QUINTET: An Experiential Learning Platform for Quantum Education

Abhishek Parakh
Computer Science Department
Kennesaw State University
Marietta, GA 30068
aparakh@kennesaw.edu

Mahadevan Subramaniam
Computer Science Department
University of Nebraska Omaha
Omaha, NE 68182
msubramaniam@unomaha.edu

***Abstract*—This paper describes the conceptual framework behind QUINTET, an experiential learning platform, for education and workforce development in secure quantum communication, quantum networks and quantum computation. The platform enables users to generate lessons using QUINTET engine while specifying learning objectives and constraints. The engine composes the required learning objects using fractional knapsack problem to generate best possible lesson(s) that satisfy the given constraints. This paper primarily describes the working of QUINTET and generation of lessons.**

***Index Terms*—Experiential learning, Quantum networking and computation, Quantum education, Learning platform, Automatic generation of lessons**

I. Introduction

Educators and students in the fields of secure quantum communications and quantum networking face distinctive challenges. Proficiency in these areas demands expertise across a wide range of disciplines, such as quantum mechanics, quantum computing, cybersecurity, computer science, computer networks, algorithms, mathematics, physics, digital communications, and cryptography. The interdisciplinary and emerging nature of these fields introduces significant structural and pedagogical difficulties. Due to the limited availability of specialized educational resources and curricula, educators often need to manually compile and curate materials from diverse sources. This process involves the complex tasks of reorganizing and reformatting the materials and identifying relevant student learning objectives (SLOs), placing a substantial burden on educators. Furthermore, students find it challenging to learn effectively from courses constructed from such a disparate collection of resources.

Traditional pedagogical methods [16] are inadequate for teaching interdisciplinary subjects [5] like secure quantum communications and quantum networking, as they offer limited hands-on learning experiences necessary for understanding the interaction among various disciplines. Experiential and project-based learning approaches, which blend these disciplines, are more effective but challenging to implement due to the need for integrating theoretical concepts from physics, mathematics, and computing with practical quantum network design principles. Additionally, the scarcity of quantum computing equipment and infrastructure hinders the incorporation of hands-on experiences.

This work is sponsored by NSF award #2324924 and #2324925.

Game-based learning using serious games offers an alternative immersive learning experience [1], [4], [11]–[13], [18]. However, studies show that serious games have limited benefits in improving learning outcomes in spatial network disciplines and may hinder reflection and conceptualization due to limited exposure and attention spans [6], [18]. Therefore, a novel approach is needed to prepare the next generation workforce for secure quantum communications and networking effectively.

This paper presents QUINTET, an advanced experiential learning platform designed to overcome the structural and pedagogical challenges in educating and developing a workforce for secure quantum communication and quantum networking. The QUINTET platform includes:

1) Innovative and interactive course modules on classical and quantum networking, networking algorithms, quantum internet components and routing, classical and quantum cryptography, and network security. These modules utilize various learning objects constructed with object-oriented principles and design patterns.
2) A Learning Objects Repository (LOR) comprising modular interactive widgets, didactic materials (text, videos, interactive widgets), exercises, quizzes, and tests. These learning objects (LOs) can be used to automatically synthesize lessons and courses that ensure alignment with the specified student learning outcome (SLOs).
3) Curricular materials developed using Kolb's experiential learning [7], [12], supporting hands-on virtual quantum network experimentation.

QUINTET learning objects are designed to build proficiency in quantum communication and networking through a series of modules. These modules range from foundational topics, such as qubits and quantum operations, to advanced principles involving quantum networking devices, routing protocols, network topologies, and quantum key distribution. The goal is to enable learners to establish secure communication channels within quantum networks. Furthermore, QUINTET provides practical experience in network design, incorporating coding exercises with quantum programming libraries from IBM, Microsoft, and Amazon. These exercises allow for remote connections to quantum computers or simulation platforms.

The Learning Objects Repository (LOR) is enriched with extensive metadata, facilitating the efficient retrieval and customization of course modules for diverse learning experiences. Expert-defined concept dependencies and learner usability patterns are utilized to identify concept prerequisites and compose cohesive lesson. Instructors and learners have the option to customize the output generated by QUINTET, creating lessons that vary in length from hour-long tutorials to full semester-long courses.

Recently, the NSA developed Clark, an educational content repository for cybersecurity [10]. However, Clark requires instructors and learners to manually retrieve and organize learning modules and their prerequisites, and its rigid content structure does not allow for customization based on students' needs and objectives. This rigidity imposes a significant burden on instructors and learners to create tailored learning experiences.

Asynchronous online courses are crucial for training the future cybersecurity workforce, especially in quantum computing and networking, due to a shortage of educators. These courses are key for IT professionals in government and industry and will likely be used to retrain the workforce in quantum technologies. However, they lack flexibility, as students cannot easily extract relevant content based on their preferences or performance, nor can they access different views of the same learning objectives. This rigidity stems from insufficient metadata for course modules, hindering dynamic course layout generation, lesson, and lesson plan synthesis.

Previous research has explored the automatic synthesis of course books using rule-based frameworks tailored to specific scenarios, learning objects, and pedagogical goals [8], [15]. Other studies [9], [14], [17], [19] have developed Learning Object Repositories (LORs) defined by concepts and tailored to learning preferences and educational backgrounds. However, these frameworks lack mechanisms to integrate hands-on labs and projects, which are essential in the cybersecurity field. Additionally, they are less agile and often do not collect or analyze learner usability data, limiting their ability to adapt to learner needs in real-time.

II. The QUINTET Platform

To address the challenge of developing educational content for quantum communication and quantum networking, QUINTET structures its content into three categories: foundational knowledge units (FKUs), bridge knowledge units (BKUs), and interdisciplinary knowledge units (IKUs). Foundational concepts within the constituent disciplines of QUINTET are identified and developed into FKUs. BKUs are created to link these foundational concepts across different disciplines, which are then used to construct interdisciplinary modules. These modules enable learners to achieve proficiency in secure quantum communication and quantum networking. Table I lists the modules and corresponding learning objects related to these areas.

The identification of foundational concepts for secure quantum communication and networking in QUINTET is expert-driven. We use established courses and texts, supplemented as needed to cover all necessary foundational concepts. Bridge blocks are created to link these concepts across disciplines.

In quantum computing and cryptography, bridge blocks include topics like quantum key distribution (preparation, measurement, entanglement-based, continuous variable, protocols). For networking and cryptography, as well as networking and quantum computing, bridge blocks cover quantum repeaters, trusted nodes, quantum routing protocols, quality of service, denial of service, key rates, transmission distances, qubit attenuation, multi-modal transmission, topology effects, eavesdropping, and multi-channel quantum key distribution.

These bridge blocks help learners from varied backgrounds achieve proficiency in constituent disciplines and understand their interactions. They are crucial for developing interdisciplinary modules, such as end-to-end quantum secure communication channels, robust and fault tolerant implementation on quantum networks, hybrid quantum-classical networking protocols, and multi-modal communication with congestion control.

A. QUINTET Learning Objects

The QUINTET framework encompasses didactic, experiential, and assessment Learning Objects (LOs). Didactic LOs in QUINTET resemble sections in traditional textbooks but are parameterized by several attributes. For example, the learning choice parameter specifies the representation used by an LO to present its concept. Supported learning choice values for didactic LOs include text, visual (image), symbolic example, numerical example, widgets, interactive/non-interactive simulations, code-IDE, and code-IDE with tests.

Each educational content is developed in multiple formats, with each format available as a separate LO to support learning through various representations. For instance, an LO with the parameter value *text* describes the qubit measurement concept textually, while another LO with the parameter value *symbolic example* presents the same concept using equations. Additional learning choice parameters are supported as needed.

