



University of
Sheffield

Description Report: **Evolutionary
Optimization of Quantum Circuits for
Automated Circuit Design**

Eryk Arkadiusz Majoch

Supervisor: Phil McMinn

COM3610
October 11, 2024

Contents

1	Introduction	1
1.1	Background	1
1.2	Aims and Objectives	1
1.3	Studied Literature	2
2	Analysis	3
2.1	Problem Decomposition	3
2.1.1	Genetic Programming Algorithm	3
2.1.2	Information Representation	4
2.1.3	Simulation Program	4
2.2	Proposed Technologies	4
3	Plan of Action	5

Chapter 1

Introduction

1.1 Background

Quantum computing stands at the forefront of technological innovation, promising to revolutionise fields such as cryptography and drug discovery. Quantum computing leverages the principles of quantum mechanics to perform computations that would not be feasible for classical computers. By utilising quantum properties such as superposition and entanglement, quantum computers have the potential to solve certain problems much faster than their classical counterparts^[1].

However, the design of effective quantum circuits remains a significant challenge, often requiring experience in both quantum mechanics and computer science. Quantum circuits, the basis of quantum algorithms, consist of a series of quantum gates applied to qubits, manipulating quantum states to perform computations^[13]. Traditional approaches to quantum circuit design rely heavily on human intuition and deep knowledge, which can be an obstacle in the development of new quantum algorithms.

Platforms like IBM's Quantum Composer^[7] have made it easier for researchers to manually create quantum circuits. However, as the complexity of quantum algorithms grows, manual design becomes increasingly difficult and time consuming. This limitation highlights the need for automated approaches to quantum circuit design.

1.2 Aims and Objectives

This project aims to address the challenges of manual quantum circuit creation by exploring the application of Genetic Programming (GP) to automate the design process. GP, a subset of evolutionary algorithms, offers a promising approach to automated program synthesis^[9], evolving computer programs by applying principles inspired by biological evolution.

The primary objectives of this project are:

- Develop a complete simulation environment for quantum circuits using genetic programming
- Define appropriate genetic operations for circuit evolution

- Define fitness functions that guide the evolution process towards the desired circuit behaviour
- Analyse the effectiveness of evolutionary optimisation for quantum circuits by comparing them to already existing solutions using publicly available benchmarks

To ensure the quality and usability of the resulting simulation environment, it is essential to define key characteristics that must be followed:

- **Simplicity:** The platform should be easily configurable for the end user, with clear parameters and intuitive interfaces.
- **Clear Output:** The platform should generate its output in a preferably standardised format, ensuring that users can easily read, understand, and potentially integrate the results with other tools.
- **Performance:** The platform should demonstrate reasonable computational efficiency.
- **Scalability:** The platform should be designed to run both locally on classical computers and on High Performance Computing (HPC) clusters, allowing for the evolution of more complex quantum circuits when greater computational resources are available.

1.3 Studied Literature

- S. Bhandari et al. “Evolving Quantum Circuits to Implement Stochastic and Deterministic Cellular Automata Rules”. In: *Cellular Automata*. Ed. by B. Chopard et al. Cham: Springer International Publishing, 2022, pp. 119–129. ISBN: 978-3-031-14926-9.
- S. Bhandari et al. *How quantum and evolutionary algorithms can help each other: two examples*. 2024. arXiv: 2408.00448 [quant-ph]. URL: <https://arxiv.org/abs/2408.00448>.
- J. Brown, M. Paternostro, and A. Ferraro. “Optimal quantum control via genetic algorithms for quantum state engineering in driven-resonator mediated networks”. In: *Quantum Science and Technology* 8.2 (Jan. 2023), p. 025004. ISSN: 2058-9565. DOI: 10.1088/2058-9565/acb2f2. URL: <http://dx.doi.org/10.1088/2058-9565/acb2f2>.
- L. Sünkel et al. *GA4QCO: Genetic Algorithm for Quantum Circuit Optimization*. 2023. arXiv: 2302.01303 [quant-ph]. URL: <https://arxiv.org/abs/2302.01303>.
- T. Wong. *Introduction to Classical and Quantum Computing*. Rooted Grove, 2022. ISBN: 9798985593105. URL: <https://books.google.co.uk/books?id=M3jqzGEACAAJ>.

Chapter 2

Analysis

2.1 Problem Decomposition

The task of optimising automated quantum circuit design can be decomposed into three main categories:

1. **Genetic Programming Algorithm**
2. **Information Representation**
3. **Simulation Program**

Each of these categories presents unique challenges that need to be addressed to achieve the desired outcome. All the identified challenges also contain possible solutions which will be considered in the final solution and explained in a later section of the dissertation. They'll be experimented on or combined in order to see which yield the best possible outcomes. This is only an initial proposal of possible solutions as more may arise through further research if generated results aren't of expected quality.

2.1.1 Genetic Programming Algorithm

Genetic programming frameworks allow us to control every step of the evolution procedure. With this, we are able to tailor the framework to our needs. Here are the key aspects which will be fine-tuned as a part of this projects research:

Fitness Functions	Selection Methods	Crossover Methods	Mutation Methods
Fidelity Function	Random Selection	Single-point Crossover	Single Gate Flip
Entanglement Function	Tournament Selection	Multi-point Crossover	Mutate Qubits
			Mutate Gates

Table 2.1: Genetic Algorithm Decomposition Table

2.1.2 Information Representation

The next significant problem concerns the representation of any data stored within the program. There needs to be a way for the user to provide their circuit requirements in a simple way when using the program. During the evolutionary process, the circuits must be represented in such way that the algorithm may perform its steps comfortably. When simulation has finished, it would be preferable to use a standardised format whilst exporting the generated circuit in order to keep the possibility of using them with other tools open.

