# Quantum Number Partitioning

Yangda Bei u7281660
3rd Year Mathematics
Advanced Studies Course SCNC3101
*ANU Research School of Physics, The Australian National University*
*ANU Mathematical Sciences Institute, The Australian National University*

The number partition problem (NPP) is an NP-hard optimisation problem. We show how to formulate the NPP as a maximum cut (max-cut) problem of a complete, weighted graph, allowing us to apply the Quantum Approximation Optimisation Algorithm (QAOA) and its variants, multi-angle QAOA (MAQAOA) and eXpressive QAOA (XQAOA), to find optimal solutions. To benchmark the performances of the ansatzes on NPP's, we derive closed-form expressions at depth $p = 1$ for NPP formulated as max-cut and compare it to the Goemans-Williamson algorithm for max-cut as well as classical NPP heuristics. Our results show that although XQAOA performs better than the Goemans-Williamson algorithm, QAOA, and MAQAOA on complete, weighted graphs, classical NPP heuristics are more effective for solving NPP's for small, problem sizes.

## I. INTRODUCTION

With the current state quantum computing hardware, known as Noisy Intermediate-Scale Quantum (NISQ) devices [1], we are still quite far off from achieving general *quantum supremacy* [2]. However, one particularly promising method to obtain quantum advantage on NISQ devices is the use of variational quantum algorithms (VQA's) [3]. Unlike traditional quantum algorithms, VQA's are a class of hybrid quantum-classical algorithms that seek to optimise an objective function by iteratively adjusting a set of quantum parameters in a quantum circuit, much like a classical neural network.

One of the most prominent VQA's is the Quantum Approximate Optimization Algorithm (QAOA), which has shown promise in solving combinatorial optimization problems [4]. QAOA has been studied extensively on the *maximum cut problem* but has also been applied to practical applications such as the *binary paint shop problem* [5] and the *tail-assignment problem* [6].

In this study, we look at the *number partitioning problem*, which has applications in multiprocessor scheduling [7], voter manipulation [8], and public key encryption schemes [9]. We present our findings on the effectiveness of quantum combinatorial optimisation methods on the number partitioning problem, which to our knowledge has not been done before. In particular, we evaluate the application of QAOA and its variants, multi-angle QAOA (MAQAOA) [10] and eXpressive QAOA (XQAOA) [11], by reformulating number partitioning problems as instances of weighted maximum cut problems.

In Section II, we review some quantum computing foundations. In Section III, we give an overview of the maximum cut problem and the number partitioning problem, and discuss the mapping of the problems to their respective Ising Hamiltonians. In Section IV, we review the Quantum Approximation Optimisation Algorithm (QAOA) and its variants. In Section V, we show our results of maximum cut formulations of number partitioning problems and provide a discussion on our findings in Section VI. We conclude with future directions in Section VII.

The repository for this project can be found here insert url

## II. PRELIMINARIES

### A. Computational Complexity

To begin, we first define the type of problems that were analysed by giving an overview of computational complexity. A *decision problem* is a problem whose inputs can be posed with a yes or no solution. A *complexity class* is a set of computational problems which uses similar computational resources that are needed to solve them. They provide a way of categorising the difficulty of various problems. We list out a few common complexity classes and their definitions:

- **P**: This class contains problems that can be solved in polynomial time with a deterministic algorithm. Equivalently, the running time of the algorithm is bounded by a polynomial function of the input size.

- **NP**: This class contains problems for which solutions can be verified in polynomial time using a non-deterministic algorithm. Equivalently, given a potential solution, it is possible to check whether it is correct or not in polynomial time.

- **NP-Hard**: This class contains problems that are at least as hard as the hardest problems in NP. Equivalently, if there exists a mapping from a problem $p$ in NP to a problem $b$ in NP-hard in polynomial time, then $a$ is also NP-hard.

- **NP-Complete**: This class contains problems that are both NP and NP-hard. Equivalently, a problem $p$ is NP-complete if it is in NP and if every problem in NP can be reduced to $p$ in polynomial time.

Although a candidate solution to an NP-complete problem can be verified in polynomial time, a solution cannot be found in polynomial time with a deterministic algorithm. That is, the most comprehensive method to obtain all solution using any currently known algorithm is by performing a brute force search which increases as the size of the problem grows. The P versus NP problem wishes to determine if these problems can be solved quickly. Whilst methods to compute solutions to NP-complete problems in polynomial time are yet to be discovered, heuristic methods and approximation algorithms are typically used to find optimal solutions.

### B. Quantum Computing

Whilst classical information is encoded in bits, we express quantum information in quantum bits, often referred to as qubits [12]. A single qubit can be thought of as a two-state system, such as a spin-half or a two-level atom. We represent the quantum state of a qubit by the vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which we call its computational basis. More generally, a quantum system has $n$ qubits if it can be represented as a superposition of orthonormal basis elements within a Hilbert space of dimension $2^n$. The $2^n$ orthonormal basis elements is typically represented by $\{|\mathbf{x}\rangle\}$, where $\mathbf{x}$ is a bitstring of length $n$. For multi-state quantum systems, a bitstring of length $n$ corresponds to the Kronecker product of the vectors represented by each bit,

that is, we represent a general quantum state as

$$\begin{aligned}|\psi\rangle &= \sum_{k=1}^{2^n} a_k |x_1\rangle \otimes \cdots \otimes |x_n\rangle \\ &= \sum_{k=1}^{2^n} a_k |x_1 \ldots x_n\rangle, \end{aligned} \tag{1}$$

where $x_i \in \{0, 1\}$ and $a_k \in \mathbb{C}$. According to the Born rule, $|a_k|^2$ is the probability of configuration $k$ being measured and $\sum_{k=1}^{2^n} |a_k|^2 = 1$. To preserve this norm, operators acting on single qubits must be $2 \times 2$ unitary matrices. Of these, the most common are the Pauli operators given by

