Instance Space Analysis of Quantum Algorithms

Supervisors: Prof. Kate Smith-Miles, Prof. Lloyd Hollenberg

Vivek Katial

2024-11-26

The University of Melbourne

Agenda

Agenda

1. Introduction
2. Background
3. Quantum Approximate Optimization Algorithm (QAOA)
4. Instance Space Analysis
5. Various Instance Spaces
6. Software for QAOA Parameter Initialisation
7. Conclusion and Future Work

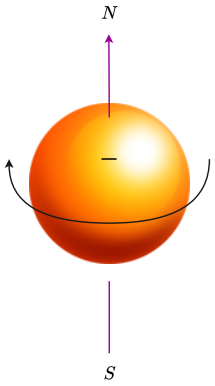
Introduction

Introduction

Why Quantum?

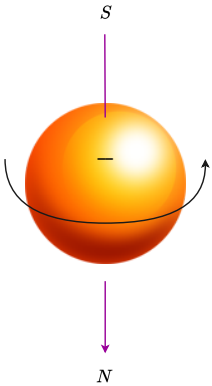
1. Quantum Computers can *theoretically solve* can solve some problems much faster than classical computers
2. What problems?
   - **Shor's algorithm** for factoring large numbers - could break RSA encryption [12]
   - **Grovers Search** - Quadratic speedup over classical search [6]
   - **Simulation of physical systems** - Quantum Chemistry, Material Science
3. What's the catch?
   - Hardware is **hard** - assuming no errors we need several 1000s of qubits
   - With current error rates - need millions of qubits + 100s of millions of gates
   - **NISQ** - Noisy Intermediate-Scale Quantum (NISQ) devices - 50-100 qubits, noisy, error-prone

How does Quantum Computing work?



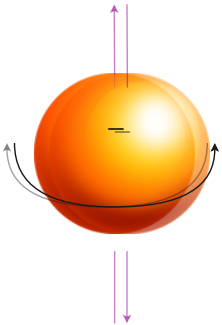
Spin Up

$|\psi\rangle = |0\rangle$



Spin Down

$|\psi\rangle = |0\rangle$



Superposition

$|\psi\rangle = a|0\rangle + b|1\rangle$

Measurement

Wave Function Collapse

When a quantum system is measured, its wavefunction collapses into a single state, and the system loses its superposition.

Probabilistic Outcome

The outcome of a measurement is probabilistic, with the probability determined by the quantum state before the measurement.

Disturbance

The act of measuring a quantum system inevitably disturbs the system, making it impossible to precisely determine the pre-measurement state.

Noisy-intermediate Scale Quantum Era

- Currently we're in the NISQ-era of Quantum Computing
- Need to design algorithms that can run on NISQ-devices

1. Can run on small (100-1000 qubit devices)
2. Solve useful problems
3. Shouldn't require extensive error correction

QAOA is a low-depth algorithm that can help solve optimisation problems

Aims of the Thesis

1. We focus on the addressing the challenges in **Variational Quantum Algorithms (VQAs)** [8]; a class of quantum algorithms that are expected to be run on NISQ devices.
2. Among prominent VQAs, **Quantum Approximate Optimization Algorithm (QAOA)** [4] and **Variational Quantum Eigensolver (VQE)** are widely studied.
3. The main area of focus in this thesis is to study the instance dependence of QAOAs to better understand and stress test its performance.

MaxCut Problem

Partition a graph $G = (V, E)$ into two sets S and $V \setminus S$ such that the number of edges between the two sets is maximised.

$$\max_{\mathbf{s}} \sum_{(i,j)\in E} w_{ij}(1 - z_i z_j)$$

where $z_i \in \{0, 1\}$ and $w_{ij} \in \mathbb{R}$ is the weight of edge (i, j).