User's Description	Circuit Representation	Output Generation
Unitary Matrix	String Encoding	Visual Circuit Diagram
Truth Table	Directed Acyclic Graph	Qiskit Class Instance
Circuit Properties	Quantum Assembly Language	

Table 2.2: Information Representation Decomposition Table

2.1.3 Simulation Program

The final significant consideration is the user's interaction with the program itself. The simulation environment needs to be versatile by allowing the user to configure all of their requirements as well as be user-friendly.

Interface	Scalability
Command Line Interface (CLI)	Local execution on personal computers
Graphical User Interface (GUI)	Distributed execution on High Performance Computing (HPC) clusters

Table 2.3: Simulation Program Decomposition Table

2.2 Proposed Technologies

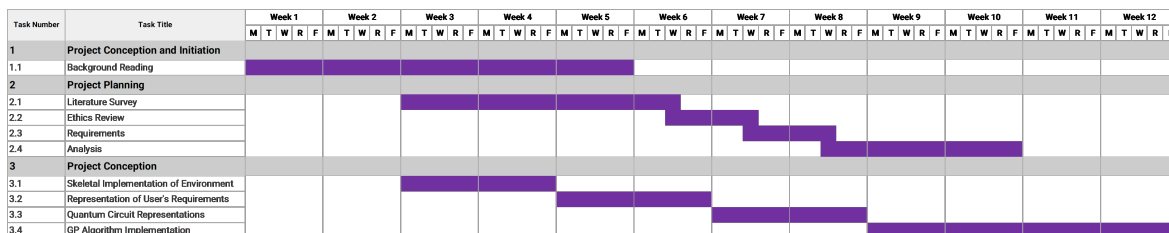
To address these challenges I will be using the following technologies. The program will be written using the Python programming language^[11] due to its popularity in machine learning tasks as well as extensive availability of third-party frameworks. The genetic programming framework of choice is PyGAD^[6] due to its customisability and for the GUI I've decided on PyQt^[5] due to my familiarity with its C++ library. I will also be using IBM's Qiskit quantum computing framework^[8] which is known for its representation, simulation and visualisation of quantum circuits.

Plan of Action

In the first semester, the focus will be on laying a solid foundation for the whole project. It will begin with extensive background reading on quantum computing and its applications. Concurrently, a literature survey will be conducted to dive deeper into the complicated process of optimising evolutionary quantum circuit design. This phase will also involve defining system requirements and conducting an analysis of the problem space.

The second semester will primarily focus on optimisation of the GP algorithm. Rigorous testing and validation of the system will be conducted throughout this stage, as well as data collection for the final report. The final weeks will be devoted to documentation, result analysis and final report editing.

Throughout the duration of the project, regular meetings with my project supervisor will be scheduled in order to ensure the project will be completed on time and to address any issues or question that arise. The Gantt charts below provide a rough visual representation of the project timeline.



5

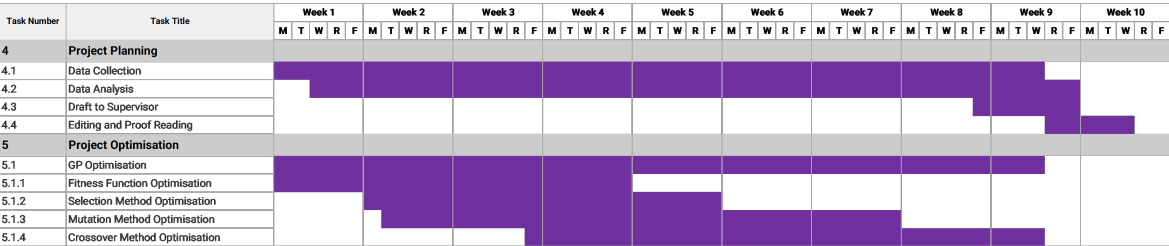


Figure 3.2: Gantt chart for semester two

Bibliography

- [1] M. Belkhir, H. Benkaouha, and E. Benkhelifa. “Quantum Vs Classical Computing: a Comparative Analysis”. In: *2022 Seventh International Conference on Fog and Mobile Edge Computing (FMEC)*. 2022, pp. 1–8. DOI: 10.1109/FMEC57183.2022.10062753.
- [5] R. Computing. “PyQt6: Python bindings for the Qt’s Company Qt application framework”. In: (2024).
- [6] A. F. Gad. “Pygad: An intuitive genetic algorithm python library”. In: *Multimedia Tools and Applications* (2023), pp. 1–14.
- [7] *IBM Quantum*. 2023. URL: <https://quantum.ibm.com/composer>.
- [8] A. Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. arXiv: 2405.08810 [quant-ph]. URL: <https://arxiv.org/abs/2405.08810>.
- [9] Y. Nagashima. “Genetic Algorithm for Program Synthesis”. In: *Fundamentals of Software Engineering*. Ed. by H. Hojjat and E. Ábrahám. Cham: Springer Nature Switzerland, 2023, pp. 104–111. ISBN: 978-3-031-42441-0.
- [11] *Welcome to Python.org*. <https://www.python.org/>. [Accessed 10-10-2024].
- [13] T. Wong. *Introduction to Classical and Quantum Computing*. Rooted Grove, 2022. ISBN: 9798985593105. URL: <https://books.google.co.uk/books?id=M3jqzgEACAAJ>.