Course Scheduling Using Quantum Computing

Mohammed Abdulsami   Moustafa Salama   Omar Abdelrasoul
Youssef AbdelWahab   Yusuf Alsawah

UG Project
Supervised by: Dr. Ahmed ElMahdy, Dr. Marwa Sorour

June 2024

Table of Contents

Table of Contents

Course Scheduling Problem

The course scheduling problem is about creating the best schedule for teachers and classes based on their available times slots.which makes it hard to find a good solution Because of its complexity.

Problem Complexity

A Problem Complexity is categorized by the required time to solve it.

Complexity Classes:

- P class contains problems that can be solved in polynomial time
  - Array sorting is in P class
- NP class contains problems that can be solved in nondeterministic polynomial time
  - Time–Table scheduling problem is in NP class

Problem Complexity

The course scheduling problem is a combinatorial optimization problem. This means we need to find the best schedule by minimizing the objective function.

Course Scheduling Problem

let us solve a simple course scheduling problem. For example, we have 2 Instructors and 2 classes also 2 available time slots. How we can present our problem?

Course Scheduling Problem

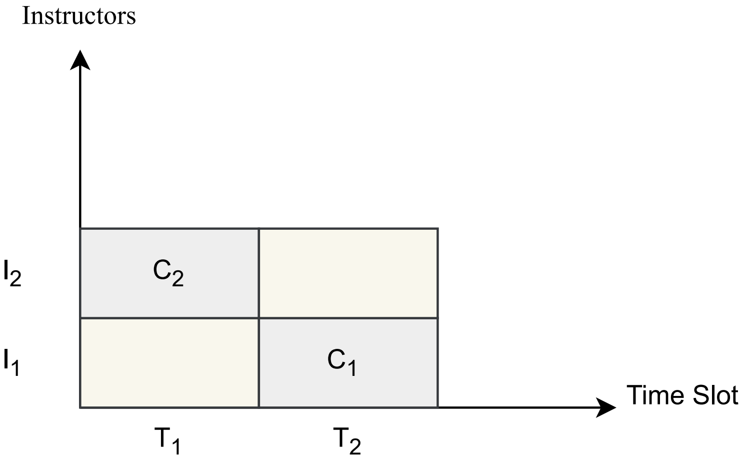


Figure: Time Table

Course Scheduling Problem

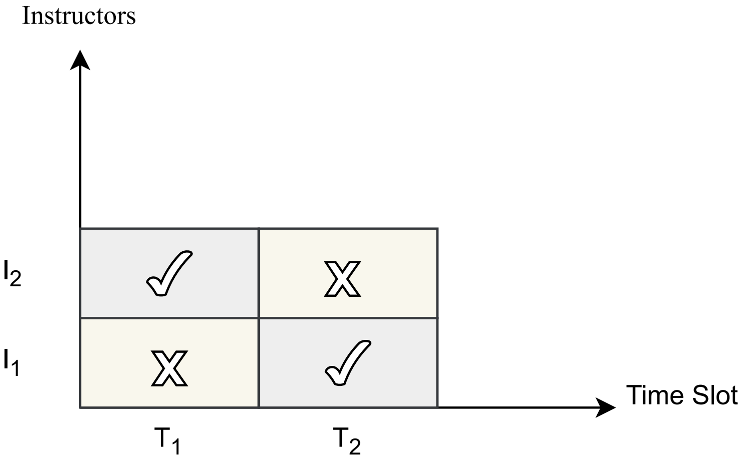


Figure: Instructor Available Time

Course Scheduling Problem

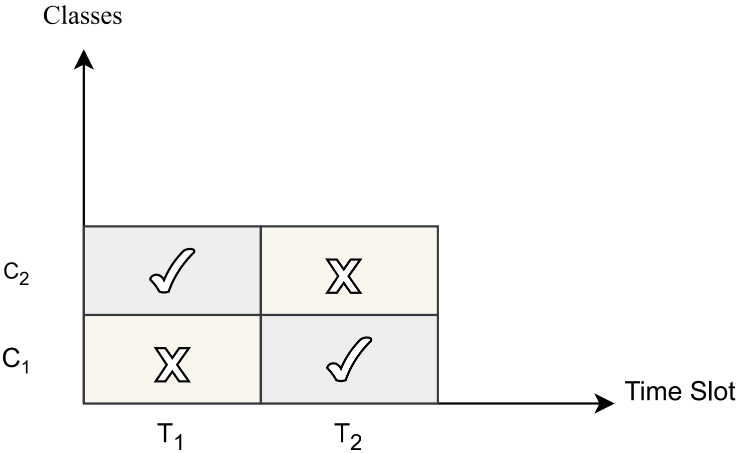


Figure: Classes Available Time

Table of Contents

Problem Formulation

- N: Number of instructors
- M: Number of classes
- T: is a set consisting of $\{T_1, T_2, ...T_N\}$, where T_i is a binary string containing the available hours for teaching for an instructor i.
- C: is a set consisting of $\{C_1, C_2, ...C_M\}$, where C_j is a binary string containing the available hours for studying for a class j.
- R: Matrix of $M * N$ size where R_{ij} denotes the number of hours that instructor i is required to teach class j

Problem Formulation

Our problem will be:

- N: 2
- M: 2