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \text{ and } \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2}$$

### C. Ising Models

The Ising model is a mathematical model that describes phase transitions. The atoms in a crystal are represented by vertices and bonds between adjacent atoms are represented by edges. The mapping is such that a solution to the problem corresponds to a spin configuration that minimizes the energy of the Ising spin glass model. Classically, an Ising model can be represented as a quadratic function of a set of $n$ spin states $s_i = \pm 1$ given by the Hamiltonian

$$H(s_1, \ldots, s_n) = -\sum_{i<j} J_{ij} s_i s_j - \sum_{i=1}^{n} h_i s_i, \tag{3}$$

where $J$ is a real, symmetric matrix with zero diagonal often referred to as the coupling matrix, and $h$ is a vector of real coefficients often referred to as the local magnetic field. The quantum version of the Ising model simply maps each spin variable $s_i$ to the Pauli-$Z$ matrix $\sigma_i^z$ which acts on the $i$-th qubit in a Hilbert space of $n$ qubits. Equation 3 can then be expressed as an operator given by $H \in \mathbb{R}^{2^n} \times \mathbb{R}^{2^n}$, where

$$H(\sigma_1^z, \ldots, \sigma_n^z) = -\sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z - \sum_{i=1}^{n} h_i \sigma_i^z. \tag{4}$$

Since the decision form of the Ising model determining whether $H_0 \leq 0$ is NP-complete [13] (where $H_0$ is the ground state of the Hamiltonian), we can map any NP-complete problem in polynomial time to solving for the ground state of an Ising spin model [14]. We will use the quantum Ising formulation approach [15] to find the ground states of combinatorial optimisation problems that we introduce next.

## III. NP-HARD PROBLEMS CONSIDERED

### A. Maximum Cut (Max-Cut)

The maximum cut (max-cut) problem is a decision problem known to be NP-complete [16]. Given an undirected graph and an integer $k$, the cut problem determines whether there is a partition of the nodes into two complementary subsets such that the number of edges between the two subsets is at least $k$. A *maximum cut* is a cut that is as large as any cut for that graph. Figure 1 shows a maximum cut for an example graph.

The optimisation version of this problem is NP-hard which aims to maximise the sum of the weights of the edges in the cut. Let $G = (V, E)$ be a weighted, undirected $n$-graph with edge weights $w_{ij} > 0$, $w_{ij} = w_{ji}$ for all $(i, j) \in E$. For each node $i$, we assign $x_i = \pm 1$ to separate the nodes into the two complementary subsets. The max-cut problem can then be formulated as a binary quadratic program of the form

$$\begin{aligned} &\text{maximise} \quad C_{MC} = \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - x_i x_j) \\ &\text{subject to} \quad x_i \in \{-1, 1\} \quad \forall i \in V. \end{aligned} \tag{5}$$

Equivalently, we can formulate the max-cut problem as an Ising spin glass model through the trivial mapping $x_i \mapsto s_i$, where $s_i \in \{-1, 1\}$. We then obtain the Ising energy function for the max-cut problem given by

$$\begin{aligned} H(s_1, \ldots, s_n) &= \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(1 - s_i s_j) \\ &= \sum_{i<j} \frac{w_{ij}}{2} - \sum_{i<j} \frac{w_{ij}}{2} s_i s_j. \end{aligned} \tag{6}$$

Setting $J_{ij} := w_{ij}/2$ gives an Ising energy function that satisfies Equation 3 up to a constant. Finally, promoting the spin variables to Pauli-$Z$ spin matrices $s_i \mapsto \sigma_i^z$ gives a cost Hamiltonian $H_{MC}$ in the form of Equation 4, which returns

$$H_{MC} = -\sum_{i<j} \frac{w_{ij}}{2} \sigma_i^z \sigma_j^z + \sum_{i<j} \frac{w_{ij}}{2}. \tag{7}$$

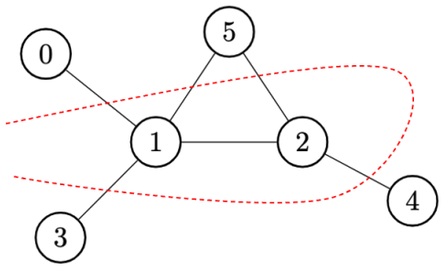



**FIG. 1:** *Unweighted graph with max-cut of 5. The cut can be represented by the disjoint sets* $S_1 = \{0, 3, 4, 5\}$ *and* $S_2 = \{1, 2\}$*, which corresponds to a bitstring encoding of* $\mathbf{x} = 100111$ *or* $\mathbf{x} = 011000$*.*

### B. Number Partitioning (NPP)

The number partitioning problem (NPP) is another well-known NP-complete problem [16]. Given a set of integers, the problem is to determine whether there exists a partition of the input set into two subsets, such that the difference between the sums of the two subsets are equal.

