Novel Variational Quantum Algorithms for MaxCut

Afonso Sequeira Azenha
afonso.azenha@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2024

Abstract

In this work, we introduce the Interpolated QAOA/QEMC (iQAQE) Framework, a novel approach inspired by the Quantum Approximate Optimization Algorithm (QAOA) and the Qubit-Efficient MaxCut Heuristic Algorithm (QEMC) for designing Variational Quantum Algorithms (VQAs) to solve the Maximum Cut (MaxCut) problem. This Framework builds on the core components of QEMC while integrating concepts from QAOA to harness the strengths of both algorithms, requiring fewer qubits and demonstrating greater resilience to statistical uncertainty associated with small shot numbers compared to QEMC. This framework provides adjustable parameters to create various VQAs, and we introduce heuristics for selecting these parameters, including the number of qubits, list cardinality, and mappings. Our evaluations show that iQAQE often matches or exceeds the performance of QEMC and can outperform classical state-of-the-art algorithms like Goemans-Williamson in specific scenarios. We also propose two alternative approaches for solving the MaxCut problem derived from our iQAQE investigations, which offer additional insights and potential research directions. Additionally, we present a small machine learning model to determine optimal mappings for specific graphs based on statistical properties of the mappings themselves. Ultimately, the iQAQE Framework serves as a versatile testbed for developing new VQAs, potentially leading to significant advancements in the field.

Keywords: Hybrid Quantum-Classical Computing (HQCC), Variational Quantum Algorithms (VQAs), Maximum Cut (MaxCut) Problem, Quantum Approximate Optimization Algorithm (QAOA), Qubit-Efficient MaxCut Heuristic Algorithm (QEMC).

1. Introduction

In recent decades, quantum computing has made significant strides [1], leveraging quantum mechanics to potentially revolutionize fields like cryptography, optimization, and machine learning. Despite these advances, current Noisy Intermediate Scale Quantum (NISQ) devices have yet to achieve quantum advantage, where quantum computers outperform classical ones in solving specific problems efficiently.

Currently, the most promising approaches for achieving meaningful quantum advantage are found in ”Hybrid Quantum-Classical Computing” (HQCC), which merges the computational power of quantum devices with the reliability of classical systems. Variational Quantum Algorithms (VQAs) have emerged as a leading area of research within this framework, primarily due to their compatibility with near-term quantum systems. They demand fewer qubits and operate with shallower circuit depths, making them highly adaptable to current quantum technology. These algorithms are intentionally designed as hybrids, leveraging classical optimizers to fine-tune the parameters of Parameterized Quantum Circuits (PQCs).

Among the applications, combinatorial optimization has seen notable progress. The Quantum Approximate Optimization Algorithm (QAOA) [2] shows potential for solving the MaxCut problem, but requires more qubits than current devices can handle when scaled to larger problem instances. This limitation has prompted the development of alternative approaches such as the Qubit-Efficient MaxCut Heuristic Algorithm (QEMC) [3], which addresses MaxCut with fewer qubits. Continued research in hybrid quantum-classical methods is essential for achieving quantum advantage.

1.1. Motivation

The search for efficient algorithms to solve combinatorial optimization problems is critical in numerous fields, such as logistics, finance, and telecommunications. The MaxCut problem, for example, has broad applications in areas like machine learning [4], statistical physics [5], circuit design [5], and data clustering [6]. Creating more efficient algorithms to solve this problem can improve the performance of these applications, driving substantial progress in their respective fields.

Moreover, there is a strong interest from the computer science community in terms of computational complexity, particularly regarding the MaxCut problem's NP-hard nature. Efficient algorithms for this problem could shed light on the classical-to-quantum computing boundary and even the $P = NP$ problem. Imagining a world where $P = NP$ is proven true, though unlikely, is fascinating. It would imply polynomial-time solutions for all NP problems, including MaxCut, Traveling Salesman Problem (TSP), and Knapsack Problem, among others. This breakthrough would revolutionize optimization, benefiting areas like vehicle routing and job scheduling. Notably, RSA encryption could be compromised if $P = NP$, highlighting the critical importance of understanding these complexities.

The aforementioned considerations drive our efforts to develop a new algorithm for solving the MaxCut problem with greater efficiency and accuracy.

1.2. Topic Overview

In this project, we propose the Interpolated QAOA/QEMC Framework. This will enable us to develop novel VQAs by leveraging the strengths of two existing VQAs, QAOA and QEMC, for improved performance in solving the MaxCut problem. QAOA requires a qubit for each graph vertex, making it difficult to scale. In contrast, QEMC uses exponentially fewer qubits by assigning one basis state to each graph node, requiring only $\log_2(n)$ qubits (for n graph vertices). However, this compression leads to limitations in QEMC's results. By interpolating both VQAs, we aim to create an algorithm that utilizes fewer qubits than QAOA and performs better than QAOA and QEMC. The new algorithm, constructed through the iQAQE Framework, assigns multiple basis states to each node, in contrast to QEMC's single basis state approach. This design tentatively allows for a more practical implementation on present-day NISQ devices, thanks to its reduced qubit requirements compared to QAOA, fewer measurement shots than QEMC, and potentially greater trainability than QAOA.

2. Theoretical Background

Here, we'll introduce some key concepts essential for grasping the foundations of this work.

2.1. Hybrid Quantum-Classical Computing

Hybrid quantum-classical computing refers to a computational approach that combines elements of both classical and quantum computing paradigms to leverage the strengths of each. In this model, classical processors and quantum processors work in tandem to solve complex problems more efficiently than either could achieve alone. This collaborative strategy aims to harness quantum computing's unique capabilities while mitigating the challenges and limitations associated with quantum systems, such as error correction and decoherence. As of today, the synergy between classical and quantum elements holds promise for addressing complex real-world problems in areas like optimization, machine learning, and cryptography [7].

2.1.1. Variational Quantum Algorithms

The essence of HQCC lies in VQAs, which use parameterized quantum circuits (ansätze). These algorithms employ classical optimization to adjust the quantum circuit parameters, similar to how neural networks are trained to minimize a cost function. This approach leverages classical optimization tools and keeps quantum circuit depth shallow, reducing noise and making VQAs suitable for the current NISQ era. VQAs have diverse applications and are considered the most promising avenue for achieving near-term quantum advantage, despite challenges in trainability, accuracy, and efficiency [7].

