

# An Analysis of Quantum Approximate Optimization Algorithm Performance Across Various Graph Types

Wouter Pennings

Fontys University of Applied Sciences

Eindhoven, Netherlands

465288@student.fontys.nl

**Abstract**—This paper presents an analysis of the Quantum Approximate Optimization Algorithm (QAOA) performance across various graph types, focusing on its application to the maxcut problem. Quantum computing, leveraging phenomena such as superposition and entanglement, offers potential advancements in solving complex computational problems more efficiently than classical methods. QAOA, a hybrid quantum algorithm, combines classical and quantum computational resources to approximate solutions to optimization problems. This study explores QAOA's effectiveness in both noiseless and noisy environments using a quantum computer simulator. The noiseless experiments involve varying graph sizes and connectivity levels, for realistic noisy quantum computing conditions experiments how different levels of noise impact its performance. Key parameters of QAOA,  $\beta$ ,  $\gamma$ , and  $P$ , are optimized using the Simultaneous Perturbation Stochastic Approximation (SPSA) method. The results indicate that QAOA performs well in identifying optimal or near-optimal max-cuts for smaller graphs but faces challenges as graph size increases. Noise significantly impacts QAOA's performance, although it still outperforms random selection methods. The study highlights the need for further research to optimize QAOA parameters and explore its scalability and robustness in practical applications. The findings highlight QAOA's potential for approximating solutions to optimization problems on small-scale graphs, while also underscoring challenges related to scalability and performance consistency in the presence of quantum noise. Limitations include the limited diversity of graphs tested, particularly in noisy experiments, and the fixed  $P$  parameter, suggesting avenues for future research. This study contributes valuable practical insights into QAOA's behavior, addressing a gap in comprehensive experimental analysis of quantum algorithms.

**Index Terms**—Quantum computing, QAOA, Quantum noise, Graphs, Maxcut, Optimization

## I. INTRODUCTION

In recent years, quantum computing has emerged as a promising field with the potential to revolutionize the way complex computational problems in fields such as medicine, material science and machine learning are approached [1], [2], [3]. Various algorithms have gotten attention for their theoretical ability to solve problems more efficiently than their classical peers; Quantum Approximate Optimization Algorithm is one of them. To understand this paper, it is important to grasp two key quantum phenomena: superposition and entanglement. Superposition allows a quantum bit (qubit) to exist in a both the 0 and 1 state simultaneously

until it is measured, at which point it collapses into a single definite state [4], [5], this mechanic is represented by a Hadamard (H) gate. Entanglement, on the other hand, describes a connection between two or more qubits where their states are intertwined, meaning the state of one qubit influences the state of another [4], [5], this mechanic is represented by a controlled-not (CNOT) gate.

Quantum Approximate Optimization Algorithm (QAOA) is a hybrid quantum algorithm designed to accurately approximate optimization problems. A hybrid quantum algorithm, means it combines both classical and quantum computational resources, leveraging the strengths of each to solve complex problems [6]. This exploration uses QAOA to address the max-cut problem. Maxcut is a graph problem where the goal is to divide the nodes into two groups so that the number of cut edges connecting the different groups is as large as possible. This paper uses the variant where all edges have the same weight. With QAOA, each node in the graph is represented by a qubit, which is put into a superposition state using a Hadamard (H) gate. The edges connecting these nodes are represented by entangling the corresponding qubits, using either a ZZ-gate or a sequence of CNOT-RZ-CNOT gates.

QAOA has three key parameters:  $\beta$ ,  $\gamma$ , and  $p$ . The parameter  $p$  sets the depth of the circuit, essentially dictating how many layers of operations are applied. The parameters  $\beta$  and  $\gamma$  are the angles that are classically optimized to find the best possible solution to our problem. The  $\beta$  angles are used in the mixing layer, also known as the mixing Hamiltonian, where they control the mixing of quantum states by rotating qubits around the X-axis. This process helps explore different configurations of the solution space. On the other hand, the  $\gamma$  angles are applied in the cost layer, or the cost Hamiltonian, where they adjust the rotations, guiding the solution towards an optimal outcome. The implementation used in this paper draws on the approach detailed in Farhi et al. [7], ensuring that the study builds on established methods while exploring new insights.

Several works provide foundational context and highlight ongoing challenges in QAOA research. The review by Blekos et al. [8] offers a recent comprehensive overview of QAOA and its variants, discussing its applicability, hardware challenges such as error susceptibility, and various parameter optimization strategies. While their work broadly surveys the landscape of QAOA, including some experimental imple-

mentations of variants, this analysis aims to address the lack of research on the practical application of QAOA, particularly how it behaves across different types of graphs. The study explores the effectiveness of QAOA on the maxcut problem by generating graphs through altering two key parameters: the number of nodes and connectivity. All experiments are conducted using a quantum computer simulator because it is more accessible than actual quantum hardware and has the unique capability to precisely control and vary noise levels. While most experiments are performed in a noiseless environment, some are conducted with increasing levels of noise to study its effect on performance. This approach builds upon existing work, such as that by Herrman et al. [9], who examined the relationship between particular graph features (e.g., symmetries, odd cycles, and density) and QAOA success by analyzing smaller graphs. Similar to their work, this paper examines how graph types influence QAOA performance for maxcut. However, our study differentiates itself by considering a wider range of graph sizes and incorporating noisy simulations. This provides a more generalized understanding of QAOA's behavior under conditions relevant to current and future quantum hardware. The study offers an experimental analysis of QAOA's practical performance amidst varied graph complexities and environmental noise.

Analyzing QAOA under noise-free conditions is important for understanding its fundamental capabilities and limitations, independent of the imperfections inherent in current quantum hardware. While it is widely acknowledged that current quantum computers operate in the Noisy Intermediate-Scale Quantum (NISQ) era, where noise significantly impacts computation, noisiness simulations offer a baseline to which more realistic experiment results can be compared to.

## II. METHODOLOGY

The objective of this study is to understand how the Quantum Approximate Optimization Algorithm (QAOA) behaves when solving the max-cut problem across various graphs in both ideal and noisy environments.