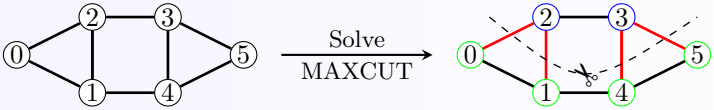


Figure 1: An example of a six-node MaxCut problem

Solution is a binary string $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ where $s_i \in \{0, 1\}$ and the optimal objective value is $C_{\max}$ where $C_{\max} = \sum_{(i,j)\in E} w_{ij}$ for the edges in our **maximum cut**.

Solving MaxCut on a Quantum Computer

We map the MaxCut problem to a *Hamiltonian* for quantum optimization.

Classical Formulation

Objective:

$$\max_{\mathbf{s}} \sum_{(i,j)\in E} w_{ij}(1 - z_i z_j)$$

State Space:

$$\mathbf{s} \in \{0,1\}^n$$

Solution:

$$\mathbf{s}^* = \operatorname*{argmax}_{\mathbf{s}} \sum_{(i,j)\in E} w_{ij}(1 - z_i z_j)$$

map

Quantum Formulation

Objective:

$$H = \sum_{(i,j)\in E} w_{ij}(I - Z_i Z_j)$$

State Space:

$$|\psi\rangle \in \mathcal{H}_2^{\otimes n}$$

Solution:

$$|\psi_{\text{ground}}\rangle = \operatorname*{argmin}_{|\psi\rangle} \langle\psi|H|\psi\rangle$$

Quantum Approximate Optimization Algorithm (QAOA)

Quantum Approximate Optimization Algorithm (QAOA)

- QAOA prepares a parameterised "trial" (ansatz) state of the form:

$$
\begin{aligned}
|\psi(\theta)\rangle &= |\psi(\vec{\gamma}, \vec{\beta})\rangle \\
&= \prod_{j=1}^{p} e^{-i\beta_j \hat{H}_B} e^{-i\gamma_j \hat{H}_P} |+\rangle^{\otimes n}
\end{aligned}
$$

- Where $\hat{H}_P = \sum_{(i,j)\in E} w_{ij}(1 - \hat{Z}_i\hat{Z}_j)$ is the problem Hamiltonian and $\hat{H}_B = \sum_{i=1}^{n} \hat{X}_i$ is the mixing Hamiltonian. - The parameters $\vec{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\vec{\beta} = (\beta_1, \ldots, \beta_p)$ are optimised to minimise the expectation value of the problem Hamiltonian.

Quantum Approximate Optimization Algorithm (QAOA)

- The QAOA Ansatz Energy is given by taking the expectation value of the problem Hamiltonian with respect to the trial state:

$$
\begin{aligned}
F_p(\vec{\gamma},\vec{\beta}) &= \langle +|^{\otimes n} \left(\prod_{j=1}^{p} e^{-i\beta_j \hat{H}_B} e^{-i\gamma_j \hat{H}_P}\right)^{\dagger} \hat{H}_P \left(\prod_{j=1}^{p} e^{-i\beta_j \hat{H}_B} e^{-i\gamma_j \hat{H}_P}\right) |+\rangle^{\otimes n} \\
&= \langle \psi(\vec{\gamma},\vec{\beta})|\hat{H}_P|\psi(\vec{\gamma},\vec{\beta})\rangle
\end{aligned}
$$

- The goal is to find the optimal parameters $\vec{\gamma}^*, \vec{\beta}^*$ that minimise the energy $F_p(\vec{\gamma},\vec{\beta})$.

$$
(\vec{\gamma}^*,\vec{\beta}^*) = \arg\min_{\vec{\gamma},\vec{\beta}} F_p(\vec{\gamma},\vec{\beta}), \qquad \alpha = \frac{F_p(\vec{\gamma}^*,\vec{\beta}^*)}{C_{\max}}
$$

Quantum Approximate Optimization Algorithm (QAOA)