- T: $\{10\ ,\ 01\}$
- C: $\{10\ ,\ 01\}$
- $R : \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Problem Formulation

- $f(i,j,h) = 1 \rightarrow h \in T_i \cap C_j$
  - The function will yield 1 if both instructor i and class j are available in h hour.
- $\sum_{h \in H} f(i,j,h) = R_{ij}$, for all $1 \leq i \leq N$ and $1 \leq j \leq M$
  - Assures that instructor i teaches class j the required number of hours during the week.

Problem Formulation

- $\sum_{i=1}^{N} f(i, j, h) \leq 1$, for all $h \in H$ and $1 \leq j \leq M$
  - Assures that each j class does not have multiple instructors simultaneously.
- $\sum_{j=1}^{M} f(i, j, h) \leq 1$, for all $1 \leq i \leq N$ and $h \in H$
  - Assures that each i instructor does not teach multiple classes simultaneously.

Table of Contents

Quadratic Unconstrained Binary Optimization

To run the previous formulation on a quantum computer we need to convert the previous mathematical problem to Quadratic Unconstrained Binary Optimization (QUBO) problem.

Quadratic Unconstrained Binary Optimization

QUBO stands for:

- Quadratic
  - Our equations are from the second degree (ie. x^2 or $x \cdot y$).
- Unconstrained
  - We will introduce slack terms to turn inequalities to equalities and do not forget to square it to avoid negative values.
- Binary
  - We use binary numbers only we can not use decimal numbers
- Optimization
  - We are trying to find the best solution either by minimizing or maximizing.

QUBO Problem Formulation

$$\sum_{h \in H} f(i,j,h) = R_{ij} \text{ , for all } 1 \le i \le N \text{ and } 1 \le j \le M \tag{1}$$

$$\downarrow$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \left(\sum_{h \in H} f(i,j,h) - R_{ij} \right)^2 \tag{2}$$

Check that every instructor i taught the required hours for class j class

QUBO Problem Formulation

$$\sum_{i=1}^{N} f(i,j,h) \leq 1 \text{ , for all } h \in H \text{ and } 1 \leq j \leq M \tag{3}$$

$$\downarrow$$

$$\sum_{h \in H} \sum_{j=1}^{M} \left(\sum_{i=1}^{N} f(i,j,h) + \tau_{jh} - 1 \right)^2 \tag{4}$$

