Summer 7-26-2024

ADAPTABLE QUANTUM EDUCATION PLATFORM USING LEARNING OBJECTS

Krishna Puja Anumula
kanumula@students.kennesaw.edu

ADAPTABLE QUANTUM EDUCATION PLATFORM USING LEARNING OBJECTS

KRISHNA PUJA ANUMULA

A REPORT SUBMITTED AS PART OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE AT THE COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

KENNESAW STATE UNIVERSITY

GEORGIA, UNITED STATES OF AMERICA

July 2024

Advisor: Dr. Abhishek Parakh

Committee Members:

Dr. Maria Valero

Dr. Md Abdullah Al Hafiz Khan

DECLARATION

I confirm that the work contained in this master's thesis project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged, and all verbatim extracts are distinguished by quotation marks.

Signed**:** Krishna Puja Anumula ***Date*****:** 06/26/2024

CONSENT OF CIRCULATION

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. Any copying from or publication of this thesis involving potential financial gain will not be allowed without written permission.

***Signed*:** Krishna Puja Anumula ***Date*:** 06/26/2024

DEDICATIONS

I dedicate this thesis to my father, Srinivas Reddy Anumula, and my mother, Nirmala Anumula. Their unwavering support, encouragement, and love have been my greatest source of strength throughout my academic journey. I am forever grateful for the sacrifices they made to ensure my education and for always being there to guide me.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Abhishek Parakh, for his invaluable guidance, support, and encouragement throughout the course of this research. His expertise and insights have been instrumental in shaping the direction of my thesis, and his patience and understanding have provided me with the confidence to overcome numerous challenges.

I am profoundly thankful for the countless hours he has dedicated to discussing ideas, reviewing drafts, and providing constructive feedback. His commitment to excellence and passion for knowledge have been truly inspiring, and I feel lucky to have had the opportunity to work under his mentorship.

I also want to express my gratitude to Dr. Mahadevan Subramaniam. His insights and advice have been invaluable, and his willingness to help and provide feedback has greatly enriched my research. His expertise and support have made a significant impact on my work, and I am very thankful for his contributions.

ABSTRACT

In recent years, the need to make classroom learning more interactive and engaging has become increasingly important. The lack of workforce in interdisciplinary fields such as quantum networking and quantum internet requires a new approach that addresses every learner's individual needs. To address this challenge, this thesis introduces an adaptive learning platform rooted in the theory of learning objects and Kolb's experiential learning model. The platform aids educators and learners in designing and utilizing various learning objects for quantum networking and quantum internet.

The platform enables educators and learners to build their own lessons and lesson plans using learning objects tagged with metadata, tailored to their interactivity levels and learning preferences. Learners engage with the platform by providing background information such as the time they can dedicate to the course, their existing knowledge, and their learning goals. Based on this information, the platform generates a customized course with content designed to maximize the learner's achievement of their learning goals, utilizing the learning object dependency graph.

This thesis also delves into a simulator exercise that helps learners visualize and understand the key generation and message transmission using the E91 and the three-stage quantum networking protocols. By exploring the design, implementation, and feasibility of the platform, this thesis highlights its potential to revolutionize the teaching and learning of complex subjects such as quantum computing and quantum networking.

Contents

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Education is a key part of our society, with countries investing a lot in their educational systems. Traditionally, education has been delivered in a one-size-fits-all way, where all students follow the same path regardless of their individual needs. However, this method often doesn't work well for everyone. As a result, there's a growing shift towards personalized learning, especially in advanced fields like quantum education.

1.1 Problems with the One-Size-Fits-All Approach

The one-size-fits-all approach has several major problems [1]:

- **Lack of Individualization**: Students have different learning preferences (visual, textual, hands-on, quiz-driven, example-driven, etc.), and a single teaching method may not be effective for all students.
- **Fixed Curriculum**: A fixed curriculum does not allow students to explore the outcomes that interest them or align with their career goals.
- **Teacher-Centric Model**: The traditional model focuses on teacher-led instruction, which can result in passive learning where students aren't actively engaged.

1.2 Motivation for Personalized Learning

These limitations highlight the need for personalized learning models that cater to each student's unique needs, strengths, and interests. Personalized learning allows students to set their own learning goals, choose how they learn, and move at their own pace. This flexibility is especially important in advanced fields like quantum education, where the material is complex and rapidly changing [1] [2] [3] [4] [5] [6].

1.3 Complexities in Quantum Education

Quantum education is particularly challenging and requires a personalized approach [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17]. The field involves complex counterintuitive theories and innovative technologies that students need to understand deeply. Additionally, quantum education combines knowledge from different areas like physics, computer science, and engineering, requiring a flexible educational approach.

Students entering quantum education often have different backgrounds and levels of knowledge, making a one-size-fits-all approach especially ineffective. Personalized learning can help meet these diverse needs and ensure all students succeed.

1.4 Objective of the Thesis

This thesis's objective is to develop and evaluate a personalized learning platform that dynamically generates customized lesson plans for students in quantum education. This platform aims to address the limitations of the traditional one-size-fits-all approach by leveraging personalized learning and adaptive learning techniques. The key objectives include:

- **Developing a Personalized Learning Tool**: Create a tool that uses structured course content, interactive quizzes, and simulators to engage students and enhance their understanding of complex concepts.
- **Dynamic Lesson Plan Generation**: Implement a lesson plan generator that tailors learning paths based on individual student details, including their existing knowledge, available study time, and desired learning outcomes.
- **Integrating Advanced Simulator**: Develop and integrate simulator, leveraging quantum networking protocols such as E91 and 3-stage protocols, to provide visual experience that deepen student’s comprehension of these advanced topics.

- **Feasibility Testing and Feedback**: Conduct feasibility tests with students to gather feedback on the effectiveness and usability of the personalized learning platform, aiming to continuously improve the tool based on user experiences.
- **Utilizing Mathematical Models**: Apply advanced mathematical models, such as the precedence-constrained knapsack problem [1], to optimize decision-making in personalized learning environments, accounting for prerequisites and uncertainties in the learning process.
- **Enhancing Engagement and Outcomes**: Ultimately, the platform aims to increase student engagement, tailor learning experiences to individual needs, and improve educational outcomes in advanced and rapidly evolving fields like quantum education.

1.5 Introduction to the Adaptable learning platform

This thesis introduces a tool that dynamically generates lesson plans customized to individual students' learning preferences and desired learning outcomes. This tool collects details from students, such as pre-requisites they are already proficient in, the time they can dedicate to studying, and the outcomes they want to achieve through an interactive UI. Based on this information, the tool creates personalized lessons (presented as a series of Jupyter notebooks) that help students achieve proficiency in an efficient manner.

1.5.1 Leveraging Existing Course Materials with Interactive Tools

This platform uses existing course material on quantum education, from Clark. Center, that includes interactive quizzes and simulators designed to make learning more engaging and effective [18]. The quizzes provide immediate feedback, helping students see what they understand and where they need more practice. The simulators allow students to explore complex concepts in a hands-on way, making learning more interactive and enjoyable.

1.5.2 Architecture of the Adaptable Learning Platform

The architecture of the adaptable learning platform consists of three main components:

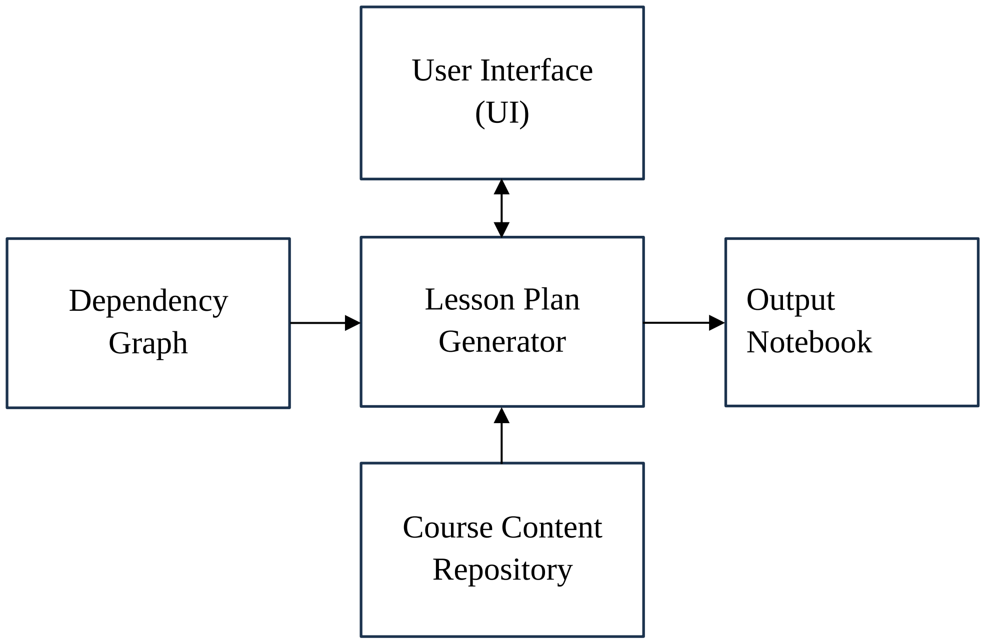


Figure 1.1: Architecture diagram for adaptive learning platform

- **User Interface (UI)**:

  The UI is where students interact with the system. They input their details, such as their existing knowledge, available study time, and learning goals. They also access interactive quizzes and simulators through the UI.

- **Course Content Repository**:

  This repository stores all the learning materials, including structured outcomes, concepts, interactive quizzes, and simulators. It serves as the content source for generating lesson plans.

- **Dependency Graph:**

The Lesson Plan Generator relies on the repository to fetch the necessary learning materials and the associated dependency graphs of the learning outcomes. These graphs are essential for the generator to create accurate and effective personalized lesson plans, ensuring that students are guided through the content in a logical sequence.

- **Lesson Plan Generator**:

  The lesson plan generator is the core component that dynamically creates personalized lesson plans based on the information collected from students and the content available in the repository. It ensures that each student receives a customized learning path tailored to their individual needs and goals.

Through extensive computational experiments, this thesis demonstrates the effectiveness of these methods in optimizing learning pathways, ensuring students can achieve their educational goals while navigating the complexities of quantum studies. Ultimately, this research contributes to the development of flexible and adaptive educational tools that can better support individualized learning in advanced and rapidly evolving fields like quantum education.

1.6 About this Thesis

This is the thesis of *Krishna Puja Anumula*, submitted as part of the requirements for the degree of Master of Science in Computer Science at the College of Computing and Software Engineering, Kennesaw State University, USA.

1.7 Structure of the Thesis

A list of all chapters within the thesis and a summary of the content are provided below.

Chapter 2 Quantum Networking Basics. This chapter provides an overview of the fundamental principles and technologies underpinning quantum networks. It covers key concepts such as qubits, entanglement and superposition. The chapter also discusses quantum repeaters and quantum communication protocols like Quantum Key Distribution (QKD).

Chapter 3 Network Simulator. This chapter details the development and implementation of a network simulator designed to model quantum communication networks. It explains the simulator's architecture, functionalities, and how it can be used to simulate key generation and message transmission using the E91 and 3-stage protocols.

Chapter 4 Notebook Generator. In this chapter, the creation of the notebook generator is discussed. This tool dynamically generates personalized lesson plans based on student inputs such as prior knowledge, available study time, and desired learning outcomes. The chapter covers the tool's architecture and the adaptive learning strategies employed to tailor educational experiences for students.

Chapter 5 Results. The chapter demonstrates use cases of the network simulator and notebook generator, showing how they enhance student engagement and understanding of complex quantum concepts through practical applications. This chapter also presents the results of feasibility tests conducted with students using the personalized learning platform. It includes quantitative and qualitative feedback on the effectiveness and usability of the tool.

Chapter 6 Future Works. The last chapter outlines potential future developments and improvements for the personalized learning platform and its components. It discusses possible enhancements in the adaptive learning algorithms, and further validation of the tools in diverse educational settings.

2 QUANTUM NETWORKING BASICS

2.1 Introduction

Quantum networking is an emerging field that combines principles of quantum mechanics with communication technology to enable secure and efficient transmission of information [19] [20]. Unlike classical networks that rely on conventional bits to transmit information, quantum networks leverage the principles of quantum mechanics, utilizing qubits to encode, process, and transmit data [21] [22]. This chapter aims to provide a foundational understanding of the key concepts and Quantum networking protocols like E91 and 3 stage [23].

2.2 Key Concepts

2.2.1 Qubits

Qubits, or quantum bits, are the fundamental units of information in quantum computing and quantum networking. Unlike classical bits, which can be either 0 or 1, qubits exploit the principles of quantum mechanics to exist in multiple states simultaneously. This unique property enables quantum networks to perform complex computations and transmit information in ways that are impossible for classical systems.

2.2.2 Superposition: The Core Principle

Superposition is one of the key principles that differentiates qubits from classical bits. A qubit can exist in a state $|0\rangle$, a state $|1\rangle$, or any quantum superposition of these states. Mathematically, this can be represented as [24]:

$$| \psi\rangle = \alpha \mid 0\rangle + \beta \mid 1\rangle$$

where $|\psi\rangle$ is the quantum state of the qubit, and α and β are complex numbers that represent the probability amplitudes of the qubit being in state $|0\rangle$ and state $|1\rangle$, respectively.

The probabilities must satisfy the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1$$

This superposition allows qubits to perform many calculations simultaneously, providing a significant computational advantage over classical bits.

2.2.3 Manipulating Qubits: Quantum Gates and Circuits

Just as classical bits are manipulated using logical gates, qubits are manipulated using quantum gates. Quantum gates are unitary operations that change the state of qubits, allowing for the execution of quantum algorithms. Some of the basic quantum gates include:

- **Pauli-X Gate (NOT Gate)**: Flips the state of the qubit ($|0\rangle$ to $|1\rangle$ and vice versa) as depicted in Figure 2.2.
- **Hadamard Gate (H Gate)**: Creates superposition states by transforming the basis states $|0\rangle$ and $|1\rangle$ into equal superpositions as demonstrated in Figure 2.3.
- **Controlled-NOT Gate (CNOT Gate)**: Entangles two qubits by flipping the state of the target qubit if the control qubit is in state $|1\rangle$.

Quantum circuits are composed of these gates, and they perform computations by applying a sequence of operations to qubits. The power of quantum computation arises from the ability to create and manipulate superpositions and entanglements through these circuits. All the figures related to? were generated using Qiskit [25].

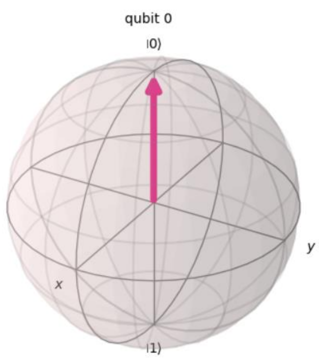


Figure 2.1Qubit State

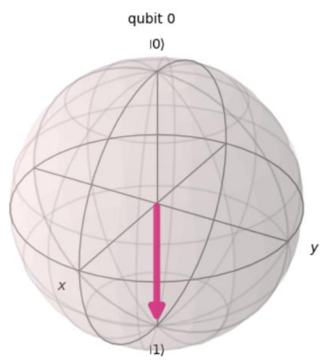


Figure 2.2 X-gate State

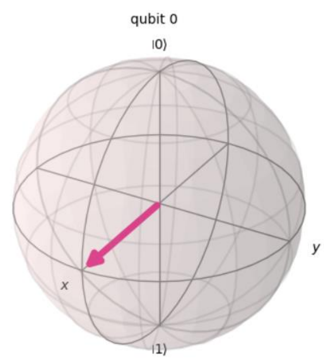


Figure 2.3 H-gate State

2.2.4 Entanglement and Creating Entangled States

Entanglement is a fundamental phenomenon in quantum mechanics where the quantum states of two or more particles become interconnected such that the state of one particle cannot be described independently of the state of the others [26].

Entanglement can be created using quantum gates. The most common entangled state is the Bell state, which is a maximally entangled two-qubit state. The Bell state can be created using a Hadamard gate followed by a CNOT gate as seen in Figure 2.4.

Bell State Circuit:

- Apply a Hadamard gate to the first qubit (Q_0) to create a superposition state.

- Apply a CNOT gate with the first qubit (Q_0) as the control and the second qubit (Q_1) as the target to entangle the two qubits.

The resulting Bell state $|\Phi^+\rangle$ is given by [24]:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

where $|00\rangle$ and $|11\rangle$ represent the entangled states of the photon pairs as shown in Figure 2.5.

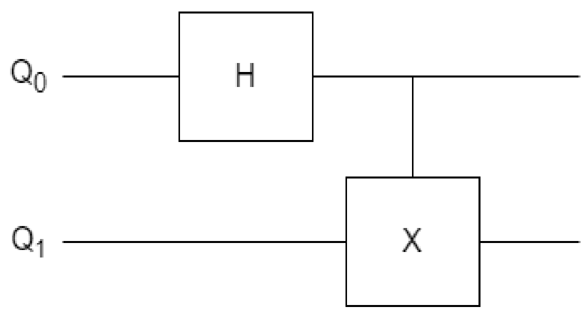


Figure 2.4 Entanglement Circuit

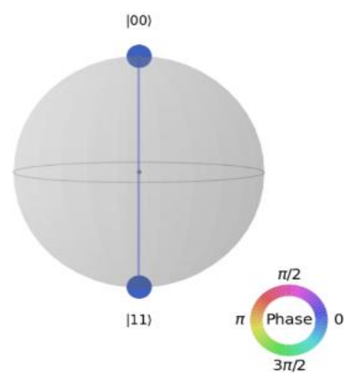


Figure 2.5 Entangled States

2.2.5 Creating Entanglement Swapping

Entanglement swapping is a process that enables two qubits that have never interacted directly to become entangled.

Steps:

- **Create Initial Entanglement**:

  Entangle Qubits Q_0 and Q_1, and Qubits Q_2 and Q_3 using the gates as shown in Figure 2.4. Qubits Q_0 and Q_1 are entangled using a Hadamard gate on

Q_0 and a CNOT gate with Q_0 as control and Q_1 as target. Similarly, Qubits Q_2 and Q_3 are entangled.

- **Bell State Measurement**:

  Perform a Bell state measurement on Qubits Q_1 and Q_2. A CNOT gate is applied with Q_1 as control and Q_2 as target, followed by a Hadamard gate on Q_1. Qubits Q_1 and Q_2 are then measured.

- **Conditional Operations**:

  Apply corrective operations on Qubit Q_3 based on the measurement outcomes of Qubits Q_1 and Q_2. the corrective operations that need to be applied to Q_3 depend on the measurement outcomes of Qubits Q_1 and Q_2. Apply both a CZ gate on Q_0 and Q_1, and a CNOT gate between Q_2 and Q_3. The Entanglement swapping circuit is built as illustrated in Figure 2.6. The Qubits Q_0 and Q_3 share an entangled state of either $|00\rangle$ or $|11\rangle$ as shown in Figure 2.7.

