UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

# Solving MaxCut problem with Quantunum Approximate Optimization Algorithm

*Author:*
Sophia ZHYROVETSKA

*Supervisor:*
Prof. Volodymyr TKACHUK

*A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science*

*in the*

Department of Computer Sciences
Faculty of Applied Sciences


Lviv 2021

# Declaration of Authorship

I, Sophia ZHYROVETSKA, declare that this thesis titled, "Solving MaxCut problem with Quantunum Approximate Optimization Algorithm" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

*"A classical computation is like a solo voice — one line of pure tones succeeding each other. A quantum computation is like a symphony — many lines of tones interfering with one another."*

Seth Lloyd

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

**Solving MaxCut problem with Quantunum Approximate Optimization Algorithm**

by Sophia ZHYROVETSKA

# *Abstract*

Given an undirected unweighted graph, the 2-MaxCut problem can be stated as the problem of partitioning the nodes of a graph into two subsets such that the number of edges between them is as large as possible. It is a well-studied NP-hard and APX-hard problem with applications in various fields, including statistical physics, machine learning and circuit layout design. This thesis investigates the Quantum Approximation Optimization Algorithm for solving the MaxCut problem. We study this approach analytically, show how to implement it on the quantum circuit, hold the experiments on the quantum simulator and the real quantum computer and test how good this algorithm works on graphs of different sizes.

# *Acknowledgements*

First of all I would like to thank my supervisor Prof. Volodymyr Tkachuk for his guidance and constant support during this work. I especially thank him for introducing me to quantum physics and showing how to connect it with computer science. I am also highly grateful to Dr. Khrystyna Hnatenko for sharing time with my supervisor and me to research the quantum optimization algorithms.

I am deeply thankful to the Ukrainian Catholic University and the Faculty of Applied Sciences for providing me an opportunity to spend the previous four challenging, yet unforgettable years in such a wonderful community. I would like to express the highest gratitude to all of my lecturers, who shared their valuable experience and knowledge during my academic journey.

I would also like to thank my family for their encouragement and support. Last, but not least, very special thanks goes to my friends, who always were there for me. Without you I could not have made it until the end ...

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**QAOA** **Q**antum **A**pproximation **O**ptimization **A**lgorithm
**QASM** **Q**antum **AS**seM**b**ly language

# Physical Constants

Reduced Planck's Constant $\hbar = 1.054 \times 10^{-34}\,\mathrm{J\,s^{-1}}$

# Chapter 1

# Introduction

According to the Moore's Law, the number of elements per unit area of a classic computer chip doubles every two years. However, it will cease when the size of the transistors becomes so small that quantum fluctuations become significant and classical electronic elements become unstable. Therefore, it is not surprising that the idea of building a computer, based on the quantum properties of chips, came up. At the same time, this brings up the question: will such quantum computers be able to perform classical algorithms? The answer is yes. In the early 1980s P. Benioff described the quantum mechanical model of the Turing Machine [2]. Based on his study, R. Feynman developed a universal quantum simulator. Besides, D. Deutsch started to work on quantum algorithms and showed that some of them can be exponentially faster than any possible deterministic classical algorithm [3]. This is referred to as quantum supremacy.

Progress has been swift. Nowadays quantum computers are already built by large companies and can be remotely used by every developer. Google currently has the largest quantum processor in the world (72 qubits) and IBM plans to have a 1000-qubit quantum computer by 2023. Furthermore, quantum supremacy was already demonstrated for the first time by Google in 2019 — only 200 seconds were needed for their quantum processor to perform a task, while a classical supercomputer needs 10 000 years to solve it [1]. In December 2020 physicists in China demonstrated another example of quantum supremacy. Their quantum computer in 200 seconds did a calculation that takes ordinary supercomputer 2.5 billion years to do [15].

Given this, the search for new quantum algorithms that work better than the classical ones is nowadays very active. The most interesting are NP-hard problems that most likely cannot be solved with classical algorithms in polynomial time. However, solving certain types of these problems can be seen from a new perspective if we take advantage of the quantum nature of the Universe.

One of such NP-hard problems is The Maximum Cut (MaxCut) optimization problem, which is simply formulated as the problem of partitioning the nodes of a graph into two subsets such that the number of edges between them is as large as possible. There also are generalized versions of MaxCut such as n-MaxCut problem, where the the nodes have to be parted into n subsets. Another generalization is a weighted MaxCut, where each edge has a real number weight and the aim is to maximise the sum of weights rather than the number of the edges. This thesis will focus on the unweighted 2-MaxCut problem.

The aim of this work is to solve the MaxCut problem on the quantum computer, using Quantum Approximation Optimization Algorithm, originally proposed by Farhi, Goldstone and Gutmann [4]. We are going to investigate analytically how this algorithm works, implement it on the quantum circuit and test how good it works on graphs of different sizes.

# Chapter 2

# MaxCut Problem

Though simple to state, MaxCut problem is among the first problem proven to be computationally intractable and is one of the Karp's 21 NP-complete problems [9]. MaxCut is also an APX hard problem meaning that, most likely, there is no polynomial time approximation algorithm to solve it [13].

## 2.1 Problem Description

Given an $n$-node undirected graph $G = (V, E)$ with vertex set $V = \{v_1, v_2, ..., v_n\}$ and edge set $E$, a *cut* $\{S_1, S_2\}$ of $G$ is defined as a partition of vertices into two disjoint subsets $S_1$ and $S_2$. The size of the cut for the unweighted graph is the number of edges between $S_1$ and $S_2$. *Maximum cut* is a cut, which size is at least the size of any other cut for the graph $G$.

In consideration of that, the *MaxCut* problem can be stated as follows: given a unweighted undirected graph G = (V, E), find a maximum cut.

Consider assigning a binary variable $x_i$ to each node $v_i$. The value of $x_i$ is given by the following

$$x_i = 1 \text{ if } i \in S_1, \quad x_i = -1 \text{ if } i \in S_2 \tag{2.1}$$

With that in mind, the MaxCut problem can be formulated as an optimization problem

$$max\left\{ f(x) = \frac{1}{2} \sum_{\{i,j\} \in E} (1 - x_i x_j) \right\}, \quad x_i \in \{-1, 1\}, \text{ for all } i \in V \tag{2.2}$$

If two vertices are in the same subset, then the sum in $f(x)$ equals to 0. Conversely, if the nodes are in different subsets, the sum equals to 1, which means that the edge between these nodes is cut and the overall value of $f(x)$ increases. The more edges are cut — the larger is value of $f(x)$. Hence, MaxCut is an optimization problem, where one tries to maximize the objective function $f(x)$.