Check that every C_j class will be available at h hour

QUBO Problem Formulation

$$\sum_{j=1}^{M} f(i,j,h) \leq 1 \text{ , for all } 1 \leq i \leq N \text{ and } h \in H \tag{5}$$

$$\downarrow$$

$$\sum_{h\in H} \sum_{i=1}^{N} \left(\sum_{j=1}^{M} f(i,j,h) + \lambda_{ih} - 1 \right)^2 \tag{6}$$

Check that every T_i teacher will be available at h hour

QUBO Formulation

$$
\begin{aligned}
\min \sum_{h\in H}\sum_{i=1}^{N}\left(\sum_{j=1}^{M} f(i,j,h) + \lambda_{ih} - 1\right)^2 + \\
\sum_{h\in H}\sum_{j=1}^{M}\left(\sum_{i=1}^{N} f(i,j,h) + \tau_{jh} - 1\right)^2 + \\
\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{h\in H} f(i,j,h) - R_{ij}\right)^2
\end{aligned}
\tag{7}
$$

Table of Contents

Quantum Computing Types

There are two main types of Quantum Computers:

- Adiabatic Quantum Computer
- Gate-based Quantum Computer

Adiabatic Quantum Computer

Adiabatic Quantum Computer works based on the quantum adiabatic theorem:

Theorem

If a quantum system starts in a ground state, so long as we evolve the state slowly, it is likely to remain in a ground state.

Table of Contents

Ground-state

The ground state of a quantum-mechanical system is its stationary state of lowest energy; the energy of the ground state is known as the zero-point energy of the system.

Ground-state

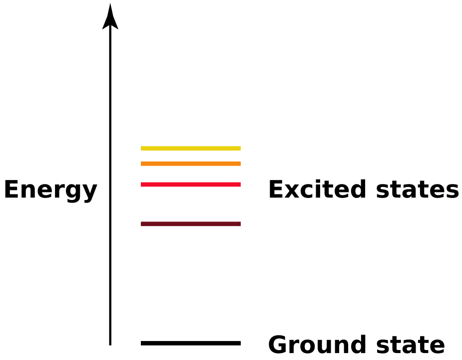
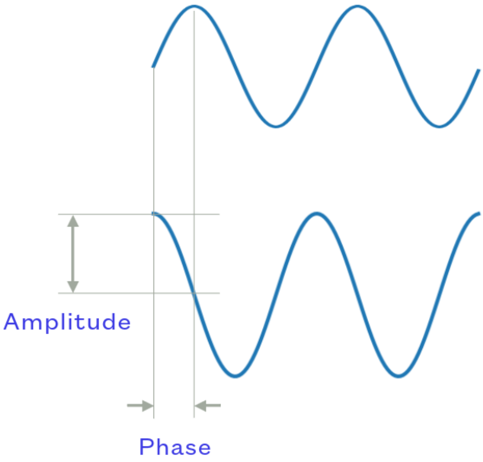


Figure: Energy Levels

Wave Function



The Wave: Complex Number

- $C = ae^{i\phi}$
- a is the amplitude
- ϕ is the phase shift

Wave Function

If we have a plane and a wave traveling through the plane. When observing point a and b at anytime:

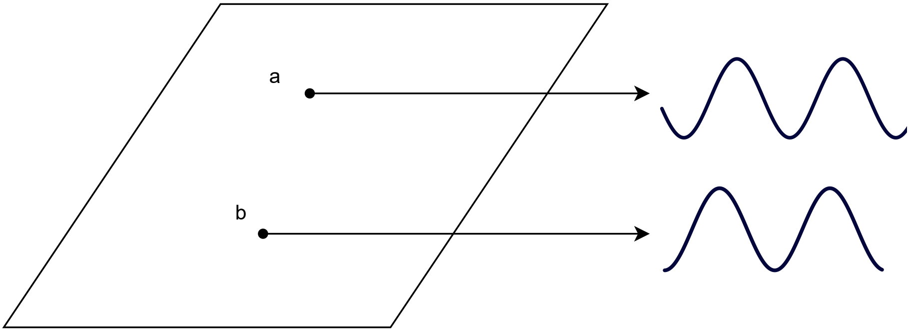


Figure: Normal Plane

Wave Function

What if the wave was time-independent? This is the stationary wave. It changes depending on the space, not the time.

