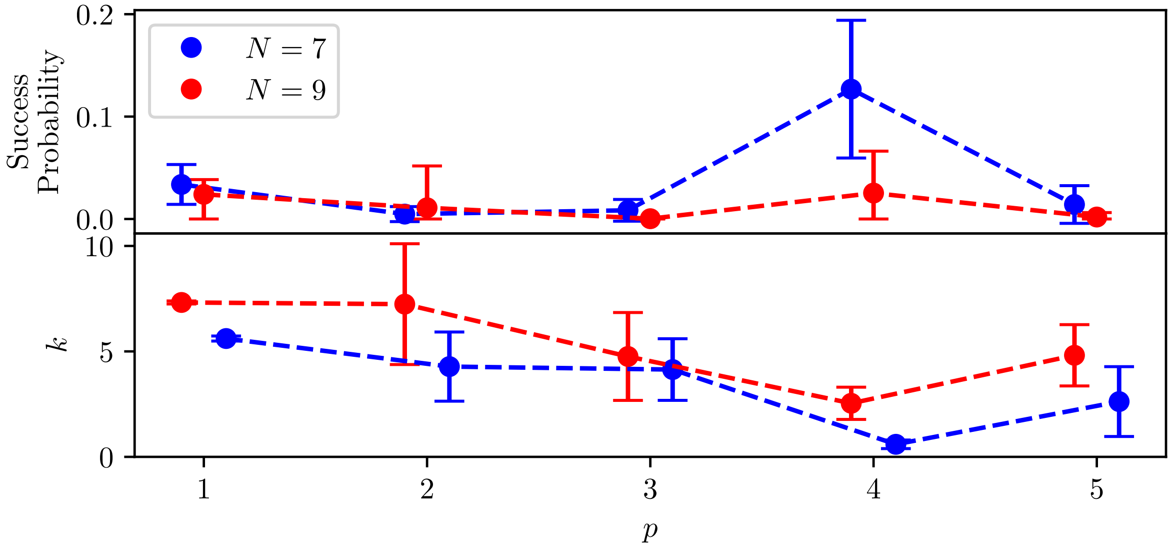


Figure 4.2: Figure 4.1 repeated for 25 random weighted graphs using the optimised angles of one single problem instance across different layers of p. Note the different axis limits for k. The standard deviation of each metric is plotted except for where this would imply a negative probability. The two plots have been offset slightly only to aid clarity.

dissimilar graphs is not overly surprising. For the MAXCUT problem, [30] finds a much better performance of recycled parameters on similar graph cases than on 'worse-case' changes. Moreover, these 'worse-case' changes affect only one node position. This is in contrast to these results which compare across entirely new graph instances.