Note that in case of MaxCut, maximization problem can be easily turned to a minimization one as follows

$$min\left\{ f(x) = \frac{1}{2} \sum_{\{i,j\} \in E} (x_i x_j - 1) \right\}, \quad x_i \in \{-1, 1\}, \text{ for all } i \in V \tag{2.3}$$

Figure 2.1 gives an example. If we cut the graph into the subsets $S_1 = \{1\}$ and $S_2 = \{2, 3\}$ or $S_1 = \{1, 2\}$ and $S_2 = \{3\}$, the size of the cut (which is the $f(x)$ value from (2.2)) equals to 1. If the subsets are $S_1 = \{1, 3\}$ and $S_2 = \{2\}$, then $f(x) = 2$, which is the maximum cut size for this graph.

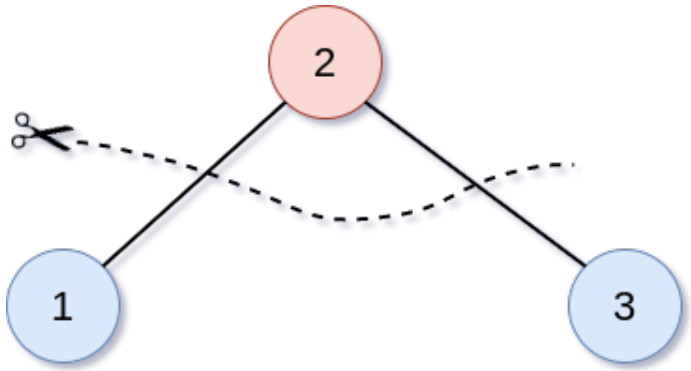


FIGURE 2.1: MaxCut example

## 2.2 Applications

While MaxCut is a well studied problem, it is still an interest of research. As well as other NP-hard problems, MaxCut has a major theoretical importance. Apart from that, it has a lot of applications in various fields, like machine learning, computer vision, network science, clustering, statistical physics and circuit layout design.

A great application of the MaxCut problem can be found in the design of very large scale integrated (VLSI) chips. One of the difficulties of this task is a minimization of vertical interconnection area (via) between the layers of the chip. In the general case, this problem is NP-hard. However, it was showed by Kajitani (1980) that the two layer via minimization problem can be solved as a maximum cut problem on a planar graph [8].

# Chapter 3

# Literature Review

As MaxCut problem is NP-hard, there exists no polynomial-time algorithm to solve it exactly, unless P = NP, which most likely is not true. Therefore, a vast majority of developed algorithms to solve MaxCut are approximation algorithms.

The first of such algorithms with ratio 0.5 was presented in 1976 by Sahni and Gonzales [14]. It is a simple greedy algorithm that iterates through the nodes and on each step decides to which subset assign a vertex $v_i$, based on which what maximizes the size of the cut. Since then some researches were done, presenting algorithms with slightly improved performance guarantees of $\frac{1}{2} + \frac{1}{2m}$, $\frac{1}{2} + \frac{n-1}{4m}$, $\frac{1}{2} + \frac{1}{2n}$, where $n$ is the number of vertices, $m$ — the number of edges.

In 1994, Goemans and Williamson(GW) developed an algorithm that achieves an approximation ratio 0.878 and has a time complexity $O(n^2 log(n)$ [6]. It is based on randomly rounding of a solution to semi definite programs and is still considered the state-of-the-art algorithm for the MaxCut problem with a proven approximation ratio.

Many further researches proposed some improvements for solving MaxCut problem. Homer and Peinado gave a parallelized version of GW [7]. Kim and Moon developed a hybrid genetic algorithm(GA) for MaxCut problem, which gives promising results, outperforming GW in many tests [10]. Kim, Yoon and Geem used harmony search algorithm for solving MaxCut, that produced even greater cut sizes than GA [11].

Apart from that, there have been good results achieved for certain types of graphs. Feige, Karpinski and Langberg considered enhancing the GW algorithm with an additional local step that moves misplaced vertices from one side of the partition to the other, until no such vertices are left. They showed that their algorithm obtains an approximation ratio of at least 0.921 for graphs of maximal degree 3, and for 3-regular graphs the approximation ratio is at least 0.924 [5].

# Chapter 4

# Quantum Computing

Before moving on to the QAOA and solving the MaxCut problem on the quantum computer, we will give a brief explanation of the quantum computing basics, needed for the further work.

## 4.1 Quantum State

States of the quantum system are represented with vectors that form a linear space. In Dirac notation they are called ket-vectors and are denoted as $|\psi\rangle$. Note, that $c|\psi\rangle$, where $c$ is a complex number, identifies the same state.

The linearity of vector space represents the principle of superposition in quantum mechanics. If the quantum system can be in the state $|\psi_1\rangle$ or $|\psi_2\rangle$, then it can also be in the state that is a linear combination of this states:

$$|\psi\rangle = c_1|\psi_1\rangle + c_2|\psi_2\rangle, \tag{4.1}$$

where $c_1$ and $c_2$ are complex numbers.

## 4.2 Operators

In quantum mechanics linear Hermitian operators, which are in fact Hermitian matrices, correspond to any observable, i.e. any quantity which can be measured in a physical experiment.

Linear mapping of the ket-vector $|\psi\rangle$ to $|\psi^{'}\rangle$ can be expressed through applying a linear operator $\hat{A}$ to the state $|\psi\rangle$

$$|\psi^{'}\rangle = \hat{A}|\psi\rangle \tag{4.2}$$

Consider the eigenvalue equation for the operator $\hat{A}$

$$\hat{A}|a\rangle = a|a\rangle \tag{4.3}$$

The quantum system in the state $|\psi\rangle$ can always be described by a superposition of eigenstates from (3.3) with some unequal coefficients.

$$|\psi\rangle = c_1|a_1\rangle + c_2|a_2\rangle + ... + c_n|a_n\rangle \tag{4.4}$$

Given that, the measurement postulate of quantum mechanics states the following.

Consider a quantum system in the state $|\psi\rangle$. Then the result of a measurement of a physical quantity $A$, which corresponds to an operator $\hat{A}$, is one of its eigenvalues

$a = a_1, a_2, ..., a_n$. At the same time, after measurement, the state vector $|\psi\rangle$ collapses to the eigenstate $|a_n\rangle$ of the $\hat{A}$ operator.

$c_n$ in (3.4) is a probability amplitude and $|c_n|^2$ is a probability that after the measurement the system will collapse to the corresponding state $|a_n\rangle$.