The development of a VQA involves defining a cost function, C, representing the problem's solution. Then, an ansatz, a parameterized quantum circuit, is proposed, with parameters $\boldsymbol{\theta}$ to be optimized. The ansatz undergoes training in a hybrid quantum-classical loop to minimize $C(\boldsymbol{\theta})$ (Eq. 1). Quantum computation estimates $C(\boldsymbol{\theta})$ and its gradient, while classical routines optimize $\boldsymbol{\theta}$. Further details for each step of the VQA framework are provided.

$$\boldsymbol{\theta}^{\star} = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) \tag{1}$$

Cost function: Defining the VQA involves constructing a cost function that assigns numerical values to trainable parameters $\boldsymbol{\theta}$. This function shapes a cost landscape for optimization. The aim is to identify its global minimum, representing the solution (Eq. 1). Typically, the cost takes the form of Eq. 2, consisting of functions $\{f_k\}$, a parameterized unitary $U(\boldsymbol{\theta})$, input states ρ_k, and observables O_k. Oftentimes, it is convenient to design the cost function to be given by the expectation value of a Hamiltonian (e.g., the cost Hamiltonian in QAOA), which is why we mention that it is beneficial for it to have this specific form.

$$C(\boldsymbol{\theta}) = \sum_k f_k \left(Tr \left[O_k U(\boldsymbol{\theta}) \rho_k U^{\dagger}(\boldsymbol{\theta}) \right] \right) \tag{2}$$

Ansatz: The ansatz (parameterized quantum circuit, $U(\boldsymbol{\theta})$) plays a crucial role in determining the parameters $\boldsymbol{\theta}$ and their optimization to minimize the cost. Ansatz design can be problem-inspired or agnostic, with each approach offering distinct advantages. Problem-agnostic ansätze are versatile but may require more parameters, complicating optimization. In contrast, problem-oriented ansätze incorporate problem-specific information, potentially reducing the parameter space and improving interpretability. Choosing between these approaches involves balancing accuracy and generality, considering resource utilization. While problem-inspired ansätze often yield superior performance, their design presents challenges. Common ansatz types include hardware-efficient, unitary coupled clustered, and quantum alternating operator ansätze, among others.

Gradients and training: After defining the cost function and ansatz, the next step involves training the parameters $\boldsymbol{\theta}$ to optimize Eq. 1. Analytically computing the gradient of the cost function is a notable advantage of many VQAs, achieved using the parameter-shift rule (Eq. 4), where $\boldsymbol{\theta}_{\pm} = \boldsymbol{\theta} \pm \alpha \boldsymbol{e_l}$, holds for any real number α. Practically, $\alpha = \pi/4$ is commonly used for accuracy [7]. The VQA workflow resembles traditional machine learning, with input states processed by an ansatz parameterized by $\boldsymbol{\theta}$, followed by measurement and cost function estimation. A classical optimizer updates $\boldsymbol{\theta}$ iteratively until convergence to find the optimal parameters $\boldsymbol{\theta}^{\star}$. This high-level overview illustrates how VQAs function within a hybrid loop. Therefore, specifying the cost function and ansatz will be sufficient to characterize them.

2.2. MaxCut Problem

The MaxCut problem, a fundamental problem in graph theory and combinatorial optimization, involves partitioning a graph $G = (V, E)$ into two disjoint subsets, S_1 and S_2, such that the number of edges connecting vertices from different subsets is maximized. Formally, the objective is to find a partition (S_1, S_2) that maximizes:

$$\text{Cut}(S_1, S_2) = \sum_{(u,v)\in E} \chi(u, v), \tag{3}$$

where $\chi(u, v) = 1$ if u and v belong to different subsets, and $\chi(u, v) = 0$ if they belong to the same subset. This quantity is referred to as the "cut" of the partition (S_1, S_2). Pictorially, this can be represented as cutting the edges of the graph (Figure 1), hence the name MaxCut. What we describe here is the un-directed, un-weighted MaxCut problem. A more general formulation would involve the specific graph's adjacency matrix, W_{ij}.

Since the MaxCut problem is NP-hard, finding an optimal solution efficiently is a significant computational challenge, especially as the graph size grows. However, researchers have developed various approximation algorithms and heuristics to approach near-optimal solutions in a reasonable time, including both classical and quantum approaches.

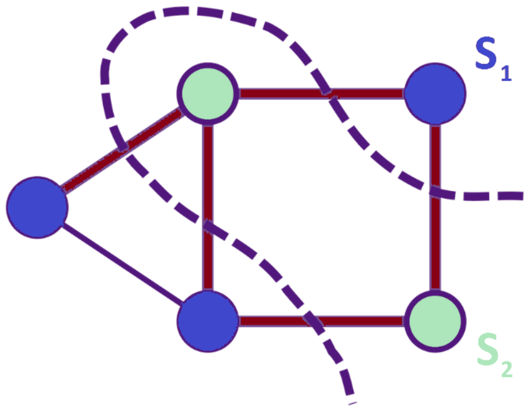

Figure 1: An example of a graph with a partition that maximizes the number of cut edges (red). Note that the MaxCut partition might not be unique.

2.3. State-of-the-Art Algorithms

Next, we introduce the state-of-the-art algorithms for tackling the MaxCut problem, encompassing both classical (Goemans-Williamson) and hybrid quantum-classical (QAOA and QEMC) approaches.

2.3.1. Goemans-Williamson Algorithm

Developed by Goemans and Williamson in 1995, this algorithm employs semidefinite programming to create an approximate solution, later refined for a near-optimal cut. Semidefinite programming optimizes a linear function over a symmetric matrix while ensuring positive semidefiniteness. It finds applications in control theory, nonlinear programming, geometry, and combinatorial optimization, as detailed in [9] and references therein.