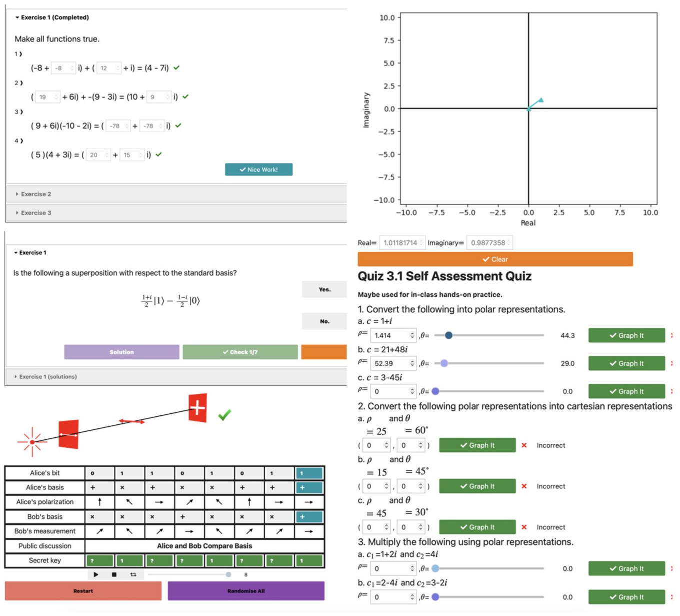
The position of a Learning Object (LO) in the QUINTET Knowledge Unit (KU) hierarchy is determined by several parameters: KU (with values FKU, BKU, IKU), learning objective (a sequence of Student Learning Objectives, or SLOs), prerequisite (with values strict, conditional, waivable), and completion (with values yes/no).

- **Strict prerequisites:** require all specified LOs to be completed before attempting the current LO.
- **Conditional prerequisites:** require at least one of the listed LOs to be completed.
- **Waivable prerequisites:** suggest background material that can be ignored but may transition to strict or conditional prerequisites based on learner usability patterns.

Successful completion of an LO is indicated by the Boolean-valued completion parameter.

QUINTET Modules	**Learning Objects**
Classical Networking	Basic networking principles, mathematical representations of networks, routing and optimization strategies, network topologies – P2P, linear, ring, lattice, tree, small-world, random, and hybrid topologies
Quantum Networking	Quantum channel, quantum processes, quantum states and channel representations, optical encoding of quantum information, quantum error correction and basic quantum routing techniques
Networking Algorithms	Shortest path, minimum-cost flow, max flow, minimum spanning tree, multicommodity flow, and routing and scheduling problems
Components of Quantum Internet	Quantum routers, quantum switches, quantum state preparation, quantum measurements, quantum state evolution during transmission, entanglement purification, quantum state teleportation, and quantum gate teleportation, super dense coding and entanglement distribution
Classical Cryptography	Secret-key cryptosystems, public-key cryptosystems, digital signatures, hashing, secure communication channels, stream ciphers, block ciphers, computational and information-theoretic security, classical cryptanalytic attacks
Quantum Cryptography	Basics of quantum cryptography, semi-quantum cryptography, quantum key distribution, and attacks on quantum cryptographic systems
Network Security Principles	Eavesdropping, masquerading, disruption, denial of service, integrity, availability, man-in-the-middle, and message rerouting
Quantum Network Security Principles	Eavesdropping on quantum channels, optical probing, blinding of detectors, fault injection, out-of-system attacks, malicious entanglement, false failure reports, quantum denial of service, dishonest quantum computation, link attacks, switching disruptions, quantum node hijacking
Secure Communication	End-to-end quantum communication protocols, robust and fault tolerant implementation on quantum networks, hybrid quantum-classical networking protocols, multi-modal communication with congestion control

TABLE I: Secure quantum communication and networking modules and corresponding learning objects.


Exercise 1 (Completed)

Make all functions true.

1) (-8 + [-8] i) + ([12] + i) = (4 - 7i) ✓

2) ([19] + 6i) + -(9 - 3i) = (10 + [9] i) ✓

3) (9 + 6i)(-10 - 2i) = ([-78] + [-78] i) ✓

4) (5)(4 + 3i) = ([20] + [15] i) ✓

Real= 1.01181714 Imaginary= 0.9877358

Exercise 1

Is the following a superposition with respect to the standard basis?

$\frac{1+i}{2}|1\rangle - \frac{1-i}{2}|0\rangle$

Alice's bit	0	1	1	0	1	0	1	1
Alice's basis	+	×	+	×	×	+	+	+
Alice's polarization	↑	↖	→	↗	↖	↑	→	→
Bob's basis	×	×	×	+	×	×	×	+
Bob's measurement	↗	↖	↗	→	↖	↗	↗	→
Public discussion	Alice and Bob Compare Basis							
Secret key	?	1	?	?	1	?	?	1

Quiz 3.1 Self Assessment Quiz

Maybe used for in-class hands-on practice.

1. Convert the following into polar representations.

a. $c = 1+i$

ρ= 1.414 ,θ= 44.3

b. $c = 21+48i$

ρ= 52.39 ,θ= 29.0

c. $c = 3-45i$

ρ= 0 ,θ= 0.0

2. Convert the following polar representations into cartesian representations

a. $\rho = 25$ and $\theta = 60^\circ$

(0 , 0) × Incorrect

b. $\rho = 15$ and $\theta = 45^\circ$

(0 , 0) × Incorrect

c. $\rho = 45$ and $\theta = 30^\circ$

(0 , 0) × Incorrect

3. Multiply the following using polar representations.

a. c_1=1+2i and c_2=4i

ρ= 0 ,θ= 0.0

b. c_1=2-4i and c_2=3-2i

ρ= 0 ,θ= 0.0

Fig. 1: Interactive learning objects in QUINTET.

Assessment LOs evaluate learner performance in the specified list of LOs and set the completion parameter values accordingly. The assessment type parameter supports various forms of evaluations, including quizzes, homework, simulations, code-IDE, code-IDE with tests, midterms, finals, reports, and virtual labs.

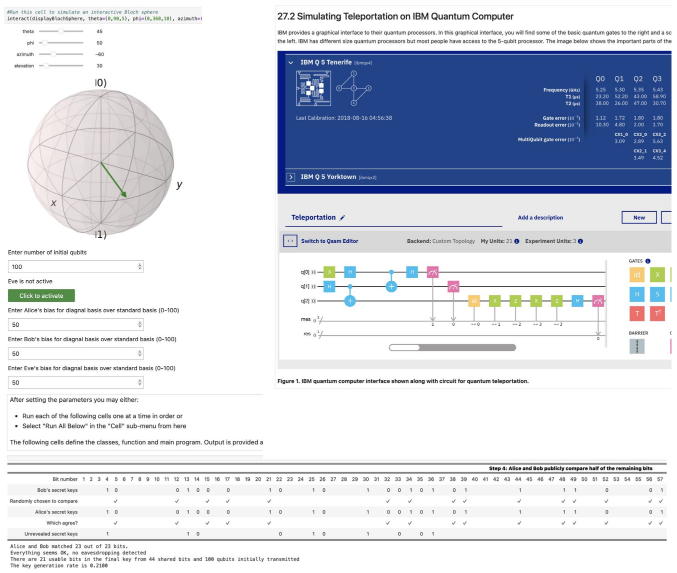
Frontend: In QUINTET, learning objects (LOs) are delivered through Jupyter notebooks, the standard for quantum computing and programming. These notebooks support the integration of code, mathematics, visual objects, simulations, widgets, and quizzes, and interface with external APIs like IBM and Microsoft quantum simulators. Python is used for both quantum programming and cybersecurity.

Jupyter notebooks consist of markdown and code cells, with each LO spanning multiple cells and its Student Learning Objectives (SLOs) distributed throughout. Notebooks can contain multiple LOs, with the notebook's SLOs being the combined SLOs from all cells. Figure 1 display a number of interactive learning objects of the type quiz, simulation, and auto-graded labs. Figure 2 displays the interactive lab for Bloch sphere representation, self contained programmatic simulation learning object and the IBM Quantum Simulation learning object. Finally, figure 3 shows the quantum network simulator within QUINTET that allows for large scale simulation of networks with different topologies, router and repeater placements and protocols. The network simulator also supports the display of qubit state in real-time using Bloch sphere as it traverses through the network and the equivalent circuit constructed for the network path chosen by the qubit.