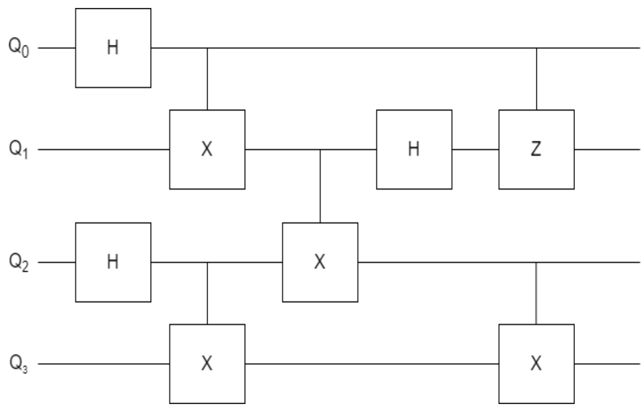


Figure 2.6 Entanglement Swapping Circuit

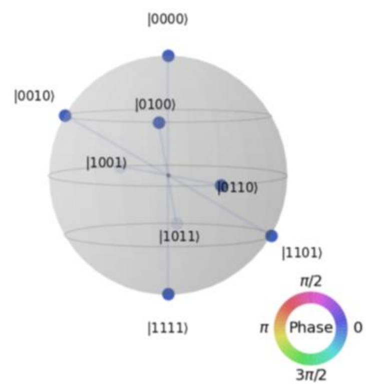


Figure 2.7 Qubits State after Swapping

2.2.6 Quantum Repeaters

Quantum repeaters are crucial components for extending the range of quantum communication networks beyond the limitations imposed by direct transmission. In classical communication, repeaters amplify signals to cover long distances, but in quantum communication, signal amplification is not possible due to the no-cloning theorem, which states that quantum information cannot be copied perfectly. Instead, quantum repeaters use a combination of entanglement swapping and entanglement purification techniques to extend the distance over which quantum information can be reliably transmitted [27] [28].

2.2.7 Quantum Teleportation

Quantum teleportation [29] allows the transmission of quantum information (the state of a qubit) from one location to another, using entanglement and classical communication. It does not transport matter but rather the state information of the qubit [30] [31].

2.3 Quantum Communication Protocols

Quantum networks rely on several key communication protocols [32] [33] that exploit quantum mechanical properties for secure information transfer. Quantum Key Distribution (QKD) [34] [35] [36] is a method for securely distributing encryption keys [37] [38]. Qubits can exist in a superposition of states, in contrast to traditional bits, which can only exist in one state at a time. Furthermore, two qubits can be coupled together by quantum entanglement so that the state of one instantly affects the other, regardless of how far apart they are. Furthermore, by guarding against eavesdropping, the qubit no-cloning theorem avoids the duplication of unknown qubits, guaranteeing secure key exchange [19] [39]. Protocols like E91and 3-Stage [40] use quantum states to detect eavesdropping and ensure that keys are exchanged securely [41] [42].

2.3.1 E91 protocol

The E91 protocol [43], developed by Artur Ekert in 1991, is a quantum key distribution (QKD) scheme that leverages quantum entanglement and the principles of Bell's inequality to ensure secure communication. In this protocol, pairs of entangled particles are shared between two parties, typically referred to as Alice and Bob [44]. By measuring these particles along randomly chosen axes, they obtain correlated results that can be used to generate a shared secret key. The security of the E91 protocol is rooted in the fact that any eavesdropping attempt by a third party (Eve) will disturb the quantum correlations, revealing the presence of the eavesdropper [45] [46].

2.3.1.1 Key generation Process

The essence of the E91 protocol lies in the unique properties of quantum entanglement, which allows pairs of particles to remain interconnected regardless of the distance separating them. By exploiting this phenomenon, the E91 protocol enables Alice and Bob to generate a shared secret key that is inherently secure against eavesdropping. This process is achieved through a series of well-defined steps involving the preparation and measurement of entangled photons, and the comparison of measurement bases.

- **Creation of Entangled Photons:**

  A special device produces pairs of entangled photons. These photons have properties that are strongly correlated, meaning the state of one photon is known if the state of the other is measured. The entangled state can be represented as:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

where $|00\rangle$ and $|11\rangle$ represent the entangled states of the photon pairs as seen in Figure 2.4.

- **Distribution of Photons:**

One photon from each entangled pair is sent to one party (Alice), while the other photon is sent to the second party (Bob). If Alice and Bob are far apart, quantum repeaters with entanglement swapping are used to maintain the entanglement.

- **Measurement Settings:**

Both parties randomly choose measurement settings from a predefined set of angles. These angles are used to measure the properties of the photons they receive.

- **Measuring the Photons:**

Alice and Bob each measure their photons using their chosen settings. The measurement results will be either 0 or 1, indicating different quantum states.

- **Announcement of Measurement Settings:**

After all measurements are completed, Alice and Bob announce the angles they used for each measurement, but they do not reveal the actual results.

- **Comparison of Measurement Settings:**

Alice and Bob compare their chosen measurement settings. They keep the results where their settings matched and discard the rest. These matching results are used to generate the raw key.

2.3.2 3 stage protocol

The three-stage quantum key distribution (QKD) [47] protocol is another method used to establish a secure communication channel between two parties, Alice, and Bob. The security of the three-stage protocol relies on the unpredictability and randomness introduced at each stage of the process, which makes it difficult for an eavesdropper (Eve) to gain information about the key without being detected [48] [49].

2.3.2.1 Unitary Matrices

A unitary matrix U is a complex square matrix that satisfies the condition:

$$U^{\dagger}U = UU^{\dagger} = I$$

where $U^{\dagger}$ is the conjugate transpose (also known as the Hermitian adjoint) of U, and I is the identity matrix.

Unitary matrices are important in quantum mechanics because they represent quantum operations that preserve the norm of quantum states, ensuring that the total probability remains 1.

2.3.2.2 Key Generation Process

The essence of the three-stage QKD protocol lies in its multi-stage process, where the security is enhanced by multiple rounds of encoding and decoding. This process is achieved through a series of well-defined steps involving the preparation, transformation, and measurement of photons.

- ***Preparation of Photons:***

  Alice prepares a series of photons in a known quantum state.

- ***First Stage: Encoding by Alice:***

  Alice encodes her secret key onto the photons by applying a quantum operation on the initial quantum states. This operation can be represented by a unitary matrix U_A:

$$| \psi_A\rangle = U_A | \psi\rangle$$

  where $| \psi\rangle$ is the initial state of the photons.

- ***Second Stage: Transformation by Bob:***

  Bob receives the encoded photons from Alice and applies his own quantum operation U_B on the received photons. This operation further transforms the quantum states, adding an additional layer of encoding:

$$| \psi_{AB}\rangle = U_B | \psi_A\rangle = U_B U_A | \psi\rangle$$

- ***Third Stage: Decoding by Alice:***

  The photons are sent back to Alice, who applies the inverse of her initial operation U_A^{-1} to the photons. This step removes her initial encoding, leaving the photons in a state that only depends on Bob's operation U_B:

$$| \psi_B\rangle = U_A^{-1} | \psi_{AB}\rangle = U_A^{-1} U_B U_A | \psi\rangle$$

- ***Measurement:***

  Alice and Bob measure the photons using different basis. The measurement results will be either $|0\rangle$ or $|1\rangle$, indicating different quantum states.

- ***Announcement of Measurement Settings:***

  After all measurements are completed, Alice and Bob announce which basis they used for each measurement. They do not reveal the actual measurement results.

- ***Comparison of Measurement Settings:***

  Alice and Bob compare their measurement results. They keep the results where their settings matched and discard the rest. These matching results are used to generate the raw key.

3 QUANTUM NETWORKING SIMULATOR

3.1 Introduction

Understanding quantum networking protocols is very challenging because they involve complex and abstract ideas. Traditional teaching methods often aren't enough to make these concepts clear. Therefore, there's a strong need for a simulator to help students learn.

A simulator provides a hands-on, interactive way for students to see and experiment with quantum networking protocols, like how secure keys are generated, how messages are sent and visualize qubit transmission. By using a simulator, students can better understand these difficult topics because they can see how things work in a visual and practical way. Using a simulator in education makes learning more engaging and effective. It allows students to learn by doing, which helps them understand and remember the material better. This approach makes the complicated ideas of quantum networking easier to grasp and more interesting to study. Thus, this thesis introduces a new educational tool: the Quantum Network Simulator. This simulator lets you interactively explore how quantum communication works, focusing on two main protocols: E91 and 3-stage.

3.1.1 Overview of the Quantum Network Simulator

The Quantum Network Simulator is designed to be user-friendly and works right in Jupyter Notebook [50]. It uses a simple interface (GUI) made with tkinter [51], where you can see a simulated quantum network displayed on a map-like background.

3.1.2 Why Simulate Quantum Networks?

Simulating quantum networks is important because:

- **Better Understanding:** It helps you visualize how the protocols communicate within a quantum network.
- **Hands-On Learning:** You can try different scenarios and see immediate results, which helps you connect theory with real-world applications.

- **Personalized Learning:** You can customize simulations to focus on what you want to learn; example, generating keys or sending messages in a quantum network.

3.1.3 Network Parameters and Assumptions

When designing and using a quantum network simulator, several key network parameters need to be considered to ensure the simulations are realistic and educational. Below are some of the essential parameters along with the assumptions made for each:

Quantum Network Simulator Parameters:

Table 3.3.1 Quantum Network Simulator Parameters for Quantum Communication

Simulator Parameters	Parameter	Description
Node Placement	Physical locations of nodes within the network.	Nodes are placed in a grid-like pattern to simplify the simulation, representing possible real-world setups.
Network Size	The total number of nodes in the network.	A default grid size of 5x5 is used, but this can be adjusted based on specific learning objectives.
Protocol Selection	Choice between E91 and 3-stage protocols.	Users select the protocol based on the specific quantum communication scenario they wish to explore.
Mode of Operation	Transmission mode or arrangement mode	In arrangement mode, users can drag and drop nodes to distinct locations to understand the impact of physical distances.
		In transmission mode, users can simulate direct qubit transfer between nodes.
Task Selection	Tasks such as generating a secure key or sending a message	Tasks are chosen to demonstrate practical applications of quantum communication.

Path Selection	The route that data takes through the network.	The shortest path is calculated using Dijkstra’s algorithm for the E91 protocol, while the 3-stage protocol involves user-guided path selection through intermediate nodes.
Entanglement and Qubit State Changes	Behavior and visualization of qubits during transmission	Qubits undergo specific operations (e.g., X gate, H gate) as they are transmitted, with their states visualized using the Qiskit library

Assumptions:

1. **Ideal Conditions:**

   The simulator assumes ideal quantum channels without noise, loss, or decoherence, simplifying the learning process [52].

2. **User Knowledge:**

   Users have a basic understanding of quantum computing principles, enabling them to make informed choices in the simulator.

3.2 Simulator Design

The Quantum Network Simulator aims to make quantum networking concepts accessible and easy to understand through interactive simulations. By focusing on user-friendly design and practical learning experiences, the simulator helps students and researchers explore complex quantum communication protocols.

3.2.1 Goals and Objectives

The primary goal of the Quantum Network Simulator is to serve as an educational aid, simplifying the learning process for quantum networking. The simulator is designed to be intuitive, allowing users of varying technical backgrounds to engage with it easily. Through interactive learning, the simulator enables users to actively explore quantum protocols, fostering a deeper understanding of these advanced concepts. Another key objective is flexibility, allowing the tool to be customized for different learning scenarios and objectives.

3.2.2 Key Features and Functionalities

The Quantum Network Simulator offers a range of features and functionalities to enhance the learning experience. The simulator provides two main modes of operation:

- **Arrangement Mode:** In Arrangement Mode, users can rearrange nodes to simulate physical distances, helping them understand the impact of node placement on network performance and later proceed to transmission Mode.
- **Transmission Mode:** In Transmission Mode, users can directly transmit information between nodes, seeing firsthand how data travels through the network.

The simulator supports two key quantum communication protocols:

- **E91 protocol:** Users can generate a random key using the E91 protocol and then send a message using classical one-time pad encoding with the generated key.
- **3-Stage protocol:** Users can generate a random key using the 3-Stage protocol and then send a message using classical one-time pad encoding with the generated key. Alternatively, users can also send a message directly using quantum communication, like how the key is generated using 3 stage process.

The simulator is highly customizable, allowing users to choose network grid size and customize the number of nodes in the network. This flexibility ensures that the simulator can cater to a wide range of educational needs and preferences.

3.2.3 User Interface Design

The user interface design of the Quantum Network Simulator focuses on simplicity and ease of use.

3.2.3.1 Welcome Screen

When users run the simulator, a launch window displays a welcome screen prompting them to select options such as mode (Arrangement or Transmission) and protocol (E91 or 3-Stage).

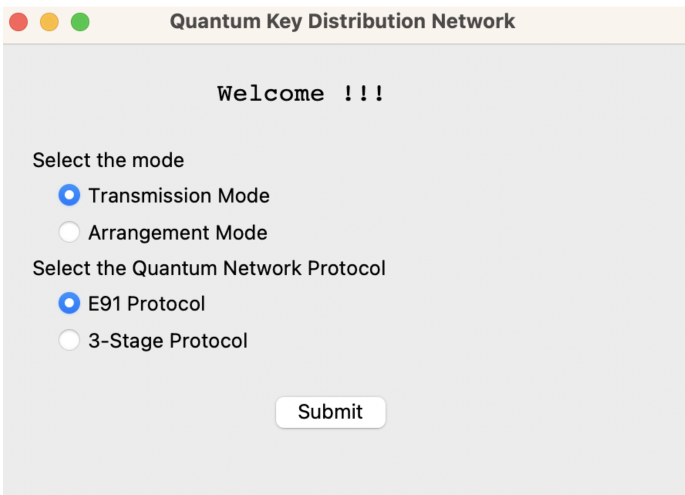


Figure 3.1 Welcome Screen

3.2.3.2 Arrangement Screen:

When the user selects Arrangement Mode, they are directed to the Arrangement screen, where a network grid defaulted to a 5x5 size is displayed. Each node in this grid can be connected to its adjacent nodes, creating a network topology that facilitates the selection of paths for transmission. On this screen, the user can drag and drop nodes and rearrange them on the map screen, with the distance between nodes referencing real-world physical distances. The control panel includes options to adjust the grid size, with an input field and a submit button for applying changes. It also features a reset button that reverts the entire screen, including node selections and transmissions, back to the original state. Additionally, there is a button to proceed to the transmission mode screen for random key generation and message transmission.

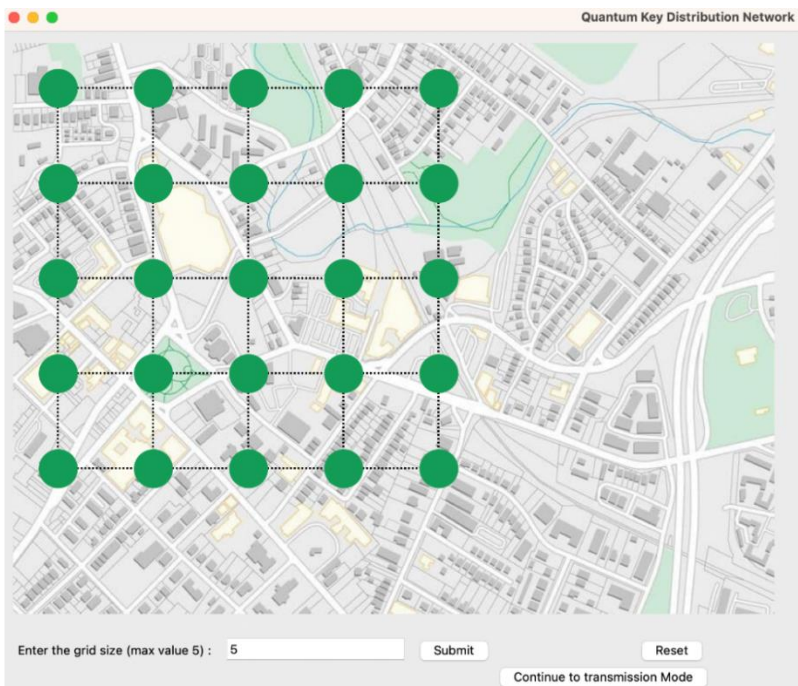


Figure 3.2 Arrangement Screen

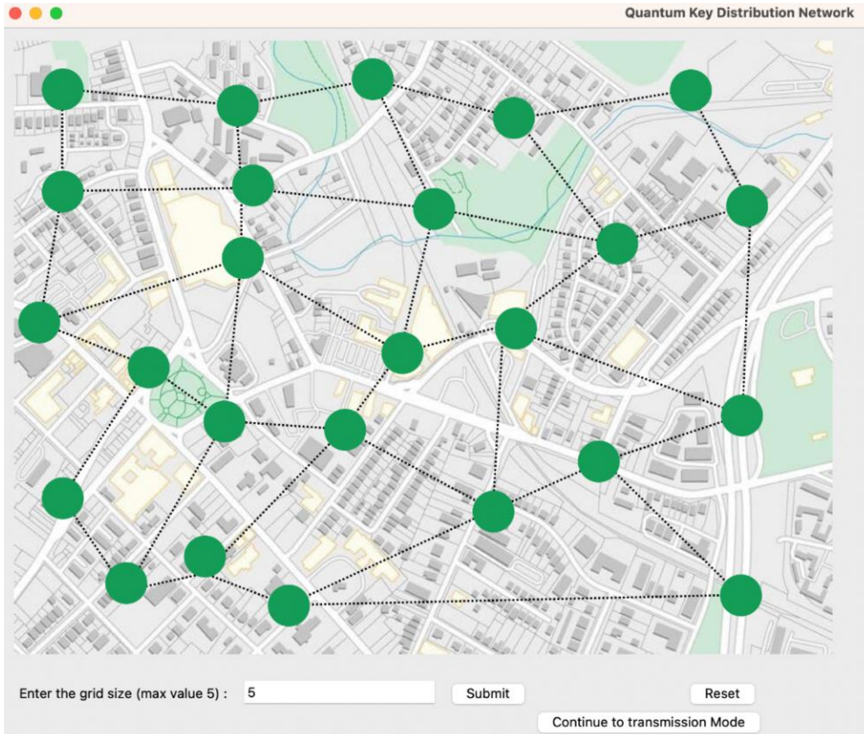


Figure 3.3 Arrangement Screen after placement

3.2.3.3 Transmission Screen:

When the user selects the Transmission mode, they are navigated to the transmission screen. Here, the user can select start and end nodes that act as Alice and Bob, generate a random key, and perform message transmission. The control panel includes an input to select or change the grid size and a submit button to apply these changes. A reset button is available to revert the entire screen, including node selection and transmissions, to its original state.