The Goemans-Williamson algorithm addresses the MaxCut problem through a series of mathematical formulations and randomized procedures. Initially, an integer quadratic program (Q) is defined to maximize the "cut" in a graph. This is represented by binary variables y_i, where S_1 and S_2

$$\frac{\partial C}{\partial \theta_l} = \sum_k \frac{1}{2\sin\alpha} \left(Tr\left[O_k U(\boldsymbol{\theta}_+)\rho_k U^{\dagger}(\boldsymbol{\theta}_+)\right] - Tr\left[O_k U(\boldsymbol{\theta}_-)\rho_k U^{\dagger}(\boldsymbol{\theta}_-)\right]\right) \tag{4}$$

correspond to the vertices with values $y_i = 1$ and $y_i = -1$, respectively. However, since solving (Q) directly is NP-hard, the algorithm instead employs a semidefinite programming relaxation (P), which relaxes constraints to optimize vectors $v_i \in S_n$ rather than y_i. This relaxation is represented as:

$$(P) \qquad \begin{aligned} &\text{Maximize } \frac{1}{2} \sum_{\substack{i<j:\\(i,j)\in E}} (1 - v_i \cdot v_j) \\ &\text{Subject to: } v_i \in S_n \qquad \forall i \in V, \end{aligned} \tag{5}$$

where v_i are vectors constrained to the n-dimensional unit sphere (S_n). The algorithm then proceeds as follows:

1. Solve (P) to obtain optimal vectors v_i;
2. Select a random vector r uniformly distributed on S_n;
3. Partition vertices into S_1 and S_2 based on whether $v_i \cdot r \geq 0$ or $v_i \cdot r < 0$.

Effectively, the algorithm divides vertices based on a randomly chosen hyperplane in n dimensions. The performance guarantee of the Goemans-Williamson algorithm is quantified by the approximation ratio $\alpha > 0.878$ [9].

2.3.2. Quantum Approximate Optimization Algorithm (QAOA)

The QAOA, renowned for its prowess in quantum-enhanced optimization, was initially crafted to tackle a range of combinatorial optimization problems such as constraint-satisfaction [10] and Max-Cut [11]. QAOA utilizes a cost Hamiltonian, denoted as H_C, which serves as the basis for extracting the objective function. This Hamiltonian is formed by associating each conventional classical variable y_i with a Pauli spin $1/2$ operator, $\boldsymbol{Z}_i$. Formally, the usual MaxCut objective $L(y) = \frac{1}{2}\sum_{\substack{i<j:\\(i,j)\in E}} (1 - y_i y_j)$, denoting the cut of partition $y = (y_1, ..., y_N)$, is transformed into the cost Hamiltonian

$$H_C = \frac{1}{2} \sum_{\substack{i<j:\\(i,j)\in E}} (1 - \boldsymbol{Z}_i \boldsymbol{Z}_j). \tag{6}$$

The Trotter-inspired QAOA ansatz involves n cycles of alternating time evolution between H_C and a mixer Hamiltonian, $H_M = \sum_{j=1}^{N} \boldsymbol{X_j}$, for N qubits and graph nodes. The role of this mixer Hamiltonian can be understood as introducing quantum fluctuations, or transitions, between different states, helping the algorithm explore the solution space more effectively, ideally preventing it from getting trapped in sub-optimal local minima. This ansatz, depicted in Figure 2, illustrates n QAOA layers, each comprising alternating applications of H_C and H_M.

The cost function $C(\boldsymbol{\gamma}, \boldsymbol{\alpha})$ is determined by the expectation value of H_C over the ansatz state $|\psi_n(\boldsymbol{\gamma}, \boldsymbol{\alpha})\rangle$, obtained at the end of the quantum circuit (Figure 2). In practice, defining $\boldsymbol{\theta} = \{\boldsymbol{\gamma}, \boldsymbol{\alpha}\}$, the cost function is $C(\boldsymbol{\gamma}, \boldsymbol{\alpha}) = \langle\psi_n(\boldsymbol{\gamma}, \boldsymbol{\alpha})| H_C |\psi_n(\boldsymbol{\gamma}, \boldsymbol{\alpha})\rangle$, with

$$|\psi_n(\boldsymbol{\gamma}, \boldsymbol{\alpha})\rangle = e^{-i\alpha_n H_M} e^{-i\gamma_n H_C} ... e^{-i\alpha_1 H_M} e^{-i\gamma_1 H_C} |\psi_0\rangle\,, \tag{7}$$

where $|\psi_0\rangle$ is the initial state entering the ansatz. In QAOA, it is customary to start with a uniform superposition over the N bit-string[1] basis states, i.e., $|\psi_0\rangle = |+\rangle^{\otimes N} = \frac{1}{\sqrt{2^N}} \sum_{z\in\{0,1\}^N} |z\rangle$. This is achieved by applying a Hadamard gate to each of the qubits, at the start of the quantum circuit.

In practice, the mixer Hamiltonian terms $e^{-i\alpha_l H_M}$ correspond to $R_x(2\alpha_l)$ and are easy to implement. The cost Hamiltonian terms, on the other hand, are a bit more tricky, requiring the use of 2 CNOT gates. Each of the terms $e^{-i\gamma_l(1-\boldsymbol{Z}_j\boldsymbol{Z}_k)/2}$ can be transpiled into a quantum circuit, as shown in Figure 3. For each edge in the graph, we'll have one of these terms, in each of the n layers of the QAOA ansatz.

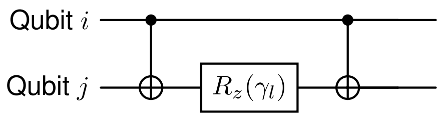

Figure 3: $e^{-i\gamma_l \boldsymbol{Z}_i \boldsymbol{Z}_j/2}$ decomposition.

Subsequently, a classical routine is employed to optimize the values of the parameters by minimizing the negative of the cost (analogous to the "cut"), therefore enabling the extraction of the Max-Cut partition.

[1]Keep in mind, there's one qubit allocated for each graph node.

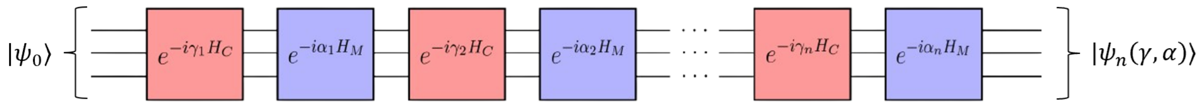

Figure 2: Quantum alternating operator ansatz. Adapted from [8].