The optimisation version of this problem is NP-hard which aims to minimise the difference in the sums of the two partitions. A solution is called a perfect partition if the difference is 0 or 1 if the sum of the entire set is even or odd respectively. Let $S = \{n_1, \ldots, n_N\}$ be a set of $n$-positive integers. Then, the number partitioning problem becomes

$$\begin{aligned} &\text{minimise} \quad C_{NPP} = \left( \sum_{i=1}^{N} n_i x_i \right)^2 \\ &\text{subject to} \quad x_i \in \{-1, 1\} \quad \forall i \in V. \end{aligned} \tag{8}$$

Again, we can equivalently formulate NPP as an Ising spin glass model by the trivial mapping $x_i \mapsto s_i$, where $s_i \in \{-1, 1\}$. We obtain the Ising energy function for NPP given by

$$\begin{aligned} H(s_1, \ldots, s_n) &= \left( \sum_{i=1}^{N} n_i s_i \right)^2 \\ &= 2 \sum_{i<j}^{N} n_i n_j s_i s_j + \sum_{i=1}^{N} n_i^2. \end{aligned} \tag{9}$$

Setting $J_{ij} := -2n_in_j$ gives and Ising energy function that satisfies Equation 3 up to a constant. Again, promotion of the spin variables to Pauli-$Z$ spin matrices gives a cost Hamiltonian $H_{NPP}$ in the form of Equation 4, which gives

$$H_{NPP} = 2\sum_{i<j}^{N} n_i n_j \sigma_i^z \sigma_j^z + \sum_{i=1}^{N} n_i^2. \tag{10}$$

Although the optimisation problem is NP-hard, there are many classical heuristics and approximation algorithms that can solve or give an approximate solution efficiently [17]. There are two factors that decide the hardness of a given problem: the size of the set $S$ denoted by $n$, and the maximum number of bits needed to represent the integers denoted by $m$. The ratio $m/n$ then predicts the difficulty of the problem, with $m/n \leq 1$ considered to be easy and $m/n > 1$ to be hard [18].

## IV. QUANTUM APPROXIMATE OPTIMISATION ALGORITHM (QAOA)

Combinatorial optimisation problems can be formulated by specifying $n$ bits and $m$ clauses, where each clause is a constraint on a subset of the bits. These clauses are satisfied for certain assignments and unsatisfied for others. The objective function $C : \{0,1\}^n \to \mathbb{Z}$ defined on $n$ bit strings, is the number of satisfied clauses given by

$$C(x) = \sum_{\mu=1}^{m} C_\mu(x), \tag{11}$$

where $x = x_1x_2\ldots x_n$ is the bit string and $C_\mu(x) = 1$ if it satisfies the clause $C_\mu$ or 0 otherwise. For approximation optimisation algorithms, there is a guaranteed ratio on how close the approximate solution is to the optimum of the problem, that is, a bitstring $x' \in \{0,1\}^n$ is guaranteed to be produced with a high probability satisfying

$$C_{max} \geq C(x') \geq rC_{max}, \tag{12}$$

where $C_{max} = \max_x C(x)$.

QAOA attempts to find an approximate solution to combinatorial optimisation problems using a variational quantum approach. Variational quantum circuits train an ansatz, a quantum operation that depend on a set of parameters $\boldsymbol{\theta}$. The ansatz is trained using a hybrid quantum-classical loop to solve the optimisation problem

$$\boldsymbol{\theta^*} = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta}). \tag{13}$$

There are three components in a QAOA mapping on a problem instance on $n$ binary variables $x \in \{0,1\}^n$ which consists of:

1. An *initial quantum state*, which is taken to be the uniform superposition of all computational basis states of $x$ given by

$$|s\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2n}} \sum_{n=0}^{2n-1} |x\rangle, \tag{14}$$

where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

2. The *problem Hamiltonian* (or *cost Hamiltonian* as this is the "cost function"), given by

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{\mu=1}^{m} e^{-i\gamma C_\mu}. \tag{15}$$

3. The *mixing Hamiltonian*, given by

$$U(B,\beta) = e^{-i\beta B} = \prod_{\nu=1}^{n} e^{-i\beta\sigma_\nu^x}. \tag{16}$$

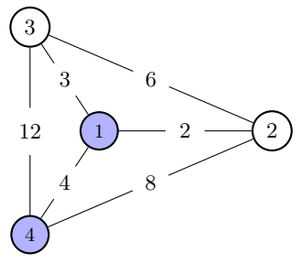



**FIG. 2:** *The set $S = \{1,2,3,4\}$ can be partitioned into subsets $S_1 = \{1,4\}$ and $S_2 = \{2,3\}$. When the partition problem is transformed into a complete graph, this partition corresponds to the maximum cut of $12 + 3 + 2 + 8 = 25$.*

For $p \in \mathbb{N}$, there are $2p$ variational parameters that govern the QAOA circuit. Since $C$ has integer eigenvalues, we take $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ where $\gamma_i \in [0, 2\pi]$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ where $\beta_i \in [0, \pi]$. The variational state obtained after depth $p$ is given by

$$|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle = U(B,\beta_p)U(C,\gamma_p)\ldots U(B,\beta_1)U(C,\gamma_1)|s\rangle, \tag{17}$$

where the variational parameters are found using a classical optimiser. The optimal parameters that maximise the expectation value is given by $(\boldsymbol{\gamma^*}, \boldsymbol{\beta^*})$, where

$$(\boldsymbol{\gamma^*}, \boldsymbol{\beta^*}) = \arg\max_{\boldsymbol{\gamma},\boldsymbol{\beta}} \langle C\rangle, \tag{18}$$

and $\langle C\rangle$ is the expectation value given by

$$\langle C\rangle = \langle\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})|C|\psi_p(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle. \tag{19}$$

