Semester Project Report

Quantum Information Theory

Kurella Anand Kasyup
EP22BTECH11012

Project Guide:
Dr. Alok Kumar Pan

Academic Year: 2024-25

Contents

Abstract

This report summarizes the study of quantum computing concepts through David McMahon's book on Quantum Computing. It explores foundational topics in quantum mechanics and information theory. Additionally, simulations of three major quantum algorithms — Grover's Algorithm, Shor's Algorithm, and the Quantum Teleportation Protocol — are presented to illustrate their functionality and potential applications.

Chapter 1

Introduction

Quantum computing leverages the principles of quantum mechanics to perform computations that are infeasible for classical computers. This project focuses on understanding the theoretical aspects of quantum information and the practical implementation of key quantum algorithms.

1.1 Objective

The objective of this project is to:

- Study foundational concepts in quantum computing using David McMahon's book.
- Simulate major quantum algorithms and analyze their significance.
- Develop intuition about the practical applications of quantum information theory.

1.2 Scope of the Project

This report encompasses:

- A review of theoretical concepts from the literature.
- Simulation and analysis of quantum algorithms using Qiskit.
- Exploration of quantum information applications, including cryptography and computation.

Chapter 2

Literature Review

The primary resource for this project was David McMahon's book on Quantum Computing. The book introduces quantum mechanics concepts, such as qubits, superposition, entanglement, and quantum gates. Key learnings include:

- Mathematical representation of qubits and quantum gates.
- Principles of quantum measurement and state collapse.
- Algorithms such as Grover's Algorithm and Shor's Algorithm, which showcase quantum speedups.

Chapter 3

Quantum Algorithms

This section provides an overview and simulation of three significant quantum algorithms. All simulations were implemented using the Qiskit library.

3.1 Grover's Algorithm

Grover's Algorithm is used for unstructured search problems, providing a quadratic speedup compared to classical algorithms.

3.1.1 Simulation

Listing 3.1: Grover's Algorithm Simulation in Qiskit

```
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import plot_histogram

# Grover's Algorithm: Single-item search in a two-qubit database
def grover_algorithm():
    qc = QuantumCircuit(2)  # 2 qubits

    # Apply Hadamard gates to create superposition
    qc.h([0, 1])

    # Oracle: Flip the amplitude of the marked state (|11>)
    qc.cz(0, 1)

    # Diffuser
    qc.h([0, 1])
    qc.x([0, 1])
    qc.cz(0, 1)
    qc.x([0, 1])
    qc.h([0, 1])

    # Measurement
    qc.measure_all()
    return qc

qc = grover_algorithm()
```

```
qc.draw('mpl')

# Simulate the circuit
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1024).result()
counts = result.get_counts()
print("Counts:", counts)
plot_histogram(counts)
```

3.2 Shor's Algorithm

Shor's Algorithm factorizes large integers in polynomial time, showcasing the potential of quantum computing in breaking RSA encryption.

3.2.1 Simulation

Listing 3.2: Shor's Algorithm Simulation in Qiskit

```
from qiskit.algorithms import Shor
from qiskit import Aer

# Simulating Shor's Algorithm for factoring 15
def shor_algorithm():
    backend = Aer.get_backend('aer_simulator')  # Quantum
       simulator backend
    shor = Shor(backend=backend)

    # Factorize 15
    result = shor.factor(15)
    return result

result = shor_algorithm()
print("Factors␣of␣15:", result.factors)
```

3.3 Quantum Teleportation Protocol

Quantum teleportation allows the transfer of a quantum state between two parties using entanglement and classical communication.

3.3.1 Simulation

Listing 3.3: Quantum Teleportation Simulation in Qiskit

```
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import plot_histogram

# Quantum Teleportation Protocol
def teleportation():
    qc = QuantumCircuit(3)  # 3 qubits: Qubit 0 (Alice's), 1 (
        entanglement), 2 (Bob's)

    # Step 1: Create an entangled pair between Qubit 1 and Qubit
        2
    qc.h(1)
    qc.cx(1, 2)

    # Step 2: Prepare Qubit 0 in an arbitrary state (|  > =  |0>
         +  |1>)
    qc.x(0)  # Example: Prepare |1> state (replace with general
        preparation if needed)

    # Step 3: Alice applies Bell measurement on Qubit 0 and Qubit
         1
    qc.cx(0, 1)
    qc.h(0)

    # Step 4: Send classical results to Bob and correct his state
    qc.barrier()
    qc.cx(1, 2)
    qc.cz(0, 2)

    # Measurement
    qc.measure_all()
    return qc

qc = teleportation()
qc.draw('mpl')

# Simulate the circuit
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1024).result()
counts = result.get_counts()
print("Counts:", counts)
plot_histogram(counts)
```

Chapter 4

Conclusion

This project provided an in-depth understanding of quantum information theory through theoretical study and practical simulations. Simulating quantum algorithms reinforced their theoretical aspects and highlighted the potential of quantum computing in various applications.

4.1 Future Work

Future research can explore advanced quantum algorithms and their optimization for specific use cases, such as quantum machine learning and cryptographic protocols.

Appendix A

Appendix

A.1 Code Files

All simulation codes are provided as Python scripts and can be accessed from the project GitHub repository: `https://https://github.com/anandk3012/EP12_SEM_PROJECT`

A.2 References

1. David McMahon, *Quantum Computing Explained*.
2. Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*.
3. IBM Qiskit Documentation, `https://qiskit.org/documentation/`.