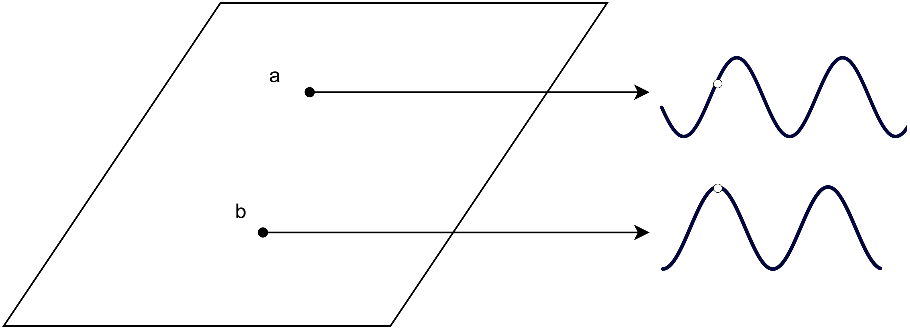


Figure: Stationary Wave Plane

Hamiltonian

Simply, the Schrödinger equation is based on Hamiltonian.

$$|\Psi_t\rangle = e^{-iHt}\,|\Psi_0\rangle$$

Hamiltonian is an operator that describes the KE and PE of a system. It operates on the input state and is a key operator in the Schrödinger equation.

Schrödinger Equation

Time-independent equation:

$$H\left|x\right\rangle = f(x)\left|x\right\rangle$$

By solving this equation we get eigenstates. Eigenstates are subsets of possible classical output and they are stationary waves. As we can see the state does not change because this equation is time dependent. Also by solving f eigenvectors that have the smallest eigenvalue.

Schrödinger Equation

Time-dependent equation:

$$|\Psi_t\rangle = e^{-iHt} |\Psi_0\rangle$$

- $|\Psi_t\rangle$ is the state at t time.
- i is the imaginary number.
- t is the time.
- $\hat{H}$ is the Hamiltonian.

Schrödinger Equation

Time-dependent equation:

$$|\Psi_t\rangle = e^{-iHt}|\Psi_0\rangle$$

This equation evolves the state over time, unlike the previous equation. We use the following table to convert terms to the corresponding Hamiltonian.

Hamiltonian

$f(x)$	H_f
x	$\frac{1}{2}I - \frac{1}{2}Z$
$\overline{x}$	$\frac{1}{2}I - \frac{1}{2}Z$
$x_1 \bigoplus x_2$	$\frac{1}{2}I - \frac{1}{2}Z$
$x_1 \wedge x_2$	$\frac{1}{4}I - \frac{1}{4}(Z_1 + Z_2 - Z_1 Z_2)$

Table: Hamiltonians Representing Basic Boolean Clauses

Adiabatic theorem

Adiabatic theorem:

- We start with initial Hamiltonian, H_i, whose ground-state is easy to prepare
- We evolve our H_i slowly until we reach the H_f
- The final Hamiltonian, H_f, whose ground-state is the solution for our problem.