### A. Applying QAOA to Number Partitioning

Since NPP and max-cut are NP-complete, we can formulate the NPP as an instance of max-cut so that we can apply QAOA to find optimal solutions. Let $S = \{n_1, \ldots, n_N\}$ be an NPP. Consider the complete weighted $N$-vertex graph $G = (V, E)$ where each $i \in V$ takes on the value $n_i$ and each edge has weight $w_{ij} = n_in_j$. Then, the optimal weighted max-cut of the graph partitions the vertices into the optimal solution to the original NPP. Figure 2 gives an example of an NPP solved using max-cut. We can represent the cost Hamiltonian of the NPP in terms of the cost Hamiltonian of max-cut as

$$\begin{aligned} H_{NPP} &= \left(\sum_{i=1}^{N} n_i\sigma_i^z\right)^2 \\ &= 2\sum_{i<j} n_in_j\sigma_i^z\sigma_j^z + \sum_{i=1}^{N} n_i^2 \\ &= -4H_{MC} + 2\sum_{i<j} n_in_j + \sum_{i=1}^{N} n_i^2 \\ &= -4H_{MC} + \left(\sum_{i=1}^{N} n_i\right)^2. \end{aligned} \tag{20}$$

For depth $p = 1$, we provide an analytical form to finding the expectation value of the cost function of QAOA on the max-cut formulation of an NPP in terms of $\beta$ and $\gamma$. We denote this cost as $\text{QAOA}_1$.

**Theorem 1.** *Consider the QAOA$_1$ state $|\gamma,\beta\rangle$ of an NPP transformed into a max-cut problem for a graph $G=(V,E)$. Then the expectation value of the cost $\langle C_{uv}\rangle$ for each edge $\{u,v\}\in E$ is given by*

$$\langle C_{uv}\rangle = \frac{w_{uv}}{2} + \frac{w_{uv}}{4}\left[\sin 4\beta \sin\gamma'_{uv}\left(\prod_{w\in F}\cos\gamma'_{wv} + \prod_{w\in F}\cos\gamma'_{uw}\right) + \sin^2 2\beta\left(\prod_{w\in F}\cos\left(\gamma'_{uw}+\gamma'_{vw}\right) - \prod_{w\in F}\cos\left(\gamma'_{uw}-\gamma'_{vw}\right)\right)\right],$$

*where $F = V\setminus\{u,v\}$ and $\gamma'_{uv} = -4w_{uv}\gamma_{uv}$.*

*Proof.* We simplify Equation 13 of [11] for a complete graph using our formulation of an NPP as a max-cut problem. For complete graphs, we have that $e = d = F$, where $e = \mathcal{N}(v)\setminus\{u\}$, $d = \mathcal{N}(u)\setminus\{v\}$, and $F = \mathcal{N}(u)\cap\mathcal{N}(v)$. By definition, those product terms become 1, returning the simplified result after renaming. □

Since Theorem 1 finds the expectation value for each edge, we simply sum the expectation values over all edges of a graph to find the overall expectation value for QAOA$_1$. For complete, $n$-vertex graphs like the instances that we are considering, when we transform an NPP into a max-cut problem, the number of edges is given by $\binom{n}{2}$, yielding $\mathcal{O}(n^2)$ operations. We also see that there are $n-2$ nodes to consider in the set $F$ for a given edge, which means that the product terms are calculated in $\mathcal{O}(n)$ time. Therefore, calculating the expectation value for a graph at depth $p=1$ can be found in $\mathcal{O}(n^3)$ time. However, we note that to find a bitstring that encodes an approximate solution to the graph requires the measurement of a quantum circuit that represents the problem (see assignment document and notebook for more details).

### B. Multi-Angle QAOA (MAQAOA)

The Multi-Angle QAOA (MAQAOA) varies from QAOA by introducing angle parameters for each summand of the cost and mixing operators as opposed to having a single angle parameter for the cost operator and a second angle for the mixing operator. The angles are chosen to provide more flexibility in optimising the circuit's parameters and improving the algorithm's performance as it is able to explore and determine the feasibility of different solutions more efficiently [10].

For MAQAOA, the problem and mixing Hamiltonians are defined respectively as

$$U(C,\boldsymbol{\gamma}) = e^{-i\sum_\mu \gamma_\mu C_\mu} = \prod_{\mu=1}^{m} e^{-i\gamma_\mu C_\mu}, \text{ and} \tag{21}$$

$$U(B,\boldsymbol{\beta}) = e^{-i\sum_\nu \beta_\nu B_\nu} = \prod_{\nu=1}^{n} e^{-i\beta_\nu \sigma_\nu^x}, \tag{22}$$

where $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_m)$ and $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_n)$. For depth $p=1$, MAQAOA would then generate a state of the form

$$\begin{aligned}|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle &= U(B,\boldsymbol{\beta})U(C,\boldsymbol{\gamma})|s\rangle \\ &= \prod_{\mu=1}^{m} e^{-i\gamma_\mu C_\mu}\prod_{\nu=1}^{n} e^{-i\beta_\nu\sigma_\nu^x}|s\rangle.\end{aligned} \tag{23}$$

Herrman et al. [10] proved that MAQAOA guarantees an exact solution as $p\to\infty$. We again present the analytical formula for the expectation value of the cost function of MAQAOA on max-cut formulation of an NPP. We denote this cost as MAQAOA$_1$.