Two types of experiments are conducted. The first type, noisy experiments, aims to understand how noise impacts QAOA's effectiveness. This is achieved by applying increasing levels of noise to a single graph until a realistic noise level is reached. The second type, noiseless or ideal experiments, investigates QAOA performance across different graphs, each with increasing numbers of nodes or edges.

For the ideal experiments, graphs consist of 4, 5, 6, 7, 8, 9, 10, or 15 nodes with randomly generated edges. Each graph size has two variants: low connectivity and high connectivity. Low connectivity implies a 40% chance of connection between nodes, while high connectivity implies an 80% chance. This results in a total of sixteen distinct graphs, all generated using NetworkX. NetworkX is a Python library for studying graphs and networks [10].

The noisy experiments utilize a single graph, referred to as the "house with a X" (Fig. 1). These experiments explore how increasing noise levels degrade QAOA's effectiveness.

The noise levels, implemented as `noise_factor`, considered are 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, and 1. Baseline experiments include randomly selected max-cut instances and an ideal circuit simulation. The noise models are based on Di Bartolomeo et al.'s implementation [11], with parameters used from Qiskit Kyiv quantum computer. These parameters represent the single-qubit depolarizing error probability ( $P$ ), amplitude damping time of a qubit in ns ( $T1$ ) and the dephasing time of a qubit in ns ( $T2$ ).  $P$  is multiplied by `noise_factor` while  $T1$  and  $T2$  are divided by `noise_factor` as they are inversely proportional to `noise_factor`.

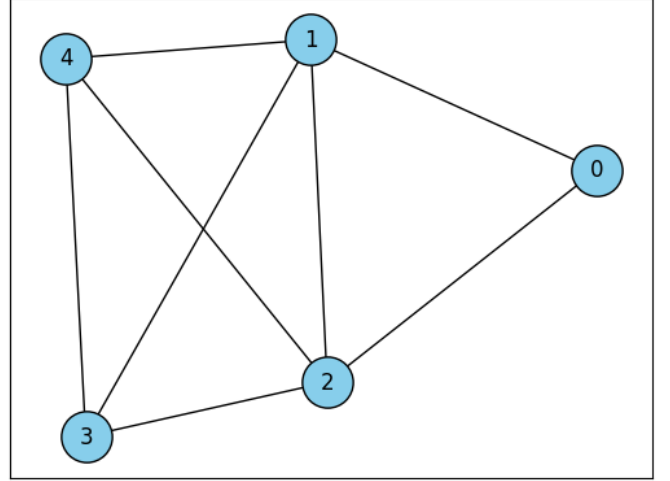


Fig. 1: Visualization of "house with a X"

Both ideal and noisy experiments employ the Simultaneous Perturbation Stochastic Approximation (SPSA) to optimize  $\gamma$  and  $\beta$  parameters, parameter  $P$  is fixed at five across all experiments. SPSA's parameters are set to default values except for iteration (set to 10), learning rate (0.2), learning rate decay (0.602), and perturbation magnitude (0.2). The initial guess used in this study of  $\gamma$  and  $\beta$  for all layers is 0.5. There are many optimizers available, such as COBYLA, however SPSA seems to perform well in context such as these [12].

To evaluate and compare the performance of the experiments, several evaluation metrics are used. The metrics are the same for both type of experiments. The metrics are:

- **mean\_random**: The average cut size based on probabilities, assuming  $P$  is 0.
- **mean**: The average cut size based on the statevector's probabilities.
- **mean\_top\_10**: The average cut size of the top ten cuts with the highest probability.
- **mean\_top\_5**: The average cut size of the top five cuts with the highest probability.
- **mean\_top\_3**: The average cut size of the top three cuts with the highest probability.
- **approximation\_ratio**:

$\frac{\text{mean\_top\_10}}{\text{maxcut\_bruteforce}} = \text{approximation\_ratio}$ . This metric is used in other papers, such as [9] and [13]

- **maxcut\_qaoa**: The cut with the highest probability from QAOA.
- **maxcut\_bruteforce**: The actual max-cut of the graph determined through classical brute-force methods.

Average cut size is a weighted mean where the weights are based on the probabilities.

In the case of noisy experiments, the statevector probabilities are also shown, which indicate the deviation in results due to noise. These metrics help determine QAOA's performance effectiveness in solving the max-cut problem.

#### A. QAOA implementation & measuring

The implementation used for the experiments in this paper and described below is based on Fahri et al [7]. The experiments are not run on actual hardware, but simulated using QuantumSim [14], an educational quantum computer simulator developed at Fontys University of Applied Sciences. This paper uses a computationally improved version which allows for faster simulation and for more qubits to be simulated.

QAOA's three main steps of this paper's implementation (Listing 1) are described below. Fig. 2 provides a visualization of the circuit.

- **Step 1**: Creating quantum circuit: Initializing a quantum circuit, each qubit represents a node of the graph.
- **Step 2**: Circuit into superposition: Putting all qubits into superposition using a Hadamard (H) gate.
- **Step 3**: Construction layers of cost and mixer Hamiltonian: Applying a ZZ gate between two qubits, which entangles them, this is the cost Hamiltonian. Using CNOT-RZ-CNOT gates to simulate a ZZ gate. Applying a RX gate to each qubit, this is the mixer Hamiltonian, the  $\beta$  value of the RX gate is multiplied by two.

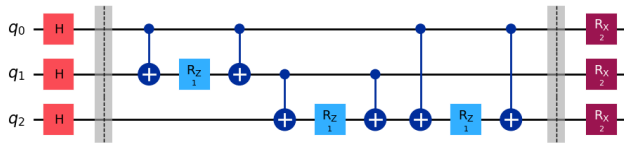


Fig. 2: The QAOA circuit was generated by Qiskit. Barriers are added before and after the cost layer to prevent Qiskit from rearranging them. The circuit has two layers, and both  $\beta$  and  $\gamma$  are  $[1, 2]$ . The graph is a basic triangle, it has three nodes, 1 is connected to 2 and 3, and 3 is also connected to 2.