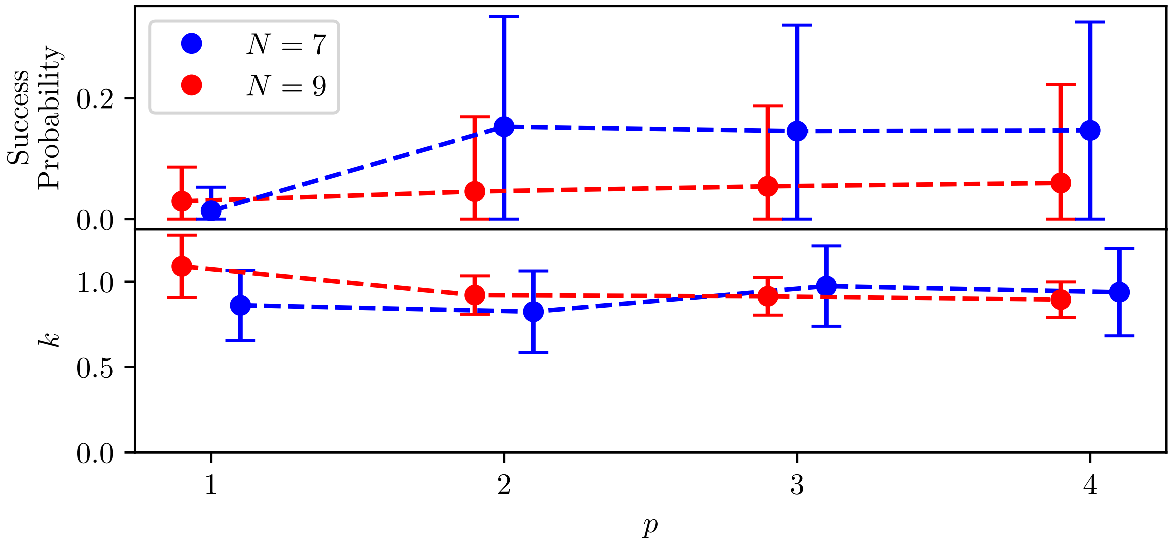


Figure 4.3: The success probability and value of k (as defined in 2.8) for 25 different weight arrangements of the same graph using the optimised angles of one single problem instance across different layers of p. Note the different axis limits for k. The standard deviation of each metric is plotted except for where this would imply a negative probability.

This leads to the question of comparing across similar problem cases. In 4.3, the performance of reusing parameters for the same graph when different weights are applied was measured.. In the use of MWIS to schedule decentralised wireless scheduling, this would model the changing amount of traffic through a network [29].

It is immediately apparent that this leads to a much better performance for QAOA, even with one less layer of p. That the success probability is lower than the unweighted case is a reflection of MIS's degeneracy: often, there is more than one optimal solution, increasing the chance of finding the groundstate. The consistency of the value of k suggests that QAOA finds a greater proportion of optimal solutions found. It is nonetheless peculiar that the values stay constant with increasing p.

While [34] explicitly reuses the same angles across different instances, [30] instead uses them as starting points for optimisation. The results here suggest that for similar graph cases there is merit to the strategy of reusing parameters.

4.2 Parameter Patterns

The feasibility of using the same values of $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ across different problem instances suggests the existence of underlying patterns in the expectation landscape. Indeed Zhou *et al.* in [24] and Crooks in [36] find such patterns after extensive searches over the parameter space for the MAXCUT problem.

MAXCUT is a similarly NP-hard combinatorial optimisation problem and is used as an example problem in much of QAOA literature. It seeks to divide the nodes of a graph into two sets such that the number of edges between the two sets - the number of 'cuts' - is maximised. When used on regular graphs (graphs where each node has the same valence) γ becomes periodic upon $[-\pi/2, \pi/2]$. In addition, the problem exhibits