A simple example of the operators in quantum mechanics are spin-$\frac{1}{2}$ operators. Spin-$\frac{1}{2}$ is an intrinsic property of elementary particles (e.g. electrons) that can be in two possible states. Spin-$\frac{1}{2}$ operators in the matrix notation looks as follows

$$s^\alpha = \frac{\hbar}{2}\sigma^\alpha, \quad \alpha = x, y, z, \tag{4.5}$$

where

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4.6}$$

are Pauli matrices.

These matrices are built upon the basis of eigenvectors of $\sigma^z$ that can be denoted as

$$|0\rangle \equiv |\uparrow\rangle \equiv |+1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv |\downarrow\rangle \equiv |-1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.7}$$

Pauli matrices act on the vectors (3.7) in the following way

$$\begin{aligned} \sigma^x|\uparrow\rangle &= |\downarrow\rangle, \quad \sigma^x|\downarrow\rangle = |\uparrow\rangle \\ \sigma^y|\uparrow\rangle &= i|\downarrow\rangle, \quad \sigma^y|\downarrow\rangle = -i|\uparrow\rangle \\ \sigma^z|\uparrow\rangle &= |\uparrow\rangle, \quad \sigma^z|\downarrow\rangle = -|\downarrow\rangle \end{aligned} \tag{4.8}$$

## 4.3 Quantum Bit

A basic unit of information in a classical computer is a bit. Classical bit is realized with the classical system that can be in two stable states, that are by convention denoted as 0 and 1.

The quantum computer operates with an information encoded by quantum bits (qubits) that are the two-state quantum systems. The qubit has states $|0\rangle$ and $|1\rangle$ which can be also viewed as column vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4.9}$$

The ideal classical bit has no other states then 0 and 1. In contrast, qubit is a quantum system for which the quantum superposition principle holds. Thus, a qubit can be in the state $|\psi\rangle$ that is a linear combination of the basis states $|0\rangle$ and $|1\rangle$.

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \tag{4.10}$$

This state can be visualized as a vector enclosed inside a Bloch sphere (Figure 3.1), which is state space of all possible qubit state vectors. The vectors are allowed to rotate anywhere on the surface of the sphere, and each vector represents a different quantum state.

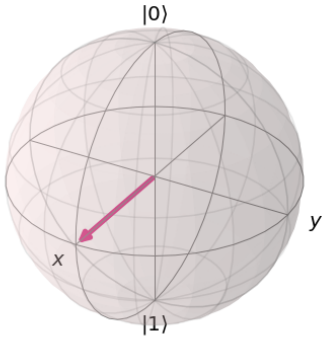


FIGURE 4.1: Bloch sphere

Two qubits can be in the state

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle \tag{4.11}$$

In this stat are encoded four binary numbers at the same time. $n$ qubits encode $2^n$ numbers. The quantum computer works with all the $2^n$ numbers simultaneously. This property is called quantum parallelism and because of that some problems can be solved on the quantum computer more effective than on the classical one.

# Chapter 5

# Methodology

## 5.1 Mapping MaxCut to the Ising Model

To solve an optimization problem on a quantum computer, we need to turn it into a problem of measurement of a quantum Hamiltonian, which is the system's total energy. Finding maximum cat of the graph can be easily mapped to a problem of maximizing the Hamiltonian of an Ising model.

The Ising model is an example for a lattice spin model. In each lattice site $k$, we consider a spin $\sigma_k$ with two possible states: $\sigma_k = \pm 1$ (corresponding to $|\uparrow\rangle$ and $|\downarrow\rangle$). Nearest sites interact with each other with an energy $J$. If there is no external field interacting with the lattice, the full energy of a spin configuration $\sigma$ is a Hamiltonian

$$H(\sigma) = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j, \tag{5.1}$$

where $\langle ij \rangle$ indicates summation over nearest neighbours.

In Chapter 2 we showed that solving the MaxCut problem is maximizing

$$f(x) = \sum_{\langle ij \rangle \in E} (1 - x_i x_j), \quad x_i \in \{-1, 1\}, \tag{5.2}$$

where each edge is counted only once. This can be also expressed as

$$f(x) = \left( - \sum_{\langle ij \rangle \in E} x_i x_j + C \right), \quad x_i \in \{-1, 1\} \tag{5.3}$$

As $C$ is a constant, maximizing $f(x)$ is equivalent to maximizing the sum

$$- \sum_{\langle ij \rangle \in E} x_i x_j, \quad x_i \in \{-1, 1\} \tag{5.4}$$

Consider assigning a spin for each vertex of the graph $G$. One can replace the binary variable $x_i$ with the eigenvalues of $\sigma^z$ spin operator, which are 1 and $-1$. Also assign energy $J$ for each edge ($J = 1$ in case of the unweighted graph). Then the graph will represent the Ising model and maximizing its Hamiltonian

$$H(\sigma) = - \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z, \quad \sigma_i^z \in \{-1, 1\} \tag{5.5}$$

will be equivalent to maximizing (5.4).

A spin configuration $\sigma$ with a maximal Hamiltonian divides all nodes into two subsets (one with $\sigma_k = -1$ and another with $\sigma_k = 1$) that are the solution for the MaxCut problem.

## 5.2 QAOA

We are going now to dive deep into the Quantum Approximation Optimization Algorithm, introduced by Farhi [4]. It is a quantum algorithm that produces approximate solutions for combinatorial optimization problems.

### 5.2.1 Overview

For MaxCut the core idea of QAOA is to find the highest energy state of the system, which is basically a state with the highest energy eigenvalue $E$

$$H|\psi\rangle = E|\psi\rangle \tag{5.6}$$

Using QAOA we try to guess how this state looks like if we can't solve the Hamiltonian for eigenstates and eigenvalues exactly. For that we can construct a "trial" (also called "guess" or "ansatz") state $|\psi^{'}\rangle$, which is basically a a linear combination of different Hamiltonian eigenstates

$$|\psi^{'}\rangle = \sum_{n=0}^{\infty} c_n|\psi_n\rangle \tag{5.7}$$

The aim is to find such $|\psi^{'}\rangle$ that the expectation value of the energy $\langle|\psi^{'}|H|\psi^{'}\rangle$ for this ansatz is as high as possible. Given that, one can introduce parameters to the trial state $|\psi_0^{'}(\gamma_1, \gamma_2, ...)\rangle$.

Then we can vary parameters within the ansatz are until the energy of the trial state is maximized. The main task in this method is to choose a good trial state with the good set of parameters.