Key Design Decisions

1. **Circuit Depth** (p)
   - Controls the expressivity of the ansatz
   - As $p \to \infty$ QAOA can find the exact solution
2. **Classical Optimizer**
   - Gradient-free: Nelder-Mead, COBYLA
   - Gradient-based: ADAM, SPSA
3. **Initial Parameters** $(\vec{\gamma}_0, \vec{\beta}_0)$
   - Various strategies for initialisation (e.g. TQA [9], INTERP [16])

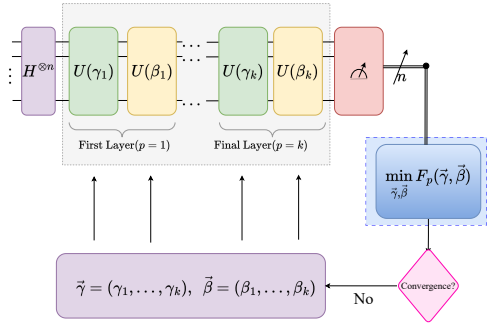


Figure 2: QAOA Circuit Architecture

Current State of QAOA Research

Key Findings

- Recent studies show optimal QAOA parameters for depth p are transferable across typical instances [2]
- Optimal QAOA depth is instance-dependent [10]
- Algorithm performance heavily influenced by:
  - Choice of classical optimizer [5]
  - Initial parameter selection [16],[7],[11],[14]

Current Limitations

- Narrow focus on specific instance classes ($d-$regular graphs, random graphs, no weights)
- Predominantly shallow depth circuits ($p \leq 3$) [10],[14]
- Limited understanding of instance feature impacts on QAOA performance and design decisions
- Lack of standardized parameter initialization frameworks [1]

Research Question

How do problem instance characteristics influence the key design decisions of the QAOA and its performance, **and** can we develop methods to automatically select optimal parameters based on these characteristics to improve QAOA's performance on diverse MaxCut instances?

Importance of Instance Diversity?

- **Parameter Sensitivity:**
  - High computational cost for repeated optimization.
- **Barren Plateaus and Rugged Landscapes:**
  - Flat optimisation landscapes hinder finding optimal parameters [2],[3].
- **Benefits of instance-based approaches?**
  - Reduces calls to quantum devices $\rightarrow$*less resource use*

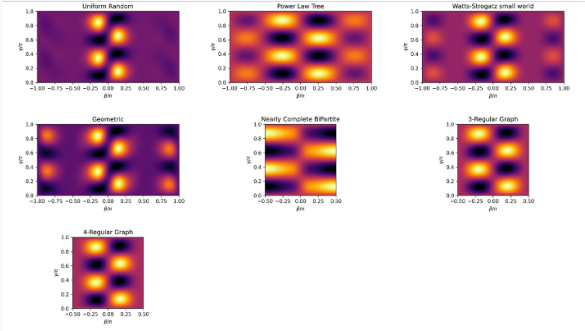


Figure 3: Optimisation Landscape for various instances of unweighted MaxCut

Instance Space Analysis

Instance Space Analysis

Instance Space Analysis [13]

- Based on the *No Free Lunch Theorem* [15]: Algorithms have strengths and weaknesses
- Identify features that differentiate instances from each other and influence algorithm performance
- Identifies which algorithms are best suited for which instances
- Visualised across a 2D plane