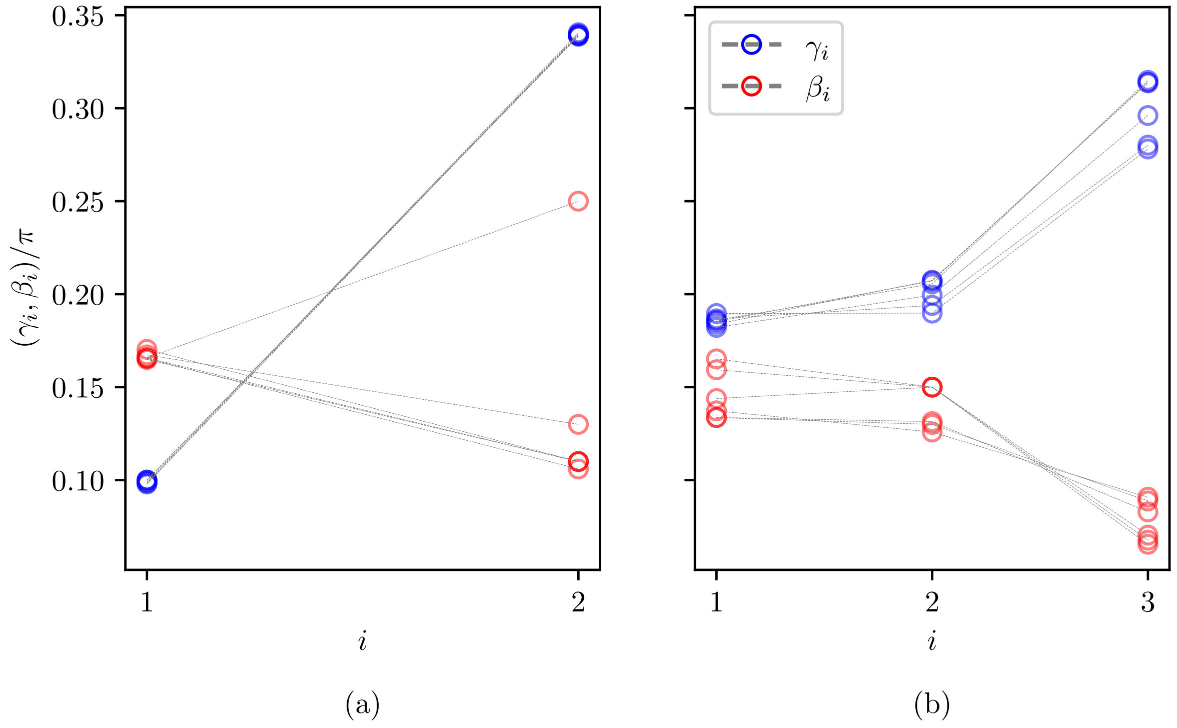


Figure 4.4: The optimal QAOA parameters for the MAXCUT problem for (a) $p = 2$ and (b) $p = 3$ on 3-Regular graphs with 12 nodes. The found optimal angles display clear agreement with the patterns found in [36] [24]. Of particular note is the resemblance of the increasing γ and decreasing β to the schedules used in and adiabatic quantum computing.

$\mathbb{Z}_2$ symmetry. This can be seen by considering the path graph featured in 2.1; the solutions of $|101\rangle$ and $|010\rangle$ are equivalent, with the result that β can be restricted to the interval $[-\pi/4, \pi/4]$. Both works uncover patterns in the optimal values of (γ, β). These simulations are repeated in 4.5 and the known patterns recovered.

A pattern that resembles the adiabatic transition emerges, where γ monotonically increases and β decreases; superficially, this resembles the state of the time-dependent Hamiltonian as it morphs from $H_D \rightarrow H_C$. From this, Zhou *et al.* inferred that there was a slowly varying continous curve for the angles $\gamma_i^\star$ and $\beta_i^\star$. This was used to build a heuristic to give good starting initialisation points for level p by interpolated between the points from $p - 1$.

For MIS, the search is carried out over a much larger parameter space; this has the consequence that each of the 10^3 random initialisation points are less likely to lead the global optimum. This causes the less palatable spread of the found parameters. Despite this, hints of interesting patterns do emerge. Of particular interest is the 'V' shaped pattern for β in (b) which shows that the monotonic decrease and increase in β and γ is seemingly unique to certain problems, and not a universal trait of optimising QAOA. [12] and [37] both showed that QAOA employed alternate methods to traverse critical points for problems with small spectral gaps; this is of particular significance as for certain problem cases, adibatic quantum computing must have an exponentially long