B. Experiential Learning in QUINTET

Experiential learning in QUINTET is facilitated through experiential learning objects (LOs).

- **Concrete Experimentation Phase:** This includes interactive simulations, code-IDE with tests, and self-graded exercises, grouped into notebooks that also function as virtual labs.
- **Reflection Phase:** LOs are grouped into reflection zones within notebooks, incorporating interactive simulations that replicate learner actions from the concrete experimentation phase. These zones allow learners to replay actions and observe outcomes, with support for student-initiated queries to deepen comprehension.
- **Conceptualization Phase:** LOs in conceptualization zones help students generalize and abstract concepts from concrete experiments and reflections, fostering computational thinking. These zones encourage students to apply learned concepts to varied scenarios involving networks



```
#Run this cell to simulate an interactive Bloch sphere
interact(displayBlochSphere, theta=(0,90,5), phi=(0,360,10), azimuth=(
```

Enter number of initial qubits: 100

Eve is not active

Enter Alice's bias for diagnal basis over standard basis (0-100): 50

Enter Bob's bias for diagnal basis over standard basis (0-100): 50

Enter Eve's bias for diagnal basis over standard basis (0-100): 50

After setting the parameters you may either:

- Run each of the following cells one at a time in order or
- Select "Run All Below" in the "Cell" sub-menu from here

The following cells define the classes, function and main program. Output is provided a

27.2 Simulating Teleportation on IBM Quantum Computer

IBM provides a graphical interface to their quantum processors. In this graphical interface, you will find some of the basic quantum gates to the right and a sc
the left. IBM has different size quantum processors but most people have access to the 5-qubit processor. The image below shows the important parts of the

Last Calibration: 2018-08-16 04:56:38

	Q0	Q1	Q2	Q3
Frequency (GHz)	5.25	5.30	5.35	5.43
T1 (μs)	23.20	52.20	43.00	58.90
T2 (μs)	38.00	26.00	47.00	30.70
Gate error (10^{-3})	1.12	1.72	1.80	1.80
Readout error (10^{-2})	10.30	4.80	2.00	1.70
MultiQubit gate error (10^{-2})		CX1_0 3.09	CX2_0 2.89	CX3_2 5.63
			CX2_1 3.49	CX3_4 4.52

Backend: Custom Topology My Units: 21 Experiment Units: 3

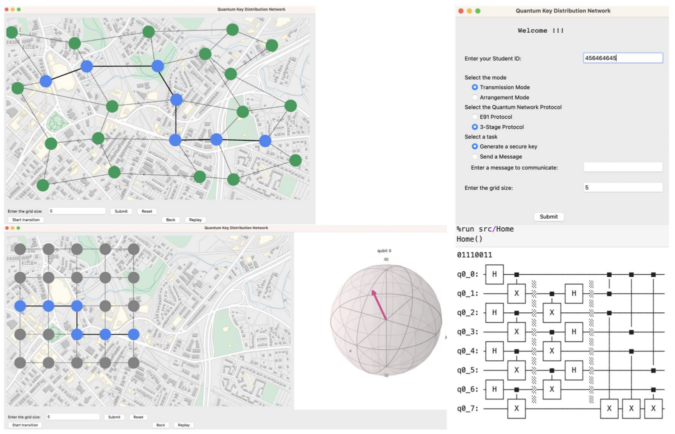
Figure 1. IBM quantum computer interface shown along with circuit for quantum teleportation.

Step 4: Alice and Bob publicly compare half of the remaining bits

Bit number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
Bob's secret keys				1	0							0	1	0	0		0				1	0			1	0				1		0	0	1	0	1		0	1					1				1	1			0				0	1
Randomly chosen to compare					✓							✓			✓		✓				✓											✓		✓				✓	✓					✓				✓	✓			✓				✓	✓
Alice's secret keys				1	0							0	1	0	0		0				1	0			1	0				1		0	0	1	0	1		0	1					1				1	1			0				0	1
Which agree?					✓							✓			✓		✓				✓											✓		✓				✓	✓					✓				✓	✓			✓				✓	✓
Unrevealed secret keys				1									1	0								0			1	0				1			0		0	1																					

```
Alice and Bob matched 23 out of 23 bits.
Everything seems OK, no eavesdropping detected
There are 21 usable bits in the final key from 44 shared bits and 100 qubits initially transmitted
The key generation rate is 0.2100
```

Fig. 2: Interactive learning object labs in QUINTET.



Quantum Key Distribution Network

Welcome !!!

Enter your Student ID: 456464645

Select the mode
[✓ Transmission Mode] Arrangement Mode

Select the Quantum Network Protocol
E91 Protocol [✓ 3-Stage Protocol]

Select a task
[✓ Generate a secure key] Send a Message

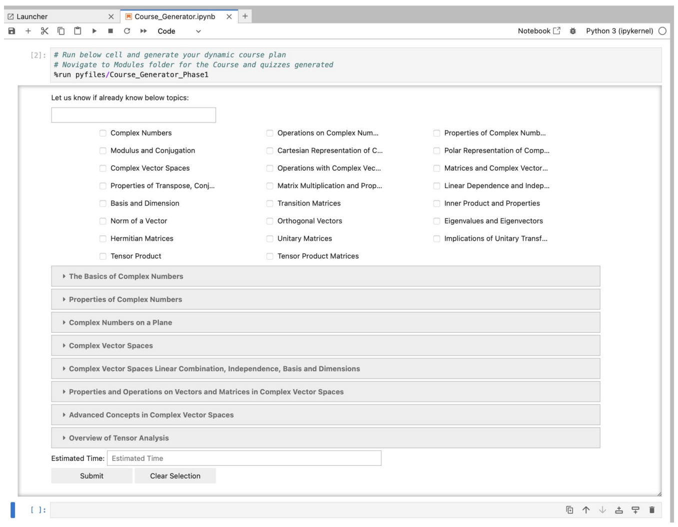
Enter a message to communicate:

Enter the grid size: 5

```
%run src/Home
Home()
```

```
01110011
```

Fig. 3: Quantum Network/Internet Simulator in QUINTET



```
[2]: # Run below cell and generate your dynamic course plan
     # Novigate to Modules folder for the Course and quizzes generated
     %run pyfiles/Course_Generator_Phase1
```

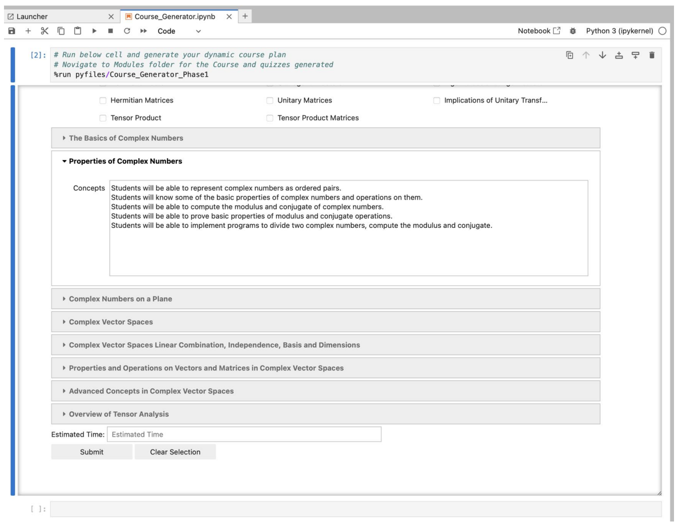
Let us know if already know below topics:

☐ Complex Numbers	☐ Operations on Complex Num...	☐ Properties of Complex Numb...
☐ Modulus and Conjugation	☐ Cartesian Representation of C...	☐ Polar Representation of Comp...
☐ Complex Vector Spaces	☐ Operations with Complex Vec...	☐ Matrices and Complex Vector...
☐ Properties of Transpose, Conj...	☐ Matrix Multiplication and Prop...	☐ Linear Dependence and Indep...
☐ Basis and Dimension	☐ Transition Matrices	☐ Inner Product and Properties
☐ Norm of a Vector	☐ Orthogonal Vectors	☐ Eigenvalues and Eigenvectors
☐ Hermitian Matrices	☐ Unitary Matrices	☐ Implications of Unitary Transf...
☐ Tensor Product	☐ Tensor Product Matrices	

- The Basics of Complex Numbers
- Properties of Complex Numbers
- Complex Numbers on a Plane
- Complex Vector Spaces
- Complex Vector Spaces Linear Combination, Independence, Basis and Dimensions
- Properties and Operations on Vectors and Matrices in Complex Vector Spaces
- Advanced Concepts in Complex Vector Spaces
- Overview of Tensor Analysis

Estimated Time: Estimated Time

Fig. 4: Entry page of QUINTET displaying linear algebra preliminary modules, and pre-requisite selection interface.



```
[2]: # Run below cell and generate your dynamic course plan
     # Novigate to Modules folder for the Course and quizzes generated
     %run pyfiles/Course_Generator_Phase1
```

☐ Hermitian Matrices	☐ Unitary Matrices	☐ Implications of Unitary Transf...
☐ Tensor Product	☐ Tensor Product Matrices	

- The Basics of Complex Numbers
- Properties of Complex Numbers

Concepts:
Students will be able to represent complex numbers as ordered pairs.
Students will know some of the basic properties of complex numbers and operations on them.
Students will be able to compute the modulus and conjugate of complex numbers.
Students will be able to prove basic properties of modulus and conjugate operations.
Students will be able to implement programs to divide two complex numbers, compute the modulus and conjugate.

- Complex Numbers on a Plane
- Complex Vector Spaces
- Complex Vector Spaces Linear Combination, Independence, Basis and Dimensions
- Properties and Operations on Vectors and Matrices in Complex Vector Spaces
- Advanced Concepts in Complex Vector Spaces
- Overview of Tensor Analysis

Estimated Time: Estimated Time

Fig. 5: Entry page of QUINTET displaying the expanded learning outcomes of module titled "Properties of Complex Numbers".

and quantum states. Interactive audio-visual scenarios and coding tasks present applications, assisting student queries with problem-solving hints.

- **Active Experimentation Phase:** Managed by assessment LOs, enabling learners to test their understanding and apply their knowledge.

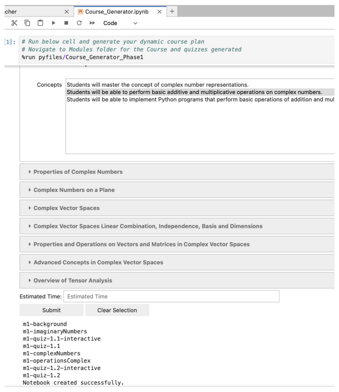
This structured approach ensures that learners can experiment, reflect, conceptualize, and actively test their knowledge through a comprehensive set of interactive and didactic tools.

III. System Overview

Figure 4 shows the initial interface of QUINTET. The top of the screen presents to the learners a list of existing high level modules and topics covered in these. A student can check off boxes for the topics that the student already knows as pre-requisites. Checking a box will eliminate learning objects contributing to that topic from the generated lesson. A convenient search box facilitates searching of topics as the topics list grows. The current screenshot in figure 4 shows the linear algebra topics that a student needs to achieve proficiency in before embarking on the topics of quantum communications, networking and computation.

Once the student has identified the appropriate pre-requisite topics that the student already knows, the student can expand various modules which will display the associated learning outcomes with that module. A student can then select one or more of the learning outcomes that they want to become proficient in (figure 5).

Figures 6a and 6b display two scenarios where a student identifies no known pre-requisites and one of the known pre-requisites (in this case complex numbers from figure 4), respectively. The student has not identified any estimated learning time they are willing to spend and therefore QUIN-



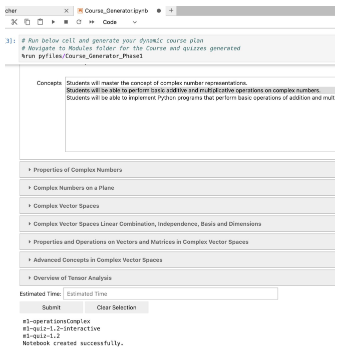
```
[1]: # Run below cell and generate your dynamic course plan
     # Novigate to Modules folder for the Course and quizzes generated
     %run pyfiles/Course_Generator_Phase1
```

Concepts
Students will master the concept of complex number representations.
Students will be able to perform basic additive and multiplicative operations on complex numbers.
Students will be able to implement Python programs that perform basic operations of addition and mul

Course_Generator.ipynb
Code
Properties of Complex Numbers
Complex Numbers on a Plane
Complex Vector Spaces
Complex Vector Spaces Linear Combination, Independence, Basis and Dimensions
Properties and Operations on Vectors and Matrices in Complex Vector Spaces
Advanced Concepts in Complex Vector Spaces
Overview of Tensor Analysis
Estimated Time:
Estimated Time
Submit
Clear Selection

```
m1-background
m1-imaginaryNumbers
m1-quiz-1.1-interactive
m1-quiz-1.1
m1-complexNumbers
m1-operationsComplex
m1-quiz-1.2-interactive
m1-quiz-1.2
Notebook created successfully.
```

(a) Without pre-requisites



```
[3]: # Run below cell and generate your dynamic course plan
     # Novigate to Modules folder for the Course and quizzes generated
     %run pyfiles/Course_Generator_Phase1
```

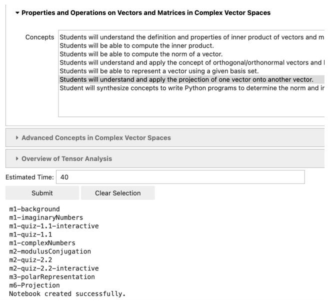
Concepts
Students will master the concept of complex number representations.
Students will be able to perform basic additive and multiplicative operations on complex numbers.
Students will be able to implement Python programs that perform basic operations of addition and mult

Course_Generator.ipynb
Code
Properties of Complex Numbers
Complex Numbers on a Plane
Complex Vector Spaces
Complex Vector Spaces Linear Combination, Independence, Basis and Dimensions
Properties and Operations on Vectors and Matrices in Complex Vector Spaces
Advanced Concepts in Complex Vector Spaces
Overview of Tensor Analysis
Estimated Time:
Estimated Time
Submit
Clear Selection

```
m1-operationsComplex
m1-quiz-1.2-interactive
m1-quiz-1.2
Notebook created successfully.
```

(b) With pre-requisites

Fig. 6: Notebook generation in QUINTET without and with pre-requisites selected for a single learning outcome.