After these steps a QAOA circuit is created and can be measured to find the maxcut of a graph. We will be directly looking at the probabilities of the simulator instead of measuring the statevector thousands of times. Looking directly at the probabilities of the statevector is only possible in simulators. It does not change the results of the quantum circuit as with enough measurements the results would reflect the probabilities of the statevector anyway. It does make the simulation process faster as measuring a circuit 100.000 times does take a significant amount of time. In the case of a noisy circuit, the circuit is executed a 30 times, and a mean of the probabilities is taken. The probabilities of the noisy circuit are indeterministic, therefore, the probabilities

can be quite different between executes; requiring several executes.

### III. RESULTS

#### A. Noisy experiments

Table I show all metric tracked for the noisy experiments. Noise factor describes the amount of noise. Mean random, Mean, Mean Top 10, Mean Top 5 and Mean Top 3 all are average cut sizes, in the case of Mean Top N it only calculates the average cut size of the outcomes with the top N highest probabilities. Maxcut QAOA and Maxcut BruteForce show how many edges were cut. Maxcut QAOA is the cut with the highest probability of all possible solutions.

TABLE I: THIS TABLE SHOWS ALL THE METRICS TRACKED FOR THE NOISY EXPERIMENTS, INCLUDING THE NOISELESS RUN, CONDUCTED ON THE "HOUSE WITH AN X" GRAPH (FIGURE 1).

Noise Factor	Mean Random	Mean All	Mean Top 10	Mean Top 5	Mean Top 3	Approx. Cut ratio	Max-Cut QAOA	Max-Cut Brute-Force
Noiseless	4.00	4.64	4.86	5.08	5.45	0.81	6	6
0.001	4.00	4.16	4.49	4.43	4.15	0.75	5	6
0.005	4.00	4.15	4.45	4.65	4.73	0.74	5	6
0.01	4.00	4.13	4.46	4.66	5.00	0.74	5	6
0.05	4.00	4.19	4.40	4.40	4.34	0.73	5	6
0.1	4.00	4.16	4.60	5.18	5.00	0.77	5	6
0.5	4.00	4.11	4.70	4.65	5.00	0.78	5	6
1.0	4.00	4.13	4.71	4.82	4.73	0.78	5	6

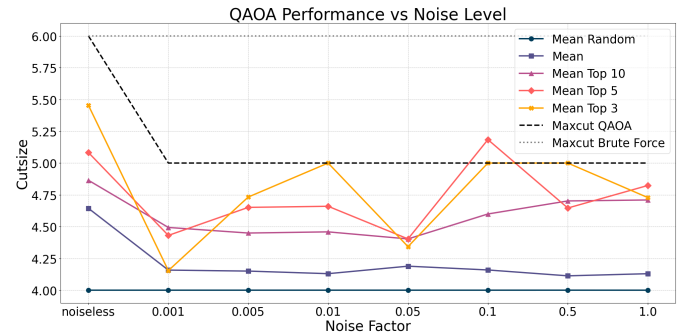


Fig. 3: This graph shows the same data as Table I.

#### a) Probability plots:

Probability plots illustrate the likelihood of each possible quantum state. The sum of all probabilities for a given plot is 1, with individual probabilities ranging from 0 to 1. These probabilities describe the chance that a specific state will be the outcome when a quantum circuit is measured.

The plots present results for various conditions:

- Fig. 4 represents the probabilities for an ideal, noiseless circuit.
- Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, and Fig. 11 show the probabilities for each experiment at its specific noise factor.
- The black bar indicates the standard deviation.

- The red and yellow arrows mark the upper and lower bounds of each possible outcome, derived from 30 executions.

At the bottom of each probability plot, labels such as  $|00111\rangle$  and  $|10001\rangle$  are presented. These labels represent the measured quantum states of the qubits in Dirac notation. For illustrative purposes, the state  $|00111\rangle$  can be decomposed as follows:

- Qubit 1 is in state 0 (Commonly noted as  $|0\rangle$ )
- Qubit 2 is in state 0 (Commonly noted as  $|0\rangle$ )
- Qubit 3 is in state 1 (Commonly noted as  $|1\rangle$ )
- Qubit 4 is in state 1 (Commonly noted as  $|1\rangle$ )
- Qubit 5 is in state 1 (Commonly noted as  $|1\rangle$ )

In the context of this study, these qubit states define the partitioning of a graph's nodes into the two groups of the cut graph. Each qubit represents a specific node within the graph. The state  $|00111\rangle$  signifies that the first and second nodes are assigned to group A, the third, fourth, and fifth nodes are assigned to group B. This allows calculation of the number of cut edges. Fig. 4 shows that  $|01100\rangle$  and  $|10011\rangle$  are the two outcomes with the highest probability. Noiseless or ideal probabilities for maxcut are always symmetrical,  $|01100\rangle$  and  $|10011\rangle$  is identical the nodes are just put into the opposite group. Noisy quantum computers have a random component to them and are indeterministic. Therefore, the probability plots of the noisy experiments, E.G. Fig. 7, is not symmetrical.

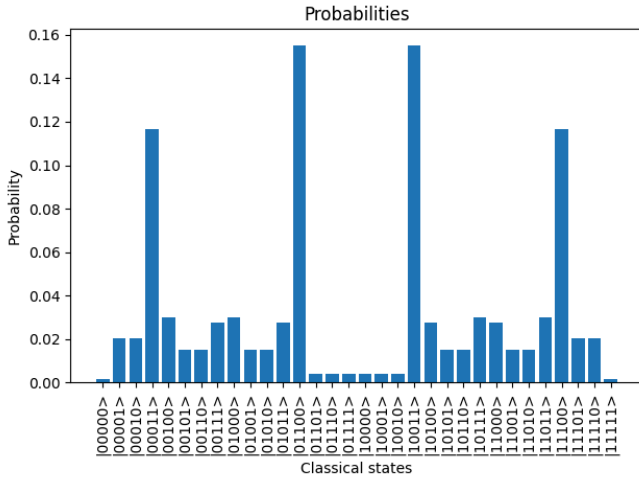


Fig. 4: Probabilities of noiseless QAOA on “House with a X”