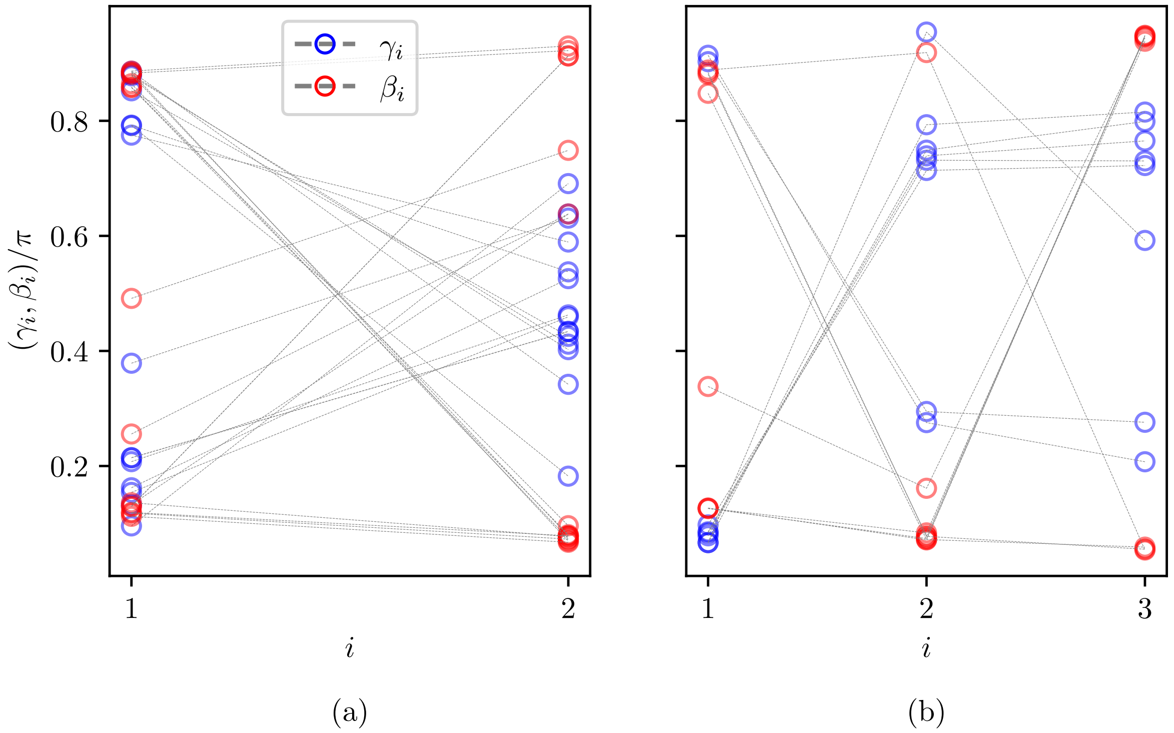


Figure 4.5: The optimal QAOA parameters for (a) $p = 2$ and (b) $p = 3$ on graphs with 10 nodes. Degeneracies in β and γ have been reducedusing the symmetry displayed in 3.2.

time evolution to stay in the groundstate.

Figure 4.6 shows the results when the same search for patterns is attempted for MWIS. Due to the lack of periodicity in γ, a wide starting range was initially chosen of $\gamma(\beta) \in [0, 3\pi]([0, \pi])$. This, combined with the fact that a lack of computing resources mean that only 10^3 random points where chosen, meant an area of 6^p more points were searched using just 10% of the number of starting points. Unsurprisingly, this did not succeed in consistently finding global optima. In [26], an approach was taken for optimisation of QAOA with $p > 1$ by using the success probability as cost function. A similar effort was carried out by first optimising for the best values of $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ and then extracting the $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ that gave the best success probability from the 10^3 minimisations. In general, appreciable divergence was found, supporting the findings of [26] that the energy and success probability local minima were not necessarily aligned.

A more successful approach would, as a basic first step, require greatly expanded computing resources than were available for this project. In addition to this, an unfortunate bug with the Qiskit software used meant that random instances of the quantum circuit froze, halting the optimisation process until a keyboard interrupt was issued. On a six-core machine working in parallel on 10^3 across 12 graphs; this had the unfortunate consequence that simulations could not be left unattended, slowing down progress.