Properties and Operations on Vectors and Matrices in Complex Vector Spaces

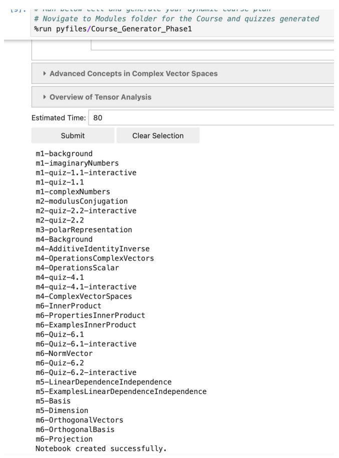
Concepts
Students will understand the definition and properties of inner product of vectors and m
Students will be able to compute the inner product.
Students will be able to compute the norm of a vector.
Students will understand and apply the concept of orthogonal/orthonormal vectors and l
Students will be able to represent a vector using a given basis set.
Students will understand and apply the projection of one vector onto another vector.
Student will synthesize concepts to write Python programs to determine the norm and ir

Advanced Concepts in Complex Vector Spaces
Overview of Tensor Analysis
Submit
Clear Selection

Estimated Time: 40

```
m1-background
m1-imaginaryNumbers
m1-quiz-1.1-interactive
m1-quiz-1.1
m1-complexNumbers
m2-modulusConjugation
m2-quiz-2.2
m2-quiz-2.2-interactive
m3-polarRepresentation
m6-Projection
Notebook created successfully.
```

(a) Without pre-requisites



```
# Novigate to Modules folder for the Course and quizzes generated
%run pyfiles/Course_Generator_Phase1
```

Advanced Concepts in Complex Vector Spaces
Overview of Tensor Analysis
Submit
Clear Selection

Estimated Time: 80

```
m1-background
m1-imaginaryNumbers
m1-quiz-1.1-interactive
m1-quiz-1.1
m1-complexNumbers
m2-modulusConjugation
m2-quiz-2.2-interactive
m2-quiz-2.2
m3-polarRepresentation
m4-Background
m4-AdditiveIdentityInverse
m4-OperationsComplexVectors
m4-OperationsScalar
m4-quiz-4.1
m4-quiz-4.1-interactive
m4-ComplexVectorSpaces
m6-InnerProduct
m6-PropertiesInnerProduct
m6-ExamplesInnerProduct
m6-Quiz-6.1
m6-Quiz-6.1-interactive
m6-NormVector
m6-Quiz-6.2
m6-Quiz-6.2-interactive
m5-LinearDependenceIndependence
m5-ExamplesLinearDependenceIndependence
m5-Basis
m5-Dimension
m6-OrthogonalVectors
m6-OrthogonalBasis
m6-Projection
Notebook created successfully.
```

(b) With pre-requisites

Fig. 7: Notebook generation in QUINTET without and with pre-requisites selected for a single learning outcome.

TET simply collects and generates a lesson that includes all the necessary learning objects to complete the identified learning objective. This fact is shown at the bottom of the screen by the list of learning objects included in the generated lesson.

Figures 7a, and 7b show the generation of QUINTET lessons for different time constraints, 40 and 80 minutes respectively, for the same learning objective identified by the students. As seen in the figures the list of learning objects synthesized into the lessons differ for the two. The 40 minute lesson is subset (fraction) of the 80 minute lesson.

A. Lesson Generation

Lessons are presented to students in the form of Jupyter notebooks. The student specifies the target learning outcome they want to achieve and the pre-requisites they already know. Along with this the student can also specifies a time constraint for the lesson. QUINTET generates lessons using fractional knapsack algorithm [3] trying to maximize the number of learning objects that fit within the given time constraint [2].

B. Fractional Knapsack Problem

The 0-1 Knapsack Problem: The 0-1 Knapsack problem is an example of a combinatorial optimization problem that can be used to illustrate decision-making and resource allocation in various contexts, including education. The 0-1 Knapsack problem involves a knapsack or a backpack with a limited capacity (weight limit) and a set of items, each with a specific weight and value. The objective is to select a combination of items to put into the knapsack so that the total weight does not exceed the capacity while the total value is maximized. The "0-1" part of the name indicates that each item can either be taken or left (i.e., you cannot take a fraction of an item). In the educational context, typically a "knapsack" is a limited available resource — such as time, budget, or attention — while the "items" represent educational activities, programs, or materials, each with a "cost" (time, effort, money) and a "value" (learning outcome, student engagement, etc.).

The Fractional Knapsack Problem: The Fractional Knapsack problem extends the principles of the classic 0-1 Knapsack problem by allowing the selection of fractions of items instead of having to choose items in their entirety. This variation introduces a level of flexibility that can be highly applicable in educational contexts. In the Fractional Knapsack problem, you are still trying to maximize the total value of items put into a knapsack with a limited capacity.

In QUINTET, we have a number of educational modules. Each module consists of a number of learning objects. A student can finish a partial module, but a learning object must be completed in its entirety. Each learning object has a time for completion and a profit associated with it. Each module has a module-level learning outcome associated with it. A student provides a target learning outcome from a list of outcomes and the maximum amount of available time they can spend on it. To complete a target learning outcome, a number of modules may need to be composed into a lesson. We formulate this problem as a fractional knapsack problem to create lessons for students with specified target learning outcome that fits within the maximum time students specify.

Variables: Below are the variables for our fractional knapsack implementation:

M: Set of all modules.
L_i: Set of learning objects in module i.
t_{ij}: Time required to complete learning object j in module i.
p_{ij}: Profit (educational value) associated with completing learning object j in module i.
x_{ij}: Binary variable where $x_{ij} = 1$ if learning object j in module i is included in the plan, 0 otherwise.
y_i: Fractional variable where y_i represents the fraction of module i included in the plan based on its learning objects included.
T: Maximum total time a student is willing to spend.
O: Target learning outcome, which is achieved by completing a specific combination of modules.
O_i: Learning outcome contribution of module i when fully completed.

C_o: Set of modules required to achieve the target learning outcome O.

Objective Function: We maximize the total educational value of the selected learning objects while aiming to fulfill the learning outcome O:

Maximize $Z = \sum_{i \in C_o} \sum_{j \in L_i} p_{ij} \cdot x_{ij}$

Time Constraint: The total time spent on selected learning objects must not exceed the available time,

$$T : \sum_{i \in C_o} \sum_{j \in L_i} t_{ij} \cdot x_{ij} \leq T$$

Module Completion Fraction: Define the fraction of each module completed based on the learning objects included. If all learning objects in a module are included $y_i = 1$; otherwise, it is proportional to the number of learning objects included:

$$y_i = \frac{\sum_{j \in L_i} x_{ij}}{|L_i|} \text{ for all } i \in C_o$$

Learning Outcome Achievement: Ensure that the combination of modules sufficiently addresses the target learning outcome O. This might translate to achieving a sufficient fraction of modules in C_o:

$$\sum_{i \in C_o} O_i \cdot y_i \geq \text{Minimum required outcome level for } O$$

Binary and Fractional Variables: Ensure that x_{ij} are binary (0 or 1) and y_i are fractional (between 0 and 1):

$$x_{ij} \in \{0, 1\} \text{ for all } j \in L_i, i \in C_o$$

$$0 \leq y_i \leq 1 \text{ for all } i \in C_o$$

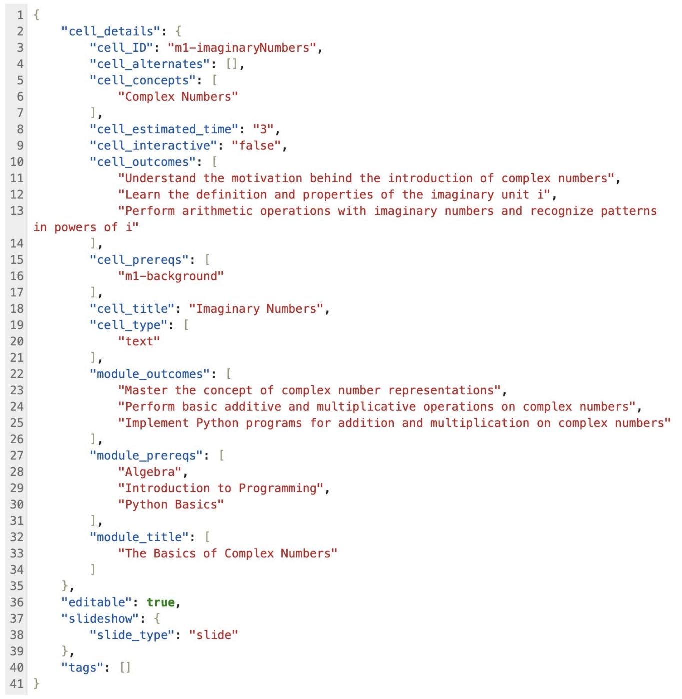
This model optimizes the selection of learning objects to maximize educational value while ensuring that the selected items do not exceed the student's available time and contribute towards achieving the specified learning outcome. This involves a mix of complete inclusion for discrete learning objects (0-1 decision) and partial completion of modules (fractional decision) based on the percentage of learning objects included from each module.

