ENGINEERS WITH
SOCIAL RESPONSIBILITY

Quantum Search for Graph Colouring

CS 405

Quantum Computation

Nisarg Suthar, Manjal Shah, Vandan Patel

(202003030 202003037 202001023)

May 20, 2023

Table of Contents

The Vertex Colouring Problem

- Given k colours and an n-vertex graph, can we colour every node such that no two adjacent nodes have the same colour?
- This is similar to searching for a solution from a list of possible configurations.
- The total possible configurations are k^n.

2-Colouring

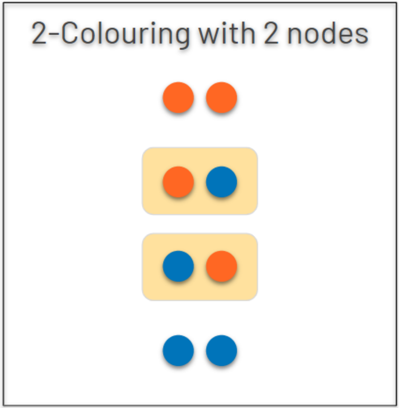

Figure: 2-Colouring Problem

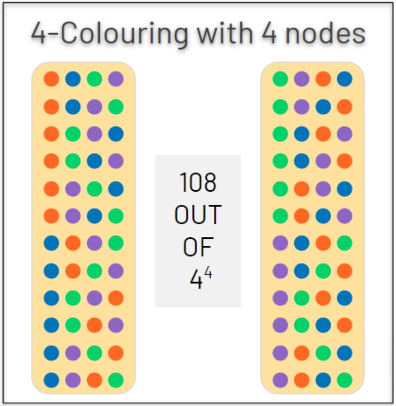

Figure: 4-Colouring Problem

Table of Contents

Introduction

- Quantum way for searching an unstructured database.
- Can provide a quadratic speedup over classical algorithms
- Main Idea:
  Amplitude amplification to increase the probability of measuring the solution state

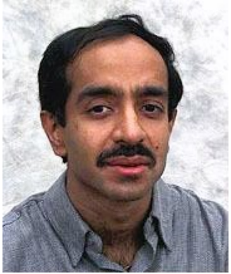

Figure: **Lov Grover**, an Indian-American Computer Scientist who created the Grover Database Search Algo in Quantum Computing. [Source]

1. Initialise a circuit with n qubits. $(|\Psi_0\rangle = |0\rangle^{\otimes^n})$
2. Apply the Hadamard gate to all qubits, which creates a uniform superposition $|\Psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |x\rangle$
3. Apply the Oracle (O_f) to mark the solution states
4. Apply the Diffuser operator $(D = H^{\otimes^n} O_d H^{\otimes^n})$ to reflect about $|\Psi_1\rangle$
5. Repeat steps 3 & 4 $P_{optimal}$ times.

$$P_{optimal} \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$$

$$|\Psi_{final}\rangle = (DO_f)^{P_{optimal}} H^{\otimes^n} |\Psi_0\rangle$$

Table of Contents

3 Qiskit API

Qiskit API

mct

- The Multi-Control Toffoli (MCT) gate can be implemented using this function
- Arguments:
  1. List of control qubits
  2. Target qubit

AmplificationProblem

- Class that implements amplitude amplification.
- Parameters:
  1. Circuit representing the oracle
  2. Circuit preparing the initial state
  3. Function to check state goodness
  4. Indices of the objective qubits

Grover

- Grover's algorithm can be implemented using this class
- Arguments:
  1. Number of iterations
  2. Optional parameter specifying the backend
- Method: amplify
  1. takes the problem as argument
  2. returns the counts of measurements

Table of Contents

Problem Setup

4 Grover for 2-Colouring

- Since there are just 2-colours we can represent each colour using a single qubit.
- When a qubit is 0, that country is colored white.
- When a qubit is 1, that country is colored black.
- This means that the correct solutions correspond to 01 (white and black) and 10 (black and white) where the incorrect solutions correspond to 00 (white and white) and 11 (black and black).

Oracle Creation

- We need to define a circuit that will flip the phase of correct solutions and leave the rest alone. In this particular case, that means:
  - If the two qubits disagree (01 or 10), we should flip the phase.
  - Otherwise (00 or 11), leave the phase as it is.
- To implement this, we will use two extra qubits:
  - 1 qubit that says there's a valid solution (01 or 10) by flipping to 1.
  - The "output" bit: 1 qubit that marks the state by flipping the phase, specifically changing from $|-\rangle$ to $-|-\rangle$.