**Theorem 2.** *Consider the MAQAOA$_1$ state $|\gamma,\beta\rangle$ for an NPP transformed into a max-cut problem for a graph $G=(V,E)$. Then, the expectation value of the cost function $\langle C_{uv}\rangle$ on an edge $\{u,v\}\in E$ is given by*

$$\langle C_{uv}\rangle = \frac{w_{uv}}{2} + \frac{w_{uv}}{2}\left[\cos 2\beta_u \sin 2\beta_v \sin\gamma'_{uv}\prod_{w\in F}\cos\gamma'_{wv} + \sin 2\beta_u\cos 2\beta_v\sin\gamma'_{uv}\prod_{w\in F}\cos\gamma'_{uw} + \frac{1}{2}\sin 2\beta_u\sin 2\beta_v\left(\prod_{f\in F}\cos\left(\gamma'_{uf}+\gamma'_{vf}\right) - \prod_{f\in F}\cos\left(\gamma_{uf}-\gamma'_{vf}\right)\right)\right],$$

*where $F = V\setminus\{u,v\}$ and $\gamma'_{uv} = -4w_{uv}\gamma_{uv}$.*

*Proof.* Following the same process as the proof to Theorem 1 yields the result. □

Similar to QAOA, finding the total expectation value for a complete graph encoding an NPP has time complexity $\mathcal{O}(n^3)$. However, compared to QAOA which has only two variational parameters, MAQAOA requires $|V|+|E| = \mathcal{O}(n)+\mathcal{O}(n^2)$ variational parameters needing to be optimised.

### C. eXpressive QAOA (XQAOA)

eXpressive QAOA (XQAOA) is a generalisation of MAQAOA, providing the algorithm with more flexibility to traverse even more possible solutions [11]. It defines a new $\boldsymbol{\alpha}$-dependent operator $U(A,\boldsymbol{\alpha})$ given by

$$U(A,\boldsymbol{\alpha}) = e^{-i\sum_j \alpha_j A_j} = \prod_{j=1}^{n} e^{-i\alpha_j\sigma_j^y}, \tag{24}$$

where $\alpha_j\in[0,\pi]$. We then let the mixing Hamiltonian be the product of $U(B,\boldsymbol{\beta})$ and $U(A,\boldsymbol{\alpha})$. For depth $p=1$, XQAOA would then generate a state of the form

$$\begin{aligned}|\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\alpha}\rangle &= U(A,\boldsymbol{\alpha})U(B,\boldsymbol{\beta})U(C,\boldsymbol{\gamma})|s\rangle \\ &= \prod_{\nu=1}^{n} e^{-i\alpha_\nu\sigma_\nu^y}e^{-i\beta_j\sigma_j^x}\prod_{\mu=1}^{m} e^{-i\gamma_\mu C_\mu}|s\rangle,\end{aligned} \tag{25}$$

where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_n)$, $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_n)$, and $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_m)$.

**Theorem 3.** *Consider the XQAOA state $|\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\alpha}\rangle$ for an NPP transformed into a max-cut problem for a graph $G=(V,E)$. Then, the expectation value of the cost function $\langle C_{uv}\rangle$ on an edge $\{u,v\}\in E$ for depth $p=1$ is given by*

$$\begin{aligned}\langle C_{uv}\rangle = \frac{w_{uv}}{2} + \frac{w_{uv}}{2}\Bigg[&\cos 2\alpha_u \cos 2\alpha_v \sin\gamma'_{uv}\left(\cos 2\beta_u \sin 2\beta_v \prod_{w\in F}\cos\gamma'_{wv} + \sin 2\beta_u \cos 2\beta_v \prod_{w\in F}\cos\gamma'_{uv}\right)\\ &- \frac{1}{2}\sin 2\alpha_u \sin 2\alpha_v\left(\prod_{w\in F}\cos(\gamma'_{uf}+\gamma'_{vf}) + \prod_{w\in F}\cos(\gamma'_{uf}-\gamma'_{vf})\right)\\ &+ \frac{1}{2}\cos 2\alpha_u \sin 2\beta_u \cos 2\alpha_v \sin 2\beta_v\left(\prod_{w\in F}\cos(\gamma'_{uf}+\gamma'_{v}f) - \prod_{w\in F}\cos(\gamma'_{uf}-\gamma'_{vf})\right)\Bigg],\end{aligned}$$

*where $F = V\setminus\{u,v\}$ and $\gamma'_{uv} = -4w_{uv}\gamma_{uv}$.*

*Proof.* Following the same process as the proof to Theorem 1 yields the result. □

Several variants of XQAOA were proposed by Vijendran et al. [11]. For our simulations with XQAOA, we apply the X=Y mixer variant XQAOA$^{\text{X=Y}}$ where we set $\alpha=\beta$ as it was found to have the best approximation ratio of max-cut for $D$-regular graphs of degree 3-10 with 128 and 256 vertices. We denote this cost as XQAOA$_1^{\text{X=Y}}$.

## V. SIMULATIONS

### A. Methods

We evaluate the performance of QAOA$_1$, MAQAOA$_1$, and XQAOA$_1^{\text{X=Y}}$ on max-cut instances of complete, weighted graphs, converted from randomly generated number partitioning problems. Problem ratios $0.2 \le m/n \le 2.0$ (for $n=10$ and $2\le m\le 20$) and $0.5\le m/n\le 4.0$ (for $2\le n\le 16$ and $m=8$) were considered.