Additionally, a key generation button allows the user to generate a random key and transmit it between Alice and Bob.

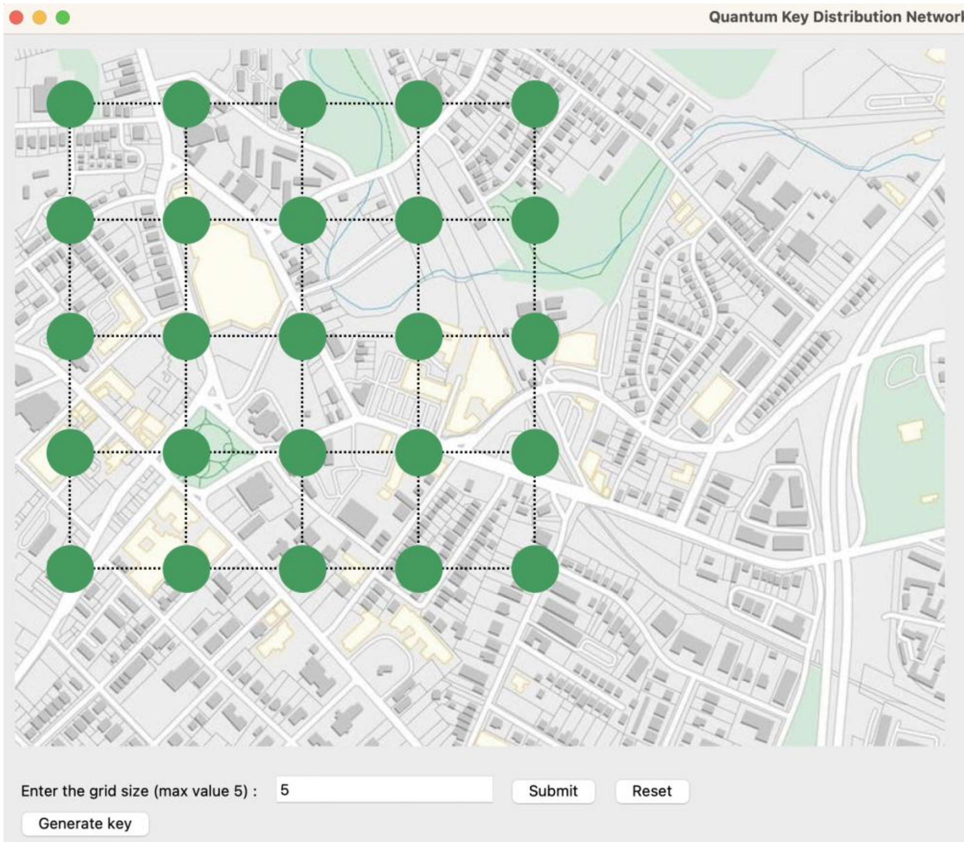


Figure 3.4 Transmission Screen

3.2.4 Node Configuration

This section details how the nodes are configured in this simulator and how students can select the Alice and Bob nodes in the network with multiple nodes and their interaction with the simulator for both the protocols, and the visualization of node selection.

A Node represents a host in a network. A host refers to any device that connects to a network and can communicate with other devices on that network. This includes computers, servers, smartphones, tablets, printers, and any other devices that use an IP address to communicate. Hosts are crucial in network operations as they serve as the endpoints for data transmission, generating, receiving, and processing network traffic. Each host is assigned a unique address, which allows it to be identified and to communicate with other devices on the network.

Each Node in this simulator has following properties:

Table 3.2 Network Node Parameters

Properties	Description
Id	Unique identifier for each node
Element_id	UI element Id tagged to the node
Status	‘SELECTION’, ‘SELECTED’, ‘DISABLED’
Position	Coordinates of the position of the node in the given area
Color	color of the node
Previous selection status	previous status of the node
Edges	Edge Ids connected to the node
Connections	Holds details of next node connection including id and coordinates.

3.2.5 Edge Configuration

Edge represents the connection between hosts. In the real world, Physical network connections usually refer to the tangible links between devices within a network, enabling communication and data exchange. These connections typically involve cables like Ethernet for wired setups, or signals like Wi-Fi and Bluetooth for wireless setups. In a wired network, devices such as computers, routers, and switches are connected via cables that transmit data through electrical signals. The goal of physical network connections is to facilitate reliable and efficient communication between various devices in a network.

Every Edge in this simulator has following properties:

Table 3.3 Network Connection Parameters

Properties	Description
Id	Unique identifier for each Edge
Element_id	UI element Id tagged to the Edge
Position	Start and end coordinates of the position of the edge in the given area
Type	'row'- connects left and right nodes
	'col' – connects top and bottom nodes
Nodes	Node Ids linked to the connection

3.3 Protocol Transmission Process

The Protocol Transmission Process in the Quantum Network Simulator provides an interactive and visual way for students to understand how quantum communication protocols work. This section covers the step-by-step description of the E91 Protocol Simulation, the 3-Stage Protocol Simulation, and the data flow and information transmission within the simulator.

3.3.1 Step-by-Step Description of the E91 Protocol Simulation

The E91 protocol is a quantum key distribution (QKD) method that uses entanglement to securely exchange cryptographic keys. Here's how the simulation process works:

3.3.1.1 Node Selection

When the user selects the E91 protocol from the welcome screen, they have the option to use Arrangement mode to rearrange the nodes. In Arrangement mode, users can interact with the network grid, moving nodes to optimize or explore different path configurations. Once satisfied with the node arrangement, they can switch to Transmission mode.

In Transmission mode, the user selects the start node (Alice), which is highlighted in blue, and the quantum circuit is initialized with a pair of qubits—one qubit is kept at Alice, while the other is sent to Bob. The user can select any node as the Bob node for transmission. The simulator will automatically calculate the shortest path between Alice and Bob based on the new arrangement.

3.3.1.2 Shortest Path Calculation

The simulator uses Dijkstra's algorithm to find the shortest path between the selected nodes. Dijkstra's algorithm is a method for finding the shortest path between nodes in a graph. The algorithm works by iteratively selecting the node with the smallest tentative distance, updating the tentative distances of its neighbouring nodes, and marking the node as processed. Initially, the distance to the source node is set to zero, and all other nodes are set to infinity. A priority queue (or similar structure) is used to keep track of nodes to be processed. At each step, the node with the smallest distance is removed from the queue, and its neighbours are updated if a shorter path is found through this node. This process continues until the destination node is reached or all nodes have been processed. The final shortest path is then reconstructed by backtracking from the destination node to the source node using a record of the path taken. The algorithm guarantees the shortest path in graphs with non-negative edge weights.

Algorithm 1: Dijkstra's Algorithm for Shortest Path

Input

- **number_of_nodes:** Total number of nodes in the graph
- **src:** Source node
- **destination:** Destination node
- **adjacency_matrix:** 2D list representing the adjacency matrix of the graph

Output

- **path:** The shortest path from the source node to the destination node

Algorithm

- **Initialize:**

```
dist ← [] * (number_of_nodes * number_of_nodes)
dist[src] ← 0
sptSet ← [False] * (number_of_nodes * number_of_nodes)
pathSet ← []
```

- **Main Loop:**

```
for count = 1 to number_of_nodes * number_of_nodes do:
    u ← minDistance(dist, sptSet)
    sptSet[u] ← True
    for v = 0 to number_of_nodes * number_of_nodes - 1 do:
        if adjacency_matrix[u][v] > 0 and sptSet  == False
        and dist[v] > dist[u] +  adjacency_matrix[u][v] then:
            pathSet.append([v, u])
            dist[v] ← dist[u] + adjacency_matrix[u][v]
```

- **Construct the Path:**

```
path ← [destination]
dest ← destination
while dest ≠ src do:
    dest ← next(x[1]
    for x in reversed(pathSet) do:
        if (x[0] == dest)
            path.append(dest)
            path.reverse()
```

- **Return Path:**
  return path

3.3.1.3 Entangled Qubits Transmission Process

After determining the shortest path, the simulator calculates the number of quantum repeaters needed to transmit the entangled qubit to Bob. The qubit created at the Alice node is first shared with the adjacent quantum repeater node on the determined shortest path. The entangled state created at Alice can be represented as:

$$|\psi_{AA`}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

where $|0\rangle$ and $|1\rangle$ are the basis states, and $A, A^{`}$ represent entangled qubits at Alice. The quantum repeater creates its own pair of entangled qubits:

$$|\psi_{RR`}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

where R and $R^{`}$ are the two qubits created at the repeater.

The repeater performs entanglement swapping between the qubit received from Alice (A) and one of the qubits it created (R). The resulting state after entanglement swapping:

$$|\psi_{AR}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

The other qubit ($R^{`}$) is sent to the next quantum repeater.

This process of entanglement swapping continues at each quantum repeater along the shortest path. Each repeater performs entanglement swapping between the received qubit and one of its own qubits, sending the remaining qubit to the next repeater. Mathematically, if there are n repeaters, the state after the n^{th} repeater would be:

$$|\psi_{AR_n}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

where R_n is the qubit at the n^{th} repeater.

The last repeater performs entanglement swapping, resulting in a final shared entangled state between Alice and Bob:

$$|\psi_{R_nB}\rangle = |\psi_{AB}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

To simulate the behavior of a quantum repeater network using the Qiskit library, each node in the network will represent a quantum circuit with a pair of entangled qubits. One of the qubits will be shared with the next node, and the process will involve entanglement swapping to maintain the entanglement over long distances. A Sample Entanglement Swapping with a Quantum Repeater is illustrated in Figure 3.5 and circuit that achieve this behavior is represented in Figure 3.6.

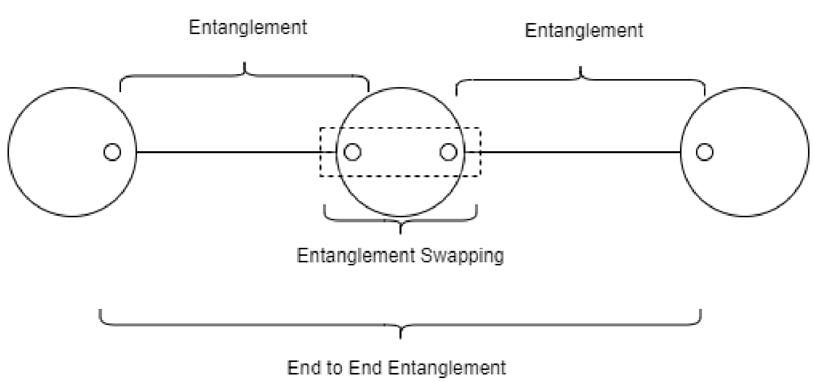


Figure 3.5 Entanglement Process between two Nodes and a Quantum Repeater

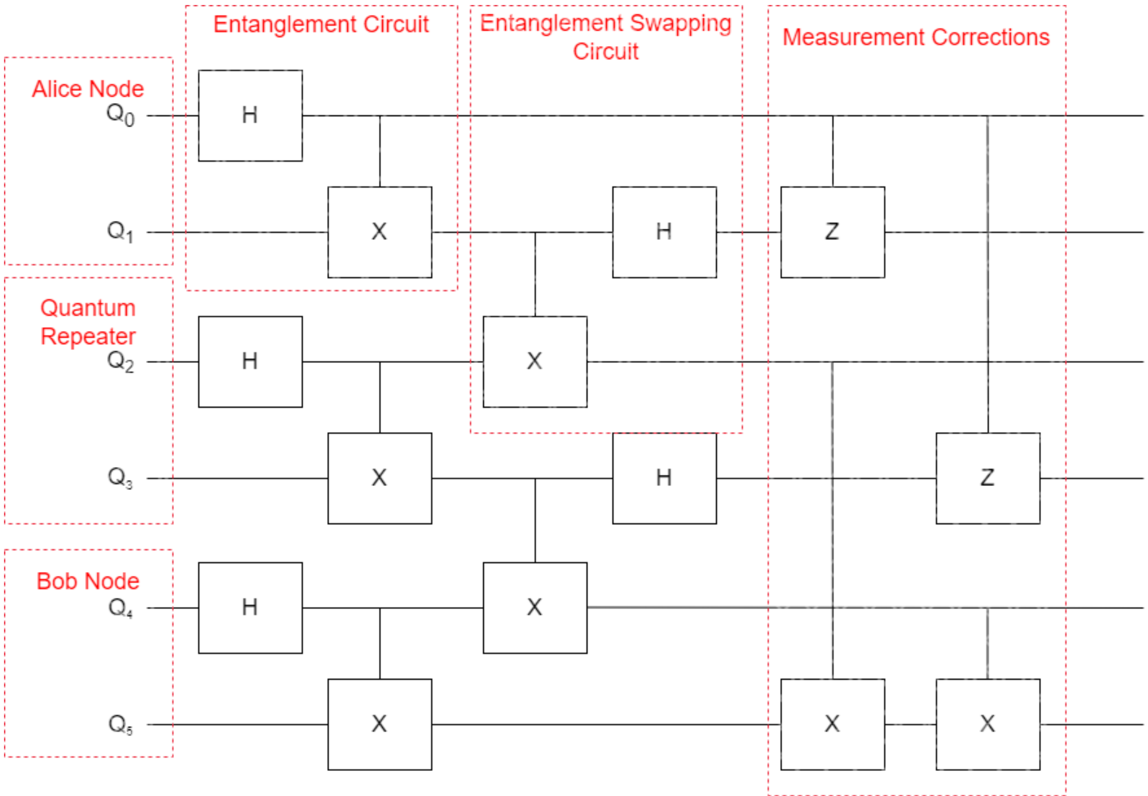


Figure 3.6 Sample Quantum Circuit with a Quantum Repeater

3.3.1.4 Random key Generation

The simulator generates a raw key by simulating the E91 protocol for 100 iterations. In each iteration, it creates an entangled pair by constructing dynamic quantum circuits at each node, measures the qubits in randomly chosen bases, and records the results.

$$\text{Alice's bases: } \theta_A \in \{0^\circ, 45^\circ, 90^\circ\}$$

$$\text{Bob's bases: } \theta_B \in \{0^\circ, 60^\circ, 90^\circ\}$$

The raw keys from Alice and Bob are filtered to retain only those bits where compatible measurement bases were used (e.g., 0°vs. 0° and 90° vs. 90°). The keys measured at Alice and Bob are identical, as the channel is assumed to be ideal and free from noise. This key can then be used for securely transmitting a message.

3.3.1.5 Message Transmission via Classical channel

One-time pad (OTP) is an encryption technique that provides theoretically unbreakable security when used correctly using classical channel. It involves combining a plaintext message with a random key of the same length using bitwise operations. The generated random key can be used for one-time pad encryption to ensure secure communication.

- **Key Generation**:

  A random key is generated using the E91 protocol. Both Alice and Bob obtain identical keys through the quantum key distribution process.

- **Message Preparation**:

  Alice prepares her plaintext message that she wants to send to Bob. The message is converted into a binary format using 8-bit encoding. Each character of the plaintext message is converted to its binary representation using 8-bit ASCII encoding. In this encoding scheme, each character is represented by an 8-bit binary number.

- **Encryption (One-Time Pad Encoding)**:

  Alice encrypts the plaintext message using the one-time pad method. This involves performing a bitwise XOR operation between each bit of the plaintext and the corresponding bit of the key.

Let's denote:

P as the plaintext binary message.

K as the binary key.

C as the resulting ciphertext.

The encryption process is represented as:

$$C = P \oplus K$$

where $\oplus$ denotes the bitwise XOR operation.

- **Transmission**:

  Alice sends the encrypted message C over the classical communication channel to Bob.

- **Decryption**:

  Upon receiving the ciphertext C, Bob decrypts it using the same one-time pad key K. The decryption process is simply another bitwise XOR operation between the ciphertext and the key.

  The decryption process is represented as:

$$P = C \oplus K$$

This operation recovers the original plaintext message because:

$$P \oplus K \oplus K = P \oplus 0 = \mathrm{P}$$

where 0 represents a binary string of all zeros.

3.3.2 Step-by-Step Description of the 3-Stage Protocol Simulation

3.3.2.1 Node Selection

The user selects the start node (Alice) on the network grid. The simulator then enables adjacent nodes for selection, guiding the user through the process.

The user continues to select nodes sequentially, creating a path from the start node to the end node. Only adjacent nodes can be selected at each step, ensuring a logical path. The intermediate nodes represent quantum repeaters.

3.3.2.2 Random Key Generation Process:

The three-stage QKD protocol involves a multi-step process where the security is enhanced by multiple rounds of encoding and decoding. Here's how the protocol works:

- **Preparation of Photons:**

  Alice prepares a series of random photons in a known quantum state using a simple quantum circuit, using the Z basis and the states are $|0\rangle$ and $|1\rangle$.

- **First Stage: Encoding by Alice:**

  Alice encodes her secret key onto the photons by applying unitary transformation U_A to the initial quantum states:

  $$|\psi_A\rangle = U_A |\psi\rangle$$

  where $|\psi\rangle$ is the initial state of the photons.

- **Second Stage: Transformation by Bob:**

  Bob receives the encoded photons from Alice and applies his own quantum operation U_B to the received photons:

  $$|\psi_{AB}\rangle = U_B U_A |\psi\rangle$$

- **Third Stage: Decoding by Alice:**

  The photons are sent back to Alice, who applies the inverse of her initial operation U_A^{-1}:

$$|\psi_B\rangle = U_A^{-1} |\psi_{AB}\rangle = U_A^{-1} U_B U_A |\psi\rangle$$

- **Measurement:**

  Bob also applies inverse operation U_B^{-1} and perform measurement using the Z basis. The measurement results will be either $|0\rangle$ or $|1\rangle$, indicating different quantum states.

- **Key Generation:**

The results of these measurements are then used to generate the shared secret key. Because the operations U_A and U_B are random and private to Alice and Bob, respectively, any eavesdropper trying to intercept the key will be detected due to the disturbance caused in the quantum states.

3.3.2.3 Message Transmission via Classical Channel:

Uses one-time-pad encryption same as message transmission in E91 protocol and transmits messages in a classical channel.

3.3.2.4 Message Transmission via Quantum Channel:

In the three-stage quantum key distribution (QKD) protocol, a message can be securely transmitted by encoding it onto qubits. The message is first converted into a binary format using 8-bit ASCII encoding, where each bit represents the state of a qubit (0 or 1). Qubits are prepared in states corresponding to the binary representation of the message. A 0 in the binary message corresponds to the $|0\rangle$ state, and a 1 corresponds to the $|1\rangle$ state. The rest of the process follows the same steps as key generation. Alice encodes the message onto the qubits by applying a Unitary quantum operation. Bob applies the second level of encoding by using Unitary quantum operation. Both Alice and Bob apply transformations subsequently and measure along Z basis. Since the channel is ideal and noiseless, the measurements will yield the same results for both parties, representing the securely transmitted message.