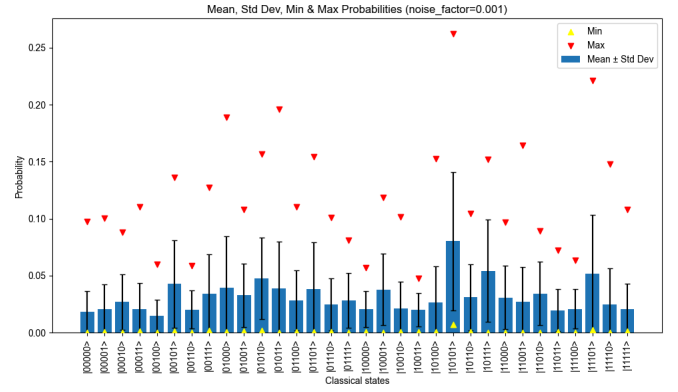


Fig. 5: Probabilities of noisy QAOA with a noise\_factor of 0.001 on “House with a X”

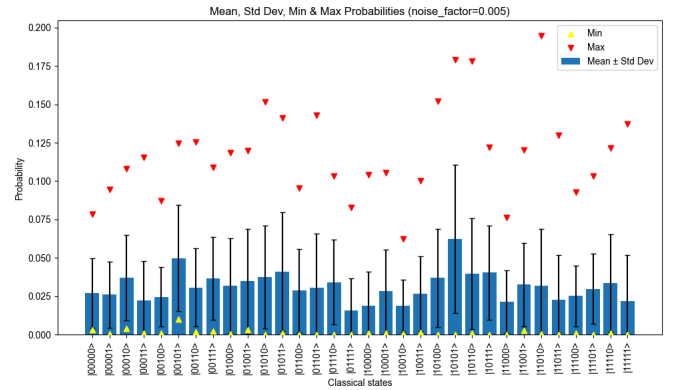


Fig. 6: Probabilities of noisy QAOA with a noise\_factor of 0.005 on “House with a X”

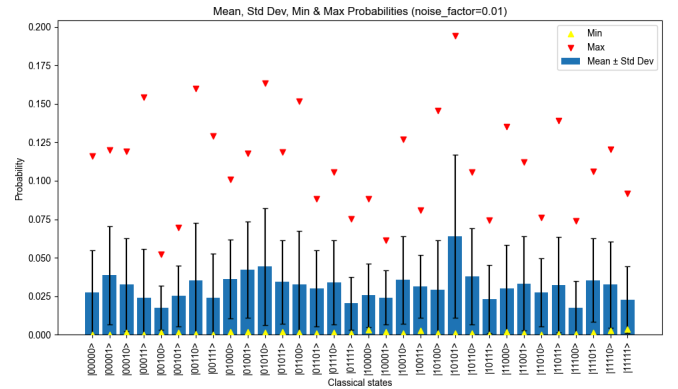


Fig. 7: Probabilities of noisy QAOA with a noise\_factor of 0.01 on “House with a X”

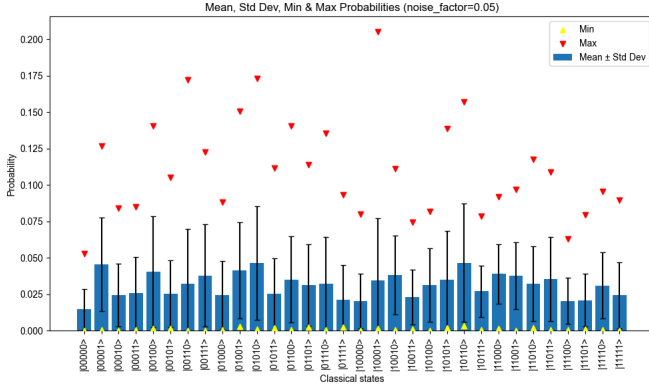


Fig. 8: Probabilities of noisy QAOA with a noise\_factor of 0.05 on “House with a X”

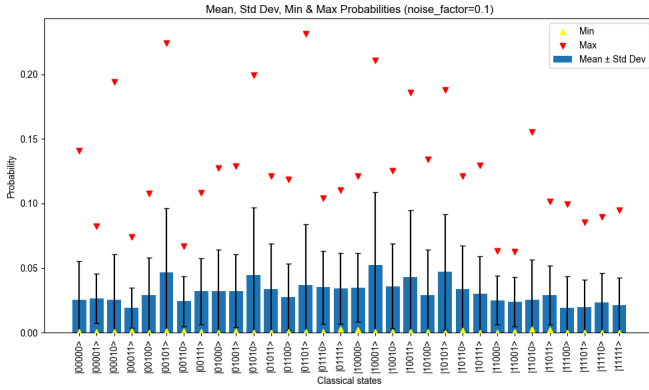


Fig. 9: Probabilities of noisy QAOA with a noise\_factor of 0.1 on “House with a X”

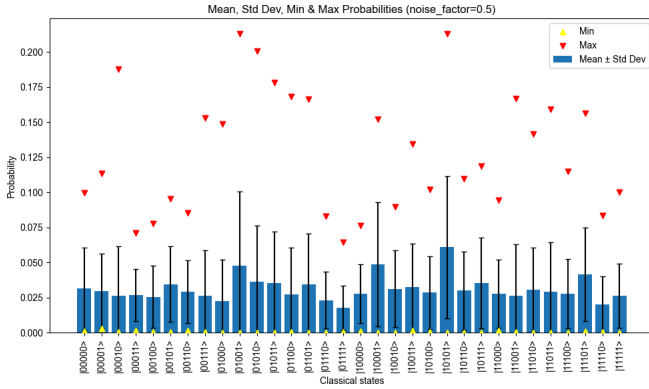


Fig. 10: Probabilities of noisy QAOA with a noise\_factor of 0.5 on “House with a X”