2.3.3. Qubit-Efficient MaxCut Heuristic Algorithm (QEMC)

QEMC [3] is somewhat similar to QAOA. However, it has a number of crucial differences. First, it only requires $n = \log_2(N)$ qubits, instead of N, where N is the number of nodes in the graph. Additionally, the QEMC algorithm is based on a novel probability threshold encoding scheme, a suitable cost function, and a parameterized unconstrained quantum circuit. Going in order:

Probability threshold encoding scheme: With n qubits, each of the $N = 2^n$ basis states will represent one of the graph's nodes. Following the sampling of the quantum circuit, a probability distribution is generated. Nodes with probabilities exceeding a certain threshold, $p_{th} = \frac{1}{2B}$, belong to "Set 1", while probabilities below this value indicate inclusion in "Set 0". This strongly diverges from how we encode set inclusion in QAOA.

Cost function: The objective function utilized in QEMC is the following:

$$L(\{p(i)\}) = \sum_{\substack{j<k: \\ (j,k)\in E}} \left[\left(d(j,k) - \frac{1}{B} \right)^2 + \left(s(j,k) - \frac{1}{B} \right)^2 \right], \quad (8)$$

where $d(j,k) = |p(j) - p(k)|$ and $s(j,k) = p(j) + p(k)$ are the absolute difference and sum of the corresponding states' probabilities. The idea is that as both $d(j,k)$ and $s(j,k)$ tend towards $1/B$, the probability of one node approaches zero (distinctive "Set 0"), while the probability of the other node approaches $1/B$ (distinctive "Set 1"), without specifying which is which. Ultimately, just like for QAOA, connections between nodes of different sets are favoured. Note, however, that this probability threshold encoding scheme assumes *a priori* that one of the sets ("Set 1") has B nodes. Nevertheless, this is not an issue, as we can efficiently iterate through all potential values of $B = 1, ..., \lfloor \frac{N}{2} \rfloor$. Frequently, it is reasonable to set $B = N/2$, and we shall use this as our default starting point.

Problem-agnostic ansatz: The QEMC circuit ansatz is agnostic to specific graph instances, a departure from QAOA where the graph structure is explicitly encoded in the quantum circuit. Instead, the graph is implicitly encoded through the cost function. As a result, the QEMC quantum circuit is not bound to any particular form and only needs to be expressive enough to approximate the optimal states in the Hilbert space. Such problem-independent ansatz approach provides considerable flexibility in ansatz selection. Frequently, the circuit ansatz known as "Strongly Entangling Layers" is employed, as depicted in Figure 4. As can be seen, this ansatz applies a series of parameterized single-qubit rotations interspersed with entangling gates to generate a highly entangled quantum state.

One notable, and perhaps unfortunate, property of QEMC is that it is efficiently simulable classically. Due to the exponential compression of the number of qubits, the algorithm can be feasibly simulated on a classical computer, even for large graphs, hence defeating its purpose as a quantum algorithm. This is something the authors of the algorithm [3] realized in hindsight. For this reason, it is now termed a quantum-inspired classical algorithm.

3. iQAQE Framework

We introduce a new framework, inspired by QAOA and QEMC, for seamlessly designing multiple distinct VQAs. This framework builds on QEMC's core components while integrating concepts from QAOA to leverage the strengths of both algorithms for improved performance. Tentatively named the iQAQE Framework, it opens the door to numerous unexplored and unknown VQAs, offering a variety of parameters to experiment with, such as the number of qubits, list cardinality, and mappings.

In iQAQE, we depart from the QEMC approach by associating each graph node with a list of basis states, in contrast to QEMC's consideration of a single basis state for each node. Each of these lists comprises $c \in [1, 2^{n-1}]$ basis states, where

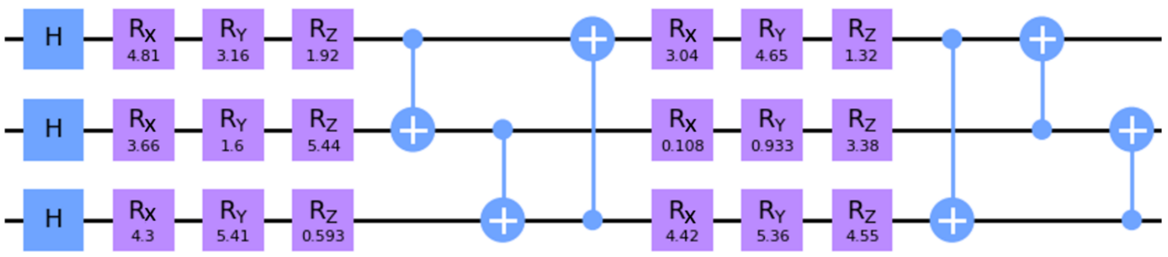

Figure 4: "Strongly Entangling Layers" circuit ansatz: case of $n = 3$ qubits and $p = 2$ layers. After a single layer of Hadamard gates, each subsequent layer consists of $3n$ single-qubit parameterized rotation gates and n CNOT gates (entangling gates). Reproduced from [3].

n represents the number of qubits. c denotes the cardinality of the lists. What we formerly referred to as "mapping" involves distributing basis states among these lists. The term "mapping" can also signify a specific allocation of basis states among the lists. This design allows for potential overlap among states from different lists. Additionally, it is important to note that the encoding of these states will utilize a qubit range expected to fall between the QEMC and QAOA requirements, specifically in $[\log_2(N), N]$, for an N-node graph.

As a default approach, we compute node probabilities straightforwardly by summing the probabilities of associated basis states and then normalizing the result. This is necessary because we utilize QEMC's probability threshold encoding scheme. Additionally, we consider iQAQE to use the same ansatz and cost function as QEMC, adjusted for the appropriate number of qubits. However, we also explore the potential for slight deviations from this formula, such as alternative methods for computing node probabilities and the adoption of problem-inspired ansatz variations.