For each ratio, 25 problem instances were generated. To find the expectation values for the problems, we used Equation 20 alongside the analytical expressions from theorems 1, 2, and 3 to find the number partitioning cost for QAOA$_1$, MAQAOA$_1$, and XQAOA$_1^{\text{X=Y}}$ respectively. We used the L-BFGS-B algorithm provided by `scipy.optimize.minimize` to optimise the variational parameters for each of the algorithms that minimises the cost. Since these algorithms are heavily dependent on their initial parameters (see Section VI A), we randomly sampled the initial parameters ($\beta\in[0,\pi]$, $\gamma\in[0,2\pi]$) 20 times and found the cost value after optimising those parameters for each problem instance, creating a distribution. We also applied the Goemans-Williamson algorithm for max-cut to our problem instances [19], where the relaxed problem was first solved with the SCS algorithm provided by `CVXPY` before generating 20 random vectors for hyperplane-rounding.

We then compare the results with the classical number partitioning heuristic algorithms, greedy and Karmarkar-Karp [17]. These algorithms are deterministic and so were only needed to be run once for each problem instance. The median cost for each ratio was then compared to the distributions obtained by QAOA$_1$, MAQAOA$_1$, XQAOA$_1^{\text{X=Y}}$, and Goemans-Williamson. The results were benchmarked with the median partition difference found by using the Gurobi solver [20] for each problem ratio which found the optimal solution for each problem.

### B. Results

Benchmarking the QAOA variants against the Goemans-Williamson, greedy, and Karmarkar-Karp algorithms, we found that the XQAOA$_1^{\text{X=Y}}$ variant consistently outperformed all of the other QAOA variants as well as the Goemans-Williamson algorithm for max-cut, but did worse than the classical heurstic greedy and Karmarkar-Karp algorithms.

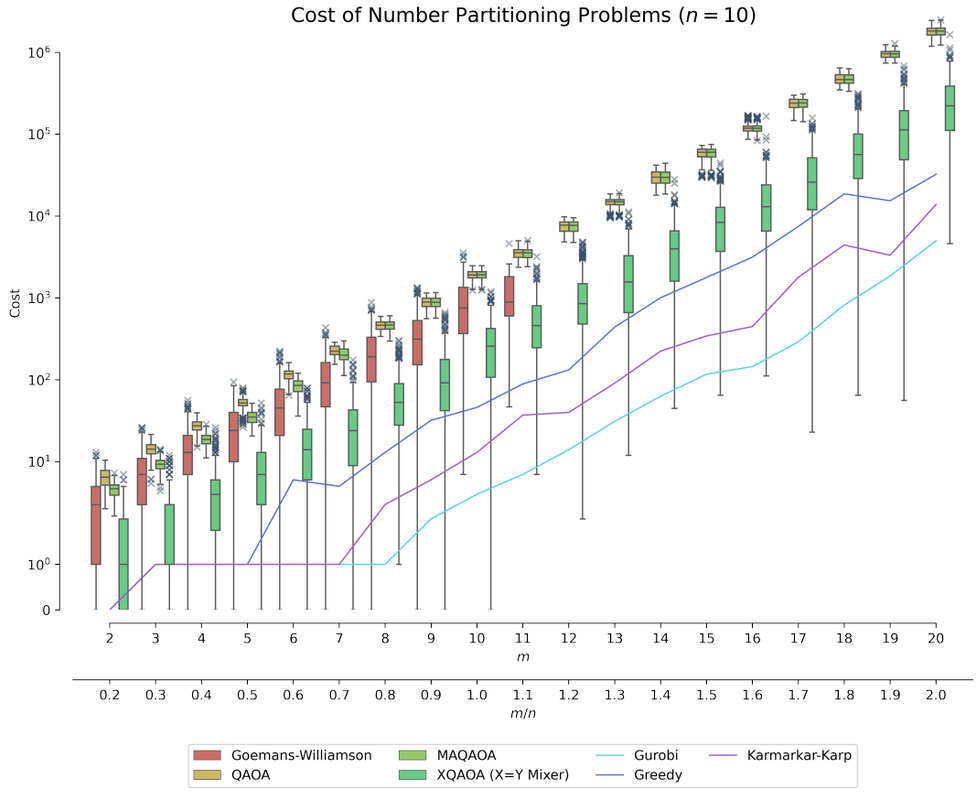
In the dataset with fixed $n=10$ (Figure 3a), we see that as $m$ increased, the median partition difference increased logarithmically for all algorithms. The boxplots reveals that the interquartile ranges of the costs of XQAOA$_1^{\text{X=Y}}$ is significantly lower for all problem ratios. We note that the SCS algorithm used in the Goemans-Williamson algorithm was not able to solve any problem instance for $m>11$ as the problems were unbounded.

In the dataset with fixed $m=8$ (Figure 3b), we see that the median costs of XQAOA$_1^{\text{X=Y}}$ is lower than the other variants as well as being lower than that of Goemans-Williamson for $n>3$. Figure 4 depicts only the probabilistic plots for varying $n$ values on a linear axis. It becomes clear then that the median values for XQAOA$_1^{\text{X=Y}}$ remains in roughly the same range as $n$ varies with positively skewed distributions for all problem instances. The performance of QAOA$_1$ and MAQAOA$_1$ are on par with each other, but struggled with problems considered "easy" ($m/n<1$).