Instance Space Analysis

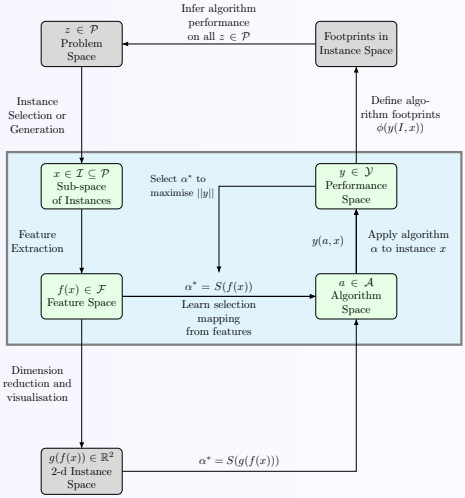
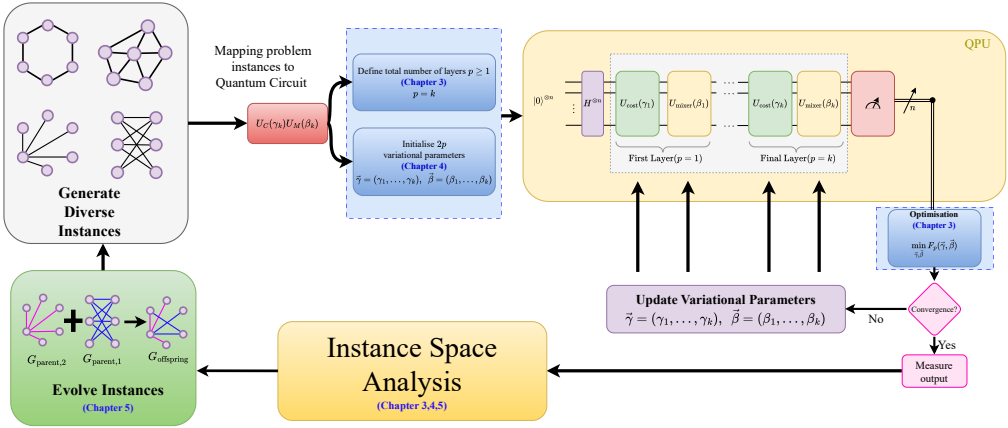


Figure 4: Instance space analysis workflow

Our Approach



Instances $\mathcal{I} \subseteq \mathcal{P}$

IS I & II: Unweighted Instances

- Random
- 3-regular
- 4-regular
- Geometric
- Watts-Strogatz
- Nearly complete bipartite

IS III & IV: Weight Distributions

- $\mathcal{U}[0, 1]$ (Uniform)
- $\mathcal{U}[-1, 1]$ (Uniform)
- $\mathrm{Exp}(\lambda)$ (Exponential)
- $\Gamma(k, \theta)$ (Gamma)
- $\mathcal{N}(\mu, \sigma^2)$ (Normal)
- $\mathrm{LogNorm}(\mu, \sigma^2)$ (Lognormal)

Total Network Classes: $7 \times 6 = 42$

Features $\mathcal{F}$

Structural Features

- Number of Edges, Nodes
- Bipartite Graph
- Clique Number
- Connected Graph
- Density
- Edge Connectivity
- Max, Min Degree
- Min. Dominating Set size
- Regular Graph
- Smallest Eigenvalue
- Vertex Connectivity

Spectral Features

- Algebraic Connectivity
- Laplacian Largest Eigenvalue
- Ratio of Two Largest Laplacian Eigenvalues
- Ratio of Two Smallest Laplacian Eigenvalues

Cycle & Path Features

- Acyclic Graph
- Average Distance
- Diameter
- Eulerian Graph
- Number of Components
- Planar Graph
- Radius

Literature

- Distance-Regular Graph
- Group Size
- Number of Cut Vertices
- Number of Minimal Odd Cycles
- Number of Orbits
- Graph Entropy: $I(G) = \frac{1}{n} \sum_i |A_i| \log |A_i|$

Weight Features

- Mean, Median, Standard Deviation, Variance
- Min/Max, IQE, Skewness, Kurtosis
- Coef. of Variation
- Weighted Avg Clustering
- Weighted Avg Path Length
- Weighted Diameter
- Weighted Radius
- Max Weighted Degree
- Min Weighted Degree

Instance Space III and IV only

Total Features: 48

Performance y

Novel Performance Metric

- **Why?** Most algorithms achieve good approximation ratios
- **Metric Components:**
  - Function evaluations
  - Approximation ratio (α)
- **Methodology:**
  - Find $\alpha_{\max}$ (best ratio achieved)
  - Set $\alpha_{\text{acceptable}} = 0.95 \times \alpha_{\max}$
  - Count iterations to reach $\alpha_{\text{acceptable}}$
  - If never reached: penalty $\kappa = 10^5$