We can postulate that such a hybrid approach might have some potential advantages. Namely, it should allow for less shots than in QEMC and fewer qubit requirements than in QAOA, in addition to, arguably, being better trainable [3]. Another notable feature of this algorithm is its tunable "quantumness." By interpolating between QAOA (a hybrid quantum-classical method) and QEMC (a classical, quantum-inspired method), we can selectively adjust the degree of quantum and classical components, which could prove to be advantageous.

We designate this as a framework due to the vast array of possibilities in qubit number, list cardinality, and mappings, which facilitate the creation of multiple unique VQAs, each with its own merits and drawbacks. In this study, we will present heuristics for parameter selection and provide a comprehensive analysis of the corresponding results.

4. Implementation

In our numerical implementation, we leveraged PennyLane [12], a Python library tailored for differentiable programming of quantum computers. Furthermore, NetworkX [13] was instrumental in generating and manipulating graphs for addressing the MaxCut problem, while CVXPY [14], tailored for convex optimization, assisted in solving the semidefinite programming relaxation of MaxCut within the Goemans–Williamson algorithm framework. Testing on actual quantum hardware was unavailable, limiting our evaluation to simulations on a personal computer. This constraint also impacted scalability and comprehensive testing, particularly for larger graphs. Simulations were conducted using the Adam optimizer [15], with learning rates fine-tuned as hyperparameters.

Moreover, it's important to elucidate the methodologies employed for benchmarking and testing the algorithms' efficacy. Central to our evaluation are the cost and approximation ratio[2] plots, essential for comparing convergence rates and final cut values across the algorithms. Additionally, recognizing the influence of initial parameters on outcomes, we adopt statistically robust metrics such as average cut values and the best-so-far (BSF) cut value[3]. Furthermore, to mitigate the impact of outliers, we incorporate the median BSF cut value in our analysis, computed from multiple training curves. Frequently, we use the average cut or approximation ratio curves generated from 10 different random initializations. This latter approach enables us to evaluate the algorithm's performance on average post-training. Essentially, we assess the likelihood of obtaining favorable results following a single run after training.

In our pursuit of optimization, grid searches[4] are conducted on hyperparameters like Adam's learning rate, the number of ansatz layers (`n_layers`), and qubits (`n_qubits`), albeit constrained by the unavailability of an HPC cluster. We also refine our evaluations by benchmarking against the Goemans-Williamson algorithm, offering insights into the relative performance of our algorithms compared to state-of-the-art solutions.

5. iQAQE Schemes and Results

Now, we present the various iQAQE schemes we have developed and the results of their numerical simulations. For a more detailed discussion, please refer to the full thesis (Chapter 5). Keep in mind that all numerical results refer to the same 8-node graph: `E=[(0,1),(0,2),(0,6),(1,2),(1,6),(3,2),(3,4),(3,5),(4,5),(4,7),(5,7),(6,7)]`.

5.1. Random iQAQE

First and foremost, we aim to understand how iQAQE compares with both QAOA and QEMC. For this comparison, n and c were randomly chosen ($n = c = 4$). Currently, the allocation of basis states to lists is also done randomly. The goal of this initial simulation is to take a preliminary look at iQAQE's performance and understand how much the results vary between finite shot number simulations and analytical ones (Figure 5). These plots generally align with our theoretical expectations. As QEMC

[2]The approximation ratio denotes the proportion of the achieved "cut" value to the MaxCut value.

[3]If a dip occurs in the graph, it is adjusted to match the previous graph value, ensuring that the final result is monotonically increasing. That's what the BSF transformation does.

[4]These are omitted here. Please refer to the full thesis.

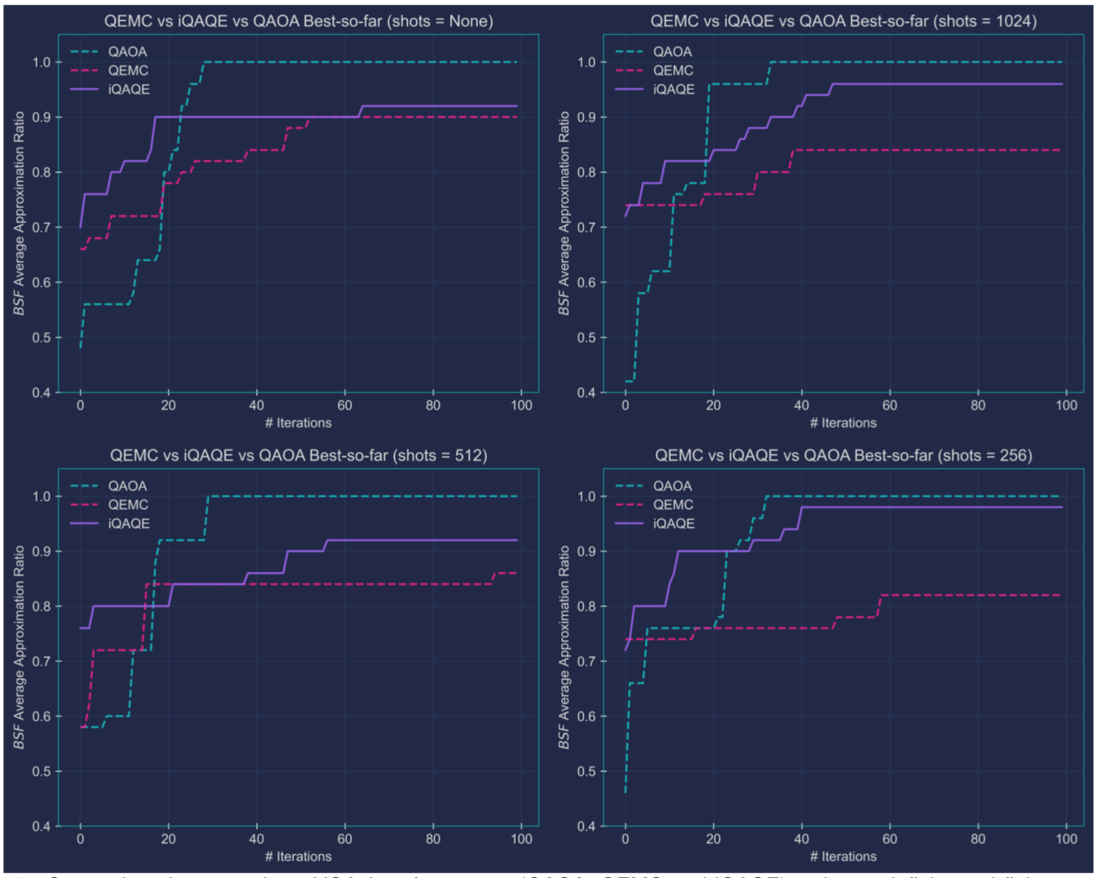

Figure 5: Comparison between the 3 VQAs' performances (QAOA, QEMC and iQAQE), using an infinite and finite number of shots. The best-so-far average is plotted (8-node graph).