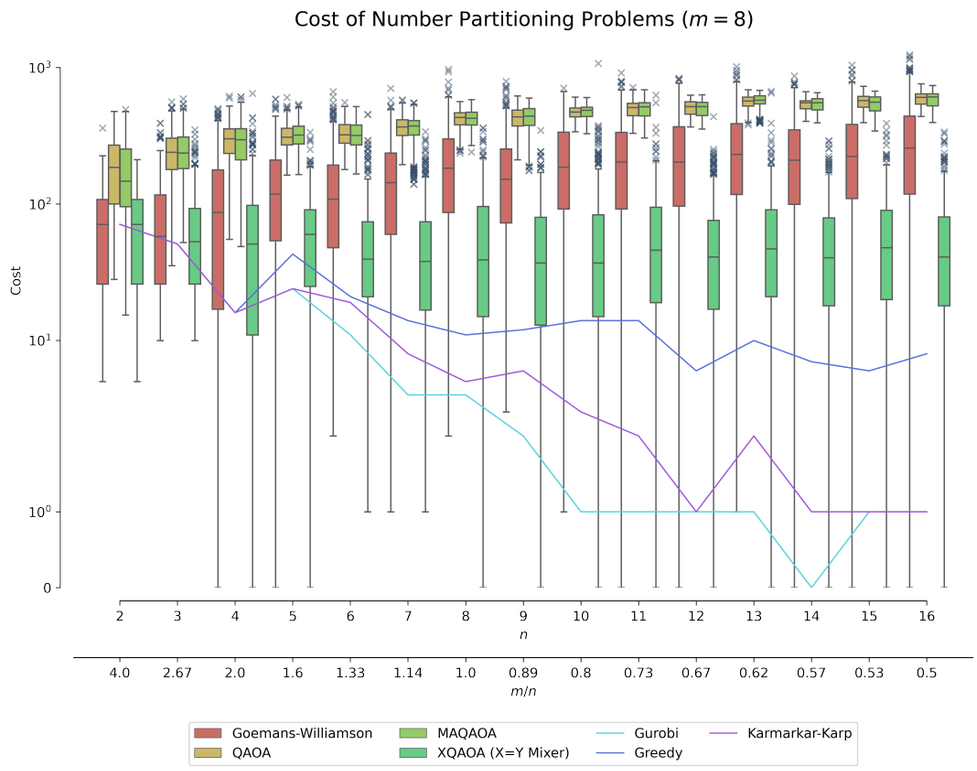
## VI. DISCUSSION

In general, lower medians and overall distribution of cost values obtained by XQAOA$_1^{\text{X=Y}}$ compared to QAOA$_1$ and MAQAOA$_1$ indicates that XQAOA$_1^{\text{X=Y}}$ is more likely to find better solutions for any random set of initial parameter values. As was the purpose of introducing the additional products in the mixing Hamiltonian, it is likely that the comparative performance of XQAOA$_1^{\text{X=Y}}$ is due to different trajectories traced through the Hilbert solution space, attributed to the Pauli-$Y$ gates governed by the variational parameter $\boldsymbol{\alpha}$. Vijendran et al. [11] also reported similar findings, however, the XQAOA$_1^{\text{X=Y}}$ outperformed all algorithms and was on par with the Gurobi optimser. Indeed, their analysis was on $D$-regular graphs with 128 and 256 vertices ($3\le D\le 10$) which are comparatively sparser graph instances than the ones analysed in this project, so our XQAOA$_1^{\text{X=Y}}$ performance may be attributed to the dependence of relatively more variational parameters.

We present the dataset for fixed $m=8$ on a linear scale in Figure 4 for ease of visualisation (visualisation for the dataset for fixed $n=10$ is not as informative on a linear scale since the medians increase exponentially with the problem ratio). From the performance max-cut algorithms QAOA$_1$, MAQAOA$_1$, XQAOA$_1^{\text{X=Y}}$, and Goemans-Williamson on the dataset for fixed $m=8$, the problem ratio was not a good indicator of the "hardness" of a given partitioning problem. Indeed, this could be due to the nature of transforming an NPP into a complete graph and solving a max-cut problem - the hardness indicated by the problem ratio simply does not translate. Gurobi was benchmarked by minimising the cost function of NPP so it exhibits similar trends to the classical heuristics.




**(a)** *Dataset for problem ratios with fixed* $n = 10$.




**(b)** *Dataset for problem ratios with fixed* $m = 8$.

**FIG. 3:** *Box plot and line plot comparing average set-difference found by analytic forms of quantum algorithms for depth* $p = 1$ *and classical algorithms for problem ratios* $m/n$. *The algorithms were benchmarked against the set-difference found by the classical optimiser Gurobi. For each problem ratio, 25 random problem instances were generated. The data used to create the box plot comes from 20 random cuts generated using the Goemans-Williamson algorithm, and 20 random initialisations of variational parameters for QAOA and its variants for each problem instance. The crosses represent outliers. The line plot for Gurobi, greedy and Karmarkar-Karp represents the median cost found across all 25 problems for each problem ratio.*

### A. Ansatz Considerations

One of the biggest challenges in using VQA's such as QAOA is the issue of barren plateaus which refers to the phenomenon where the gradients of the objective function become exponentially vanshing with system size [21], making it difficult to find the optimal solution using gradient-based optimisation methods. Because the ansatz is problem-specific, one needs to explore an expo-