Binary Classification

Label algorithm as:

- **"good":** if κ within 10% of best
- **"bad":** otherwise

Example for 4-reg graph (N=12):

- $p = 2$: bad (κ=100k)
- $p = 5$: bad (κ=100k)
- $p = 15$: good (κ=4,796)
- $p = 20$: bad (κ=7,029)

Combines function evaluations and approximation ratio for more nuanced performance assessment

Algorithm Design Choices $\mathcal{A}$

Design Choices Across Independent Studies

- **IS I:** Layer Depth (p)
  - Circuit depth and expressivity
  - $p \in 2, 5, 10, 15, 20$
- **IS II:** Classical Optimiser Selection
  - Which optimiser requires fewer calls to the quantum device?
  - Nelder-Mead, Conjugate Gradient, Powell, SLSQP, L-BFGS-B
- **IS III/IV:** Initialisation Technique
  - Can we have faster convergence to optimal parameters?
  - Random, TQA, INTERP, Constant, QIBPI, Three-Regular

Constants

- Features
- Instances
- Performance Metrics

All choices represent design decisions

Instance Space III: Parameter Initialisation

Instance Space III: Parameter Initialisation

$$\mathbf{Z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} -0.5225 & 0.2301 \\ -0.5939 & 0.7398 \\ 0.3977 & -0.2637 \\ -0.1423 & -0.2023 \\ -0.0091 & 0.5056 \\ 0.4226 & -0.019 \\ 0.0843 & 0.6528 \\ -0.0033 & -0.0937 \\ -0.2002 & -0.3513 \\ 0.3448 & -0.3839 \end{pmatrix}^{\top} \begin{pmatrix} \text{algebraic connectivity} \\ \text{average distance} \\ \text{clique number} \\ \text{diameter} \\ \text{maximum degree} \\ \text{maximum weighted degree} \\ \text{number of edges} \\ \text{radius} \\ \text{skewness weight} \\ \text{weighted average clustering} \end{pmatrix} \tag{1}$$

Instance Space III - Features

Instance Space III - Algorithms

Instance Space III - Summary

Evolving Instances

Software for QAOA Parameter Initialisation

Appendix

Instance Space I: Layer Depth (p)

Instance Space I: Layer Depth (p)

$$\mathbf{Z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} -0.0773 & 0.6222 \\ 0.0745 & 0.2121 \\ -0.2304 & -0.315 \\ 0.3981 & -0.2411 \\ 0.0285 & 0.0749 \\ 0.2204 & 0.2713 \\ -0.1895 & 0.3176 \\ 0.3328 & -0.1081 \\ 0.221 & -0.0415 \\ 0.2822 & -0.1511 \end{pmatrix}^{\top} \begin{pmatrix} \text{ratio of two largest laplacian eigenvalues} \\ \text{average distance} \\ \text{regular} \\ \text{density} \\ \text{laplacian second largest eigenvalue} \\ \text{number of orbits} \\ \text{diameter} \\ \text{maximum degree} \\ \text{number of minimal odd cycles} \\ \text{number of edges} \end{pmatrix} \tag{2}$$

Instance Space I - Features

Instance Space I - Algorithms

Instance Space I - Summary

Instance Space II: Classical Optimiser Selection

Instance Space II: Classical Optimiser Selection

$$\mathbf{Z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} -0.1144 & 0.5049 \\ -0.0706 & 0.4874 \\ 0.133 & -0.6791 \\ 0.3027 & 0.1872 \\ -0.2388 & -0.5887 \\ -0.558 & -0.1196 \\ -0.878 & -0.3254 \\ -0.0545 & -0.7915 \end{pmatrix}^{\top} \begin{pmatrix} \text{regular} \\ \text{maximum degree} \\ \text{average distance} \\ \text{laplacian second largest eigenvalue} \\ \text{laplacian largest eigenvalue} \\ \text{density} \\ \text{algebraic connectivity} \\ \text{number of edges} \end{pmatrix} \tag{3}$$

References

[1] Amira Abbas et al. **"Quantum Optimization: Potential, Challenges, and the Path Forward".** In: *arXiv preprint arXiv:2312.02279* (2023).