relies heavily on a large number of shots, transitioning to a finite shot number naturally impacts the results. Surprisingly, however, iQAQE seems to exhibit slightly greater resilience to this effect, despite sharing QEMC's cost function and ansatz. This resilience might be attributed to the presence of multiple basis states associated with each graph node, potentially reducing the need for exhaustive sampling.

5.2. Polynomial Compression-type Encodings

These schemes fix one or more qubits to 1 (or 0) while letting the others vary[5]. Fixing k qubits achieves polynomial compression of order k. The total number of nodes encoded is $\binom{n}{k}$, the number of ways to select k qubits from n. We select the smallest n such that $\binom{n}{k} \geq N$, resulting in $N = \mathcal{O}(n^k)$. We then decide how many basis states to use for each node, chosen from 2^{n-k}. We may use all 2^{n-k} states or choose a subset $c \leq 2^{n-k}$. Upon selecting the mapping, the algorithm uses QEMC's cost function and ansatz. Polynomial compression ensures $n < N$ unless $k = 1$.

[5]Fixing qubits means we only permit basis states where certain qubits are set to either 0 or 1. Each node has specific qubits fixed, and these fixed qubits differ between nodes, forming the basis for our mappings.

5.2.1. Basic Polynomial Compression-type iQAQE

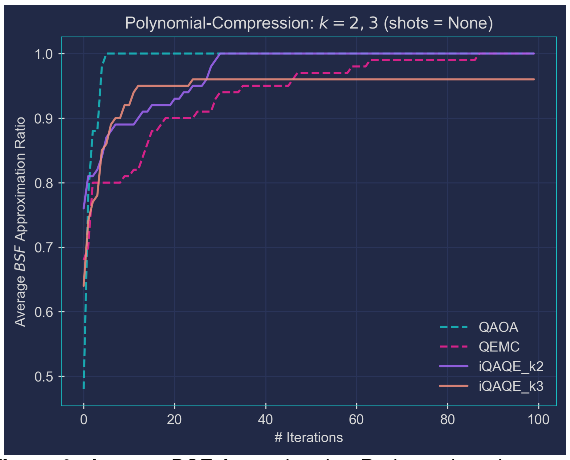

Figure 6: Average BSF Approximation Ratio *vs.* iteration number for the tested VQAs. The number of layers considered were 4 for $k = 2$ and 5 for $k = 3$ (8-node graph).

This is the simplest type of polynomial compression-based iQAQE scheme. It involves fixing k qubits to 1 instead of 0 (e.g., $\{|11\mathrm{xxx}\rangle\}$, for $k = 2$). The number of fixed qubits (k) is determined by the desired order of compression, as previously described. The results for $k = 2$ and $k = 3$ are shown in Figure 6. Results-wise, `iQAQE_k2` achieves a perfect average BSF approximation ratio, although it requires more iterations than QAOA. Our method outperforms QEMC for $k = 2$, but yields less promising results for $k = 3$. Nevertheless, both mappings use fewer qubits ($n = 5$) compared to QAOA ($n = 8$), which is advantageous for implementation on practical (NISQ) quantum computers.

5.2.2. Correlation-based iQAQE

This scheme closely resembles what was previously discussed in subsection 5.2.1. The key distinction lies in the inclusion of the possibility for both 0's and 1's to be fixed (e.g., $\{|11\text{xxx}\rangle, |00\text{xxx}\rangle\}$, for $k = 2$; recall, this encodes a single node). Consequently, the scheme aims to identify correlations among the different qubits, wherein "correlations" denote identical colors. This approach was inspired by the notion that by identifying correlations between nodes, it becomes feasible to color the entire graph as long as one node is initially colored. We present the results of numerical simulations applied to the standard 8-node graph using this correlation-based iQAQE approach (Figure 7).

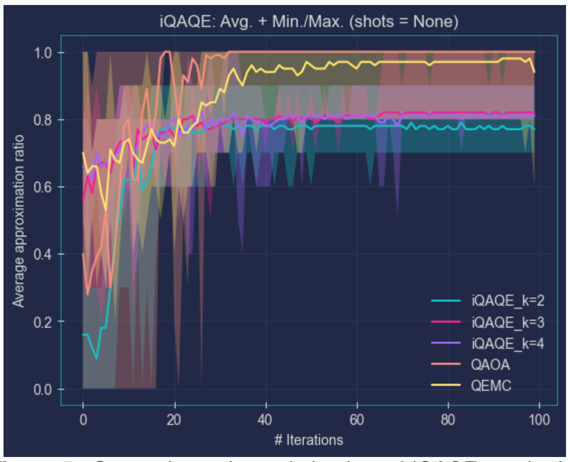

Figure 7: Comparison of correlation-based iQAQE results for $k = 2, 3$, and 4 with outcomes from QAOA and QEMC (8-node graph).

The underperformance may stem from doubling the number of basis states in each sub-list, resulting in a 50% overlap between pairs of sub-lists, akin to Basic Polynomial Compression-type iQAQE. However, this time, with twice the basis states, the (absolute) overlap is significantly higher, leading to more shared states among nodes. When overlap is excessive, adjusting one node's color without affecting others becomes more challenging, likely causing the underperformance.