### 5.2.2 Constructing a Trial State for the MaxCat

Consider the initial state of the quantum system, which is a uniform superposition over the basis states of $n$ spins, where $n$ is a number of vertices of the graph

$$|+_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{z\in\{-1,1\}^n} |z\rangle. \tag{5.8}$$

To construct a parameter dependent trial state we choose two parameters: $\alpha$ and $\beta$.

Define a unitary operator $U(C, \alpha)$ that depends on the angle $\alpha$,

$$U(C, \alpha) = e^{-i\alpha C} \tag{5.9}$$

Here $C$ denotes an cost function to be maximized. Recall that an objective function for the MaxCut problem is a Hamiltonian, that we defined in (5.5).

$$U(C, \alpha) = \prod_{\langle i,j\rangle} e^{-i\alpha\sigma_i^z\sigma_j^z} \tag{5.10}$$

Besides, define the operator $B$, which is the sum of all single bit $\sigma^x$ operators,

$$B = \sum_{j=1}^{n} \sigma_j^x \tag{5.11}$$

Now we can define a unitary operator $U(B, \beta)$ that depends on an angle $\beta$ and which we will refer as to the mixer operator

$$U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta\sigma_j^x} \tag{5.12}$$

Given that, we can construct a trial state,

$$|\alpha, \beta\rangle = U(B, \beta)U(C, \alpha)|+_n\rangle \tag{5.13}$$

Now we can find an approximate solution to the MaxCut problem through maximizing the expectation value of the Hamiltonian

$$\langle H\rangle = \langle\alpha, \beta|H|\alpha, \beta\rangle \tag{5.14}$$

### 5.2.3 Example for the 2-vertices Graph

Consider assigning a spin for each node of the 2-vertices graph. Note that if we want to solve MaxCut on the quantum computer, it is, in general, equivalent to assigning a qubit to each vertex.

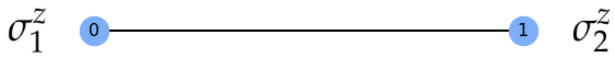


FIGURE 5.1: Spins assigned to the 2-vertices graph

The Hamiltonian (5.5) for it looks as follows:

$$H(\sigma) = -\sigma_1^z\sigma_2^z \tag{5.15}$$

Maximum of H is at:

$$\sigma_1^z = 1,\ \sigma_2^z = -1 \quad \text{or} \quad \sigma_1^z = -1,\ \sigma_2^z = 1 \tag{5.16}$$

Solving MaxCut problem for this graph is trivial but we can use it to explicitly show how QAOA works.

Recall that the eigenstates of $\sigma^z$ are demoted as

$$|0\rangle \equiv |\uparrow\rangle \equiv |+1\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle \equiv |\downarrow\rangle \equiv |-1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \tag{5.17}$$

Consider now the initial state for the system of two spins

$$\begin{aligned} |+,+\rangle &= \frac{1}{2}\left(|\uparrow\rangle_1 + |\downarrow\rangle_1\right)\left(|\uparrow\rangle_2 + |\downarrow\rangle_2\right) = \\ &= \frac{1}{2}\left(|\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + |\downarrow\downarrow\rangle\right) \end{aligned} \tag{5.18}$$

Note that (5.18) is a superposition of all the eigenstates of the Hamiltonian (5.15) and among them there is one or more state which are the solution to the MaxCut problem ($|\uparrow\downarrow\rangle$ and $|\downarrow\uparrow\rangle$ in case of our graph).

Given that, the trial state (5.13) for this graph will look as follows:

$$|\alpha,\beta\rangle = e^{-i\beta(\sigma_1^x+\sigma_2^x)}e^{-i\alpha\sigma_1^z\sigma_2^z}|+,+\rangle \tag{5.19}$$

Recall that

$$\begin{aligned} \sigma^x|\uparrow\rangle &= |\downarrow\rangle, \quad \sigma^x|\downarrow\rangle = |\uparrow\rangle \\ \sigma^z|\uparrow\rangle &= |\uparrow\rangle, \quad \sigma^z|\downarrow\rangle = -|\downarrow\rangle \end{aligned} \tag{5.20}$$

Also consider the fact that, with the use of the Taylor series, we can derive

$$\begin{aligned} e^{i\beta\sigma^z} &= \sum_{k=0}^{\infty}\frac{1}{k!}(i\beta\sigma^z)^k = \\ &= I + i\beta\sigma^z - \frac{\beta^2 I}{2!} - i\frac{\beta^3\sigma^z}{3!} + ... = \\ &= Icos\beta + i\sigma^z sin(\beta) \end{aligned} \tag{5.21}$$

With that in mind, one can calculate (5.19), applying the operators one by one.

**Step 1**

$$\begin{aligned} |\alpha,\beta\rangle &= \frac{1}{2}e^{-i\alpha\sigma_1^z\sigma_2^z}(|\uparrow\uparrow\rangle + |\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle + |\downarrow\downarrow\rangle) = \\ &= \frac{1}{2}(e^{-i\alpha}|\uparrow\uparrow\rangle + e^{i\alpha}|\uparrow\downarrow\rangle + e^{i\alpha}|\downarrow\uparrow\rangle + e^{-i\alpha}|\downarrow\downarrow\rangle) \end{aligned} \tag{5.22}$$

**Step 2**

$$\begin{aligned} |\alpha,\beta\rangle = \frac{1}{2}e^{-i\beta(\sigma_1^x+\sigma_2^x)}(e^{-i\alpha}|\uparrow\uparrow\rangle + e^{i\alpha}|\uparrow\downarrow\rangle + e^{i\alpha}|\downarrow\uparrow\rangle + e^{i\alpha}|\downarrow\downarrow\rangle) = \\ = \frac{1}{2}\Big((e^{-i\alpha}cos2\beta - ie^{i\alpha}sin2\beta)|\uparrow\uparrow\rangle + \\ +(e^{i\alpha}cos2\beta - ie^{-i\alpha}sin2\beta)|\uparrow\downarrow\rangle + \\ +(e^{i\alpha}cos2\beta - ie^{-i\alpha}sin2\beta)|\downarrow\uparrow\rangle + \\ +(e^{-i\alpha}cos2\beta - ie^{i\alpha}sin2\beta)|\downarrow\downarrow\rangle\Big) \end{aligned} \tag{5.23}$$

(5.23) reveals that applying the operators to the initial state, which is a uniform superposition, changes the coefficient before the each state. Recall that these coefficients are probability amplitudes $A_{\phi_n}$ and $|A_{\phi_n}|^2$ is a probability that the system will be in the state $\phi_n$ after the measurement.