3.4 Technical Implementation

The technical implementation of the Quantum Network Simulator involves various software tools, coding and scripting details, and integration with Jupyter Notebooks. This section breaks down these components to provide a clear understanding of how the simulator was built and operates.

3.4.1 Software and Tools Used

- **Python:**

  The primary programming language used for developing the Quantum Network Simulator. Python was chosen for its simplicity, readability, and extensive library support for scientific computing and visualization.

- **tkinter:**

  A standard Python library used to create the graphical user interface (GUI) for the simulator. It provides tools to build interactive windows and handle user inputs.

- **Qiskit:**

  An open-source quantum computing software development framework by IBM. It is used to simulate quantum circuits, visualize qubit states, and perform quantum operations.

- **Jupyter Notebooks:**

  An open-source web application that allows the creation and sharing of documents containing live code, equations, visualizations, and narrative text. It provides the platform for integrating the simulator and creating dynamic lesson plans.

3.5 Educational Benefits

3.5.1 Enhancing Understanding of Quantum Networking

The Quantum Network Simulator provides students with a hands-on tool to explore and understand complex quantum networking concepts. By simulating the E91 and 3-stage protocols, students can visualize how quantum information is transmitted and the principles behind quantum entanglement and secure communication. This interactive approach helps demystify abstract theories and makes them more accessible and engaging.

3.5.2 Hands-On Learning Experience

The simulator offers a practical learning experience where students can experiment with different scenarios and see the immediate effects of their actions. They can manipulate network nodes, test various protocols, and observe the results in real time. This experiential learning aligns with Kolb's theory, reinforcing concepts through active experimentation and reflection.

3.5.3 Bridging Theory and Practical Application

One of the significant advantages of the Quantum Network Simulator is its ability to bridge the gap between theoretical knowledge and practical application. By allowing students to build and test their own quantum networks, the simulator connects textbook concepts with real-world implementation. This hands-on practice is crucial for deepening understanding and preparing students for advanced studies or careers in quantum computing and networking.

3.6 Challenges and Solutions

3.6.1 Complexity of Quantum Algorithms:

Implementing quantum algorithms like the E91 protocol and the 3-stage protocol required a thorough understanding of quantum mechanics and precise coding to ensure accurate simulations.

Solution: Extensive research and collaboration with experts in quantum computing helped to accurately translate theoretical algorithms into functional code. Utilizing libraries like Qiskit provided reliable tools for quantum circuit simulations.

3.6.2 Integration with Jupyter Notebooks and Qiskit:

Ensuring seamless integration between the simulator's interactive UI and Jupyter Notebooks posed challenges in maintaining user-friendly interfaces and robust backend processing and Qiskit integration with Jupyter Notebooks.

Solution: Modular coding practices were employed to separate the UI components from the core simulation logic. This approach facilitated easier integration and debugging.

3.6.3 Real-Time Visualization:

Providing real-time visualization of quantum state along network paths required efficient rendering techniques to avoid performance lags.

Solution: Optimized algorithms and efficient use of visualization libraries like matplotlib and Qiskit ensured smooth and responsive visual feedback for users.

The Quantum Network Simulator not only enhances the educational experience but also contributes to the broader goal of making quantum computing and networking accessible to a wider audience.

4 ADAPTIVE NOTEBOOKS GENERATOR

4.1 Introduction

4.1.1 Overview of Adaptive Learning

Adaptive learning is an educational approach that uses technology and student background information to tailor instruction to each student's needs. This method adjusts the learning path and pace based on the learner's performance, preferences, and goals, ensuring a personalized educational experience. By continuously analyzing student interactions, adaptive learning systems can provide customized resources and activities, enhancing the overall learning process.

4.1.2 Importance of Personalized Education

Personalized education is crucial in today's diverse learning environment. It recognizes that each student has unique strengths, weaknesses, and learning styles. By offering tailored learning experiences, personalized education helps students engage more deeply, understand concepts better, and achieve their academic goals more effectively. This approach not only addresses individual learning needs but also fosters a more inclusive and supportive educational environment.

4.1.3 Objectives of the Lesson Plan Generator

The primary goal of the Lesson Plan Generator is to create customized lesson plans that align with each student's prior knowledge, learning preferences, and available study time. The generator aims to:

- Exclude already known concepts to avoid redundancy and save time.
- Allow users to select specific lessons and learning outcomes from a predefined list.

- Generate lesson plans dynamically by choosing the most relevant and beneficial concepts that fit their schedule using the Precedence Controlled Fractional Knapsack algorithm.
- Ensure that each lesson includes clear learning objectives and outcomes, grouped into coherent concepts for better understanding.
- Provide a flexible and adaptive learning experience that caters to individual student needs, enhancing the overall effectiveness of the educational process.

4.2 Input Prerequisites and User Preferences

Before generating a personalized lesson plan, it's crucial to gather detailed information about the student's current knowledge and learning preferences. The system assesses the student's prerequisite knowledge by identifying concepts and topics they have already mastered. This helps in excluding redundant content and focusing on areas where the student needs further reinforcement.

Once the prerequisite information is collected, the system allows students to select the lessons and learning outcomes they aim to achieve. This phase involves presenting a curated list of available lessons, each clearly defined with specific learning outcomes. Students can then prioritize these lessons based on their academic interests, career goals, or areas they wish to strengthen. This customization empowers students to take ownership of their learning journey and ensures that the lesson plan is tailored to their individual needs.

In addition to selecting lessons, estimating available study time is essential for creating a realistic and effective lesson plan. The system prompts students to estimate the total number of hours they can dedicate to study. This includes determining the preferred length and frequency of study sessions that fit their schedule and learning style. By considering these factors, the system optimizes the allocation of learning resources that are most relevant and that most benefit out of it to student and ensures that the lesson plan is feasible within the student's time constraints. To achieve this algorithm utilizes Jupyter notebook metadata.

The tool addresses the challenge of dynamically selecting the most pertinent learning objects based on their prerequisites and estimated time requirements. Each notebook cell represents a learning object with defined prerequisite cell and time estimates. The generator employs a precedence-based knapsack algorithm to optimize the selection of cells that fit within specified time constraints, enabling the creation of tailored lesson plans that maximize the benefit out of the learning outcome. By automatically synthesizing lesson plans through interactive Jupyter notebooks, the generator enhances educational efficiency by prioritizing relevant concepts and ensuring comprehensive coverage of learning objectives. This personalized approach not only enhances student engagement but also fosters a more efficient and productive learning experience tailored to individual preferences and capabilities.

4.3 Precedence-Based Knapsack Problem in Educational Modules

The knapsack problem is a well-known optimization problem in which a set of items, each with a weight and a value, must be selected to maximize the total value without exceeding a specified weight limit [1]. [53] [54]The precedence-based knapsack problem is a variant that introduces additional complexity by requiring certain items to be included only if specific other items are also included. These precedence constraints add a layer of dependency among items, making the problem more challenging and applicable to real-world scenarios where certain tasks or items must follow a particular order [55].

In the context of educational modules, we can apply the precedence-based knapsack problem to optimize the learning experience based on the time a student can dedicate to studying. Here, the total weight corresponds to the total time the student can allocate, each concept has some prerequisite concepts as precedence constraints, and the weight of each concept item is the total time needed to complete that concept.

4.3.1 Problem Statement

Given: A set of concepts, each with a required time to complete t_i and an educational profit value p_i.

The total available time T that the student can dedicate.

A set of precedence constraints where if concept i is prerequisite for concept j, then concept j can only be included if concept i is also included.

The objective is to maximize the total educational value within the given time, ensuring all prerequisites constraints are satisfied.

4.3.2 Mathematical Formulation

Let x_i be a binary variable that represents whether concept i is included in the study plan ($x_i = 1$) or not ($x_i = 0$).

The objective is to maximize the total educational value:

$$Maximize \sum_{i=1}^{n} p_i \, x_i$$

Subject to the constraints:

- Time constraint:

$$\sum_{i=1}^{n} t_i \, x_i \leq T$$

- Precedence prerequisites constraints:

$$x_j \leq x_i \text{ for each precedence prerequisite pair } (i, j)$$

4.4 Course Structure and Dependency mapping.

4.4.1 Course Structure

The course is structured into two units, with each unit comprising several chapters represented in Jupyter Notebooks. Each chapter begins by outlining specific learning outcomes using Bloom's Taxonomy, ensuring clarity and alignment with educational goals. These learning outcomes serve as the guiding framework for the chapter's content, which is organized into individual notebook cells. Each cell encapsulates a distinct topic or concept relevant to the chapter's objectives. This modular approach allows for granular control over the sequencing and presentation of educational content, facilitating a structured and comprehensive learning experience. By associating learning outcomes directly with specific topics within the notebooks, the course framework ensures coherence and relevance, guiding students through a systematic exploration of key concepts in an interactive and engaging manner.

4.4.2 Cell Metadata

Each cell in Jupyter Notebook can be enriched with metadata to specify its purpose, learning objectives, and additional details. These fields provide context and educational goals specific to the content within the cell [7]. The metadata includes the following details:

Table 4.1Cell Metadata parameters

Metadata	Description
Cell_Id	Unique identifier for each cell in the Notebook.
Cell_alternates	Alternative cell that performs the same action but with different visualization.
Cell_concepts	Concepts involved in the cell, detailing the core learning content.

Cell_estimated_time	Estimated time required to learn the concept presented in the cell.
Cell_interactive	Indicates if the cell contains interactive elements or activities.
Cell_outcomes	Expected learning outcomes from studying the cell's content.
Cell_prereqs	Identifies the Cell_Id of prerequisite cells that should be learned before this cell.
Cell_type	Specifies the type of content within the cell, such as text, quiz, or interactive elements.
Module_outcomes	Overall learning outcomes associated with the module that includes the cell.
Module_prereqs	Prerequisites necessary to begin studying the module that includes the cell.
Module_title	Title of the module within which the cell is situated.

Sample metadata for a cell is as follows:


```
"cell_details": {
        "cell_ID": "m10-ClassicalBits",
        "cell_alternates": [],
        "cell_concepts": [ "Classical Bits"],
        "cell_estimated_time": "2",
        "cell_interactive": "false",
        "cell_outcomes": [
            "Understand the concept of classical bits as a two-
dimensional system",
            "Learn different physical representations of bits",
            "Grasp how to represent the knowledge of a system's
state using probability vectors."
        ],
        "cell_prereqs": [ "m9-ProbabilisticModel"],
        "cell_type": ["text"],
        "module_outcomes": [
            "Understand the basics of classical computing systems
and their representation",
            "Apply the concept of probability vectors to
represent states of a classical bit",
            "Differentiate between classical and quantum systems
in terms of state representation and physical realization"
        ],
        "module_prereqs": [
            "Linear Algebra",
            "Complex Numbers",
            "Introduction to Quantum Mechanics"
        ],
        "module_title": [
            "Basics of Quantum Computing/Cryptography"
        ]
    }
```

4.4.3 Module Metadata

Metadata at the lesson level can provide overarching information and objectives for the entire lesson plan. At the lesson level, metadata encompasses broader details. The Notebook/Module Metadata contains below details:

Table 4.2 Module metadata Parameters

Module Metadata	Description
module_concepts	Concepts covered by the module, providing an overview of the main topics addressed.
module_outcomes	Learning outcomes associated with the module, specifying the skills or knowledge to be gained.
module_outcomes_mapping	Mapping of specific cell outcomes to overall module outcomes, illustrating their alignment.
module_prereqs	Prerequisites required for students to have before beginning the module.
module_title	Title of the module, summarizing its content and focus within the course structure.

Sample notebook metadata is as follows:

```
    "module_details": {
        "module_concepts": [ "Classical Bits", "Qubits", "Ket
Notation", "Qubits and Measurement", "Linear Combination”,
"Superposition", "Basis" ],
        "module_outcomes": [
            "Students will be able to model two level quantum systems
using vector and ket representations.",
            "Students will understand the notions qubit, amplitude
and probability of collapse.",
            "Students will understand the notion of superposition.",
            "Students will be able to change between basis
representations for a given qubit.",
            "Students will be able to determine if it is a
Superposition or Not Superposition"
        ],
"module_outcomes_mapping": [
            [
                "m10-ClassicalBits",
                "m10-Qubits",
                "m10-KetNotation"
            ],
            [
                "m10-QubitsAndMeasurement",
                "m10-LinearCombinationAndExample",
                "m10-quiz-10.1",
                "m10-quiz-10.1-interactive"
            ],
```

```
            [
                "m10-Superposition"
            ],
            [
                "m10-Basis"
            ],
            [
                "m10-SuperpositionOrNotSuperposition",
                "m10-quiz-10.2",
                "m10-quiz-10.2-interactive"
            ]
        ],
        "module_prereqs": [
            "Linear Algebra",
            "Complex Numbers",
            "Introduction to Quantum Mechanics"
        ],
        "module_title": "Basics of Quantum Computing/Cryptography"
    }
```


4.4.4 Dependency Mapping

Each module has predefined learning outcomes, which are specified in the first cell and within the module_outcomes key in the module metadata. All cells corresponding to a particular learning outcome are mapped to the same index element in the module_outcomes_mapping key. This mapping allows for the selection of a specific learning outcome and the corresponding cells. At the cell level, each cell can have prerequisite cells that also need to be included in the lesson plan to achieve the learning outcome. Using this information, the selection process ensures that all relevant cells and their prerequisites are incorporated to fulfill the desired learning objectives. By defining learning outcomes and mapping cells to these outcomes, students can follow a structured pathway through the material. This ensures they encounter the content in a logical order that builds on prior knowledge. Sample dependency mapping for a module is as follows:

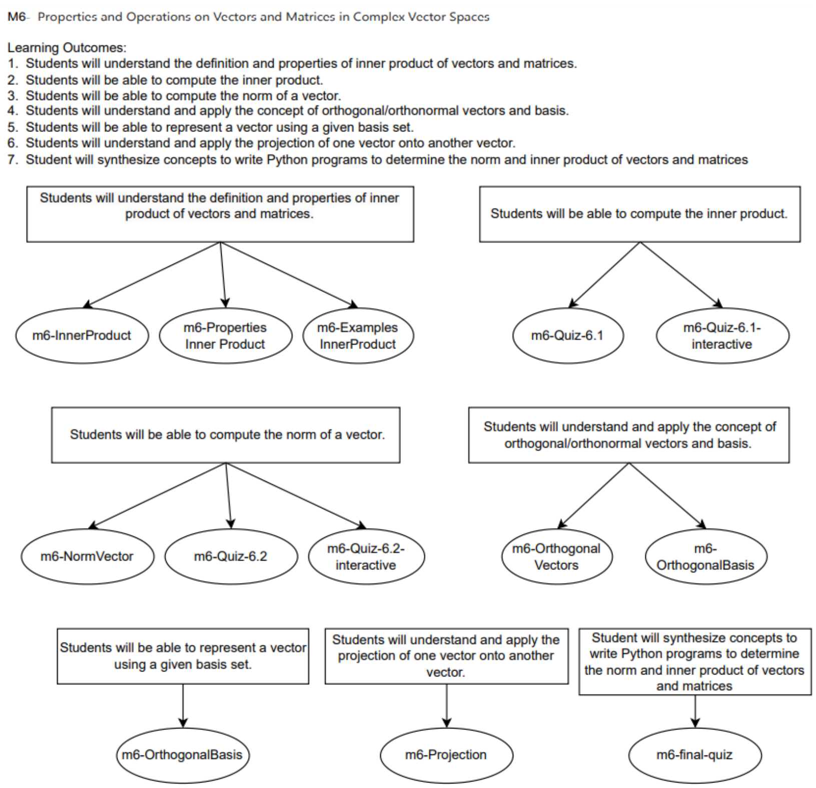


Figure 4.1Sample dependency mapping for a module

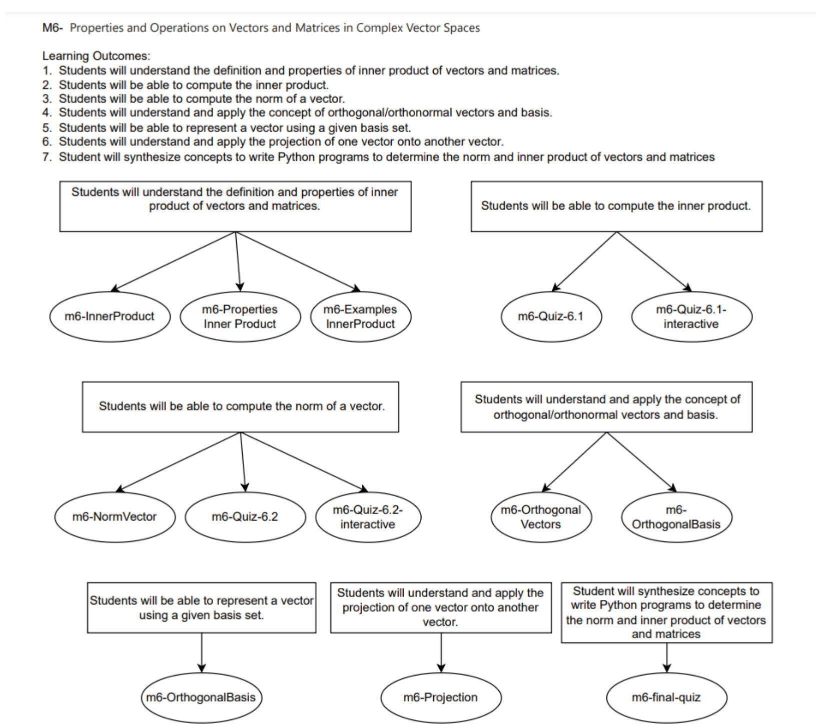


Figure 4.2 Sample learning outcome mapping for a module

4.5 Dynamic Lesson Plan Generation

The Dynamic Lesson Plan Generator utilizes metadata information to create lesson plans that satisfy constraints such as removing prerequisite concepts and accommodating time availability. This is achieved using a Precedence-Controlled Fractional Knapsack Algorithm.

4.5.1 Precedence Controlled Fractional Knapsack Algorithm

The Precedence Controlled Fractional Knapsack Algorithm is utilized in the lesson plan generator to efficiently allocate learning objectives based on their priorities and constraints. This algorithm optimizes the selection process to maximize the educational value of the lesson plan within given limitations.