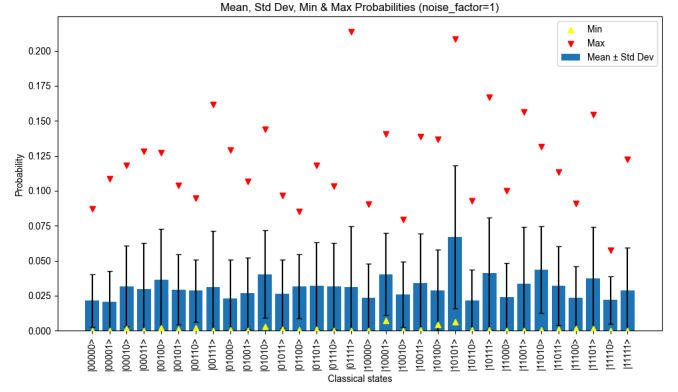


Fig. 11: Probabilities of noisy QAOA with a noise\_factor of 1 on “House with a X”

### B. Noiseless experiments

Table II and Fig. 12 show all metric tracked for the low connectivity graphs. Table III and Fig. 13 show all metric tracked for the high connectivity graphs. Nodes describes how many nodes there are in the graph used in the experiment. Mean random, Mean, Mean Top 10, Mean Top 5 and Mean Top 3 all are average cut sizes, in the case of Mean Top N it only calculates the average cut size of the outcomes with the top N highest probabilities. Maxcut QAOA and Maxcut BruteForce show how many edges were cut. Maxcut QAOA is the cut with the highest probability of all possible solutions. Figure is a reference to the graph which are in the Section VII

#### a) Low connectivity:

TABLE II: THIS TABLE SHOWS ALL THE METRICS TRACKED FOR THE NOISELESS EXPERIMENTS WITH LOW CONNECTIVITY

Nodes	Mean Random	Mean All	Mean Top 10	Mean Top 5	Mean Top 3	Ap-prox.ratio	Max-Cut QAOA	Max-Cut Brute Force	Figure
4	1.50	2.65	2.66	2.89	2.96	0.89	3	3	Fig. 14
5	2.50	3.17	3.61	4.00	4.00	0.90	4	4	Fig. 16
6	3.00	4.48	5.00	5.00	5.00	0.90	5	5	Fig. 18
7	3.00	4.71	5.48	5.69	5.87	0.91	6	6	Fig. 20
8	6.00	6.65	7.58	7.56	7.30	0.76	7	10	Fig. 22
9	8.00	8.83	10.14	10.37	10.56	0.79	11	13	Fig. 24
10	8.00	9.02	12.86	13.00	13.00	0.99	13	13	Fig. 26
15	24.50	27.54	32.95	32.56	32.30	0.94	32	35	Fig. 28



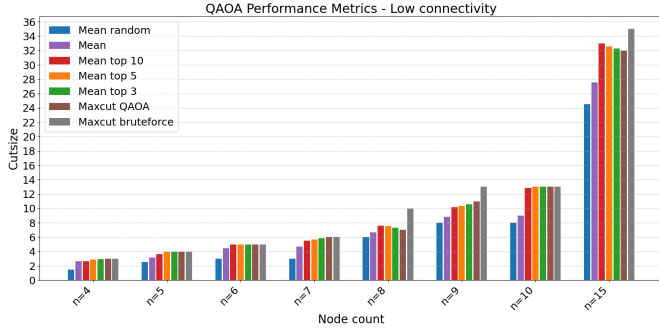


Fig. 12: This graph show how well QAOA performance a maxcut on a variety of graphs will different sizes with low connectivity by comparing multiple metrics to Maxcut brute force. Same data as Table II

#### b) High connectivity:

TABLE III: THIS TABLE SHOWS ALL THE METRICS TRACKED FOR THE NOISELESS EXPERIMENTS WITH HIGH CONNECTIVITY

Nodes	Mean Random	Mean All	Mean Top 10	Mean Top 5	Mean Top 3	Ap-prox.Cut ratio	Max-approx.Cut QAOA	Max-approx.Cut Brute Force	Figure
4	2.50	3.65	3.78	3.87	3.95	0.95	4	4	Fig. 15
5	3.50	4.37	4.85	5.00	5.00	0.97	5	5	Fig. 17
6	7.00	7.98	8.85	9.00	9.00	0.98	9	9	Fig. 19
7	9.00	9.64	11.63	12.00	12.00	0.97	12	12	Fig. 21
8	11.00	11.63	12.40	13.33	13.83	0.82	15	15	Fig. 23
9	16.00	16.69	17.60	17.43	17.69	0.88	18	20	Fig. 25
10	21.50	21.99	20.78	20.00	20.00	0.83	20	25	Fig. 27
15	41.00	42.16	48.38	48.80	48.24	0.91	48	53	Fig. 29

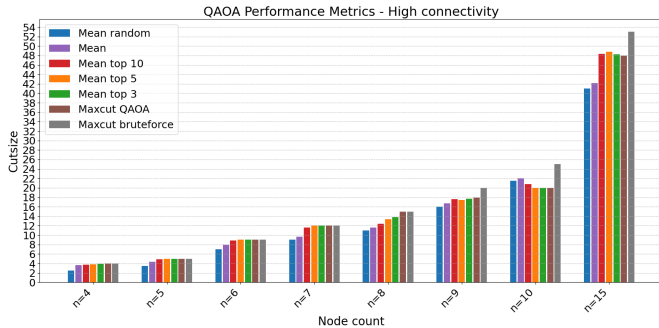


Fig. 13: This graph show how well QAOA performance a maxcut on a variety of graphs will different sizes with high connectivity by comparing multiple metrics to Maxcut brute force. Same data as Table III

## IV. DISCUSSION

### A. Noisy experiments

The experiments conducted with noise, present insights into how the QAOA performs in more realistic conditions. The noiseless experiment(Fig. 4) was the only one that found the exact maxcut of the graph. Its probability plot (Figure 4) showed confidence in the solution, with the quantum states  $|01100\rangle$  and  $|10011\rangle$  having the highest probabilities. In contrast, all noisy experiments resulted in a maxcut of 5, which is close to optimal but not the true maximum of 6.