Table of Contents

Quantum Approximate Optimization Algorithm

Quantum Approximate Optimization Algorithm (QAOA) relies on trotterization to find approximate solutions. Our Hamiltonian:

$$H = H_i + H_p$$

- H_i: Initial Hamiltonian
- H_p: Problem Hamiltonian

Quantum Approximate Optimization Algorithm

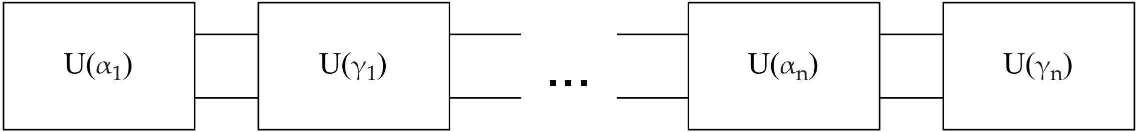


Figure: Trotterization

Quantum Approximate Optimization Algorithm

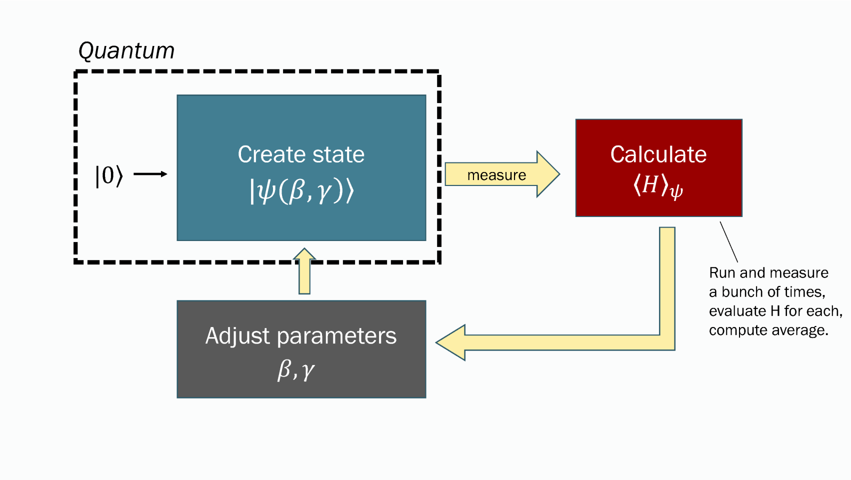


Figure: QAOA Overview (NC State University)

Hamiltonian construction

Conversion steps:

- We expand an equation
- We select the appropriate role from the table
- We raise this role to e^{-iH}
- Simplify the equation
- Finally find corresponding gates.

Hamiltonian construction

Let's expand equation 2:

$$\sum_{i=1}^{2} \sum_{j=1}^{2} \left(\sum_{h \in H} f(i, j, h) - R_{ij} \right)^2$$

Hamiltonian construction

f equation after expansion:

$$f(1,1,1)^2 + 2f(1,1,1)f(1,1,2) - 2f(1,1,1) + f(1,1,2)^2 + ...$$

- 1 is a trivial term
- other terms are non-trivial terms

When we convert trivial they result in an identity matrix that does not affect the output so it is better to discard them.

Hamiltonian construction

Let's take $-2f(1,1,1)$:

- Coefficient: -2
- Used role: $x = \frac{1}{2}I - \frac{1}{2}Z$

Conversion:

$$-2f(1,1,1) \rightarrow \frac{1}{2}I - \frac{-2}{2}Z \rightarrow \frac{1}{2}I + Z \rightarrow Z$$

Hamiltonian conversion

Let's take $-2f(1,1,1)$:

- Coefficient: -2
- Used role: $x = \frac{1}{2}I - \frac{1}{2}Z$
- Rz(λ) gate: $e^{-i\frac{\lambda}{2}Z}$

Conversion:

$$Z \rightarrow e^{-iZ} \rightarrow Rz(-2)$$

Generated circuit

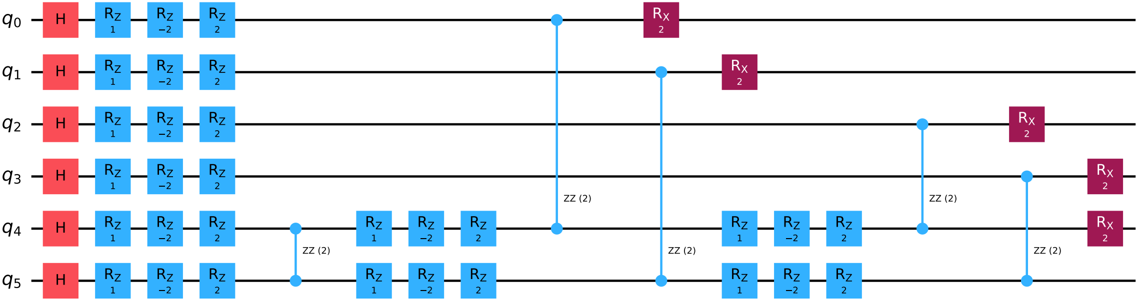


Figure: Simple circuit

Table of Contents

System Overview

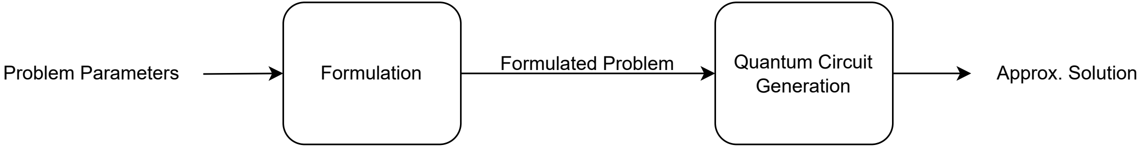


Figure: System Overview

System Overview

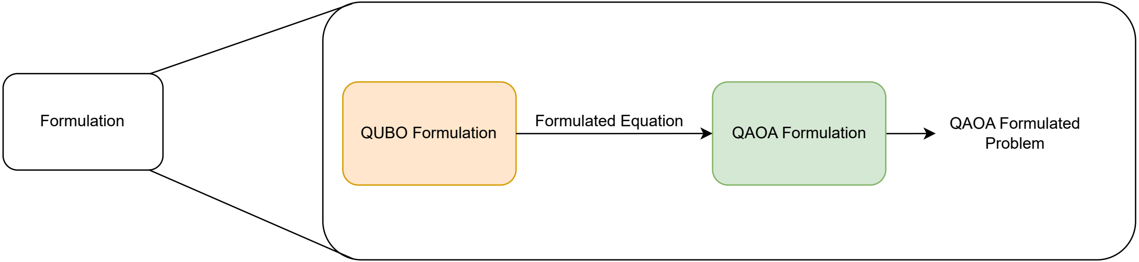


Figure: Formulation Component

System Overview

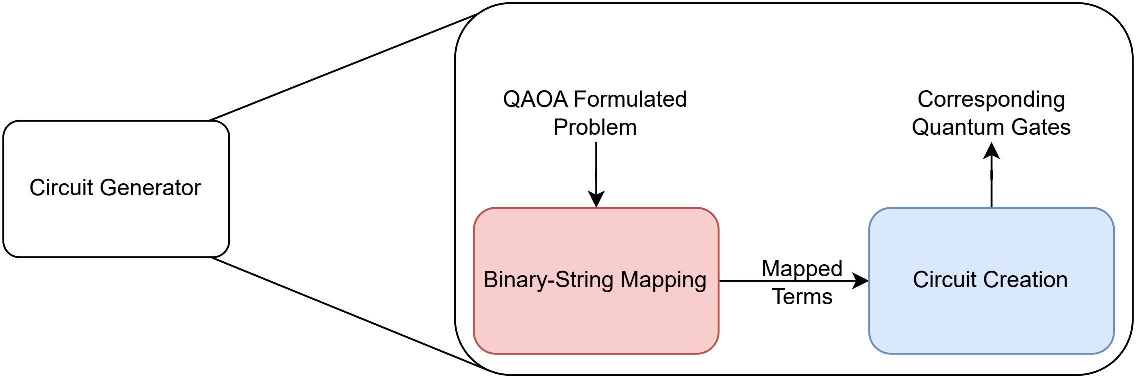


Figure: Circuit Generator Copmponet

Hamiltonian conversion

In the previous slide, we discussed how we map variables to the corresponding quantum gate. It is a straightforward process but it becomes infeasible with a large number of terms.