4.5.2 Algorithm: Dynamic Lesson Plan Generator

Algorithm 2: Dynamic Lesson Plan Generator

Inputs

- **selected_concepts:** List of selected concepts
- **total_time:** Total available time
- **pre_req_concepts:** List of prerequisite concepts

Outputs

- **selected_notebook_cells:** List of selected notebook cells

Algorithm

- **Calculate Total Profit for Each Node**

  Function: dfsWeightCalc(cellId, curr_profit)

  Input: cellId, curr_profit

  Output: totlfracProfit

Process:

Initialize totlfracProfit to 0.

Get neighbours of cellId from cell_prereqs.

For each neighbour in neighbours:

    Initialize fracProfit to 0.

    Get neighbour_cell.

    If neighbour not in visited and neighbour_cell.concepts not in pre_req_concepts:

        Add neighbour to visited.

        Set next_profit to curr_profit.

        Calculate fracProfit as next_profit / estimated_time + dfsWeightCalc(neighbour, next_profit).

        Set neighbour_cell.profit to next_profit.

        Set neighbour_cell.fracProfit to fracProfit.

        Add fracProfit to totlfracProfit.

    Else if neighbour in visited:

        Add neighbour_cell.fracProfit to totlfracProfit.

    Return totlfracProfit.

For each concept in selected_concepts:

    Get concept_cell for concept.

    If concept not in visited and concept_cell.concepts not in pre_req_concepts:

        Add concept to visited.

Set curr_profit to 1.

Calculate fracProfit using dfsWeightCalc(concept, curr_profit).

Set concept_cell.profit to curr_profit.

Set concept_cell.fracProfit to fracProfit + curr_profit / estimated_time.

- **Select Cells Based on Available Time**

Function: dfs(cellId, total_time)

Input: cellId, total_time

Output: Updated total_time

Process:

Get neighbours of cellId from cell_prereqs.

Create neighbour_cells list from neighbours.

Sort neighbour_cells by fracProfit in descending order.

For each neighbour_cell in neighbour_cells:

Get neighbour from neighbour_cell.

If total_time > 0 and neighbour not in seen and neighbour_cell.concepts not in pre_req_concepts:

Add neighbour to seen.

total_time = total_time + dfs(neighbour, total_time).

Get cell_time from neighbour_cell.estimated_time.

If cell_time <= total_time:

Subtract cell_time from total_time.

```
                Add neighbour_cell to selected_notebook_cells.
    Return total_time.
    Create concept_cells list from selected_concepts.
    Sort concept_cells by fracProfit in descending order.
    For each concept_cell in concept_cells:
        Get concept from concept_cell.
        If total_time > 0 and concept not in seen and
        concept_cell.concepts not in pre_req_concepts:
            Add concept to seen.
            total_time = total_time + dfs(concept, total_time).
            Get cell_time from concept_cell.estimated_time.
            If cell_time <= total_time:
                Subtract cell_time from total_time.
                Add concept_cell to selected_notebook_cells.
```

4.5.3 Algorithm Explanation: Dynamic Lesson Plan Generator

4.5.3.1 Calculating Total Profit for Each Node

- **Initialize Total Profit**:
    - Set the total fractional profit to zero.
- **Depth-First Search with Backtracking**:
    - For the current cell, retrieve its neighbouring nodes (child cells).
    - Iterate through the neighbouring nodes.
- **Process Each Neighbour**:

    - Initialize the fractional profit to zero.
    - Retrieve the neighbour’s details.
    - If the neighbour hasn't been visited and its concept is not a prerequisite:
        - Mark the neighbour as visited.
        - Set the next profit to the current profit.
        - Calculate the fractional profit as the sum of the next profit divided by the estimated time and the recursive call to the profit calculation function.
        - Assign the calculated profit values to the neighbour cell.
        - Add the fractional profit to the total fractional profit.
    - If the neighbour has already been visited:
        - Add its fractional profit to the total fractional profit.
    - Return the total fractional profit.

- **Loop Through Selected Concepts**:
    - For each selected concept, retrieve the corresponding cell details.
    - If the concept hasn't been visited and its concept is not a prerequisite:
        - Mark it as visited.
        - Set the current profit to one.
        - Calculate the fractional profit using the profit calculation function.
        - Assign the calculated profit values to the concept cell.

4.5.3.2 Selecting Cells Based on Available Time

- **Initialize Selection Process**:
  - After calculating the total profit for each node, select cells based on available time using another depth-first search function.
- **Retrieve and Sort Neighbouring Nodes**:
  - Retrieve the neighbouring nodes of the current cell.
  - Create a list of neighbour cells.
  - Sort these cells by their fractional profit in descending order.
- **Iterate Through Neighbour Cells**:
  - For each neighbour cell, retrieve the neighbour's identifier.
  - Check if the available time is greater than zero.
  - If the neighbour hasn't been seen and its concept is not a prerequisite:
    - Mark the neighbour as seen.
    - Update the available time by recursively calling the cell selection function.
- **Select Cells Based on Time**:
  - Retrieve the cell time from the neighbour cell's estimated time.
  - If the cell time is less than or equal to the available time:
    - Subtract the cell time from the available time.
    - Add the neighbour cell to the list of selected cells.
- **Finalize Cell Selection**:
  - Create a list of concept cells from the selected concepts.

- Sort them by their fractional profit in descending order.
- Iterate through each concept cell, retrieving the concept identifier.
- Check if the available time is greater than zero.
- If the concept hasn't been seen and its concept is not a prerequisite:
    - Mark it as seen.
    - Update the available time by calling the cell selection function.
    - Retrieve the cell time from the concept cell's estimated time.
    - If the cell time is less than or equal to the available time:
        - Subtract the cell time from the available time.
        - Add the concept cell to the list of selected cells.

In summary, this algorithm ensures that the most relevant and beneficial cells are selected for the lesson plan, optimizing for both the available time and the prerequisite concepts required to achieve the learning outcomes.

4.6 Generator User Interface Design

The user interface (UI) of the lesson plan generator is designed to be intuitive and user-friendly, providing several interactive elements to customize the learning experience:

- **Checkbox List with Search Functionality**: The UI includes a checkbox list that displays all available concepts. Users can also search through these concepts and select the ones they already know, allowing the generator to ignore these topics when creating the lesson plan.
- **Accordion Selection for Modules**: Each module is represented as an accordion, which users can expand to view and select the specific learning outcomes they wish to achieve.

Users can select one or multiple learning outcomes within each module, providing flexibility in tailoring their learning journey.

- **Timing Constraint Input**: To accommodate users' time availability, the UI features an input field where users can enter their estimated available time. This ensures that the generated lesson plan fits within their specified time schedule, making the learning process efficient and manageable.

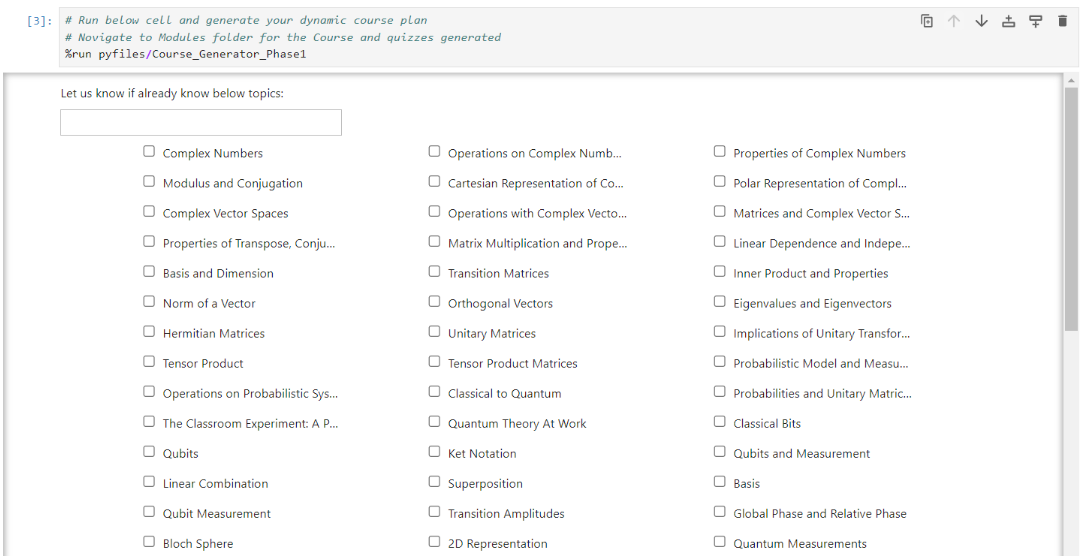


Figure 4.3 Notebook Generator Prerequisite Selection Screen

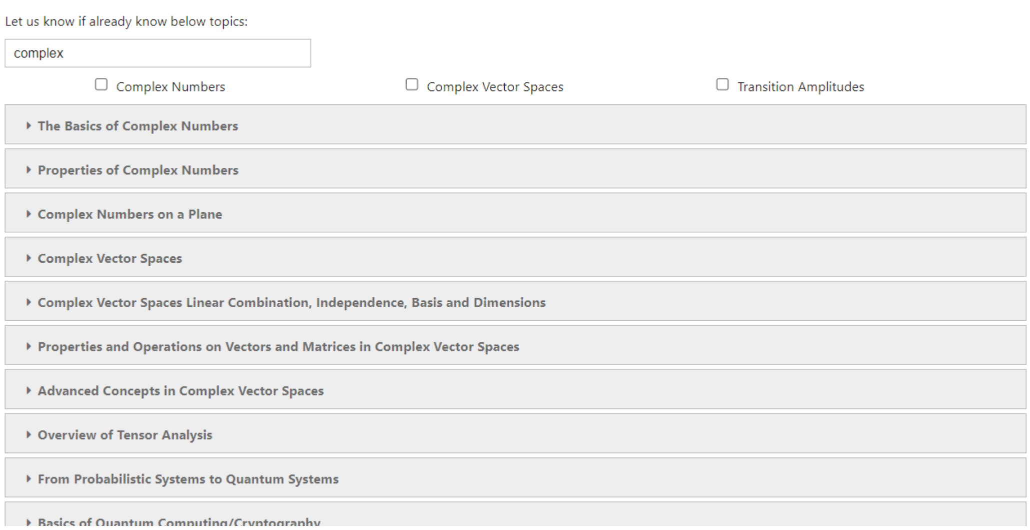


Figure 4.4 Notebook Generator Prerequisite filtering screen

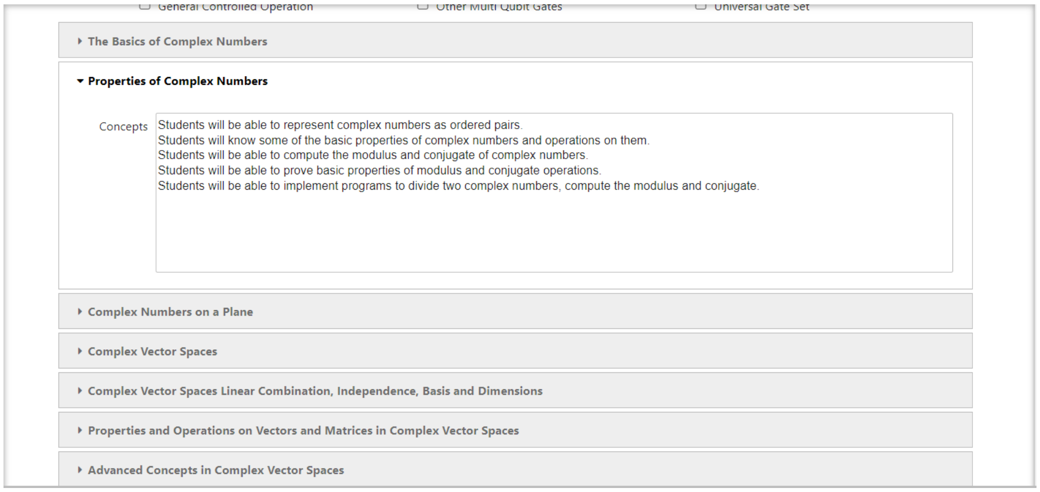


Figure 4.5 Notebook Generator Outcome selection screen

The combination of these features allows users to personalize their lesson plans based on their prior knowledge, learning goals, and time constraints, ensuring a more effective and customized educational experience.

4.6.1 Output

The lesson plan generator produces two key outputs:

- **Console Output:** The generated lesson plan displays the cell IDs of the selected cells in the console, providing a quick overview of the included content.
- **Notebook File:** A Jupyter Notebook file is created containing all the content from the selected cells. This file provides a structured and interactive format for the lesson plan, integrating explanations, code, and visualizations in a cohesive manner.

These outputs ensure that users have both a summary of the selected content and a detailed, interactive notebook for their learning activities.

4.7 Implementation Details

4.7.1 Software and Tools Used

The implementation of the lesson plan generator leverages several software tools and technologies to achieve its functionality:

- **Python**: Python programming language is used because of the ease and versatility of various libraries.
- **Jupyter Notebooks**: Jupyter Notebooks provide an interactive development environment.
- **PyWidgets**: PyWidgets, also known as ipywidgets, is a Python library that adds interactive HTML widgets to Jupyter Notebooks.

4.8 Benefits of the Adaptive Lesson Plan Generator

4.8.1 Personalized Learning Experience

The adaptive lesson plan generator enhances the educational experience by tailoring content to individual student needs and preferences. By utilizing metadata and algorithms, the generator customizes lesson plans based on student prerequisites, learning objectives, and available study time. This personalization fosters engagement and improves comprehension as students' progress through adaptive learning paths.

4.8.2 Efficiency and Time Management

Through its algorithmic approach, the lesson plan generator optimizes time management for educators and learners alike. By automating the selection and sequencing of learning objectives, it streamlines curriculum development and delivery. Educators can focus on instructional strategies while students benefit from structured, efficient learning experiences tailored to their pace and proficiency.

4.8.3 Enhanced Learning Outcomes

By mapping learning objectives to educational concepts and prioritizing objectives based on their importance, the generator promotes comprehensive understanding and mastery of subject matter. Adaptive lesson plans enable students to grasp complex concepts gradually, reinforcing knowledge through personalized exercises and assessments. This approach enhances retention and facilitates deeper learning outcomes.

4.9 Challenges and Solutions

4.9.1 Handling Diverse Learning Preferences

Challenge: Accommodating diverse learning preferences and styles poses a challenge in creating universally effective lesson plans.

Solution: The lesson plan generator employs adaptive algorithms that consider individual learning profiles, preferences, and feedback. By offering varied instructional methods—such as visual aids, interactive exercises, and textual explanations—it ensures inclusivity and engagement across diverse student demographics.

4.9.2 Algorithm Efficiency and Scalability

Challenge: Ensuring the efficiency and scalability of algorithms used for generating adaptive lesson plans is crucial to support large-scale educational implementations.

Solution: Continuous optimization and refinement of algorithms, such as the Precedence Controlled Fractional Knapsack Algorithm, enhance performance and scalability. Parallel processing and cloud computing resources further boost efficiency, enabling real-time adaptation of lesson plans to evolve educational needs and technological advancements.

4.9.3 Ensuring Accurate Mapping of Objectives to Concepts

Challenge: Accurately mapping learning objectives to educational concepts is essential for aligning lesson plans with curriculum goals and student learning outcomes.

Solution: The lesson plan generator integrates robust metadata structures that categorize learning objectives by topic, complexity, and prerequisite knowledge. Automated validation and feedback mechanisms ensure coherence and alignment between objectives, concepts, and instructional content, fostering clarity and effectiveness in educational delivery.

By addressing these challenges through innovative solutions, the adaptive lesson plan generator enhances educational practices, empowers educators, and enriches student learning experiences across diverse educational settings.

5 RESULTS AND VALIDATIONS

5.1 Quantum Network Simulator Results

5.1.1 Use Case 1: E91 protocol simulation

- User chooses arrangement mode and selects E91 protocol to simulate transmission

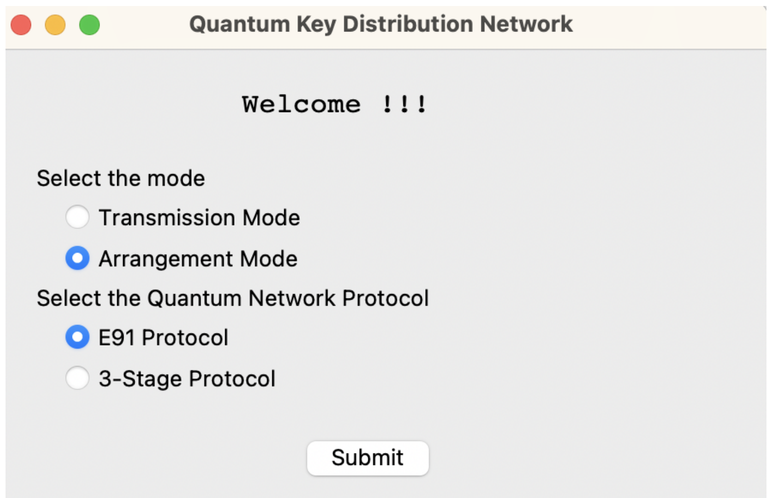


Figure 5.1 E91 protocol Selection

- User rearranges the placement of nodes in arrangement mode

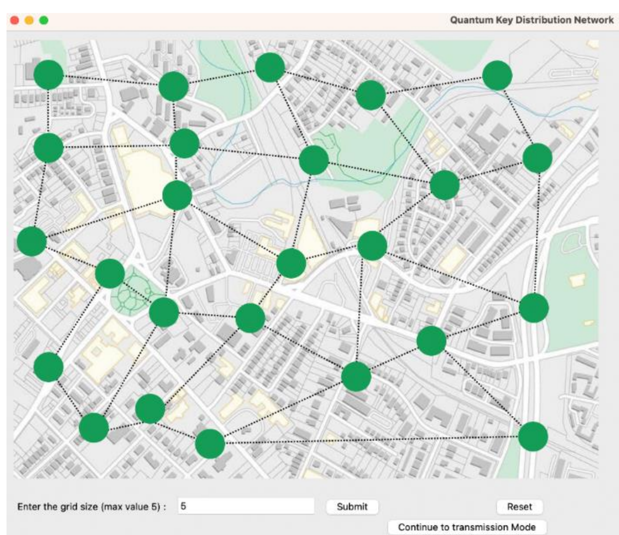


Figure 5.2 E91 protocol rearrangement of nodes placement

- User selects Start Node (Alice) in the transmission mode and a quantum circuit is initialized.