Final Circuit

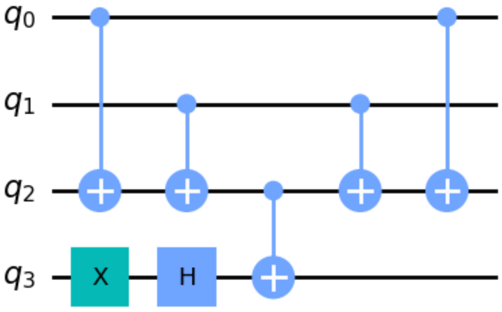

Figure: Oracle circuit

Generalisation

- We have created generalized functions for solving any problems that can be solved using 2-colouring. They take the following inputs and create the circuit for the initial state and oracle:
  1. variable qubits (list of qubits which represents different countries)
  2. check qubits (list of qubits used for checking valid solutions)
  3. output qubit (1 qubit that marks the state by flipping the phase)
  4. disagree list (list of countries which needs to be coloured differently)
- After defining the problem, we will run Grover's algorithm for some number of iterations and plot the results.

Example

Country 0	Country 1	Country 2

- Countries 0 and 1 must be coloured differently.
- Countries 1 and 2 must be coloured differently.
- The correct solutions are: 010 and 101

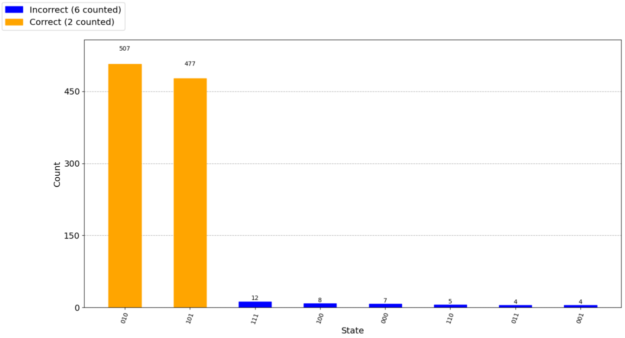

Figure: Histogram

Table of Contents

Representation

5 Grover for 4-Colouring

- To represent 4 possible colours per Country, we need to use 2 qubits. So now, each Country can be in any of these four states. For example:
  1. 00: White
  2. 01: Red
  3. 10: Orange
  4. 11: Black
- To ensure that the two countries are coloured differently, we will compare each digit and ensure at least one difference.
- This approach will require ancilla (extra) qubits.

First Approach

- Just like 2-colouring case, we need to define a circuit that will flip the phase of correct solutions and leave the rest alone. In this particular case, that means:
  - If at least one pair of two qubits disagree (00.01 or 01.10), we should flip the phase.
  - Otherwise (00.00 or 01.01), leave the phase as it is.
- To implement this, we will use three extra qubits:
  - 2 qubits that says there's a valid solution pair (01 or 10) by flipping to 1.
  - The "output" bit: 1 qubit that marks the state by flipping the phase, specifically changing from $|-\rangle$ to $-|-\rangle$.

First Approach Circuit

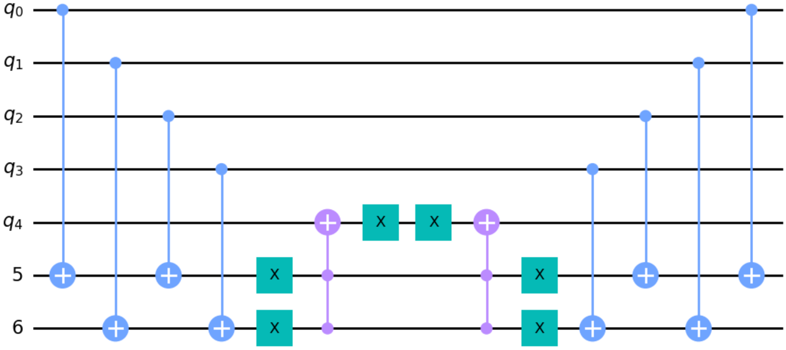

Figure: Oracle circuit

Optimisation

5 Grover for 4-Colouring

- we will optimize the oracle here is to eliminate the need for ancilla qubits as they effectively double the size and complexity of the problem.
- For encoding the idea of pairs of qubits disagreeing, there are only ever 4 cases where all 4 qubits agree with each other in case of 4 coluring.
    1. 11 11
    2. 10 10
    3. 01 01
    4. 00 00
- All (12) other states are acceptable! So, we've been using general logic to eliminate 4/16 or just 1/4 of the possibilities. Instead, let us specifically target these 4 states and check if the qubits are in one of these 4 states or not.