The probability plots for noisy QAOA (Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10, and Fig. 11) reveal large differences between executions, indicated by the wide standard deviation, minimum, and maximum values. For example, the state  $|01111\rangle$  in Fig. 11 sometimes had a probability of 0 and sometimes over 0.2. This variability clearly shows the unpredictable nature of quantum circuits when noise is present.

Interestingly, the actual amount of noise did not consistently worsen the quality of the output. Fig. 3 even shows that the Mean Top 10 metric improved as the noise factor increased, although, that is likely a coincidence. While Mean Top 5 and Mean Top 3 fluctuated, the Mean and Mean Random values remained consistently low. This suggests that the presence of any noise was more impactful than the specific level of noise; which is against expectations. Nevertheless, all metrics measured in the noisy experiments were still better than selecting a solution randomly, as Mean Random is always the lowest value.

### B. Noiseless experiments

The performance of QAOA without noise is detailed in the graphs and tables (Fig. 12, Table II, Fig. 13, Table III). The high connectivity graph with 10 nodes was an exception, as its Mean random value was higher than other Mean Top N metrics. Therefore, this particular case was excluded from the following general observations.

Key findings from the noiseless experiments include:

- QAOA consistently outperforms random max-cut selections, even in scenarios where its performance is suboptimal.
- For both low and high connectivity graphs, QAOA found the most optimal maxcut for graphs with up to 7 nodes.
- For graphs with 8 nodes or more, QAOA's results often differed from the true maximum cut found by brute force, with the exception of the low connectivity graph with 10 nodes, where QAOA was effective.
- Mean did not accurately show QAOA's effectiveness. It was more insightfull to look at the average cut size of the top N (Mean Top N) solutions with the highest probabilities. This does indicate the the outcomes with the highest amount of cut edges are also the ones with the highest probabilities.
- As graphs grew larger, for both high and low connectivity, there was a trend where focusing on a smaller number of top solutions (Mean Top N with smaller N) did not always lead to a higher average cut size. This indicates that the best cuts are not always the most probable ones.
- QAOA's performance was generally better with high-connectivity graphs compared to the low connectivity graphs. This is likely to do with the fact that to even have a valid maxcut on a high connectivity graph you likely have to cut as more edges, relatively speaking compared to low connectivity graphs. That is also why Mean random is closer to Maxcut Brute force for high connectivity graphs compared to low connectivity graphs.

Why  $N=10$  with high connectivity is such an outlier is not understood. Similarly,  $N=8$  for low connectivity is also an outlier as Mean Top 5 and Mean Top 3 are both lower than Mean Top 10, even though this is not the case for  $N=9$  and  $N=10$ . There is no existing literature on the inconsistent performance of QAOA. The fact that QAOA has such worse results with these two graphs without there being an explanation is concerning, as that makes QAOA unreliable.

In summary, these experiments indicate that QAOA provides a good approximation of the optimal Max-cut solution for small graph sizes, irrespective of connectivity levels. However, QAOA's effectiveness diminishes as graph sizes increase, while connectivity appears to have a limited impact.

### C. QAOA P Count

In this study, the  $p$  parameter for QAOA was set to a constant value of 5 for all experiments. QAOA showed its weakest performance with the largest graphs (15 nodes), and this trend was also seen with graphs of 9 and 10 nodes. Increasing the  $P$  parameter could potentially allow QAOA to perform more iterations, which might help it find a more effective maxcut solution. Increasing  $P$  might not help for noisy quantum computers, as deeper circuits suffer from more noise [15], [16], which might limit its effectiveness.

Instead of choosing a static  $P$ , it is also an option to optimize  $P$  together with  $\beta$  and  $\gamma$ . This does complicate the parameter optimization as this adds extra dimensions to optimize.

### D. Limitations

This study has several limitations:

- **Graph diversity:** Only one type of graph was used for each experiment. QAOA can behave differently with various graphs, so averaging results from multiple graph types might have reduced the impact of unusual behaviors, which likely caused the outlier results for low connectivity graphs with 8 nodes and high connectivity graphs with 10 nodes.
- **Real-world and common graph structures:** The paper did not explore how QAOA performs on real-life and common graph structures, like social networks, which have distinct features that might affect QAOA's performance.
- **Noisy experiment graph:** The noisy experiments used only one graph. As seen in the noiseless experiments, some graph sizes can be outliers, and the "house with X" graph used for noisy tests might also be such an outlier, other graphs might perform better or worse.
- **Number of noisy experiment executions:** The noisy experiments involved 30 executions per experiment. Results varied greatly between runs, a few skewed executions could heavily influence the outcome. More executions, around a hundred or more, would likely provide more consistent results, however further testing is needed.
- **Noise model and parameters:** This study used one approach to model noise in quantum computer simulations with a single set of parameters (Qiskit Kyiv).

Exploring other noise implementations and/or different noise parameters could lead to very different results.

- **Optimizer configuration:** The SPSA optimization function, used to find the best  $\beta$  and  $\gamma$  values, might not be ideally set up to find optimal values in a noisy quantum computer environment

### E. Conceptual problem: Parameter selection

The strategy of optimizing QAOA parameters individually for each graph raises a conceptual concern: using a classical optimizer, such as SPSA, to determine parameters that yield a good maxcut solution effectively involves solving the problem in advance. This introduces an apparent circularity to the approach. The process involves sampling a large set of parameter combinations and selecting those that yield favorable results. However, this raises a concern: if the goal is simply to identify high-quality solutions through classical optimization, would it not be more efficient to use a classical heuristic, such as simulated annealing, to solve maxcut directly?

A response to this concern might be that exploring a wide range of QAOA parameters and selecting the best performing configurations is more effective or computationally efficient than relying on purely classical heuristics. Nonetheless, this highlights an area where further research is needed. QAOA, and quantum computing more broadly, remains largely theoretical and is still far from practical deployment for real world problems. It would be interesting to investigate whether a set of "universal parameters" could be found that perform reasonably well across many graph instances, even if not optimally for each one. In this context, Muller et al. [16] suggest a degree of universality of optimal  $\beta$  and  $\gamma$  parameters for QAOA, which supports the feasibility of this direction.