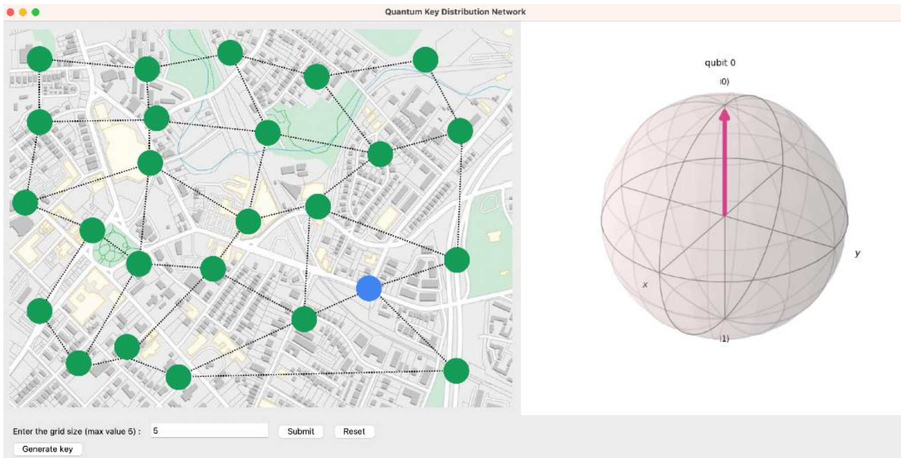


Figure 5.3 E91 protocol - Alice Node selection and quantum circuit initialization

- The user selects the End Node (Bob), and simulator establishes a connection between Alice and Bob along the shortest Path.

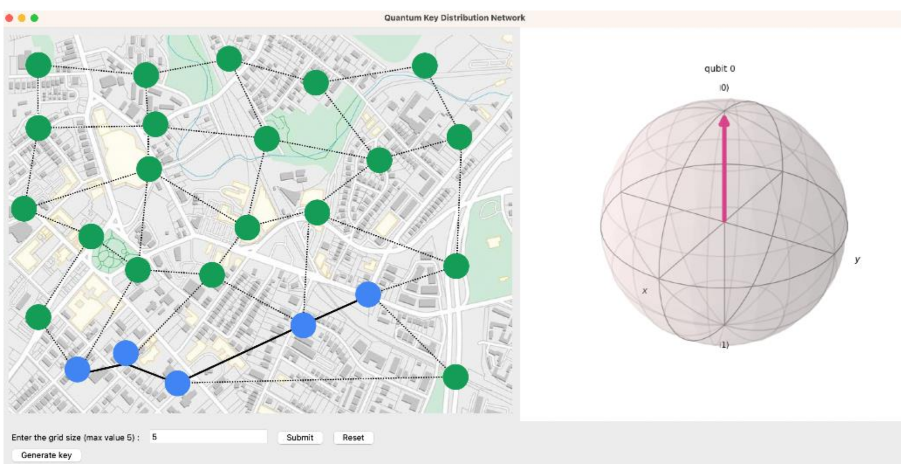


Figure 5.4 E91 protocol - Bob Node selection and shortest Path calculation

- On key generation, qubit transmission is visualized, and the generated key is displayed.

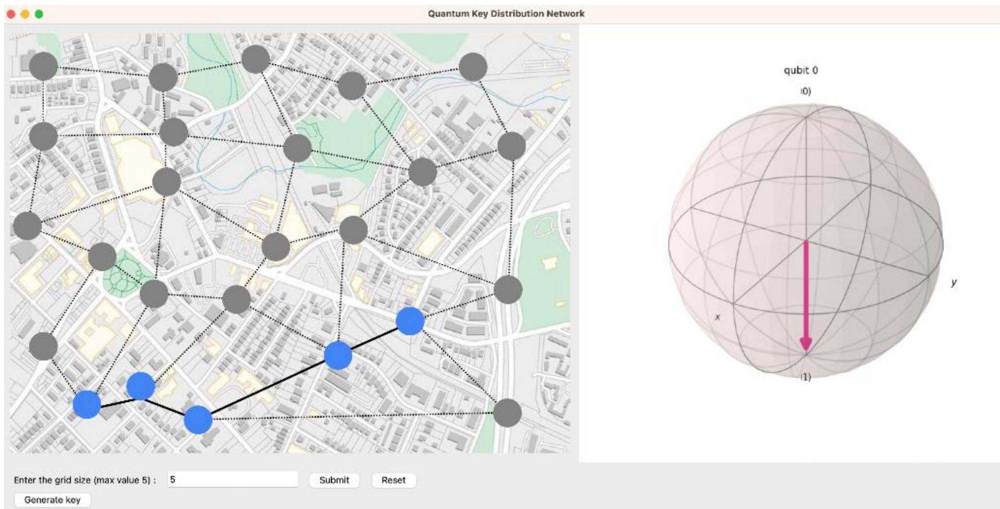


Figure 5.5 Intermediate qubit state while transmission using E91 protocol

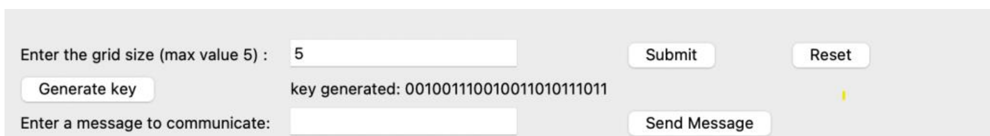


Figure 5.6 Random key generated using E91 protocol

- Message can be transmitted via classical channel and the ASCII value and encoded message strings are displayed to user.

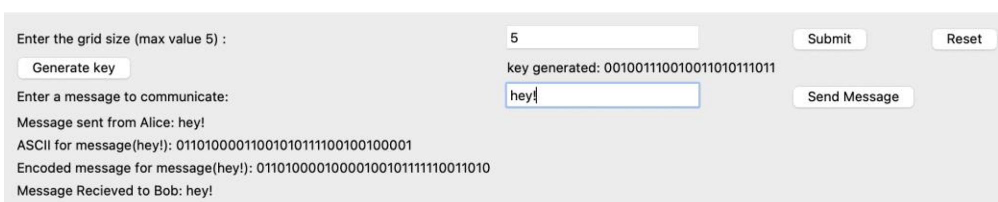


Figure 5.7 Message transmitted using E91 protocol

5.1.2 Use Case 2: 3-Stage protocol simulation

- User chooses transmission mode and selects 3 Stage protocol to simulate transmission.

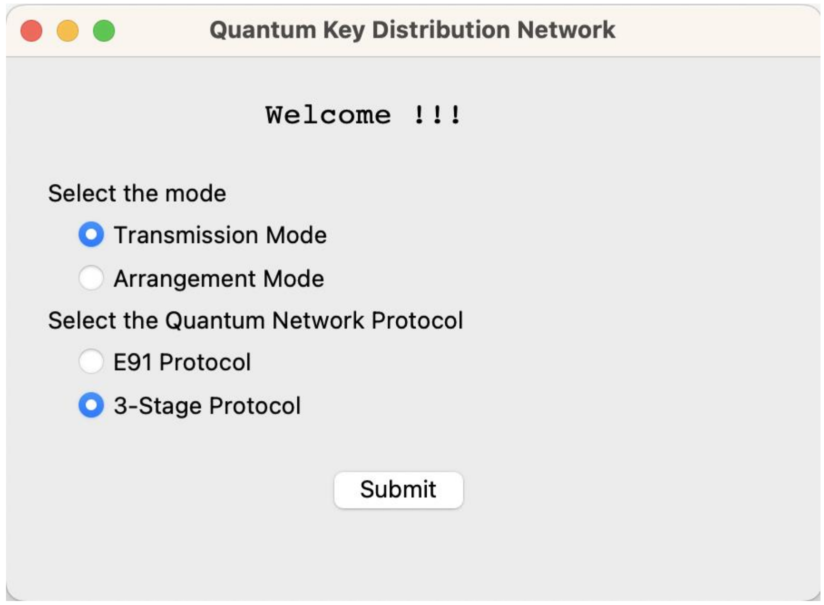


Figure 5.8 3 Stage protocol selection

- User selects the start Node (Alice), the simulator disables all other nodes except the adjacent nodes to start node to navigate user through further selection and quantum circuit is initialized.

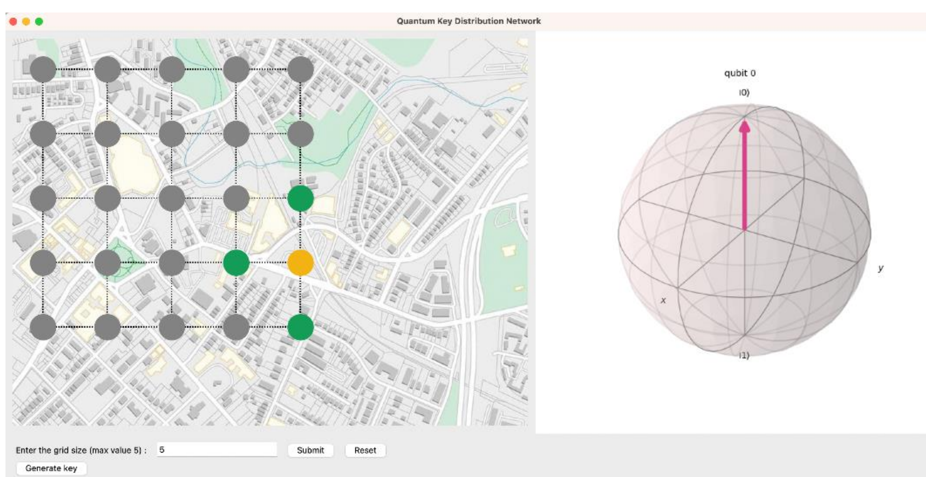


Figure 5.9 3 Stage - Alice Node selection and quantum circuit initialization

- User selects intermediate nodes along the path manually and reaches end node.

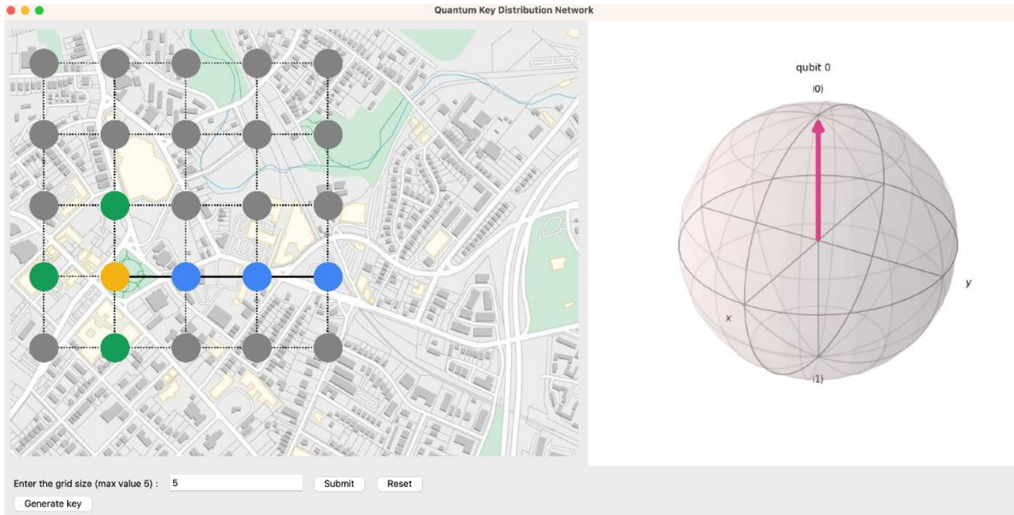


Figure 5.10 3 Stage – Bob Node selection

- On key generation, qubit transmission is visualized. The encoded qubit transmits from Alice to Bob. Bob double encodes it and sends it back to Alice. Alice applies inverse operation and sends back to Bob and generated key is shown to the user.

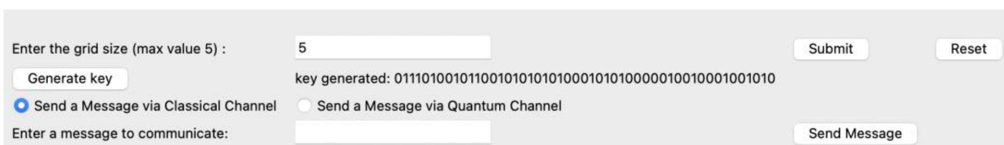


Figure 5.11 Random key generation using 3 stage protocol

- Message transmission via Classical channel uses one-pad encryption and the ASCII value and encoded message is displayed to user.

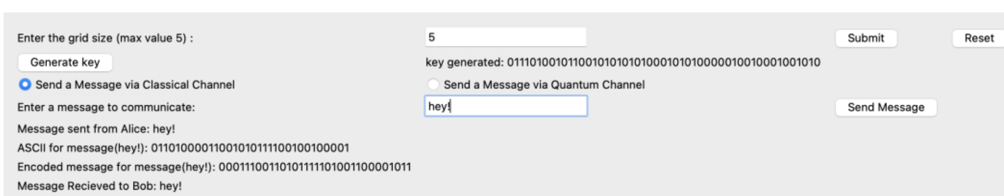


Figure 5.12 Message transmission via classical channel using 3 stage protocol

- Message transmission via Quantum channel uses the same process as key generation and message received at Bob is displayed to user.

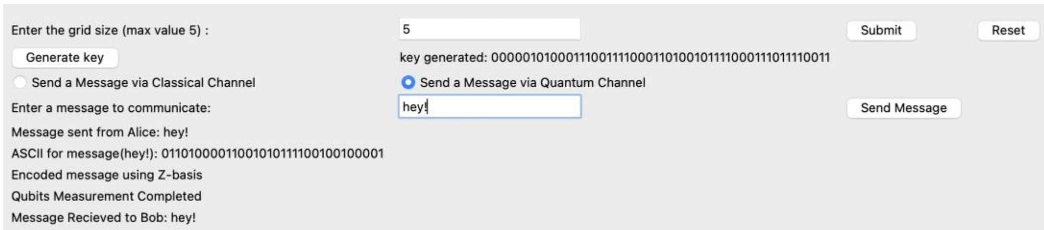


Figure 5.13 Message transmission via Quantum channel using 3 stage protocol

5.2 Notebook generator Results

5.2.1 Use Case 1: Generate lesson with no Constraints

Constraints: no time constraint and no prerequisites.

User does not select any prerequisite concepts as spotted in Figure 5.14, selects the outcome as shown in Figure 5.15 and generate the lesson. A complete lesson to achieve selected outcome is generated as represented in Figure 5.16 and Figure 5.17