For Example:

- 3 Qubits will result in 15 terms.
- 6 Qubits will result in 30 terms.
- 175 Qubits will result in 1925 terms.
- 1440 Qubits will result in 27820 terms.

Hamiltonian conversion

The number of terms to be converted to Quantum gate increases rapidly, so converting them manually would be an error-prone solution in addition to its inconvenience.

Circuit generator

As a solution, we created a simple circuit generator that facilitates the conversion process.

Circuit generator

- Takes the circuit configuration as a JSON file.
- Parses the input file.
- Output the result circuit.

Table of Contents

Experiment configuration

These are the used configurations to produce these results.

- Random seed = 10
- Shots = 10240
- p-layers = 5

Result decoding

When we run a quantum circuit it outputs the result as a bit-string it may be gibberish until you understand it.

Result decoding

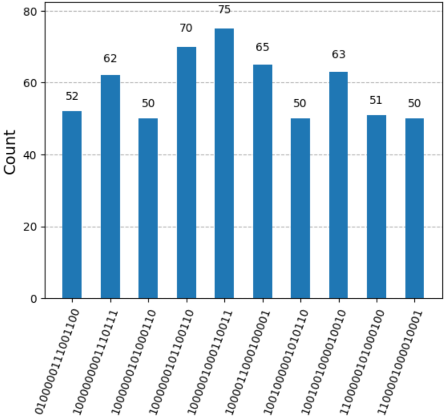


Figure: 16-qubit circuit results

Result decoding

$$\underbrace{\boxed{1}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{1}}_{f(i,j,h)}\underbrace{\boxed{0}\boxed{1}}_{\lambda_2}\underbrace{\boxed{1}\boxed{0}}_{\lambda_1}\underbrace{\boxed{0}\boxed{1}}_{\tau_2}\underbrace{\boxed{1}\boxed{0}}_{\tau_1}$$

Figure: Bit-string

Result decoding

$$\underbrace{\boxed{1}\boxed{0}\boxed{0}\boxed{0}}_{f(2,j,h)}\underbrace{\boxed{0}\boxed{0}\boxed{0}\boxed{1}}_{f(1,j,h)}\underbrace{\boxed{0}\boxed{1}}_{\lambda_2}\underbrace{\boxed{1}\boxed{0}}_{\lambda_1}\underbrace{\boxed{0}\boxed{1}}_{\tau_2}\underbrace{\boxed{1}\boxed{0}}_{\tau_1}$$

Figure: Bit-string

Result decoding

$$\overbrace{1\;0}^{f(2,2,h)}\underbrace{0\;0}_{f(2,1,h)}\overbrace{0\;0}^{f(1,2,h)}\underbrace{0\;1}_{f(1,1,h)}\underbrace{0\;1}_{\lambda_2}\underbrace{1\;0}_{\lambda_1}\underbrace{0\;1}_{\tau_2}\underbrace{1\;0}_{\tau_1}$$