For (5.23) the probabilities are:

$$|A_{\uparrow\uparrow}|^2 = |A_{\downarrow\downarrow}|^2 = \frac{1}{4}(1 + sin(2\alpha)sin(4\beta)) \tag{5.24}$$
$$|A_{\uparrow\downarrow}|^2 = |A_{\downarrow\uparrow}|^2 = \frac{1}{4}(1 - sin(2\alpha)sin(4\beta))$$

Note that the sum of all amplitudes squared, obviously, equals to 1:

$$|A_{\uparrow\uparrow}|^2 + |A_{\uparrow\downarrow}|^2 + |A_{\downarrow\uparrow}|^2 + |A_{\downarrow\downarrow}|^2 = 1 \tag{5.25}$$

The expectation value of the Hamiltonian for (5.23) is

$$\langle H \rangle = \langle \alpha, \beta | H | \alpha, \beta \rangle = -sin(2\alpha)sin(4\beta) \tag{5.26}$$

One has now to maximize $\langle H \rangle$ to find the state where the energy is the highest. For more complicated graphs we will use some classical optimization algorithms, but as this graph is simple, we can select proper $\alpha$ and $\beta$ manually.

$$max\langle H \rangle = 1 \tag{5.27}$$

when

$$2\alpha = \tfrac{\pi}{2}, \quad 4\beta = -\tfrac{\pi}{2}$$
or
$$2\alpha = -\tfrac{\pi}{2}, \quad 4\beta = \tfrac{\pi}{2}$$

Given the parameters values, we can now calculate probabilities for each state of the quantum system

$$|A_{\uparrow\uparrow}|^2 = |A_{\downarrow\downarrow}|^2 = 0, \quad |A_{\uparrow\downarrow}|^2 = |A_{\downarrow\uparrow}|^2 = \frac{1}{2} \tag{5.28}$$

Therefore, after the measurement, with equal probability the system will be in the state $|\uparrow\downarrow\rangle$ or $|\downarrow\uparrow\rangle$, that are both the solution for the MaxCut problem on the 2-vertices graph.

## 5.3 Implementing QAOA on the Quantum Computer

For implementation of the MaxCut problem solution on the quantum computer we used Python language and Qiskit, which is an open source SDK for working with quantum computers and composing quantum programs at the level of circuits and pulses. We were also using an IBM Quantum Lab for holding our experiments on the simulators and real quantum devices.

### 5.3.1 Gates

Recall that the state vector for the quantum computer is a system of qubits. To do some calculations one has to apply appropriate unitary operators called gates to the initial state. The final vector state is the result of the calculations. A sequence of quantum gates applied to the n-qubit register is called a quantum circuit.

To build a quantum circuit for the MaxCut problem we will use the following gates

**The Hadamard Gate**

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5.29}$$

Applied to a single qubit in a state $|0\rangle$ or $|1\rangle$ it creates a superposition of it.

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{5.30}$$

**The Pauli Gates**

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{5.31}$$

These gates correspond to the accordant Pauli matrices. Applied to a single qubit, they perform a rotation by $\pi$ around the corresponding axis.

$$\begin{aligned} X|0\rangle &= |1\rangle, \quad X|1\rangle = |0\rangle \\ Y|0\rangle &= i|1\rangle, \quad Y|1\rangle = -i|0\rangle \\ Z|0\rangle &= |0\rangle, \quad Z|1\rangle = -|1\rangle \end{aligned} \tag{5.32}$$

**The CNOT (XOR) Gate**

$$\begin{aligned} U_{CNOT}|00\rangle &= |00\rangle \\ U_{CNOT}|01\rangle &= |01\rangle \\ U_{CNOT}|10\rangle &= |11\rangle \\ U_{CNOT}|11\rangle &= |10\rangle \end{aligned} \tag{5.33}$$

(5.34)

Applied to the pair of qubits, this gate performs an X-gate on the second qubit (target), if the state of the first qubit (control) is $|1\rangle$.

**The $R_z(\theta)$ Gate**

$$R_z(\theta) = exp(-i\frac{\theta}{2}Z) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{5.35}$$

Applied to a single qubit, performs a rotation by angle $\theta$ around the z-axis.

**The $R_x(\theta)$ Gate**

$$R_x(\theta) = exp(-i\frac{\theta}{2}X) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{5.36}$$

Applied to a single qubit, performs a rotation by angle $\theta$ around the x-axis.

### 5.3.2 The Circuit

Now we can implement a quantum circuit to solve the MaxCut problem for a 2-vertices graph from section (5.2.3).

Recall that the cost operator we use looks as

$$U(C, \alpha) = e^{-i\alpha ZZ}, \tag{5.37}$$

where $Z$ is a gate corresponding to the $\sigma^z$. In section (5.3.1) we showed that the operator $e^{-i\alpha\hat{Z}}$ can be implemented in gates as the $R_z(2\alpha)$ gate. Equation (5.22) shows that

$$\begin{aligned} e^{-i\alpha ZZ}|00\rangle &= e^{-i\alpha}|00\rangle \\ e^{-i\alpha ZZ}|00\rangle &= e^{i\alpha}|01\rangle \\ e^{-i\alpha ZZ}|00\rangle &= e^{i\alpha}|10\rangle \\ e^{-i\alpha ZZ}|00\rangle &= e^{-i\alpha}|11\rangle \end{aligned} \tag{5.38}$$

Given that, one can derive

$$e^{-i\alpha ZZ}|ab\rangle = e^{-i(-1)^{a\oplus b}\alpha}|ab\rangle, \tag{5.39}$$

which means that the sign of the exponent depends on the parity of qubits. In the gates this can be implemented as consecutive application of CNOT, $R_z(2\alpha)$ and CNOT gates.

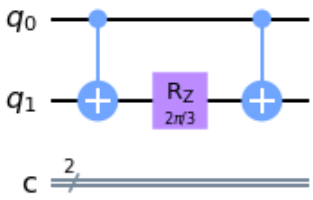


FIGURE 5.2: Cost operator gates for a 2-vertices graph

The mixer operator we use looks as follows

$$U(B, \beta) = -i\beta(X + X) \tag{5.40}$$

It can be easily implemented in gates just by applying the $R_x(2\alpha)$ to the each qubit

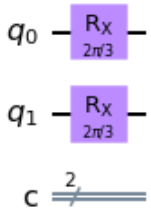


FIGURE 5.3: Mixer operator gates for the 2-vertices graph

Now one can implement the whole circuit. Recall that the initial state has to be a uniform superposition over all qubits. This can be achieved through applying the Hadamard gate to the each qubit. After applying all the gates, the measurements have to be done.