[2] Fernando G. S. L. Brandão et al. **"For Fixed Control Parameters the Quantum Approximate Optimization Algorithm's Objective Function Value Concentrates for Typical Instances".** Version Submitted. In: *arXiv* (Aug. 2023). DOI: 10.48550/arXiv.1812.04170.

[3] Sergey Bravyi, David Gosset, and Robert König. **"Quantum advantage with shallow circuits".** In: *Science* 362.6412 (2018), pp. 308–311. DOI: 10.1126/science.aar3106. eprint: https://www.science.org/doi/pdf/10.1126/science.aar3106. URL: https://www.science.org/doi/abs/10.1126/science.aar3106.

[4] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. ***A Quantum Approximate Optimization Algorithm.*** 2014. arXiv: 1411.4028 [quant-ph].

[5] Mario Fernández-Pendás et al. **"A study of the performance of classical minimizers in the Quantum Approximate Optimization Algorithm".** In: *Journal of Computational and Applied Mathematics* 404 (2022), p. 113388. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2021.113388. URL: https://www.sciencedirect.com/science/article/pii/S0377042721000078.

[6] Lov 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* (1996). DOI: 10.1145/237814.237866. URL: http://dx.doi.org/10.1145/237814.237866.

[7] X. Lee et al. **"Parameters Fixing Strategy for Quantum Approximate Optimization Algorithm".** In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2021, pp. 10–16. DOI: 10.1109/QCE52317.2021.00016. URL: https://doi.ieeecomputersociety.org/10.1109/QCE52317.2021.00016.

[8] Alberto Peruzzo et al. **"A variational eigenvalue solver on a photonic quantum processor".** In: *Nature Communications* 5.1 (2014), p. 4213. DOI: 10.1038/ncomms5213. URL: https://doi.org/10.1038/ncomms5213.

[9] Stefan H. Sack and Maksym Serbyn. **"Quantum annealing initialization of the quantum approximate optimization algorithm".** In: *Quantum* 5 (July 2021), p. 491. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-01-491. URL: https://doi.org/10.22331/q-2021-07-01-491.

[10] Ruslan Shaydulin et al. **"Classical symmetries and the quantum approximate optimization algorithm".** In: *Quantum Information Processing* 20 (2021), pp. 1–28.

[11] Ruslan Shaydulin et al. **"QAOAKit: A Toolkit for Reproducible Study, Application, and Verification of the QAOA".** In: *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*. IEEE, Nov. 2021. DOI: 10.1109/qcs54837.2021.00011. URL: https://doi.org/10.1109/qcs54837.2021.00011.

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

[13] Kate Smith-Miles and Mario Andrés Muñoz. **"Instance space analysis for algorithm testing: Methodology and software tools".** In: *ACM Computing Surveys* 55.12 (2023), pp. 1–31.

[14] Shree Hari Sureshbabu et al. **"Parameter Setting in Quantum Approximate Optimization of Weighted Problems".** In: *Quantum* 8 (Jan. 2024), p. 1231. ISSN: 2521-327X. DOI: 10.22331/q-2024-01-18-1231. URL: https://doi.org/10.22331/q-2024-01-18-1231.

[15] David H Wolpert and William G Macready. **"No free lunch theorems for optimization".** In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

[16] Leo Zhou et al. **"Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices".** In: *Phys. Rev. X* 10 (2 June 2020), p. 021067. DOI: 10.1103/PhysRevX.10.021067. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.021067.