Figure: Bit-string

Result

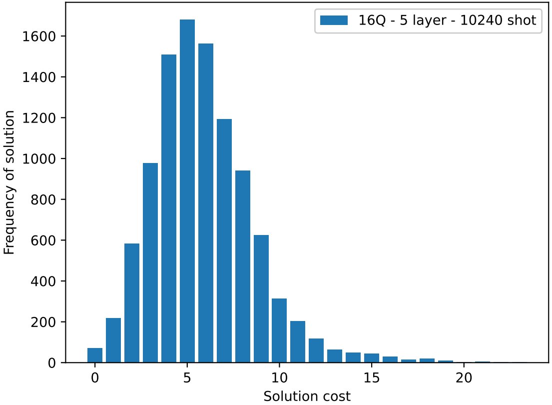


Figure: 16-Qubit Solution Distribution

Result

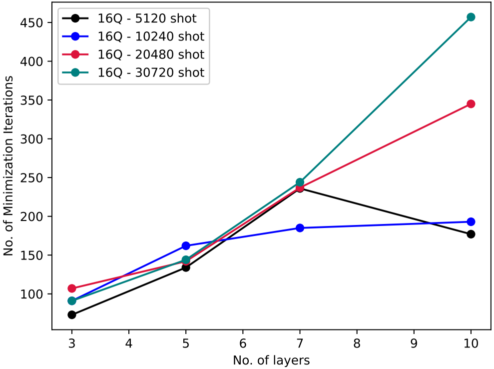


Figure: Effect of changing the no of layers on the no of minimization steps for 16Q

Result

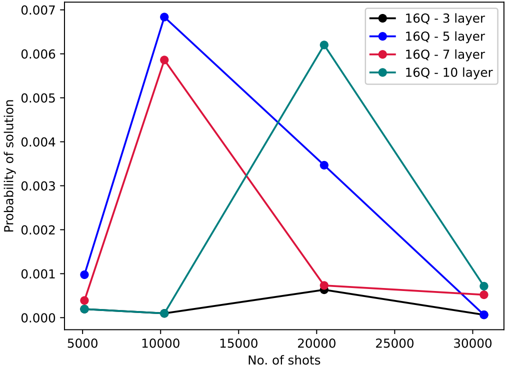


Figure: 16-Qubit No. Shots and Solution Probability

Result

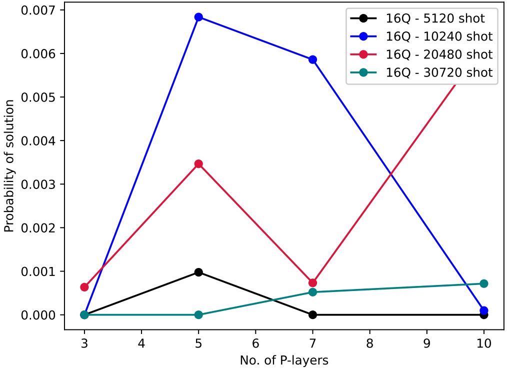


Figure: 16-Qubit No. p-layers and Solution Probability

Current Work

- Currently, experimenting with the formulation of the Flexible Open Shop Scheduling Problem to see if it produces better results.
- Waiting for feedback on our paper "On the Practicality of Restricted Time-Table Problem Solution Using Quantum Approximate", which was submitted to IEEE Quantum Week. The results should return by the 15^{th} of July.

Future Work

- Better initialize the problem variables to make the classical optimizer less time-consuming.
- Investigate grouping, to minimize/optimize the number of qubits needed for the circuit.
- Revisit the mathematical formulation of our Hamiltonians.
- Running more simulations on real Quantum Computers

Thank you!

References I

- D. Reppy, "Quantum Computing Lecture Notes," University of Cambridge, [Online]. Available: `https://www.cl.cam.ac.uk/teaching/2223/QuantComp/materials.html`. [Accessed: Jun. 3, 2024].
- IBM, "IBM Quantum Experience," IBM, [Online]. Available: `https://learning.quantum.ibm.com/`. [Accessed: Jun. 3, 2024].
- IBM, "Qiskit: An open-source quantum computing software development framework," IBM, [Online]. Available: `https://www.ibm.com/quantum/qiskit`. [Accessed: Jun. 3, 2024].

References II

- S. Hadfield, ‘On the Representation of Boolean and Real Functions as Hamiltonians for Quantum Computing’, ACM Transactions on Quantum Computing, vol. 2, no. 4, p. 3, Dec. 2021.
- QAOA Combinatorial Optimization, ECE 592/CSC 591, NC State University, Fall 2019