5.2.3. Fixed-Parity iQAQE

Now, we present another heuristic method for mapping the basis states to the nodes. Although it is similar to the previous one, this time we fixed the parity of the selected k qubits to be even. Parity is determined by the number of 1's: if the count is even, the parity is even; otherwise, it is odd. For instance, for $k = 3$ (with `n_qubits = 5`), the lists would take the form: (Keep in mind that we are using an 8-node graph.)

1. $\{|000\text{xx}\rangle, |011\text{xx}\rangle, |101\text{xx}\rangle, |110\text{xx}\rangle\}$;
2. $\{|00\text{xx}0\rangle, |01\text{xx}1\rangle, |10\text{xx}1\rangle, |11\text{xx}0\rangle\}$;
3. $\{|0\text{xx}00\rangle, |0\text{xx}11\rangle, |1\text{xx}01\rangle, |1\text{xx}10\rangle\}$;
4. $\{|\text{xx}000\rangle, |\text{xx}011\rangle, |\text{xx}101\rangle, |\text{xx}110\rangle\}$, etc.

We have presented only the first four nodes. Numerical simulations were performed, and the obtained results are presented in Figure 8. While performance still lags behind QAOA, it is notably better than the previous scenario. For $k = 2$, the results match the previous correlation-based scheme, as requiring pairs to be even is equivalent to requiring them to be the same.

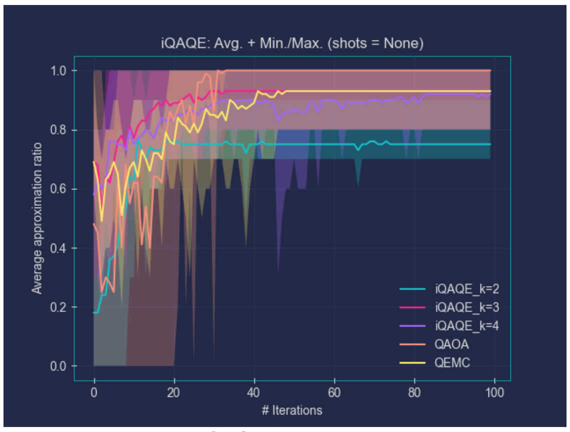

Figure 8: Fixed-parity iQAQE for $k = 2, 3$, and 4 compared with the results from QAOA and QEMC (8-node graph).

5.3. Unmodified Extended-QEMC

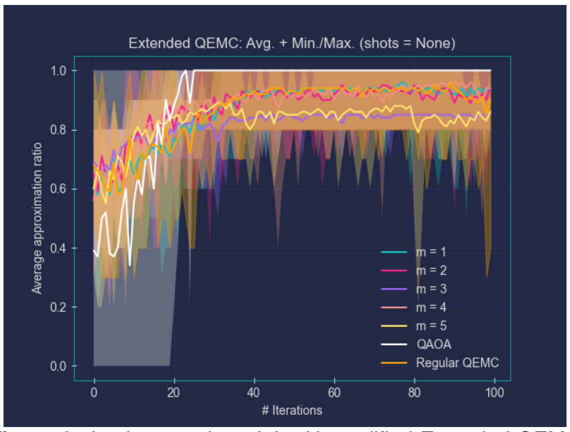

Figure 9: Implementation of the Unmodified Extended-QEMC scheme for the usual 8-node graph: comparison with QAOA and regular QEMC for various values of m. Note that m is not extended beyond 5, as this would exceed the number of qubits used in QAOA. Additionally, we utilize `n_layers = 3`, `step_size = 0.95` and `B = 4`.

Amid the various schemes we've attempted, I've devised an extension to the standard QEMC scheme. This new approach enhances QEMC by incorporating m additional qubits beyond the usual $\lceil \log_2(n) \rceil$ qubits required for n graph nodes. This increase in qubits prevents the scheme from being easily classically simulable and allows for more basis states to be associated with each graph node. This scheme assigns 2^m basis states per node. Importantly, similar to QEMC, the sets of basis states associated with different nodes do not overlap. The assignment of states to each list is currently arbitrary. For simplicity, we use a straightforward partition: the first 2^m basis states go to the first list, the next 2^m to the second list, and so on. Now, I present the results of applying this scheme to the usual 8-node graph (Figure 9).

5.4. Alternative ansätze

After considering whether QEMC and iQAQE could be affected by a generic ansatz, I pondered if tailored ansätze and excess entanglement might impact performance. To address this, I proposed implementing non-deterministic CNOT gates (ND-CNOTs[6]), adjusting entanglement dynamically. We achieved this by adding parameterized R_x gates before each CNOT's control qubit. Following numerical testing, we discovered that despite increasing complexity in the optimization landscape, this seems to substantially improve performance, especially when coupled to QEMC.

5.5. Goemans-Williamson and Bigger Graphs

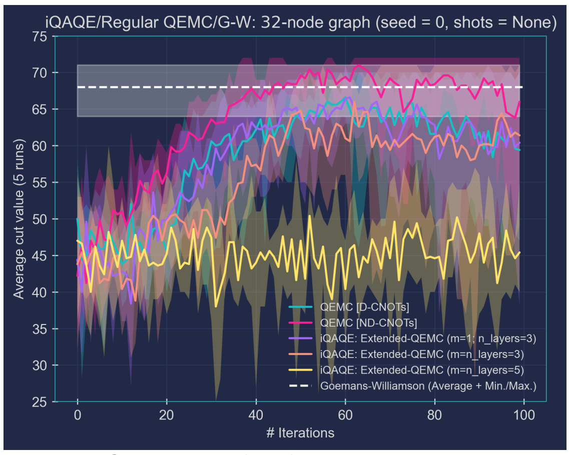

Figure 10: Comparison of performance (average cut values) among various VQAs for a 32-node graph.

To assess the scalability of these algorithms, we conducted tests on larger graphs, facilitating more accurate comparisons with classical state-of-the-art methods (GW). The results (32-node graph) for the most promising algorithms are depicted in Figure 10. Remarkably, the average performance of ND-CNOT-based QEMC competes closely with GW, occasionally surpassing it. Moreover, the maximum performance of the same ND-CNOT QEMC exceeds that of GW, as seen in Figure 10 (shaded regions). This marks the first instance in our study where one of our heuristics outperforms classical state-of-the-art algorithms, highlighting significant progress and indicating the potential for further exploration within the proposed iQAQE Framework. Other heuristics were explored in this project but are not included here due to space limitations. We have presented the most significant results, though all findings are important and should not be overlooked. For a thorough discussion, please consult the full thesis (Chapter 5).