Alternatively, future work could explore whether good parameter choices can be heuristically "guessed" based on structural properties of the graph, such as its size, degree distribution, or symmetry [17]. Such approaches could help mitigate the inefficiencies and circularity of instance-specific optimization, potentially making QAOA a more scalable and practical algorithm.

## V. CONCLUSION

In noiseless environments, QAOA consistently outperformed random selection for the maxcut problem on the graphs tested. For smaller graphs (up to 7 nodes), QAOA was effective, often finding the true maximum cut. As the graph size increased beyond 7 nodes, QAOA's ability to identify the exact maxcut diminished, regardless of graph connectivity. However, metrics still demonstrated that QAOA is relatively effective at approximating an optimal maxcut. In general, these experiments demonstrate current limitation of QAOA's scalability for larger problem instances on ideal quantum computers.

The noisy experiments provided insights into QAOA's behavior in more realistic scenarios. While the noiseless QAOA experiment found the optimal maxcut, the noisy implementations consistently resulted in a near-optimal cut,

showing a degradation from noiseless performance. A significant observation was the high variability in outcome probabilities in the presence of noise. Still, noisy QAOA performed better than random selection, demonstrating its potential even when affected by errors.

#### ACKNOWLEDGMENTS

Qin Zhao and Nico Kuijpers were my supervisor during the process of writing paper

#### VII. APPENDIX

```
def qaoa_circuit(gamma:list[float], beta:list[float], nodes:list,
edges:list, p:int) -> Circuit:
    # Consistency check
    if len(gamma) != p or len(beta) != p:
        raise ValueError(f"Lists gamma and beta should be
of length p = {p}")

    # Create circuit with n qubits, where n is the number
of nodes
    n = len(nodes)
    circuit = Circuit(n)

    # Initialize circuit by applying the Hadamard gate to all
qubits
    for q in range(n):
        circuit.hadamard(q)

    # Construct p alternating cost and mixer layers
    for i in range(p):

        # Construct cost layer with parameter gamma[i]
        for edge in edges:
            circuit.cnot(edge[0], edge[1])
            circuit.rotate_z(2 * gamma[i], edge[1])
            circuit.cnot(edge[0], edge[1])

        # Construct mixer layer with parameter beta[i]
        for q in range(n):
            circuit.rotate_x(2 * beta[i], q)

    return circuit
```