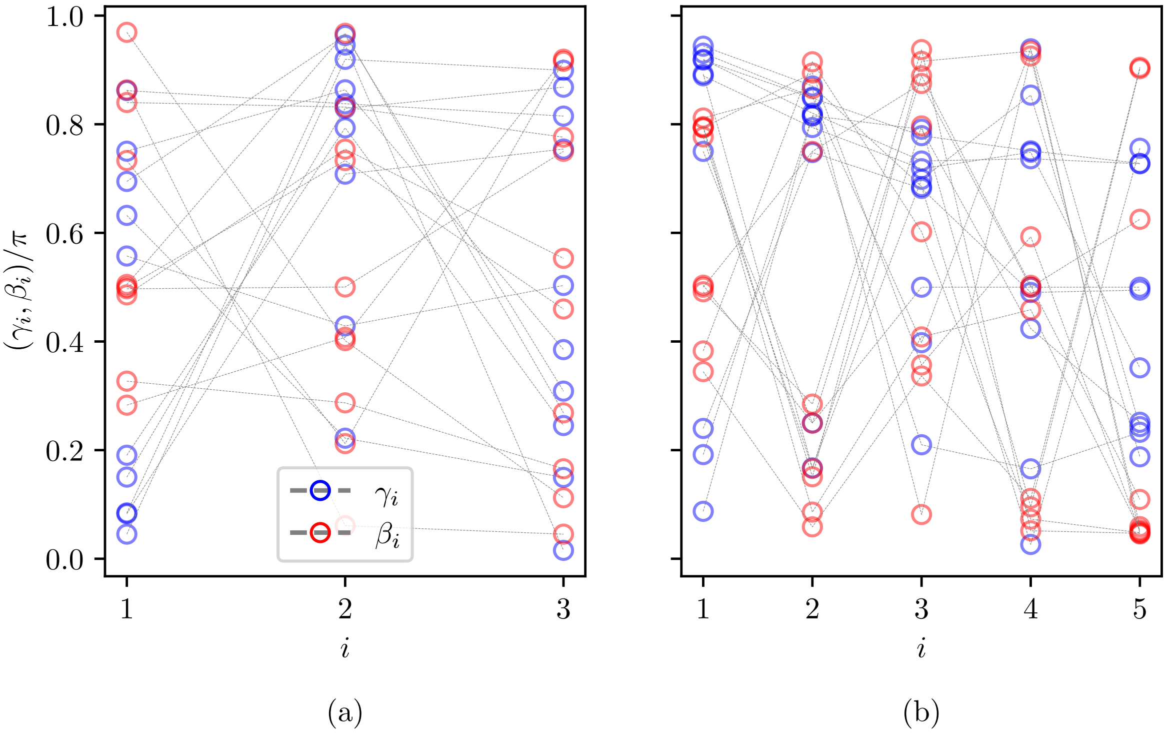


Figure 4.6: The optimal QAOA parameters for (a) $p = 2$ and (b) $p = 5$ on weighted graphs with $N = 10$ nodes. No obvious pattern presents itself. However, this is likely a consequence of the angles being stuck in sub-optimal minima.

5. Discussion and Concluding Remarks

In general, the generality of this project has been curtailed by the small range of graph sizes and layers of p explored. QAOA is an algorithm that has been explored at layers up to $p = 50$ for graphs with up to $N = 22$ nodes [24] [26]. A continuation of the study with more muscular and reliable hardware could determine with greater rigour whether patterns in the optimal parameters for MWIS problem instances exist and whether heuristics that aim to improve the speed of wavefunction optimisation could be increased.

Despite this, a number of interesting results found on certain QAOA problem instances were shown to be exhibited by the MWIS and MIS problem instances. For instance, that areas of highest success probability did not always correspond to areas found by optimising $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$ shows agreement with [26]. This suggests an adaptive strategy for optimisation: first beginning through a search for $F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})$, then at a pre-deterimined threshold (adjusted by the quality of the solution desired) switching to rewarding only values above a certain energy level. Techniques of this sort have been proposed in [38], and are a interesting topic to explore.

An exploration of the use of QAOA on real quantum hardware is of obvious interest. This study was simulated the entire statevector. On an actual quantum system, a picture of the state is instead built up through repeated runs (termed 'shots')of the circuit [39]. This process is inherently stochastic and so determining the suitability of optimisation techniques that reflect this would form an interesting point for further study. Indeed a number of techniques specifically designed for this task have recently been devisied [40]. Benchmarking these with simulations that include realistic quantum noise models would form a natural extension to the study. It was recently shown that in [41] [42] that quantum noise would lead to exponentially-small gradients in the expectation landscapes. These so called 'barren plateaus' paralyse common gradient-based optimisation techniques, such as the one used in this study. This is a major outstanding obstacle to the effectiveness of QAOA and the wider class of variational quantum algorithms it belongs too.