C. Implementation

A high-level list of steps for the algorithmic implementation of fractional knapsack problem for generating lessons using QUINTET is given below. The steps below provide a structured approach for constructing an educational lesson that optimizes learning object selections within a given time frame to meet a desired learning outcome identified by a student.

1) Initialize a dynamic programming (DP) table that stores the maximum value achievable with a given time.
2) Fill the DP table. This step prepares each module by sorting its learning objects based on efficiency.
3) Use sorted lists to update the DP table using a reverse traversal to prevent reuse of the same learning object in the current iteration.
4) Backtrack from the end of the DP table to determine which learning objects were used to achieve the DP table's maximum value at time T. This also helps to determine the fraction of each module completed.
5) Calculate how much of each module is completed based on the selected learning objects.
6) Check if the learning outcomes meet the target goal, suggesting adjustments if necessary.
7) Output the final selections and their details.

The algorithm implementing the above procedure for generation of QUINTET lessons is given in Algorithm 1.



```
{
    "cell_details": {
        "cell_ID": "m1-imaginaryNumbers",
        "cell_alternates": [],
        "cell_concepts": [
            "Complex Numbers"
        ],
        "cell_estimated_time": "3",
        "cell_interactive": "false",
        "cell_outcomes": [
            "Understand the motivation behind the introduction of complex numbers",
            "Learn the definition and properties of the imaginary unit i",
            "Perform arithmetic operations with imaginary numbers and recognize patterns in powers of i"
        ],
        "cell_prereqs": [
            "m1-background"
        ],
        "cell_title": "Imaginary Numbers",
        "cell_type": [
            "text"
        ],
        "module_outcomes": [
            "Master the concept of complex number representations",
            "Perform basic additive and multiplicative operations on complex numbers",
            "Implement Python programs for addition and multiplication on complex numbers"
        ],
        "module_prereqs": [
            "Algebra",
            "Introduction to Programming",
            "Python Basics"
        ],
        "module_title": [
            "The Basics of Complex Numbers"
        ]
    },
    "editable": true,
    "slideshow": {
        "slide_type": "slide"
    },
    "tags": []
}
```

Fig. 8: Example metadata for learning objects.

D. Proof of Correctness of Algorithm 1

To prove the correctness of Algorithm 1, we need to show that it correctly finds the optimal set of learning objects that maximize the educational value while respecting the given time constraint T and ensuring the target learning outcome is met.

1. Initialization

The algorithm initializes a dynamic programming (DP) table `dp` where `dp[k]` represents the maximum educational value achievable with k units of time. This initialization step ensures that the algorithm starts with a base case where no time is used, and no educational value is accumulated.

2. Learning Object Preparation and Sorting

For each module in the set C_O (modules required to meet the target learning outcome), the algorithm:

- Lists all learning objects.
- Sorts these learning objects by their profit-to-time ratio $\frac{p_{ij}}{t_{ij}}$ in descending order.

Algorithm 1 Fractional Knapsack Based Lesson Generation with Time Constraints

```
Require: Modules $M$, learning objects per module $L_i$, times $t_{ij}$, profits $p_{ij}$, maximum time $T$, target learning outcome $O$, required modules $C_O$
Ensure: Selected learning objects, their modules, total educational value, total time used
 1: Initialize DP table: $dp[0..T] \leftarrow 0$ // DP array with size $T+1$, initialized to 0
 2: for each module $i$ in $C_O$ do
 3:     Create a list of learning objects $LO$ with $(j, p_{ij}, t_{ij})$
 4:     for each learning object $j$ in $L_i$ do
 5:         $LO$.append($(j, p_{ij}, t_{ij})$)
 6:     end for
 7:     Sort $LO$ based on $\frac{p_{ij}}{t_{ij}}$ ratio in descending order
 8: end for
 9: for each module $i$ in $C_O$ do
10:     for each learning object $(j, p, t)$ in $LO$ do
11:         for $k$ from $T$ down to $t$ do
12:             $dp[k] = \max(dp[k], dp[k-t] + p)$
13:         end for
14:     end for
15: end for
16: Backtrack to determine selected learning objects and calculate module fractions:
17: Initialize $selected\_objects$ to an empty list
18: $current\_time \leftarrow T$
19: while $current\_time > 0$ do
20:     for each module $i$ in reverse order of $C_O$ do
21:         for each $(j, p, t)$ in reverse order of $LO$ do
22:             if $current\_time \geq t$ and $dp[current\_time] == dp[current\_time - t] + p$ then
23:                 $selected\_objects$.append($(i, j)$)
24:                 $current\_time -= t$
25:                 break
26:             end if
27:         end for
28:     end for
29: end while
30: Calculate the completion fraction $y_i$ for each module:
31: Initialize $completion\_fraction[i] = 0$ for each $i$ in $C_O$
32: for each $(i, j)$ in $selected\_objects$ do
33:     $completion\_fraction[i] += \frac{1}{|L_i|}$
34: end for
35: Check if the outcome $O$ is achieved:
36: $total\_outcome\_value = \sum_{i \in C_O} O_i \cdot completion\_fraction[i]$
37: if $total\_outcome\_value <$ required outcome level for $O$ then
38:     Print "Adjust learning objects or increase time $T$"
39: end if
40: Algorithm continued below =0
```

```
 1: Algorithm continued from above
 2: Output the results:
 3: Print"Selected learning objects and modules:"
 4: for each $(i, j)$ in $selected\_objects$ do
 5:     Print "Module", $i$, "Learning Object", $j$
 6: end for
 7: Print "Total educational value:", $dp[T]$
 8: Print "Total time used:", $T - current\_time$
 9: for each $i$ in $C_O$ do
10:     Print "Completion fraction of Module", $i$, ":", $completion\_fraction[i]$
11: end for=0
```

Reasoning: Sorting learning objects by their profit-to-time ratio ensures that we consider the most efficient (in terms of educational value per unit of time) learning objects first. This is a greedy approach that is optimal for the fractional knapsack problem.

3. Dynamic Programming Table Update

The algorithm iterates through each learning object and updates the DP table. For each learning object (j, p, t) in the sorted list, it updates the DP table backwards from T down to t:

$$\texttt{dp[k]} = \max(\texttt{dp[k]}, \texttt{dp[k - t]} + p)$$

Reasoning: This step ensures that for each possible time k, the algorithm considers whether including the current learning object would increase the total educational value. The backward iteration prevents the same learning object from being considered multiple times in the current iteration.

4. Backtracking

After filling the DP table, the algorithm backtracks to determine which learning objects were selected:

- Starting from `dp[T]`, it traces back through the DP table to find the selected learning objects.
- It calculates the fractions of modules completed based on the selected learning objects.

Reasoning: Backtracking ensures that we can reconstruct the set of learning objects that contributed to the maximum educational value stored in `dp[T]`. By tracing the updates, we can identify which learning objects were included in the optimal solution.

5. Outcome Verification

The algorithm verifies if the total educational value meets the target learning outcome:

$$\text{total\_outcome\_value} = \sum_{i \in C_O} O_i \cdot \text{completion\_fraction}[i]$$

If the total outcome value is less than the required level, the algorithm suggests adjustments.

Reasoning: This step ensures that the selected learning objects not only maximize educational value but also contribute to achieving the target learning outcome. The verification step ensures that the solution meets the problem's constraints.