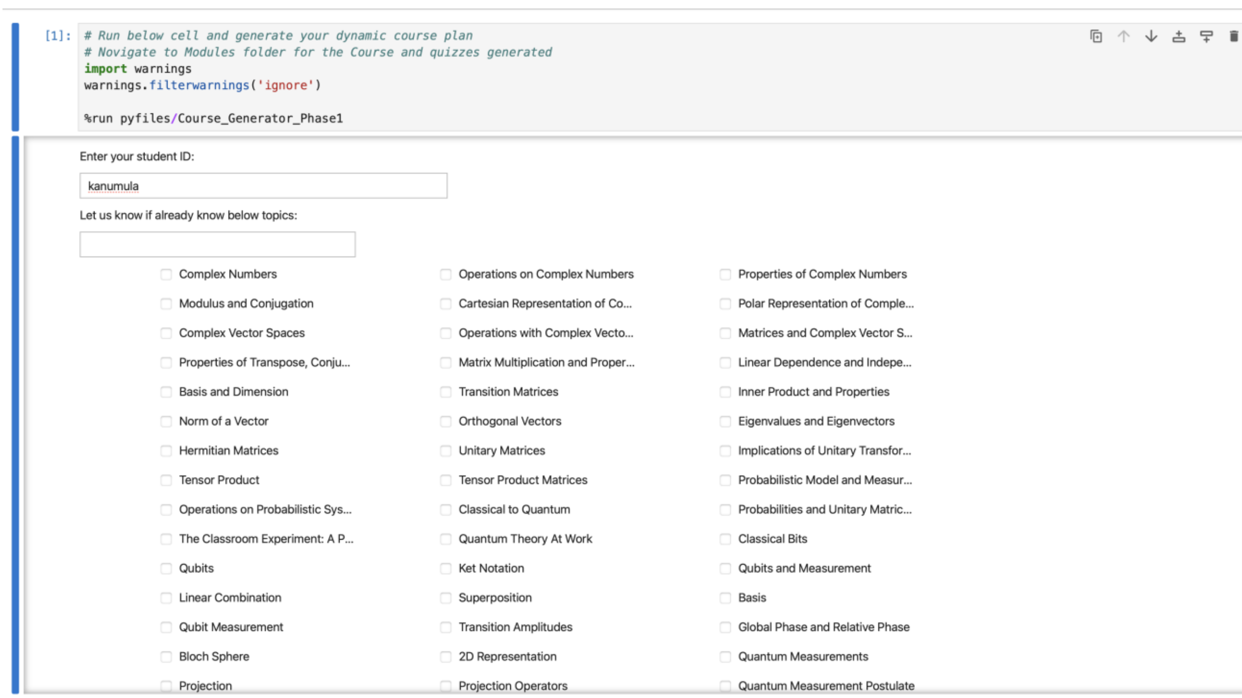


Figure 5.14 Generate Lesson Plan with no Constraints – No Prerequisite Selection

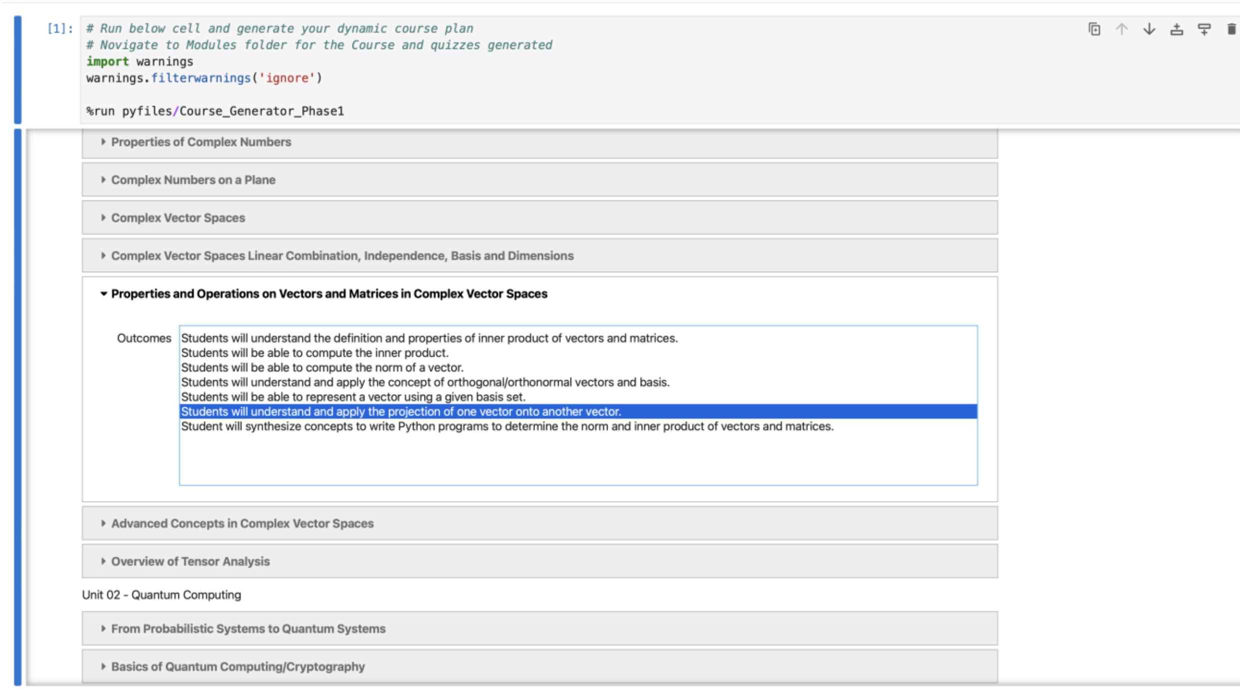


Figure 5.15 Generate Lesson Plan with no Constraints – Outcome Selection

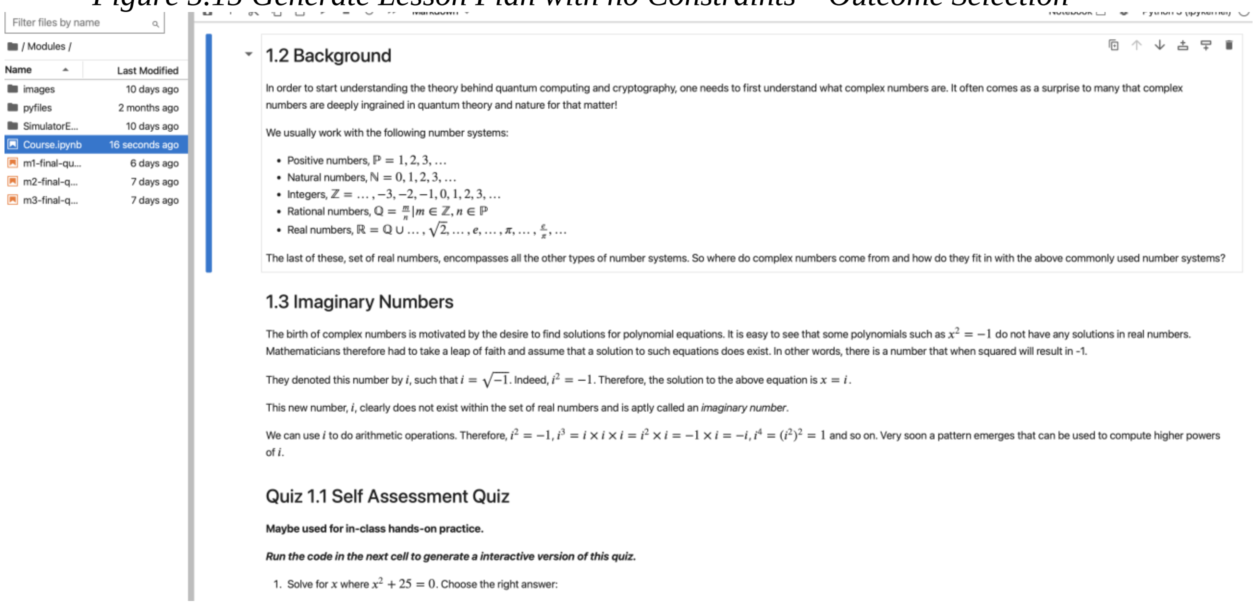


Figure 5.16 Generate Lesson Plan with no Constraints – Complete Lesson Begin Screen

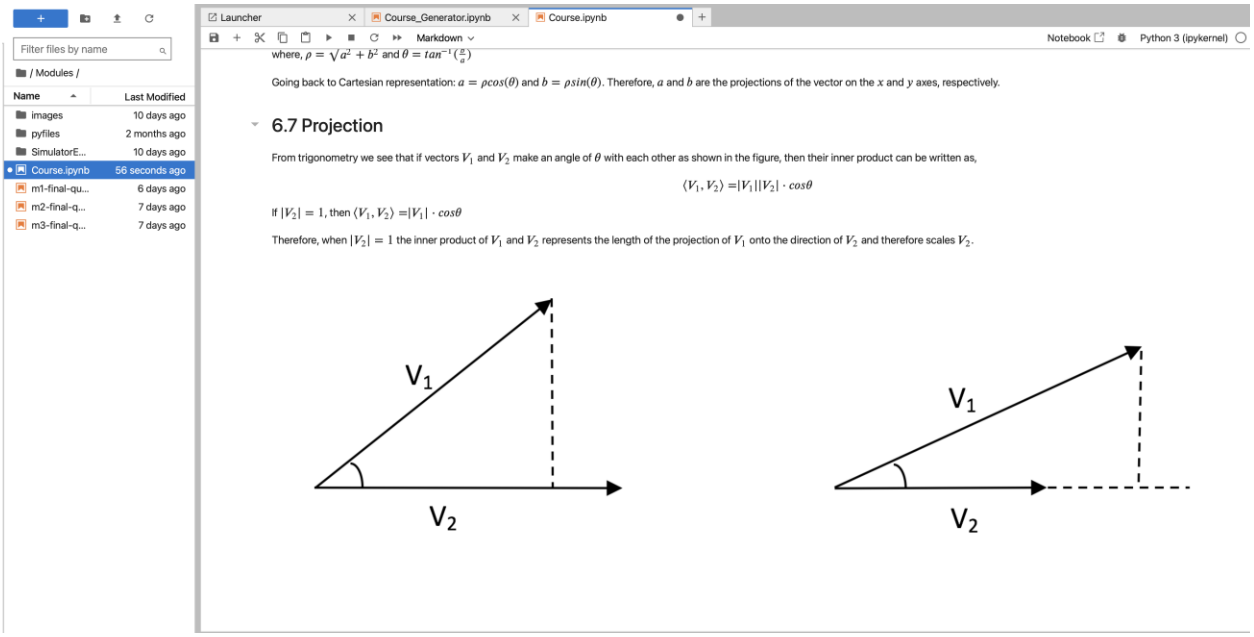


Figure 5.17 Generate Lesson Plan with no Constraints – Complete Lesson Plan End Screen

5.2.2 Use Case 2: Generate Lesson Plan with Time Constraint

Constraints: Timeframe shorter than estimated time and no prerequisites.

Total time needed to complete the outcome: 78 minutes

Provided estimated time: 30 minutes

User does not select any prerequisite concepts as spotted in Figure 5.14, selects the outcome as shown in Figure 5.18 and provide the estimated time in mins (30 mins) as depicted in Figure 5.19 and generate the lesson. A partial lesson to achieve partial outcome is generated as represented in Figure 5.20 and Figure 5.21

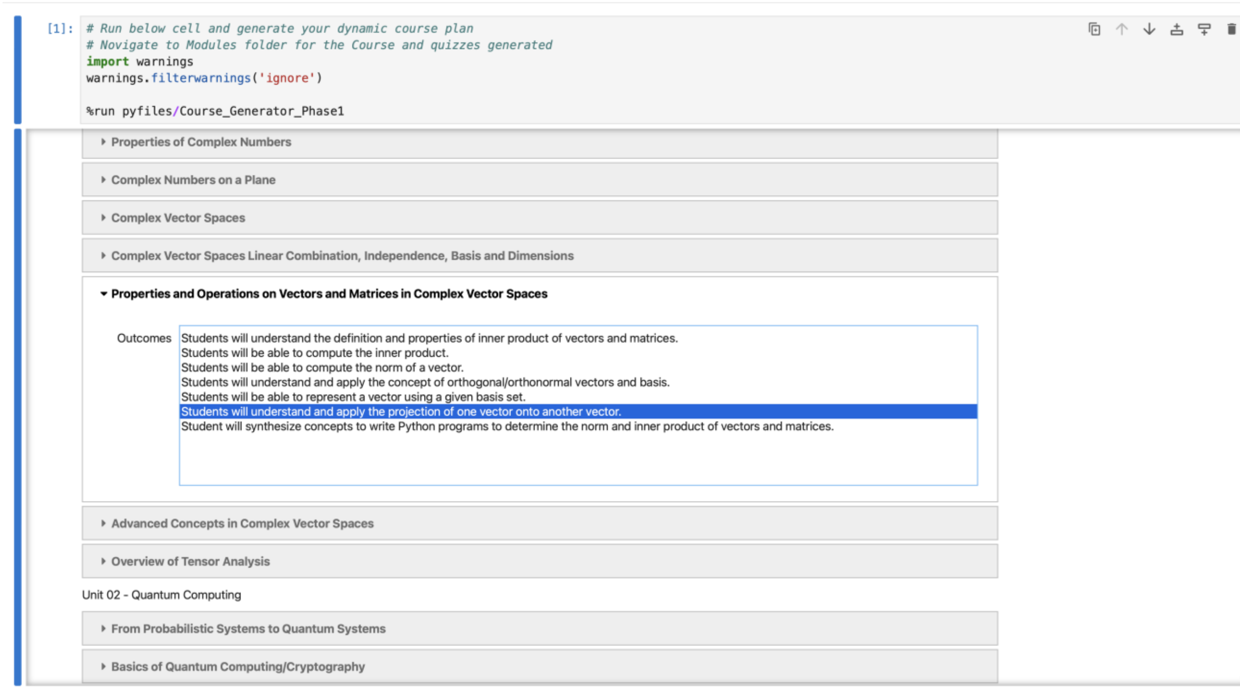


Figure 5.18 Generate Lesson Plan with Time Constraints – Outcome Selection

Figure 5.19 Generate Lesson Plan with Time Constraints – Estimated Time and Submission

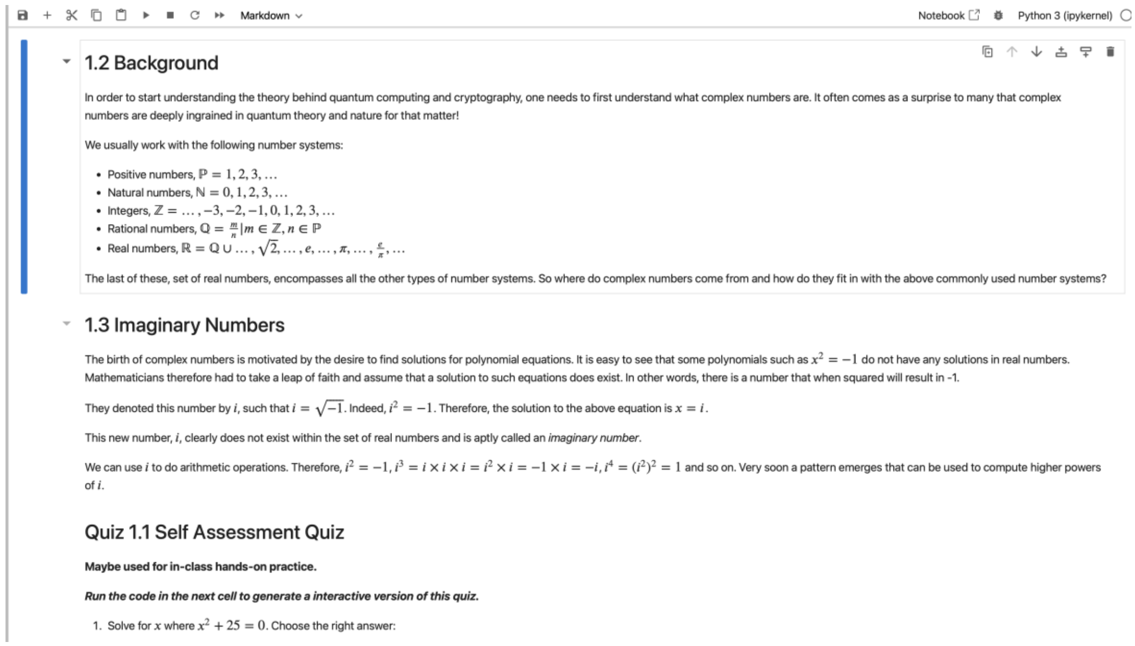


Figure 5.20 Generate Lesson Plan with Time Constraints – Partial Lesson Start Screen

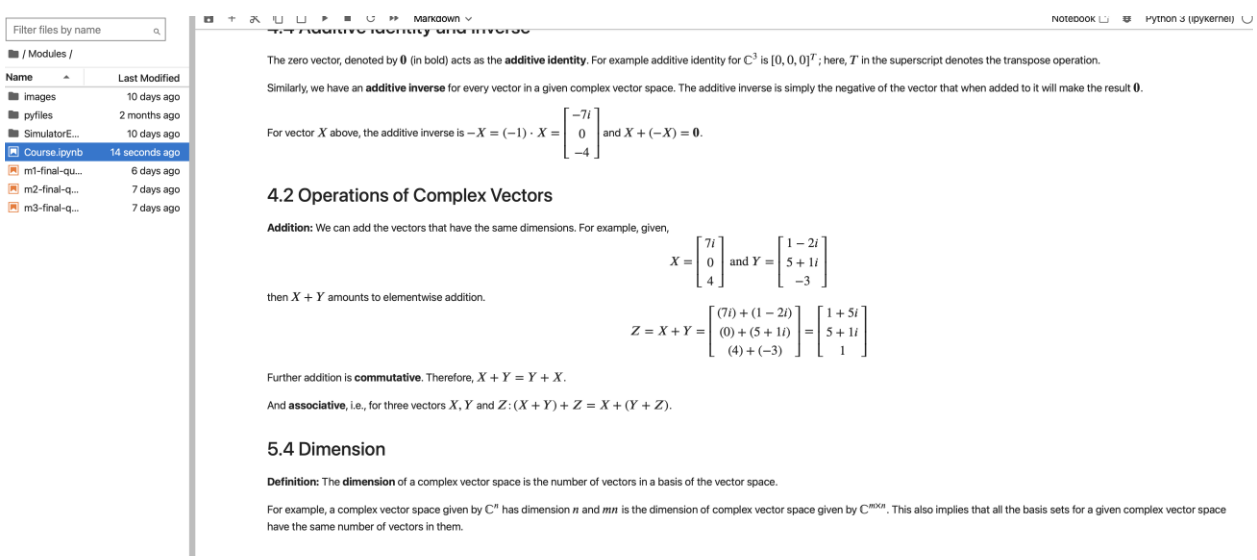


Figure 5.21 Generate Lesson Plan with Time Constraints – Partial Lesson End Screen

5.2.3 Use Case 3: Generate Lesson Plan with Prerequisite Constraint

Constraints: ‘complex numbers’ as prerequisite concept.

Total time needed to complete the outcome: 68 minutes

Provided estimated time: No time provided

User selects ‘complex numbers’ concept as prerequisite concepts that they already have a knowledge on the concept as witnessed in Figure 5.22, selects the outcome generate the lesson. A lesson generated ignoring the ‘complex numbers’ concepts as represented in Figure 5.23

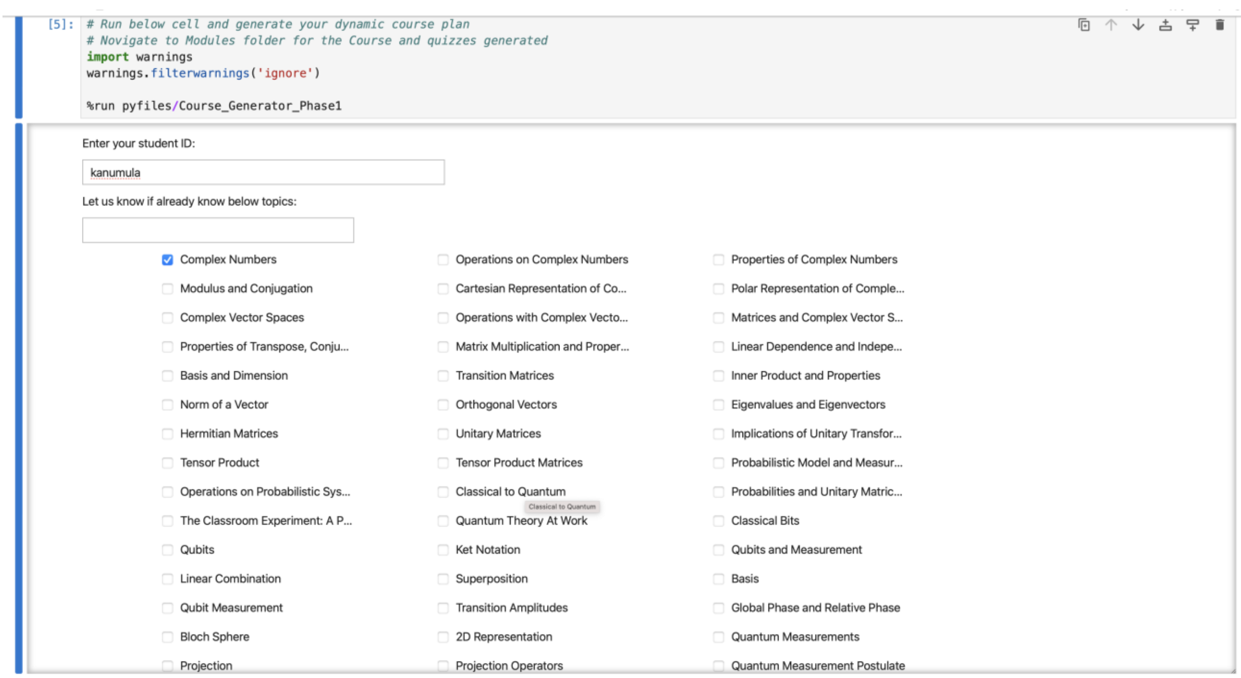
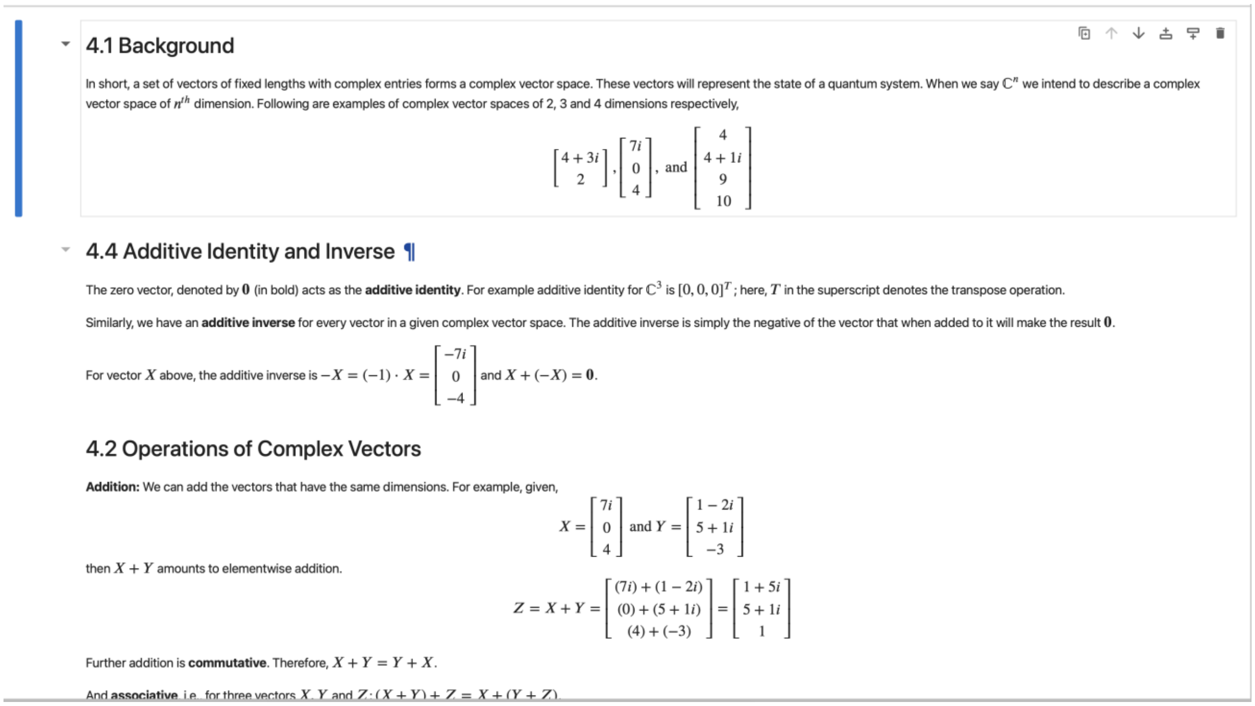


Figure 5.22 Generate Lesson Plan with Prerequisite Constraints – Complex Numbers Prerequisite Selection



4.1 Background

In short, a set of vectors of fixed lengths with complex entries forms a complex vector space. These vectors will represent the state of a quantum system. When we say $\mathbb{C}^n$ we intend to describe a complex vector space of n^{th} dimension. Following are examples of complex vector spaces of 2, 3 and 4 dimensions respectively,

$$\begin{bmatrix} 4+3i \\ 2 \end{bmatrix}, \begin{bmatrix} 7i \\ 0 \\ 4 \end{bmatrix}, \text{ and } \begin{bmatrix} 4 \\ 4+1i \\ 9 \\ 10 \end{bmatrix}$$

4.4 Additive Identity and Inverse ¶

The zero vector, denoted by $\mathbf{0}$ (in bold) acts as the **additive identity**. For example additive identity for $\mathbb{C}^3$ is $[0,0,0]^T$; here, T in the superscript denotes the transpose operation.