A comparison between the performance of QAOA on MWIS against the best greedy classical algorithms would have been enlightening. Choi *et al.* [31] found that levels above $p > 8$ gave better performance than random search and greedy search algorithms. Indeed, the comparison between different methods used in our respective studies is an interesting path to explore. In [31] J_{ij} was redefined as the average of the node weights c_i and c_j multiplied by a penalty value : $(J_{ij} \to \rho(c_1 + c_2)/2)$ node weights were not scaled [1].

The difference in approach serves to highlight some potential shortcomings of my own method. If the technique of weight scaling were to be combined with the setting of J_{ij} by node weights, it is likely that such small fractions entering the quantum gates would lead to wide expanses between optimal angle values; this would make optimisation even harder. In this case, the angle values fed into the quantum gates were dominated by the constant size of J_{ij}. This scheme would cope increasingly poorly with larger variations in node weight that the range $c_i \in [1, 10]$ used in this study. In particular, the realities of quantum hardware make administering precise rotations difficult (to achieve a prescribed accuracy requires a large overhead in gates [43]) and so the the scheme is more vulnerable to such errors. Nonetheless, it is curious that such a scheme works and with the luxury of time, the limits of its applicability could be more accurately assessed.

5.1 Conclusion

This study has outlined how the principles of quantum mechanics can be harnessed for calculation and the origin of an algorithm that may demonstrate this in the near future. A scheme for approximately solving the NP-hard maximum weighted independent set problem was developed and its implementation as a circuit demonstrated. The terrain of the $p = 1$ level of the algorithm was explored and the implications for $p > 1$ described. Similarities in the optimal angles between different problem instances were discovered at the $p = 1$ level, and the possibility of reusing these parameters for both similar and dissimilar graphs checked. Known patterns for the MAXCUT problem were found to assess the angle searching method, which was then used to search for parameter patterns at higher p.

[1] I have not included details of my comparisons between the two methods as I have only been able to guess at the exact settings - in particular, the values of ρ - used. An email of enquiry was sent but no response had been received by time of writing

References

[1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1982.

[2] J. A. Jones and M. Mosca, "Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer," *The Journal of Chemical Physics*, vol. 109, p. 1648–1653, Aug 1998.

[3] A. K. B. R. e. a. Arute, F., "Quantum supremacy using a programmable superconducting processor.," *Nature*, vol. 574, pp. 505–510, 2019.

[4] "Bloch Sphere Diagram." `https://tex.stackexchange.com/questions/345420/how-to-draw-a-bloch-sphere`. Accessed: 2021-03-04.

[5] V. Kulkarni, M. Kulkarni, and A. Pant, "Quantum computing methods for supervised learning," 2020.

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* USA: Cambridge University Press, 10th ed., 2011.

[7] M. Saeedi and I. L. Markov, "Synthesis and optimization of reversible circuits—a survey," *ACM Comput. Surv.*, vol. 45, no. 2, 2013.

[8] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[9] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.

[10] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), p. 212–219, Association for Computing Machinery, 1996.

[11] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, Aug 2018.

[12] A. J., A. Adedoyin, J. Ambrosiano, P. Anisimov, A. Bärtschi, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, N. Lemons, S. Lin, A. Malyzhenkov, D. Mascarenas, S. Mniszewski, B. Nadiga, D. O'Malley, D. Oyen, S. Pakin, L. Prasad, R. Roberts, P. Romero, N. Santhi, N. Sinitsyn, P. J. Swart, J. G. Wendelberger, B. Yoon, R. Zamora, W. Zhu, S. Eidenbenz, P. J. Coles, M. Vuffray, and A. Y. Lokhov, "Quantum algorithm implementations for beginners," 2020.

[13] D. A. Lidar and T. A. Brun, *Introduction to decoherence and noise in open quantum systems*, p. 3–45. Cambridge University Press, 2013.

[14] P. Shor, "Fault-tolerant quantum computation," in *Proceedings of 37th Conference on Foundations of Computer Science*, pp. 56–65, 1996.