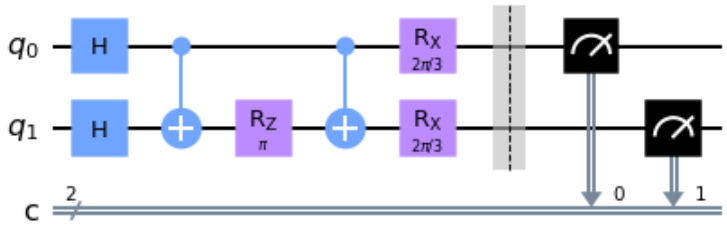


FIGURE 5.4: Quantum circuit for a 2-vertices graph

# Chapter 6

# Results

## 6.1 Quantum Simulator

The following experiments were done on QASM simulator, provided by Qiskit, that emulates execution of a quantum circuits on a real quantum computer and returns measurement counts.

### 6.1.1 2-vertices Graph

We now can run the circuit we constructed in the previous chapter for a 2-vertices graph. After 1024 iterations of running the circuit with the random parameters, we received the following distribution of states.

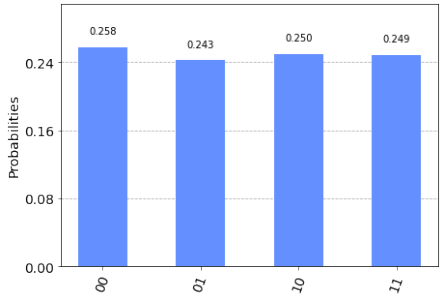


FIGURE 6.1: Distribution of states for the 2-vertices graph with random parameters

This is, obviously, not the correct solution to the problem. Recall that we now have to maximize the objective function, which is the Hamiltonian in case of MaxCut problem to get the optimal parameters. After that one can run the circuit with the optimal parameters.

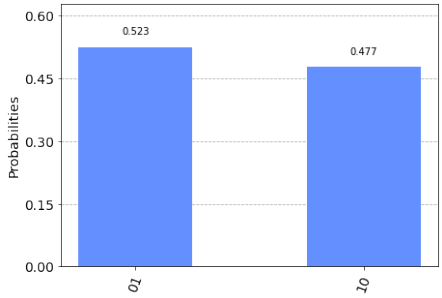


FIGURE 6.2: Distribution of states for the 2-vertices graph with optimal parameters

Figure 6.2 shows that running the circuit with optimal parameters gives either state 01 or 10 which are the exact solutions for the MaxCut problem for the given graph. These results match with those we received in (5.28), when calculating the probabilities manually.

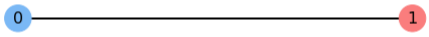


FIGURE 6.3: Maximum cut solution for the 2-vertices graph

### 6.1.2 5-vertices Graph

We are going now to construct the circuit for the following 5-vertices graph. The exact size of the maximum cut for this graph is 5.

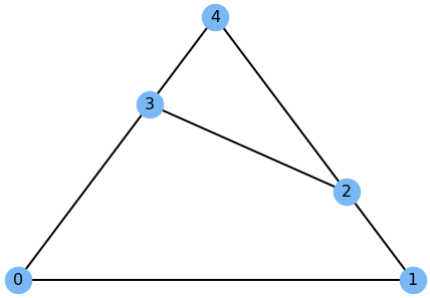


FIGURE 6.4: 5-vertices graph

The circuit for 5 qubits looks as follows

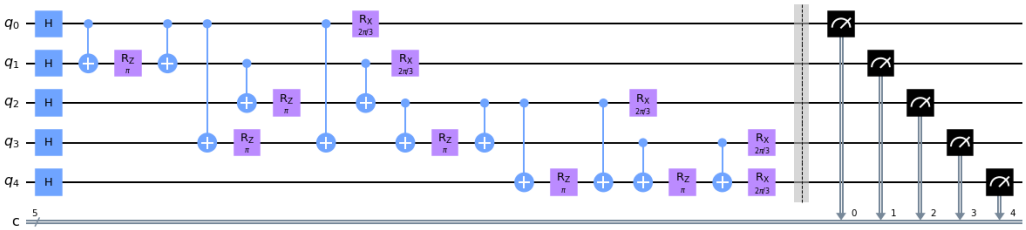


FIGURE 6.5: Quantum circuit for a 5-vertices graph

After maximization of the Hamiltonian, running the circuit with optimal parameters gives the distribution of states as on the Figure 6.6.

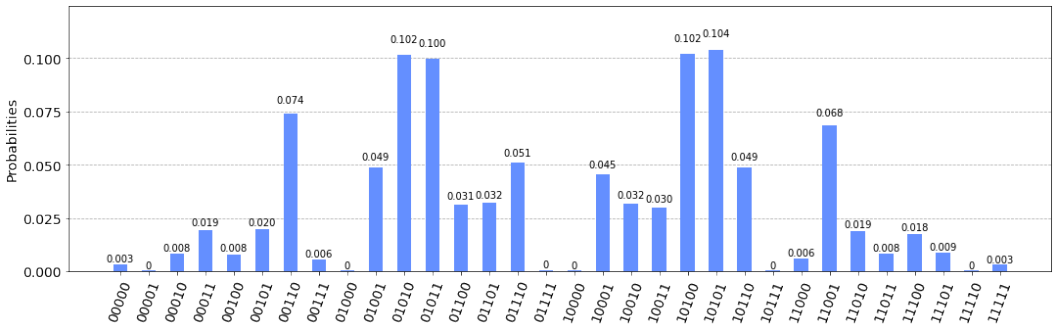


FIGURE 6.6: Distribution of states for the 5-vertices graph with optimal parameters

Given that, the maximum cut here is the cut for the most frequent state (10101).

$$cut_{max} = 5$$

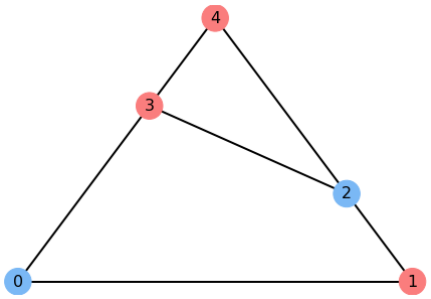


FIGURE 6.7: Maximum cut solution for the 5-vertices graph

We can also calculate the average cut size from all iterations. For this circuit

$$cut_{avg} = 4$$

Recall that in case of the 2-vertices graph, QAOA gave a perfect solution, where

$$cut_{max} = cut_{avg},$$

meaning that for all of the executions of the circuit the final state was the exact solution for the MaxCut problem. For the 5-vertices graph the picture is more mixed. The most frequent states are still those, that give the correct maximum cut value (5). However, there is also a relatively high probability to receive states 00110 and 11001, for which the maximum cut value is 4. It is easy to verify that the lower is a probability to receive a state — the lower is a cut size value for it.