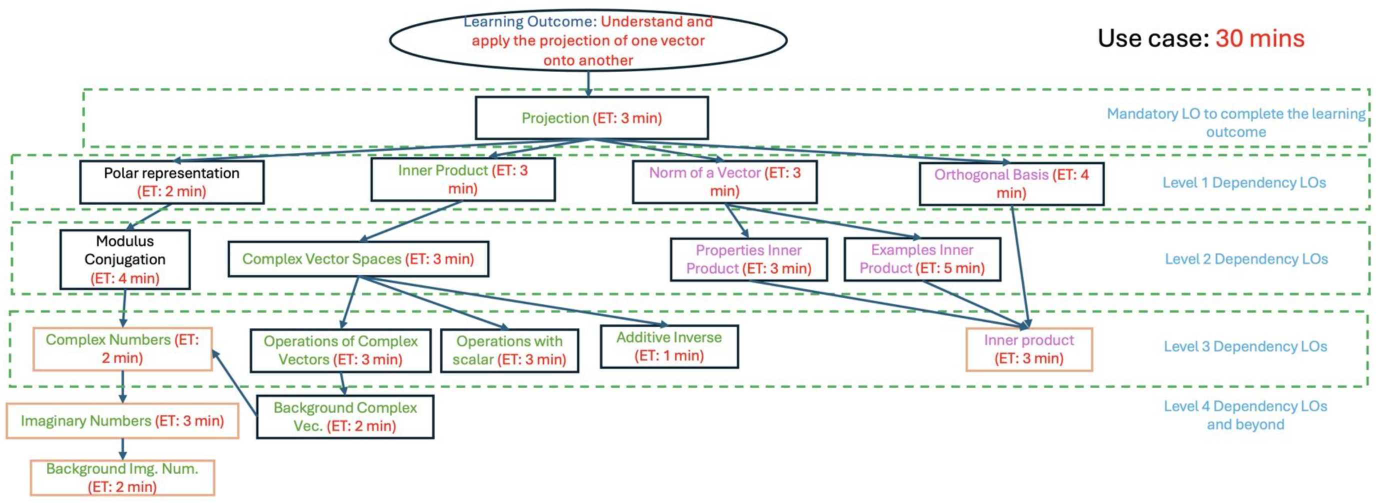

Learning Outcome: Understand and apply the projection of one vector onto another
Use case: 30 mins
Projection (ET: 3 min)
Mandatory LO to complete the learning outcome
Polar representation (ET: 2 min)
Inner Product (ET: 3 min)
Norm of a Vector (ET: 3 min)
Orthogonal Basis (ET: 4 min)
Level 1 Dependency LOs
Modulus Conjugation (ET: 4 min)
Complex Vector Spaces (ET: 3 min)
Properties Inner Product (ET: 3 min)
Examples Inner Product (ET: 5 min)
Level 2 Dependency LOs
Complex Numbers (ET: 2 min)
Operations of Complex Vectors (ET: 3 min)
Operations with scalar (ET: 3 min)
Additive Inverse (ET: 1 min)
Inner product (ET: 3 min)
Level 3 Dependency LOs
Imaginary Numbers (ET: 3 min)
Background Complex Vec. (ET: 2 min)
Level 4 Dependency LOs and beyond
Background Img. Num. (ET: 2 min)

Fig. 9: Use case with a single learning outcome and time constraint of 30 mins.

6. *Output*

The algorithm outputs:

- The selected learning objects and their respective modules.
- The total educational value.
- The total time used.
- The completion fraction of each module.

Reasoning: Providing detailed output ensures that the solution is transparent and that all selected components are clearly identified. This helps in verifying and understanding the optimal solution.

To summarize the correctness argument follows the following steps:

- The initialization step correctly sets up the DP table.
- Sorting learning objects by profit-to-time ratio ensures that we consider the most valuable learning objects first.
- The DP update step ensures that we build the optimal solution incrementally.
- Backtracking correctly reconstructs the optimal solution.
- Outcome verification ensures that the solution meets all problem constraints.

By these steps, the algorithm ensures that the optimal set of learning objects is selected to maximize educational value within the given time constraint while achieving the target learning outcome. This combination of greedy sorting, dynamic programming, and backtracking forms a correct and optimal solution to the problem.

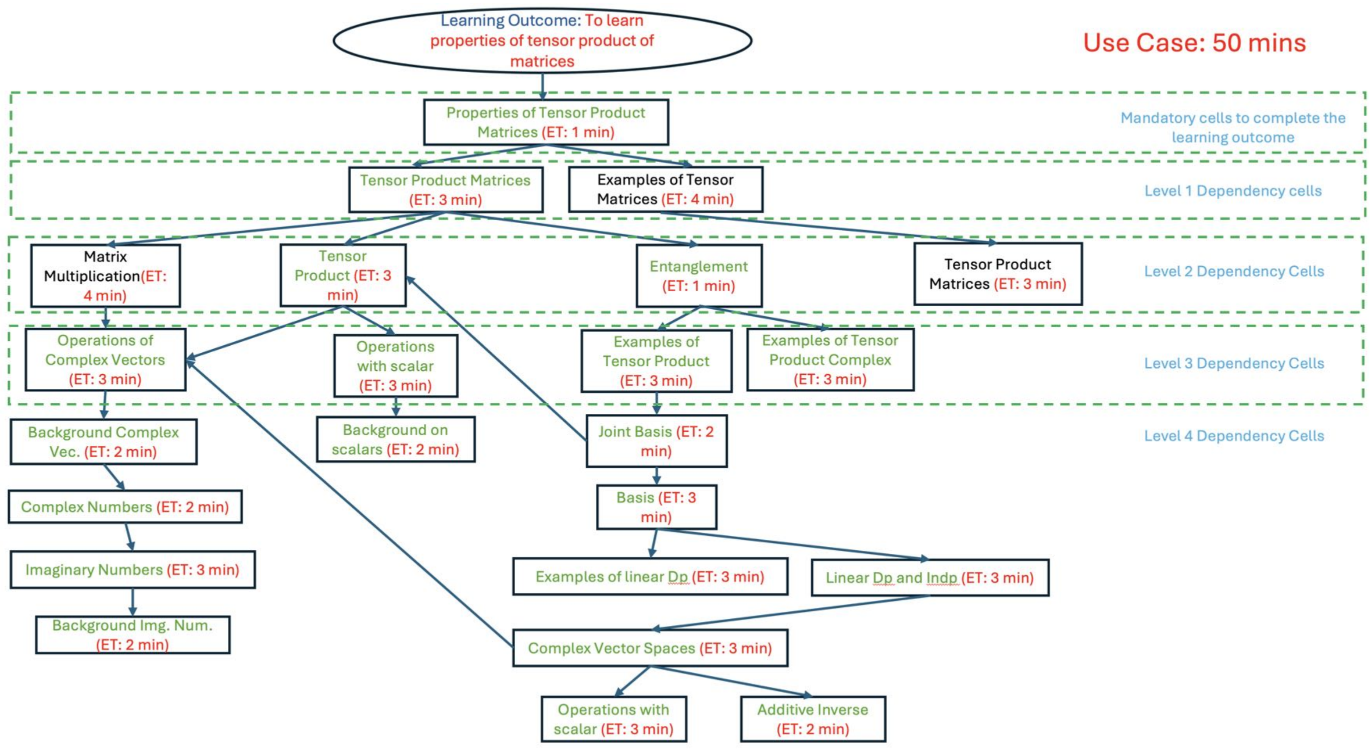
IV. Use Cases

We walk through a couple of use cases to illustrate the working of the fractional knapsack algorithm in QUINTET. Figure 9 shows a case when the student chooses a learning outcome *Understand and apply the projection of one vector onto another*. QUINTET engine maintains a dependency graph for various learning objects. Using this dependency graph the QUINTET engine first identifies the learning objects that a student needs to achieve proficiency in to meet the target learning outcome. Each learning object is instrumented with metadata (eg. figure 8) that contains an attribute for the estimated time to achieve proficiency. This estimated time is subject matter expert driven. As we can see from figure 9 that 30 minutes is not enough to meet the desired learning outcome in its entirety. As a result, the largest possible fraction of the learning outcome will be covered. The algorithm first finds the longest chain of learning object dependencies to that fit within the specified time constraint. This is denoted by green font and adds to 25 minutes. The learning objects shown in light brown color are common between two different paths of dependencies. Completing the entire LO *Inner Product* and the high level LO *Projection* leaves only 5 minutes out of 30. At this point the algorithm can pick the next largest learning object *Modulus Conjugation* and present a 29 minute lesson consisting of LOs in green font and one additional learning object *modulus conjugation*.