5.6. Randomized iQAQE benchmarking – Machine learning-based approach

We also proposed a scheme for determining the optimal mapping for a specific graph based on statistical properties of the mappings themselves, using a small machine learning model. Such a model necessitates defining input (independent variables - IVs) and output (dependent variable - DV). Our IVs will encompass statistically significant properties[7] extracted from each node's specific sub-list. Alternatively, including the basis states' mapping directly as the model's input presents challenges concerning memory-efficient encoding. Traditional one-hot encoding methods are inadequate due to the vast number of potential basis states available (which scales exponentially with the number of qubits). Additionally, the median BSF cut value after training was selected as the DV. Initially, we utilized a multiple linear regression model. Upon observing significant multicollinearity among some IVs, we transitioned to principal component analysis (PCA), followed by regression (PCR). Although the results were not entirely satisfactory, we believe this approach can inspire future research, building on this idea and our initial work. For a more detailed discussion, please refer to the full thesis (Chapter 5).

6. Alternative Schemes

We also developed alternative schemes that, although not entirely based on the iQAQE Framework, emerged from our research on these topics. These are different algorithms proposed to solve the MaxCut problem. Two such alternatives are the Parity-like QAOA and the Batch-based Oracle coloring scheme. The former involves "reducing" QAOA's problem Hamiltonian to use fewer qubits than usual, by using parity encodings, which al-

[6]We recognize that the term "non-deterministic CNOT gates" is employed in linear optical quantum computing (KLM protocol), yet we use it differently in this context.

[7]Examples: number of basis states, average Hamming weight, average pair-wise Hamming distance, etc.

low us to compress the representation of N nodes into $n < N$ qubits (with $N = \mathcal{O}(n^k)$, for any k, as long as $\binom{n}{k} \geq N$). The latter could be implemented with a trainable Oracle, which would receive $k < n$ qubits, and return the colors of k nodes. Afterwards, we would run $\frac{n}{k}$ batches of this, to obtain all the n nodes' colors. Due to space constraints, we won't discuss these schemes in detail in this extended abstract; we simply present them. For a detailed discussion, please see Chapter 6 of the thesis.

7. Conclusions

The primary accomplishment of this study is the development of the iQAQE Framework, offering promising competition with current state-of-the-art algorithms. We've showcased its advantages over QAOA and QEMC, particularly in terms of reduced qubit requirements and enhanced resilience to statistical uncertainty. Our introduced heuristics have generally shown competitive results, even surpassing classical state-of-the-art (GW) in certain scenarios, like ND-CNOT-based QEMC for a 32-node graph. This success underscores the potential of the iQAQE Framework. Moreover, we proposed a method for determining optimal mappings based on statistical properties, utilizing a small machine learning model. While initial results were not entirely satisfactory, we believe this approach can inspire further research.

Future work could focus on enhancing the machine learning model, developing new heuristics, and testing on larger graphs. Exploring neural networks, alternative input variables, different heuristics configurations, and problem-inspired ansätze could yield valuable insights. Additionally, testing the framework on real-world quantum computers is essential for assessing its practical viability. Further refinement and testing of proposed heuristics will also be crucial for continued improvement. Another research direction is to continue developing the alternative schemes presented in section 6, which could provide valuable insights and potential avenues for future study. By addressing these areas, future research can build upon the foundation laid by this work, potentially leading to significant advancements in VQAs.

Acknowledgments

This work was developed in collaboration with Zoltán Zimborás and Bence Bakó, from the Wigner Research Centre for Physics (Hungary), in the context of project: *HQCC – Hybrid Quantum-Classical Computing*, supported by the EU QuantERA ERA-NET Co-fund in Quantum Technologies and by FCT – Fundação para a Ciência e a Tecnologia (QuantERA/004/2021).

References

[1] John Preskill. Quantum computing 40 years later, 2023, 2106.10522.

[2] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014, 1411.4028.

[3] Yovav Tene-Cohen, Tomer Kelman, Ohad Lev, and Adi Makmal. A variational qubit-efficient maxcut heuristic algorithm, 2023, 2308.10383.

[4] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112 vol.1, 2001.

[5] Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

[6] Jan Poland and Thomas Zeugmann. Clustering pairwise distances with missing data: Maximum cuts versus normalized cuts. In Ljupčo Todorovski, Nada Lavrač, and Klaus P. Jantke, editors, *Discovery Science*, pages 197–208, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

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

[8] Intro to QAOA. `https://pennylane.ai/qml/demos/tutorial_qaoa_intro/`, 2024. Accessed: 2023-12-18.

[9] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995.

[10] Cedric Yen-Yu Lin and Yechao Zhu. Performance of qaoa on typical instances of constraint satisfaction problems with bounded degree, 2016, 1601.01744.

[11] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel. Quantum approximate optimization algorithm for maxcut: A fermionic view. *Phys. Rev. A*, 97:022304, Feb 2018.

[12] PennyLane Website. `https://pennylane.ai/`, 2024. Accessed: 2024-05-13.

[13] NetworkX Website. `https://networkx.org/`, 2024. Accessed: 2024-05-13.

[14] CVXPY Website. `https://www.cvxpy.org/`, 2024. Accessed: 2024-05-13.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017, 1412.6980.

[16] Zsolt Tabi, Kareem H. El-Safty, Zsofia Kallus, Peter Haga, Tamas Kozsik, Adam Glos, and Zoltan Zimboras. Quantum optimization for the graph coloring problem with space-efficient embedding. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, October 2020.

[17] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[18] Marco Sciorilli, Lucas Borges, Taylor L. Patti, Diego García-Martín, Giancarlo Camilo, Anima Anandkumar, and Leandro Aolita. Towards large-scale quantum optimization solvers with few qubits, 2024, 2401.09421.

[19] Richard M. Karp. *Reducibility Among Combinatorial Problems*, pages 219–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[20] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2, 2014.