### 6.1.3 Other Graphs

We held the experiments also for the 10-, 20- and 25-vertices graphs. The results are presented in the Table 6.1.

| Number of vertices | Exact max cut | QAOA max cut | QAOA average cut |
|---|---|---|---|
| 2 | 1 | 1 | 1 |
| 5 | 5 | 5 | 4 |
| 10 | 13 | 13 | 11 |
| 20 | 25 | 25 | 19 |
| 25 | 32 | 32 | 24 |

TABLE 6.1: QAOA results for diferent graphs

For all of the graphs that we tested, we received a correct maximum cut value. However it can be noticed, that the larger is graph — the larger is the difference between the maximum cut size value and the average cut size from all executions of the circuit. This means that for the really large graphs, there will be a high probability that the final state, received after the measurement, won't be the exact solution for the MaxCut problem.

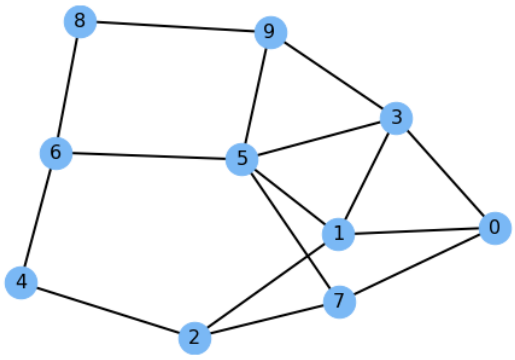


FIGURE 6.8: 10-vertices graph

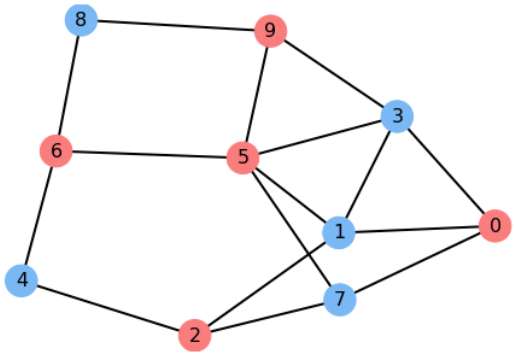


FIGURE 6.9: Maximum cut solution for the 10-vertices graph

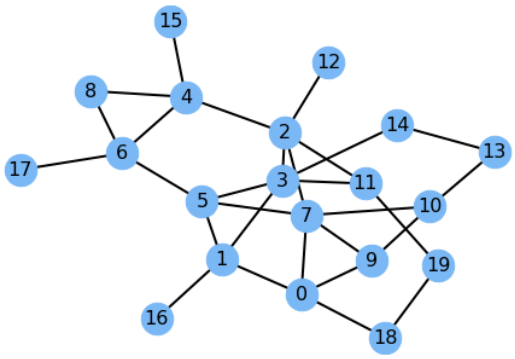


FIGURE 6.10: 20-vertices graph

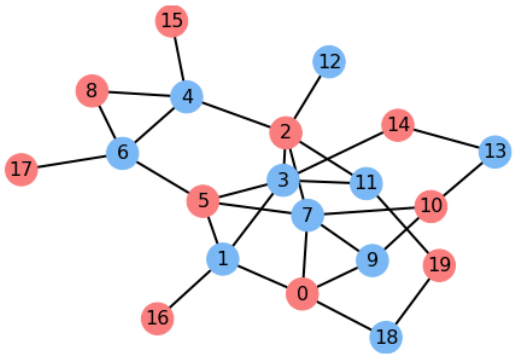


FIGURE 6.11: Maximum cut solution for the 20-vertices graph

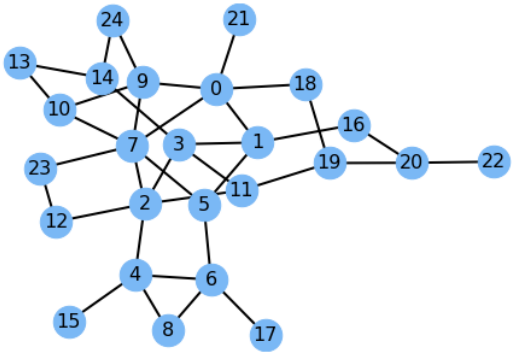


FIGURE 6.12: 25-vertices graph

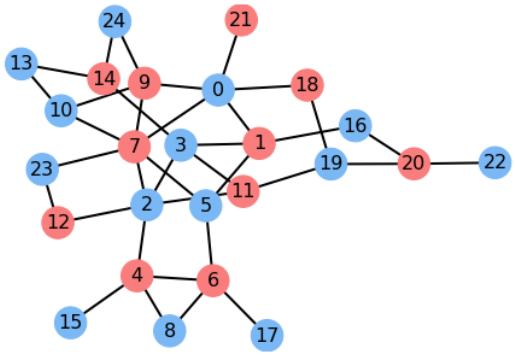


FIGURE 6.13: Maximum cut solution for the 25-vertices graph

## 6.2 Real Quantum Computer

Recall the circuit we constructed for the 5-vertices graph. After minimization we executed it with the optimal parameters on the real 5-qubit quantum device available on IBM. After 8192 execution iterations we received the following distribution of states