[15] N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, "Layered architecture for quantum computing," *Phys. Rev. X*, vol. 2, p. 031007, Jul 2012.

[16] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," *arXiv preprint arXiv:1802.06002*, 2018.

[17] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, pp. 1–7, 2014.

[18] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014.

[19] D. E. Knuth, "Postscript about np-hard problems," *SIGACT News*, vol. 6, p. 15–16, Apr. 1974.

[20] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," 2019.

[21] B. H. Korte, J. Vygen, B. Korte, and J. Vygen, *Combinatorial optimization*, vol. 1. Springer, 2011.

[22] M. Born and V. Fock, "Beweis des adiabatensatzes," *Zeitschrift für Physik*, vol. 51, pp. 165–180.

[23] Y. Sun, J.-Y. Zhang, M. S. Byrd, and L.-A. Wu, "Adiabatic quantum simulation using trotterization," *arXiv preprint arXiv:1805.11568*, 2018.

[24] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Physical Review X*, vol. 10, no. 2, p. 021067, 2020.

[25] Z. Jiang, E. G. Rieffel, and Z. Wang, "Near-optimal quantum circuit for grover's unstructured search using a transverse field," *Physical Review A*, vol. 95, no. 6, p. 062317, 2017.

[26] M. Willsch, D. Willsch, F. Jin, H. De Raedt, and K. Michielsen, "Benchmarking the quantum approximate optimization algorithm," *Quantum Information Processing*, vol. 19, pp. 1–24, 2020.

[27] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[28] V. Choi, "Adiabatic quantum algorithms for the np-complete maximum-weight independent set, exact cover and 3sat problems," tech. rep., 2010.

[29] P. Du and Y. Zhang, "A new distributed approximation algorithm for the maximum weight independent set problem," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[30] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–8, IEEE, 2019.

[31] J. Choi, S. Oh, and J. Kim, "Energy-efficient cluster head selection via quantum approximate optimization," *Electronics*, vol. 9, no. 10, p. 1669, 2020.

[32] H. Abraham, AduOffei, and R. A. *et al.*, "Qiskit: An open-source framework for quantum computing," 2019.

[33] S. Lamm, C. Schulz, D. Strash, R. Williger, and H. Zhang, "Exactly solving the maximum weight independent set problem on large real-world graphs," in *Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments, ALENEX 2019*, pp. 144–158, SIAM, 2019.

[34] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," *arXiv preprint arXiv:1812.04170*, 2018.

[35] P. Virtanen and t. Gommers, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[36] G. Crooks and N. Rubin, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," *Bulletin of the American Physical Society*, vol. 64, 2019.

[37] M. Streif and M. Leib, "Comparison of qaoa with quantum and simulated annealing," *arXiv preprint arXiv:1901.01903*, 2019.

[38] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving variational quantum optimization using cvar," *Quantum*, vol. 4, p. 256, 2020.

[39] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *Nature Physics*, vol. 17, no. 3, pp. 332–336, 2021.

[40] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, "Classical optimizers for noisy intermediate-scale quantum devices," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 267–277, IEEE, 2020.

[41] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, pp. 1–6, 2018.

[42] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. Coles, "Noise-induced barren plateaus in variational quantum algorithms," *Bulletin of the American Physical Society*, 2021.

[43] A. Asfaw, L. Bello, Y. Ben-Haim, S. Bravyi, N. Bronn, L. Capelluto, A. C. Vazquez, J. Ceroni, R. Chen, A. Frisch, J. Gambetta, S. Garion, L. Gil, S. D. L. P. Gonzalez, F. Harkins, T. Imamichi, D. McKay, A. Mezzacapo, Z. Minev, R. Movassagh, G. Nannicni, P. Nation, A. Phan, M. Pistoia, A. Rattew, J. Schaefer, J. Shabani, J. Smolin, J. Stenger, K. Temme, M. Tod, S. Wood, and J. Wootton., "Learn quantum computation using qiskit," 2020.