Listing 1: Example implementation of QAOA, implemented in QuantumSim

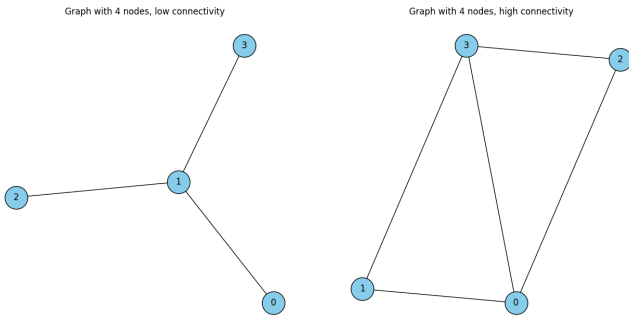


Fig. 14: Graph with 4 nodes and low connectivity

Fig. 15: Graph with 4 nodes and high connectivity

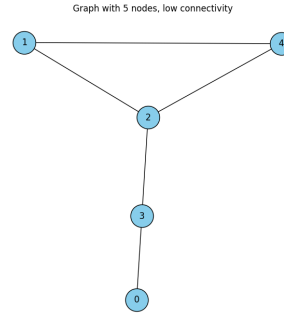


Fig. 16: Graph with 5 nodes and low connectivity

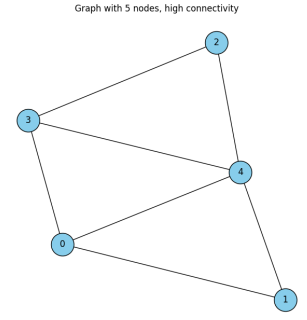


Fig. 17: Graph with 5 nodes and high connectivity

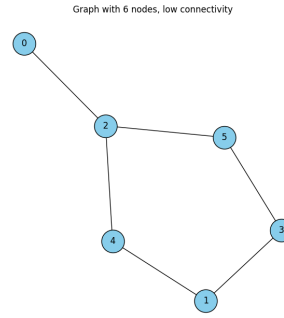


Fig. 18: Graph with 6 nodes and low connectivity

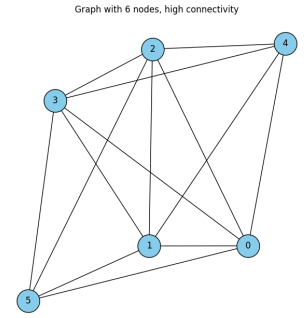


Fig. 19: Graph with 6 nodes and high connectivity

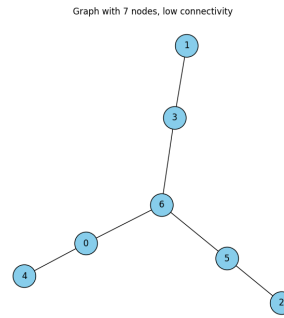


Fig. 20: Graph with 7 nodes and low connectivity

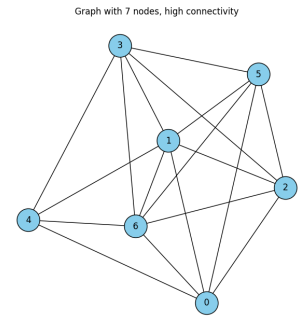


Fig. 21: Graph with 7 nodes and high connectivity

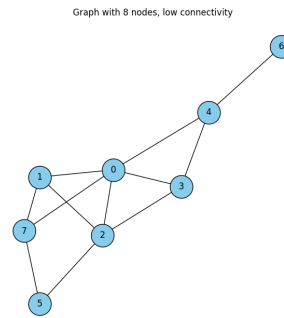


Fig. 22: Graph with 8 nodes and low connectivity

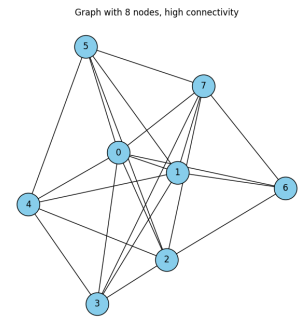


Fig. 23: Graph with 8 nodes and high connectivity



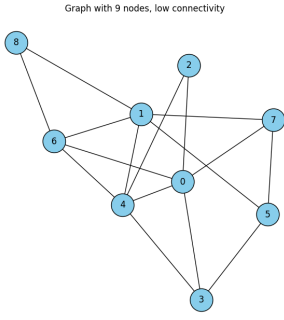


Fig. 24: Graph with 9 nodes and low connectivity

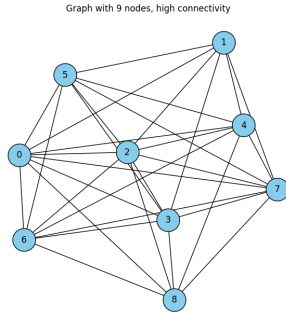


Fig. 25: Graph with 9 nodes and high connectivity

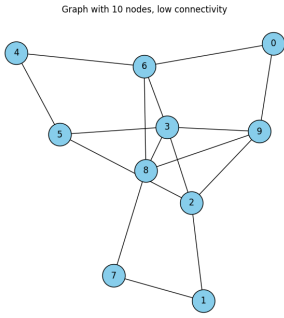


Fig. 26: Graph with 10 nodes and low connectivity

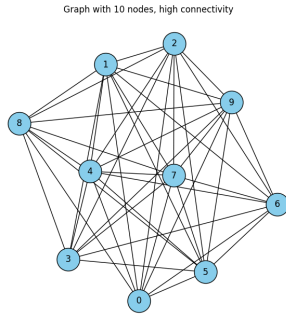


Fig. 27: Graph with 10 nodes and high connectivity

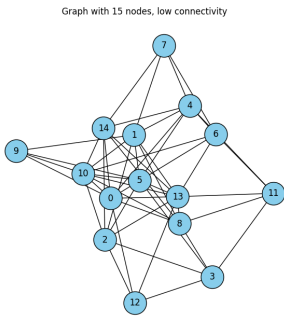


Fig. 28: Graph with 15 nodes and low connectivity

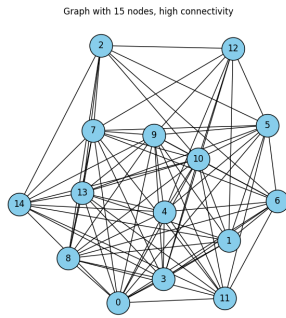


Fig. 29: Graph with 15 nodes and high connectivity

## REFERENCES

- [1] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, "Improving Quantum Algorithms for Quantum Chemistry." Accessed: Jun. 19, 2025. [Online]. Available: <http://arxiv.org/abs/1403.1539>
- [2] K. M. Svore and M. Troyer, "The Quantum Future of Computation," *Computer*, vol. 49, no. 9, pp. 21–30, Sep. 2016, doi: 10.1109/MC.2016.293.
- [3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, doi: 10.1038/nature23474.
- [4] N. S. Yanofsky, "An Introduction to Quantum Computing." Accessed: Jun. 19, 2025. [Online]. Available: <https://arxiv.org/abs/0708.0261v1>
- [5] Prashant, "A Study on the basics of Quantum Computing." Accessed: Jun. 19, 2025. [Online]. Available: <http://arxiv.org/abs/quant-ph/0511061>
- [6] R. Campos, "Hybrid Quantum-Classical Algorithms." Accessed: Jun. 19, 2025. [Online]. Available: <http://arxiv.org/abs/2406.12371>

- [7] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm." Accessed: May 09, 2025. [Online]. Available: <http://arxiv.org/abs/1411.4028>
- [8] K. Blekos *et al.*, "A Review on Quantum Approximate Optimization Algorithm and its Variants," *Physics Reports*, vol. 1068, pp. 1–66, Jun. 2024, doi: 10.1016/j.physrep.2024.03.002.
- [9] R. Herrman, L. Treffert, J. Ostrowski, P. C. Lotshaw, T. S. Humble, and G. Siopsis, "Impact of graph structures for QAOA on MaxCut," *Quantum Information Processing*, vol. 20, no. 9, p. 289, Sep. 2021, doi: 10.1007/s11128-021-03232-8.
- [10] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, Aug. 2008, pp. 11–15.
- [11] G. Di Bartolomeo *et al.*, "Noisy gates for simulating quantum computers," *Physical Review Research*, vol. 5, no. 4, p. 43210, Dec. 2023, doi: 10.1103/PhysRevResearch.5.043210.
- [12] A. Pellow-Jarman, I. Sinayskiy, A. Pillay, and F. Petruccione, "A Comparison of Various Classical Optimizers for a Variational Quantum Linear Solver." Accessed: Jun. 19, 2025. [Online]. Available: <https://arxiv.org/abs/2106.08682v1>
- [13] "MaxCut quantum approximate optimization algorithm performance guarantees for  $p > 1$ ," *ResearchGate*, doi: 10.1103/PhysRevA.103.042612.
- [14] N. Kuijpers, "QuantumSim." [Online]. Available: <https://github.com/nicokuijpers/QuantumSim>
- [15] M. Ivezic, "The Quantum Approximate Optimization Algorithm (QAOA) – A Beginner's Guide." [Online]. Available: <https://postquantum.com/quantum-computing/quantum-approximate-optimization-algorithm-qaoa/>
- [16] T. Müller, A. Singh, F. K. Wilhelm, and T. Bode, "Limitations of quantum approximate optimization in solving generic higher-order constraint-satisfaction problems," *Physical Review Research*, vol. 7, no. 2, p. 23165, May 2025, doi: 10.1103/PhysRevResearch.7.023165.
- [17] S. H. Sureshbabu *et al.*, "Parameter Setting in Quantum Approximate Optimization of Weighted Problems," *Quantum*, vol. 8, p. 1231, Jan. 2024, doi: 10.22331/q-2024-01-18-1231.