Optimized Circuit

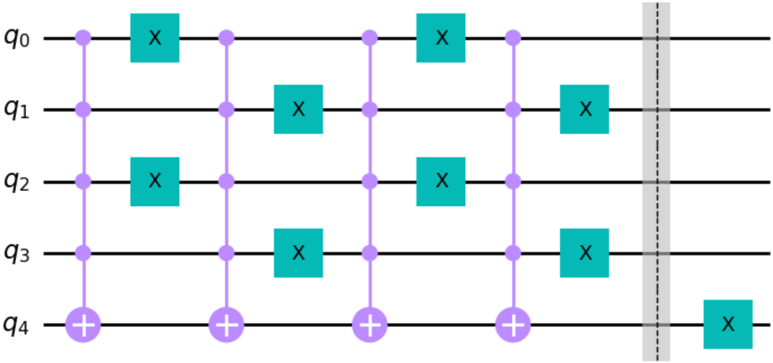

Figure: Oracle circuit

Generalisation

- We have created generalized functions for solving any problems that can be solved using 4-colouring. They take the following inputs and create the circuit for the initial state and oracle:
  1. variable qubits (list of qubits which represents different countries)
  2. check qubits (list of qubits used for checking valid solutions)
  3. output qubit (1 qubit that marks the state by flipping the phase)
  4. disagree list (list of countries which needs to be coloured differently)
- After defining the problem, we will run Grover's algorithm for some number of iterations and plot the results.

Example

<table>
<tr><td colspan="4" align="center">Country 3</td></tr>
<tr><td>Country 3</td><td>Country 0</td><td>Country 1</td><td>Country 3</td></tr>
<tr><td>Country 3</td><td colspan="2" align="center">Country 2</td><td>Country 3</td></tr>
<tr><td colspan="4" align="center">Country 3</td></tr>
</table>

- The list of countries that must be coloured differently are as follows:
  1. 0 and 1
  2. 0 and 2
  3. 0 and 3
  4. 1 and 2
  5. 1 and 3
  6. 2 and 3
- The total number of correct solutions is $4! = 24$.

Example

5 Grover for 4-Colouring

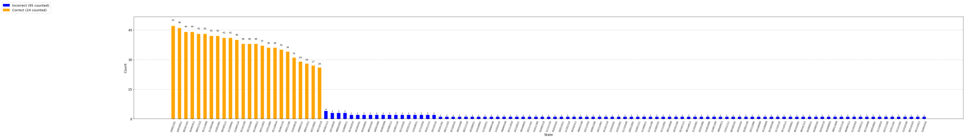

Figure: Histogram

Table of Contents

Scheduling Problem

Scheduling Aeroplanes

Given a schedule of flying routes and k aeroplanes available, create an assignment of aeroplanes to flights and timeslots.

Time—Route	Boston to DC (2 hr)	DC to LA (6 hrs)
9 AM	Route 0	Route 1
12 PM	Route 2	Route 3

Setup

We can use Graph Coloring to solve this problem!

- Instead of k colors, we have k airplanes.
- Instead of borders, we have restrictions because each aeroplane can only fly one route at a time. This means:
    1. Any aeroplanes taking off at the same time must be different.
    2. An aeroplane can be used for multiple routes, but only if it has landed after any previous flights. So, the takeoff times and route times must be considered.

Solution

Since there are 4 possible planes to use, this is fundamentally a 4 Coloring Problem. So, we can use our Grover solver from above.
The key to doing this is encoding the restrictions correctly:

1. Route 0 and Route 1 must use different planes since they take off simultaneously.
2. Route 0 and Route 2 must use different planes since the plane flying Route 0 will not have time to return to Boston for Route 2.
3. Route 1 and Route 2 must use different planes since the plane flying Route 1 will still be in the air when Route 2 is meant to start.
4. Route 1 and Route 3 must use different planes for the same reason.
5. Routes 2 and 3 must use different planes since they take off simultaneously.
6. The only acceptable reuse of a plane is for Route 0 and 3 since the plane flying Route 0 will have been able to land in DC before Route 3 takes off.

Quantum Search for Graph Colouring

Thank you so much!

Nisarg Suthar	*202003030*
Manjal Shah	*202003037*
Vandan Patel	*202001023*

This template is a based on SINTEF Presentation from Federico Zenith and its derivation Beamer-LaTeX-Themes from Liu Qilong