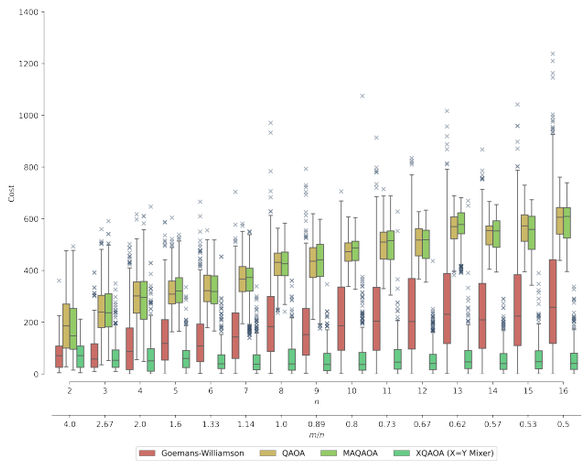



**FIG. 4:** *Box plots for problem instances with fixed $m = 8$ on a linear scale.*

nentially large space to find a solution, and so, with random initialisations of variational parameters, the probability of finding the solution is exponentially small. If the initial point is near a local optimum or resides on a barren plateau, gradient-based optimisers will converge on a local optima. We see that $\text{QAOA}_1$, $\text{MAQAOA}_1$, and $\text{XQAOA}_1^{\text{X=Y}}$ return a wide range of approximate solutions for the same problem, however, with the right choice of parameters, $\text{XQAOA}_1^{\text{X=Y}}$ can often achieve a perfect or near-perfect cut on par with the Gurobi optimiser.

We also note the large spread of distributions attained by the Goemans-Williamson algorithm. Although Goemans-Williamson achieves an approximation ratio of 0.878 in the worst case at the asymptotic limit [19], randomly vectors are generated to perform its hyperplane rounding. This means that problem instances can vary significantly, resulting in distributions that increase in spread as $n$ and $m$ grows. To find better approximations using Goemans-Williamson, the precision (i.e., repetitions for a problem instance) would need to increase exponentially with the size of the problem.

### B. Phase Transition

The *phase transition* for partitioning problems refers to the transition from problems with high multiplicity of optimal partitions to problems requiring an exhaustive traversal of all possible combinations to guarantee an exact result [18, 22]. From Equations 14 and 15 of [22], for partitioning problems of size $n = 10$, the phase transition is $\kappa = 0.88$. As mentioned earlier, the problem ratio $m/n$ was not a good indicator of hardness for the non-classical heuristics, however classical heuristics reflected this phase transition.

## VII. FUTURE DIRECTIONS

Potential directions for future work could include experimenting with other variants of XQAOA proposed by Vijendran et al. [11] as their performance on complete weighted graphs might fare better than the X=Y mixer. Furthermore, determination of the cost function landscape for max-cut on complete weighted graphs encoding a number partitioning problem would be insightful. This would involve testing different ansatz initialisation strategies and using gradient-free optimisation methods [23]. Extending the partitioning problem to multiway partitioning [17] can also be considered which will require formulating a new Ising model to encode the cost function. In addition, implementation on near-term quantum hardware [24, 25] can hopefully verify simulations performed in this study.

**Acknowledgements**

I would like to thank Syed Assad for supervising this project, as well as Aritra Das and V Vijendran for their support. I would also like to thank Jeffrey Liang for his continual positivity and resilience, which allowed us to complete the project in high spirits.

[1] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018. ISSN 2521-327X. doi: 10.22331/q-2018-08-06-79. URL https://doi.org/10.22331/q-2018-08-06-79.

[2] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.

[3] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[4] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[5] Michael Streif, Sheir Yarkoni, Andrea Skolik, Florian Neukart, and Martin Leib. Beating classical heuristics for the binary paint shop problem with the quantum approximate optimization algorithm. *Physical Review A*, 104(1):012403, 2021.

[6] Pontus Vikstål, Mattias Grönkvist, Marika Svensson, Martin Andersson, Göran Johansson, and Giulia Ferrini. Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Physical Review Applied*, 14(3):034009, 2020.

[7] Vivek Sarkar. *Partitioning and scheduling parallel programs for execution on multiprocessors*. Stanford University, 1987.

[8] Toby Walsh. Where are the really hard manipulation problems? the phase transition in manipulating the veto rule. *arXiv preprint arXiv:0905.3720*, 2009.

[9] Ralph Merkle and Martin Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE transactions on Information Theory*, 24(5):525–530, 1978.

[10] Rebekah Herrman, Phillip C Lotshaw, James Ostrowski, Travis S Humble, and George Siopsis. Multi-angle quantum approximate optimization algorithm. *Scientific Reports*, 12(1):6781, 2022.

[11] V Vijendran, Aritra Das, Dax Enshan Koh, Syed M Assad, and Ping Koy Lam. An expressive ansatz for low-depth quantum optimisation. *arXiv preprint arXiv:2302.04479*, 2023.

[12] Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738, 1995.

[13] Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

[14] Andrew Lucas. Ising formulations of many np problems. *Frontiers in physics*, 2:5, 2014.

[15] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220 (4598):671–680, 1983.

[16] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

[17] Richard Earl Korf. Multi-way number partitioning. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[18] Brian Hayes. Computing science: The easiest hard problem. *American Scientist*, 90(2):113–117, 2002.

[19] Michel X Goemans and David P Williamson. . 879-approximation algorithms for max cut and max 2sat. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 422–431, 1994.

[20] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com.

[21] 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):4812, 2018.

[22] Stephan Mertens. Phase transition in the number partitioning problem. *Physical Review Letters*, 81(20):4281, 1998.

[23] 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):1791, 2021.

[24] Andreas Bengtsson, Pontus Vikstål, Christopher Warren, Marika Svensson, Xiu Gu, Anton Frisk Kockum, Philip Krantz, Christian Krizan, Daryoush Shiri, Ida-Maria Svensson, et al. Quantum approximate optimization of the exact-cover problem on a superconducting quantum processor. *arXiv preprint arXiv:1912.10495*, 2019.

[25] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2):021067, 2020.