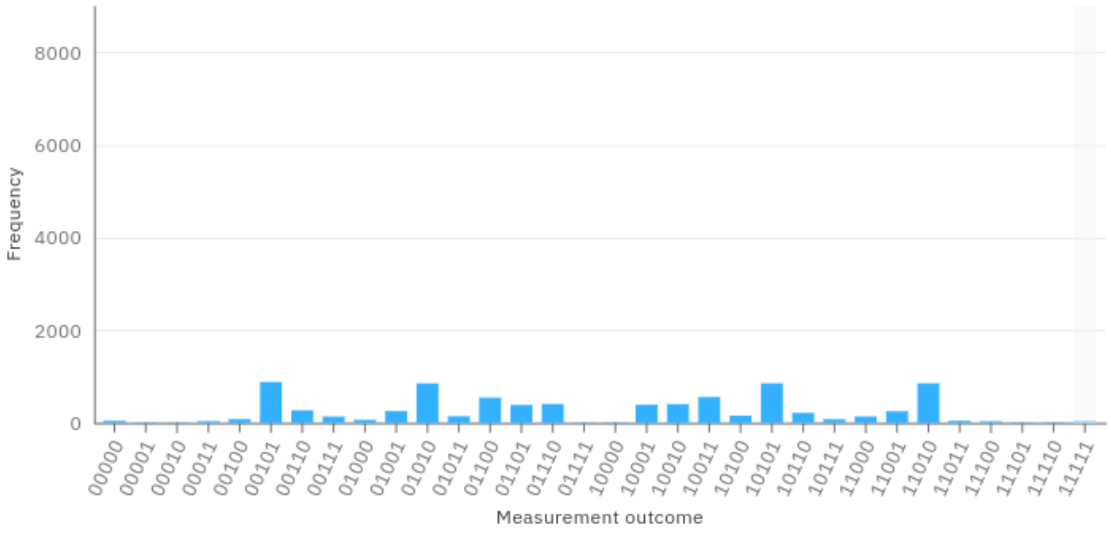


FIGURE 6.14: Distribution of states for the 5-vertices graph with optimal parameters on real quantum computer

Note that Qiskit uses an ordering where zeroth qubit is the rightmost (the least significant bit in the string). With this remark, we can compare these results with the states distribution from the simulator (Figure 6.6). It is straightforward to verify that the most frequent states are the same for both cases and they are the exact solutions to the MaxCut problem.

# Chapter 7

# Conclusion

This thesis has investigated the Quantum Approximation Optimization Algorithm for solving the unweighted 2-MaxCut problem. We showed in detail how it theoretically works on the 2-vertices graph. To verify the theory experimentally, we implemented a quantum circuit for this graph and executed it on the quantum computer simulator. The results received, coincide with our calculations. In this work we also held the experiments for the 5- 10- 20- and 25-vertices graphs. For all the graphs, the maximum cut size received after running the QAOA is the exact solution for the MaxCut problem. However, our experiments showed that the larger is the graph — the higher is a probability that the final state, received after the measurement, won't be the best solution for the MaxCut problem. Nevertheless, the probability distribution for the states demonstrated that, most likely, the maximum cut size obtained, will be still close to the best one. This is a satisfactory result for an approximation algorithm, proving that QAOA has great potential in solving the MaxCut problem.

This thesis was limited in several ways. Firstly, we have only examined how QAOA can be used for solving the unweighted 2-MaxCut problem. However, this approach also fits for the n-MaxCut and weighted graphs. Secondly, we used only one set of parameters $\alpha$ and $\beta$, while in the original paper, Farhi and Goldstone propose to use $p$ sets of parameters and construct the trial state as follows

$$|\psi\rangle = U(B, \beta_p)U(C, \alpha_p)...U(B, \beta_1))U(C, \alpha_1)|+_n\rangle \tag{7.1}$$

They claim that the quality of the approximation improves as $p$ is increased, so this may become a great modification to the approach, we implemented in this work.

There is also a large space for other future studies on the current topic. One can change or add new operators for constructing the trial state and investigate whether this will lead to better results. These researches should focus on the simplifying the objective function we maximize. Also it can be investigated how different maximization algorithms work with QAOA and how to choose the initial point so that to receive the best optimal parameters. Besides, one can examine how QAOA works on really large graphs. Finally, it is worth highlighting that Quantum Approximation Optimization Algorithm fits for solving not only MaxCut, but also a lot of other combinatorial optimization problems, including various NP-hard problems. Quantum computing is a fast-growing area, where a lot of researches are still to be done and we hope that, among other things, they may help us to see the well-known problems from the new perspective.

# Appendix A

# Implementation

The Qiskit implementation for the QAOA can be found here:
https://github.com/SophiaZhyrovetska/qaoa-maxcut

# Bibliography


[1] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019), pp. 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5.

[2] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22 (1980), pp. 563–591. DOI: 10.1007/BF01011339.

[3] David Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". In: 400 (1985), pp. 97–117.

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

[5] Uriel Feige, Marek Karpinski, and Michael Langberg. "Improved approximation of MAX-CUT on graphs of bounded degree". In: *Journal of Algorithms* 43 (May 2002), pp. 201–219. DOI: 10.1016/S0196-6774(02)00005-6.

[6] Michel X. Goemans and David P. Williamson. "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming". In: *J. ACM* 42.6 (Nov. 1995), 1115–1145. ISSN: 0004-5411. DOI: 10.1145/227683.227684.

[7] Steven Homer and Marcus Peinado. "Design and Performance of Parallel and Distributed Approximation Algorithms for Maxcut". In: *Journal of Parallel and Distributed Computing* 46.1 (1997), pp. 48–61. ISSN: 0743-7315. DOI: 10.1006/jpdc.1997.1381.

[8] Y. KAJITANI. "On via hole minimization of routing on a 2-layer board". In: *Proceeding of the IEEE International Conference on Circuits and Computer ICCC 80* (1980), pp. 295–298.

[9] Richard M Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*. Ed. by Raymond E Miller, James W Thatcher, and Jean D Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9.

[10] Su-Hyang Kim, Yong-Hyuk Kim, and Byung-Ro Moon. "A Hybrid Genetic Algorithm for the MAX CUT Problem". In: *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. GECCO'01. San Francisco, California: Morgan Kaufmann Publishers Inc., 2001, 416–423. ISBN: 1558607749.

[11] Yong-Hyuk Kim, Yourim Yoon, and Zong Woo Geem. "A comparison study of harmony search and genetic algorithm for the max-cut problem". In: *Swarm and Evolutionary Computation* 44 (2019), pp. 130–135. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2018.01.004.

[12] Fenella McAndrew. "Adiabatic Quantum Computing to solve the MaxCut graph problem". MA thesis. The University of Melbourne, 2020.

[13] Christos H Papadimitriou and Mihalis Yannakakis. "Optimization, approximation, and complexity classes". In: *Journal of Computer and System Sciences* 43.3 (1991), pp. 425–440. ISSN: 0022-0000. DOI: 10.1016/0022-0000(91)90023-X.

[14] Sartaj Sahni and Teofilo Gonzalez. "P-Complete Approximation Problems". In: *J. ACM* 23.3 (July 1976), 555–565. ISSN: 0004-5411. DOI: 10.1145/321958.321975.

[15] Han-Sen Zhong et al. "Quantum computational advantage using photons". In: *Science* (2020). ISSN: 0036-8075. DOI: 10.1126/science.abe8770.