Similarly, we have an **additive inverse** for every vector in a given complex vector space. The additive inverse is simply the negative of the vector that when added to it will make the result $\mathbf{0}$.

For vector X above, the additive inverse is $-X = (-1)\cdot X = \begin{bmatrix} -7i \\ 0 \\ -4 \end{bmatrix}$ and $X + (-X) = \mathbf{0}$.

4.2 Operations of Complex Vectors

Addition: We can add the vectors that have the same dimensions. For example, given,

$$X = \begin{bmatrix} 7i \\ 0 \\ 4 \end{bmatrix} \text{ and } Y = \begin{bmatrix} 1-2i \\ 5+1i \\ -3 \end{bmatrix}$$

then $X + Y$ amounts to elementwise addition.

$$Z = X + Y = \begin{bmatrix} (7i)+(1-2i) \\ (0)+(5+1i) \\ (4)+(-3) \end{bmatrix} = \begin{bmatrix} 1+5i \\ 5+1i \\ 1 \end{bmatrix}$$

Further addition is **commutative**. Therefore, $X + Y = Y + X$.

And **associative**, i.e., for three vectors X, Y and Z: $(X+Y)+Z = X+(Y+Z)$.

Figure 5.23 Generate Lesson Plan with Prerequisite Constraints – Partial Lesson Start Screen without Complex Numbers Concepts

5.3 Feasibility Test Results

To validate this approach, a feasibility test was conducted with a group of students. These students used the tool and provided feedback on their experiences. The results were overwhelmingly positive. Students appreciated the personalized learning paths and found that the interactive quizzes and simulators significantly enhanced their understanding of complex concepts. They reported increased engagement and a better alignment of their study time with their learning goals.

Student Feedback

Positive Feedback:

- The personalized learning feature received high praise, with students expressing eagerness to use it regularly.

- Students found that the interactive quizzes and simulators significantly enhanced their understanding of complex concepts.
- Students especially appreciated the time and prerequisite constraint solution, as it allows them to generate lesson plans according to their needs.
- Students appreciated the overall approach and found it engaging.
- The interactive nature of the tool, especially the simulator visualizing key generation using E91 and 3-stage protocols, was particularly well-received.

Constructive Feedback:

- Students expressed the need for a more interactive outcome selection process.
- A quiz-driven approach was suggested to allow users to test their knowledge effectively. Based on quiz results, students recommended incorporating personalized content to enhance learning outcomes.

6 FUTURE WORK

In this section, we discuss some of the possible improvements to the tool developed in this thesis. Some future works incorporate the suggestions given during the feasibility testing stage.

Enhanced Interactive Features:

- Develop a more interactive outcome selection process to provide a dynamic and engaging learning experience.
- Explore additional interactive elements that could be integrated into the learning platform to further enhance student engagement.

Quiz-Driven Learning Paths:

- Implement a feature to allow users to attempt quizzes and test their knowledge on the topics.
- Use quiz results to create personalized learning paths, ensuring that content is tailored to the individual needs and knowledge levels of each student.

Advanced Personalization:

- Continue to refine and expand the personalized learning feature based on ongoing student feedback and usage data.
- Incorporate adaptive learning technologies to further personalize the educational experience, adjusting the difficulty and content based on real-time student performance.

Time and Prerequisite Constraint Solutions:

- Improve the time and prerequisite constraint solution to provide even more accurate and user-friendly lesson plans.
- Integrate advanced scheduling algorithms like using dynamic programming to optimize lesson planning based on student availability and prior knowledge.

User Feedback Integration:

- Establish a continuous feedback loop with students to gather insights and suggestions for ongoing improvements.
- Regularly update the platform based on user feedback to maintain high levels of satisfaction and effectiveness.

Scalability and Accessibility:

- Work on scaling the platform to accommodate more users without compromising performance.
- Ensure that the platform is accessible to students with diverse needs and abilities, following best practices in inclusive design.

Expanding Feasibility to Other Courses:

- Extend the tool's capabilities to support a wider range of courses beyond quantum cybersecurity, ensuring the system is versatile and applicable to various fields of study.
- Collaborate with subject matter experts from different disciplines to integrate their courses into the platform, enriching the content library and expanding the tool's applicability.

Integration of Machine Learning Models:

- Investigate the integration of machine learning models to enhance the adaptive learning capabilities of the platform.
- Utilize predictive analytics to anticipate student needs and provide tailored recommendations, improving the overall effectiveness of the learning experience.
- Implement natural language processing (NLP) models to analyse student feedback and interactions, enabling more nuanced and responsive updates to the platform.

- Explore the use of reinforcement learning algorithms to continuously optimize the learning paths based on student progress and performance.

Automated Topic Selection Based on Student Inputs:

- Develop a system to store student inputs and automatically mark those topics as already known, streamlining the learning process.
- Utilize this stored data to adjust future learning paths and recommendations, ensuring that students are not retaught material they have already mastered.
- Continuously update the system based on student interactions and feedback to maintain an up-to-date and relevant knowledge base.

7 REFERENCES

[1] A. Aslan, E. Ursavas and W. Romeijnders, "A precedence constrained knapsack problem with uncertain item weights for personalized learning systems.," *Omega,* vol. 115, 2023.

[2] J. F. Pane and et al., "Informing Progress: Insights on Personalized Learning Implementation and Effects," *Rand Corporation,* 2017.

[3] V. Bommanapally, M. Subramaniam, P. Chundi and A. Parakh, "Navigation Hints in Serious Games," in *Immersive Learning Research Network (iLRN 2018)*, Missoula, 2018.

[4] E. Ostler, A. Parakh and M. Subramaniam, "QuaSim: The development of a virtual simulator for teaching topics in quantum cryptography," in *Society for Information Technology & Teacher Education International Conference*, Washington, DC, 2018.

[5] S. McDermott, S. Vadla, V. Bommanapally, A. Parakh, M. Subramaniam and E. Ostler, "Teaching quantum cryptography using a virtual 3D educator: QuaSim," in *National Cyber Summit (NCS'17)*, Huntsville, 2017.

[6] A. Parakh, M. Subramaniam and E. Ostler, "QuaSim: A virtual quantum cryptography educator," in *IEEE International Conference on Electro Information Technology (EIT)*, Lincoln, 2017.

[7] R. Mallipeddi, C. Schaaf, M. Subramaniam, A. Parakh and S. Weitl-Harms, "A Framework for an Intelligent Adaptive Education Platform for Quantum Cybersecurity," in *10.1109/FIE58773.2023.10343010*, 2023.

[8] A. Parakh, M. Subramaniam and P. Chundi, "A Framework for Incorporating Serious Games into Learning Object Repositories through Experiential Learning," in *HICSS-55*, 2022.

[9] A. Parakh and M. Subramaniam, "Galore: A Platform for Experiential Learning. The Colloquium for Information Systems Security Education," in *CISSE*, 2021.

[10] A. Parakh, V. Bommanapally, P. Chundi and M. Subramaniam, "Quantum Cryptography Exercise Schedules with Concept Dependencies," *The Colloquium for Information Systems Security Education*, 2020.

[11] A. Bommanapally, M. Subramaniam, A. Parakh, P. Chundi and V. M. Puppala, "Learning Objects Based Adaptive Textbooks with Dynamic Traversal for Quantum Cryptography," *Second Workshop on Intelligent Textbooks, In conjunction with 21st International Conference on Artificial Intelligence in Education*, 2020.

[12] S. Vadla, A. Parakh, P. Chundi and M. Subramaniam, "Colloquium for Information Systems Security Education," in *CISSE*, New Orleans, Louisiana, 2018.

[13] V. Bommanapally, M. Subramaniam and A. Parakh, "Embedding a Problem Graph into Serious Games for Efficient Traversal Through Game Space," *IEEE Frontiers in Education Conference (FIE),* pp. 1-5, 2023.

[14] A. Parakh, M. Subramaniam, P. Chundi and E. Ostler, "A Novel Approach for Embedding and Traversing Problems in Serious Games," in *21st Annual ACM SIGITE Conference on Information Technology Education*, 2020.

[15] S. Vadla, A. Parakh, C. P. and M. Subramaniam, "QUASIM: A Multi-dimensional Quantum Cryptography Game for Cyber Security, Journal of The Colloquium for Information Systems Security Education," vol. 6, 2019.

[16] A. Parakh, P. Chundi and M. Subramaniam, "An Approach Towards Designing Problem Networks in Serious Games," in *IEEE Conference on Games (IEEE CoG)*, London, 2019.

[17] A. Dilanga, V. Bommanapally, S. Vadlla, M. Subramaniam, P. Chundi and A. Parakh, "Analyzing and Predicting Player Performance in a Quantum Cryptography Serious Game," in *Games and Learning Alliance conference (GALA 2018)*, 2018.

[18] P. A, "Quantum Computing and Cryptography curriculum," 14 March 2019. [Online]. Available: https://clark.center/users/aparakh.

[19] P. K. Verma, El Rifai M. and K. W. C. Chan, Multi-photon Quantum Secure Communication, Singapore: Springer, 2019.

[20] D. Nelson and A. Parakh, "Quantum Cryptography: Opportunities and Challenges," in *6th Central Area Networking and Security Workshop*, Lawrence, 2014.

[21] N. Jha, A. Parakh and M. Subramaniam, "Effect of noise and topologies on multi-photon quantum protocols," in *SPIE Photonics West*, 2024.

[22] J. Burr, A. Parakh and M. Subramaniam, "Quantum Internet," *ACM Ubiquity,* vol. 2022, pp. 1-14, 2022.

[23] S. Mishra, K. Thapliyal and A. e. a. Parakh, "Quantum anonymous veto: a set of new protocols," in *EPJ Quantum Technol. 9, 14 (2022). https://doi.org/10.1140/epjqt/s40507-022-00133-2 (ARXIV: https://arxiv.org/abs/2109.06260).*, 2022.

[24] N. S. Yanofsky and M. A. Mannucci, "Quantum computing for computer scientists," *Cambridge University Press.,* 2008.

[25] IBM, *Qiskit 0.45 Release Notes,* https://docs.quantum.ibm.com/api/qiskit/release-notes/0.45, Accessed: 2023.

[26] R. Jozsa and N. Linden, "On the role of entanglement in quantum computational speed-up," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences,* p. 2011– 2032, 2003.

[27] H. J. Briegel, "Quantum Repeaters: The Role of Imperfect Local Operations," *Physical Review Letters,* p. 5932– 5935, 1998.

[28] J. Burr, A. Parakh and M. Subramaniam., "Quantum Internet," *ACM Ubiquity 2022*, pp. 1-14, 2022.

[29] A. Parakh, "Quantum teleportation for key-less cryptography," in *Quantum information and computation XIII, 950005, SPIE 9500, DSS 2015.* , 2015.

[30] A. Parakh, "Quantum Teleportation with One Classical Bit," in *Nature Sci Rep 12*, *3392, https://www.nature.com/articles/s41598-022-06853-w.*, 2022.

[31] A. Parakh and M. Subramaniam, "Network routing protocols for multi-photon quantum cryptography," in *Quantum Communications and Quantum Imaging XIX*, 2021.

[32] A. Parakh, "A new protocol for quantum public-key cryptography," in *IEEE International Conference on Advanced Networks and Telecommunications Systems*, 2015.

[33] A. Parakh, "A quantum oblivious transfer protocol," in *SPIE 8832, The Nature of Light: What are Photons? V, 883204*, San Diego, CA, 2013.

[34] C. H. Bennett and G. Brassard, "An update on quantum cryptography," in *Workshop on the theory and application of cryptographic techniques*, Berlin, Heidelberg, 1984.

[35] J. Burr, A. Parakh and M. Subramaniam, "Evaluating different topologies for multi-photon quantum key distribution," in *SPIE Defense + Commercial Sensing*, Orlando, 2022.

[36] A. Parakh and M. Subramaniam, "Bootstrapped QKD: improving key rate and multiphoton resistance," in *SPIE Security + Defence: Quantum Technologies and Quantum Information Science*, Berlin, 2018.

[37] M. Sayonnha and A. Parakh, "Implementing Diffie-Hellman key exchange using quantum EPR pairs," in *Quantum information and computation XIII, 950006, SPIE 9500, DSS 2015.*, 2015.

[38] P. Subramaniam and A. Parakh, "A quantum Diffie-Hellman protocol," in *11th IEEE Conference on Mobile Ad and Sensor Systems*, 2014.

[39] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information," *Cambridge University Press,* 2010.

[40] A. Parakh, "Using fewer qubits to correct errors in three-state QKD protocol," in *SPIE Security + Defence: Quantum Technologies and Quantum Information Science*, Berlin, 2018.

[41] A. Parakh, "Quantifying the security of a QKD protocol," in *IEEE International Conference on Advanced Networks and Telecommunications Systems*, Kolkata, 2015.

[42] P. Subramaniam and A. Parakh, "Limits on detecting eavesdropper in QKD protocols," in *IEEE ANTS 2014*, New Delhi, 2014.

[43] A. K. Ekert, "Quantum cryptography and Bell's theorem," *Quantum Measurements in Optics,* pp. 413-418, 1991.

[44] A. Parakh and P. Verma, "Improving the efficiency of entanglement based quantum key exchange," in *23rd IEEE International Conference on Computer Communication and Networks (IEEE ICCCN 2014)*, Shanghai, China, 2014.

[45] A. Parakh, "Providing variable levels of security in quantum cryptography," in *Quantum Communications and Quantum Imaging XVI*, San Diego, 2017.

[46] S. Noles and A. Parakh, "Position-Based Quantum Cryptography for Multi-located Prover and Single Verifier," in *7th International Conference on Quantum Cryptography (QCrypt 2017)*, Cambridge, 2017.

[47] S. Kak, " A three-stage quantum cryptography protocol," *Foundations of Physics Letters,* vol. 19, pp. 293-296, 2006.

[48] A. Parakh and J. Joel vanBrandwijk, " Correcting rotational errors in three stage QKD," in *23rd IEEE International Conference on Telecommunication (ICT 2016)*, Thessaloniki, 2016.

[49] A. Parakh and J. Joel vanBrandwijk, "Rotational error correction in three stage QKD," in *IEEE Communication Theory Workshop* , Nafplio, Greece, 2016.

[50] T. Kluyver, J. D. Team and et al., "Jupyter Notebooks – a publishing format for reproducible computational workflows.," *F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas,* pp. 87-90, 2016.

[51] F. Lundh, "An introduction to tkinter," *Www. Pythonware. Com/Library/Tkinter/Introduction/Index.,* 1999.

[52] J. Joel vanBrandwijk and A. Parakh, "Simulating security of quantum protocols under channel error condition," in *6th International Workshop on Privacy, Security and Trust in Mobile and Wireless Systems*, Hawaii, 2016.

[53] N. Boland, A. Bley and et al., "Clique-based facets for the precedence constrained knapsack problem," *Mathematical Programming,* vol. 133, 2012.

[54] E. Boyd, "Polyhedral results for the precedence-constrained knapsack problem," *Discrete Applied Mathematics,* vol. 41, no. 3, pp. 185-201, 1993.

[55] D. S. Johnson and K. A. Niemi, "On Knapsacks, Partitions, and a New Dynamic Programming Technique for Trees.," *Mathematics of Operations Research* , vol. 8, no. 1, pp. 1-14, 1983.

[56] N. Samphaiboon and T. Yamada, "Heuristic and exact algorithms for the precedence-constrained knapsack problem," *Journal of Optimization Theory and Applications,* vol. 105, no. 3, pp. 659-676, 2000.

[57] R. van de Leensel and et al., "Lifting valid inequalities for the precedence constrained knapsack problem," *Mathematical Programming,* vol. 86, pp. 161-185, 1999.

[58] B. You and T. Yamada, "A pegging approach to the precedence-constrained knapsack problem," *European Journal of Operational Research,* vol. 183, no. 2, pp. 618-632, 2007.

[59] P. Subramaniam and A. Parakh, "A quantum Diffie-Hellman protocol," in *11th IEEE Conference on Mobile Ad and Sensor Systems*, 2014.

[60] A. Parakh, " A quantum oblivious transfer protocol," in *SPIE 8832,* San Diego, 2013.