Another option for the algorithm is to complete the learning objects *Norm of a Vector* and *Orthogonal Basis* consuming 21 minutes and additional learning objects of *Complex Numbers*, *Imaginary Numbers* and *Background Img. Num.* leading to a lesson of 28 minutes. Since this falls short of the 29 minute lesson we prioritize and present the first option discussed above.

In the case of a tie, we prioritize the lesson that covers the most number of learning objects in a complete manner.

Figure 10 shows another dependency graph of learning objects for the learning outcome *To learn the properties of Tensor product of matrices*. The learning objects in green font can be covered in the student specified time of 50 minutes. The algorithm favors fully completing the longest chain of learning object dependencies that fit within the specified amount of time.



Learning Outcome: To learn properties of tensor product of matrices
Use Case: 50 mins
Properties of Tensor Product Matrices (ET: 1 min)
Mandatory cells to complete the learning outcome
Tensor Product Matrices (ET: 3 min)
Examples of Tensor Matrices (ET: 4 min)
Level 1 Dependency cells
Matrix Multiplication(ET: 4 min)
Tensor Product (ET: 3 min)
Entanglement (ET: 1 min)
Tensor Product Matrices (ET: 3 min)
Level 2 Dependency Cells
Operations of Complex Vectors (ET: 3 min)
Operations with scalar (ET: 3 min)
Examples of Tensor Product (ET: 3 min)
Examples of Tensor Product Complex (ET: 3 min)
Level 3 Dependency Cells
Background Complex Vec. (ET: 2 min)
Background on scalars (ET: 2 min)
Joint Basis (ET: 2 min)
Level 4 Dependency Cells
Complex Numbers (ET: 2 min)
Basis (ET: 3 min)
Imaginary Numbers (ET: 3 min)
Examples of linear Dp (ET: 3 min)
Linear Dp and Indp (ET: 3 min)
Background Img. Num. (ET: 2 min)
Complex Vector Spaces (ET: 3 min)
Operations with scalar (ET: 3 min)
Additive Inverse (ET: 2 min)

Fig. 10: Use case with time constraint of 50 minutes.

V. Conclusions

QUINTET introduces a first of its kind experiential learning platform designed to address the unique challenges in education and workforce development for quantum communication, quantum networking, and quantum computing. It is built on a robust learning objects repository, featuring modular interactive widgets, didactic materials (text, videos, interactive widgets), exercises, quizzes, and tests. These learning objects (LOs) are used to automatically create lessons that align with specified student learning outcomes. Lessons are structured using Kolb's experiential learning model, incorporating hands-on virtual network experimentation. The learning objects are developed offering multiple representations of the same concept.

Future work involves developing more learning objects in QUINTET and allow for additional constraints such as learning preferences (theory driven, example driven, quiz driven, hands-on exercise driven, etc.). It will also support generation of multiple lessons of specified time limit or breaking up a longer lesson into smaller chunks.

Acknowledgment

The authors thank Krishna Puja Anumula for implementing parts of the QUINTET engine.

References

[1] D. Abeyrathna, S. Vadla, V. Bommanapally, M. Subramaniam, P. Chundi, and A. Parakh. Analyzing and predicting player performance in a quantum cryptography serious game. In H. Söbke, M. Gentile, and M. Allegra, editors, *Games and Learning Alliance - 7th International Conference, GALA 2018, Proceedings*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 267–276. Springer Verlag, 2019. 7th International Conference on Games and Learning Alliance, GALA 2018 ; Conference date: 05-12-2018 Through 07-12-2018.

[2] A. Aslan, E. Ursavas, and W. Romeijnders. A precedence constrained knapsack problem with uncertain item weights for personalized learning systems. *Omega*, 115:102779, 2023.

[3] N. Z. Atashbar and F. Rahimi. Optimization of educational systems using knapsack problem. *International Journal of Machine Learning and Computing*, pages 552–555, 2011.

[4] K. Chan, K. Wan, and V. King. Performance over enjoyment? effect of game-based learning on learning outcome and flow experience. *Frontiers in Education*, 6, 2021.

[5] Y. Chen, T. A. Daamen, E. W. T. M. Heurkens, and W. J. Verheul. Interdisciplinary and experiential learning in urban development management education. *International Journal of Technology and Design Education*, 30(5):919–936, Nov 2020.

[6] S. Greipl, K. Moeller, and M. Ninaus. Potential and limits of game-based learning. *International Journal of Technology Enhanced Learning*, 12(4):363–389, 2020.

[7] D. Kolb. *Experiential Learning: Experience As The Source Of Learning And Development*, volume 1. 01 1984.

[8] E. Melis, E. Andres, J. Budenbender, A. Frischauf, G. Goduadze, P. Libbrecht, M. Pollet, and C. Ullrich. ActiveMath: A Generic and Adaptive Web-Based Learning Environment. *International Journal of Artificial Intelligence in Education*, 12:385–407, 2001. Part I of the Special Issue on Adaptive and Intelligent Web-Based Systems (editors Peter Brusilovsky and Christoph Peylo).

[9] MERLOT. Learning Objects. https://www.merlot.org/merlot/, 2024. [Online; accessed 27-May-2024].

[10] NSA. Clark. https://clark.center/, 2024. [Online; accessed 27-May-2024].

[11] A. Parakh, P. Chundi, and M. Subramaniam. An approach towards designing problem networks in serious games. In *2019 IEEE Conference on Games (CoG)*, page 1–8. IEEE Press, 2019.

[12] A. Parakh, M. Subramaniam, and P. Chundi. A framework for incorporating serious games into learning object repositories through

experiential learning. In *Hawaii International Conference on System Sciences, HICSS 2022*, 01 2022.

[13] A. Parakh, M. Subramaniam, P. Chundi, and E. Ostler. A novel approach for embedding and traversing problems in serious games. In *Proceedings of the 21st Annual Conference on Information Technology Education*, SIGITE '20, page 229–235, New York, NY, USA, 2020. Association for Computing Machinery.

[14] V. Rossano, M. Joy, T. Roselli, and E. Sutinen. A taxonomy for definitions and applications of los: A meta-analysis of icalt papers. *Educational Technology & Society*, 8:148–160, 01 2005.

[15] G. Weber and P. Brusilovsky. ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education*, 12:351–384, 2001. Part I of the Special Issue on Adaptive and Intelligent Web-Based Systems (editors Peter Brusilovsky and Christoph Peylo).

[16] G. P. Wiggins and J. McTighe. *Understanding by design*. Association for Supervision and Curriculum Development, Alexandria, VA, expanded 2nd edition edition, 2005.

[17] Wisc-Online. About Learning Objects. https://www.wisc-online.com/about-learning-objects, 2024. [Online; accessed 27-May-2024].

[18] J. Zhang, S. Hutt, J. Ocumpaugh, N. Henderson, A. Goslen, J. P. Rowe, K. E. Boyer, E. Wiebe, B. Mott, and J. Lester. Investigating student interest and engagement in game-based learning environments. In M. M. Rodrigo, N. Matsuda, A. I. Cristea, and V. Dimitrova, editors, *Artificial Intelligence in Education*, pages 711–716, Cham, 2022. Springer International Publishing.

[19] V. Štuikys, R. Burbaitė, K. Bespalova, and G. Ziberkas. Model-driven processes and tools to design robot-based generative learning objects for computer science education. *Science of Computer Programming*, 129:48–71, 2016. Special issue on eLearning Software Architectures.