science, engineering and health studies

Research Article

Science, Engineering and Health Studies
https://lio1.tci-thaijo.org/index.php/sehs
ISSN (Online): 2630-0087

Automatic question generation system for learning to create linear programming models

Kannika Boonkasem[1], Tasanawan Soonklang[1], and Thepchai Supnithi[2]*

[1] Faculty of Science, Silpakorn University, Nakhon Pathom 73000, Thailand
[2] National Electronics and Computer Technology Center (NECTEC), Pathum Thani 12120, Thailand

***Corresponding author**: Tasanawan Soonklang soonklang_t@su.ac.th*

***Received**: 19 January 2024*
***Revised**: 18 November 2024*
***Accepted**: 20 November 2024*
***Published**: 17 October 2025*

***Citation**: Boonkasem, K., Soonklang, T., & Supnithi, T. (2025). Automatic question generation system for learning to create linear programming models. Science, Engineering and Health Studies, 19, 25020004.*

ABSTRACT

Automatic question generation systems play a key role in enhancing the efficiency of teaching and learning, particularly in fields that involve complex problem-solving, such as linear programming (LP). This study presents the development and evaluation of a system designed to generate questions and answers related to business product mix problems in LP. Aimed at enhancing LP modeling skills, the system was tested on 132 undergraduate business students enrolled in a quantitative analysis course. The evaluation involved pre- and post-learning achievement tests, with data analyzed using t-tests and normalized gain (g) to measure learning progress. Results showed significant improvements in students' performance on identical (t = 14.94, p<0.05) and different tests (t = 8.95), along with a moderate learning gain (g = 0.59). These findings indicate that the system not only reduces the instructional workload but also effectively enhances students' understanding and application of LP concepts, making it a valuable tool in education.

Keywords: question generation; learning analytics; linear programming model

1. INTRODUCTION

Automatic question generation (AQG) systems are tools that generate questions based on various topics, ideas, or contextual information in natural language, derived from text paragraphs or images. AQG has become increasingly prevalent in educational settings (Mulla & Gharpure, 2023; Susanti et al., 2018) and has gained remarkable popularity in the business sector and education (Kumar et al., 2019). Research in education (Steuer et al., 2022) has demonstrated the positive impact of automatically generated questions on the learning outcomes of participants.

AQG systems can produce a variety of question types, including wh-questions, true/false questions, and gap-fill questions, also known as fill-in-the-blank or cloze questions (Panchal et al., 2021; Das et al., 2021; Van Campenhout et al., 2021; Kumar et al., 2019; Maurya & Desarkar, 2020; Steuer et al., 2022). Each question type serves specific purposes, such as aiding language comprehension (Fung et al., 2023) and assess reading comprehension (Zou et al., 2022).

Methods for creating automatic questions can be classified into three main categories (Kurdi et al., 2020; Kusuma et al., 2022; Soni et al., 2019): 1) syntax-based, 2) semantic-based, and 3) template-based. One approach within the semantic-based category is the use of ontology technology, which is applied to generate a variety of question types in the educational domain (Kusuma et al., 2020).

Most past and current research on AQG has focused on creating questions in English (Kusuma et al., 2020; Panchal et al., 2021; Zou et al., 2022; Mulla & Gharpure, 2023). A notable gap in the literature is the lack of studies on AQG in other languages, including Thai (Kurdi et al., 2020). However, despite the limited research on Thai AQG, notable developments, including generating questions and answers from Thai sentences, improving word processing, and utilizing models such as MT5 and Transformer (Wiwatbutsiri et al., 2022), have been observed. Additionally, knowledge-augmented approaches have been applied to Thai language models (Ruangchutiphophan et al., 2023). These advancements aim to enhance the accuracy of question generation and answer evaluation by integrating deep learning technology (Chotirat & Meesad, 2021) and natural language processing (NLP) techniques (Zhu et al., 2022). The complexity of the Thai language, including its grammar, diverse word usage, and flexible sentence structure, poses remarkable challenges in developing accurate AQG models (Chotirat & Meesad, 2022). Additionally, the lack of large-scale training data for Thai remains a major challenge compared to English (Wiwatbutsiri et al., 2022; Zhu et al., 2022; Phakmongkol & Vateekul, 2021; Ruangchutiphophan et al., 2023). Tasks such as understanding context, interpreting sentence meaning, word segmentation, and question classification are complex processes that require further development (Phatthiyaphaibun et al., 2023). While progress has been made in generating Thai language questions, improvements are still necessary in the diversity and naturalness of the generated questions (Phakmongkol & Vateekul, 2021).

Generative AI models, such as ChatGPT, have become popular tools for generating various types of questions and answers, offering flexibility without relying on a fixed format. However, these models require substantial data and training resources, and the generated questions and answers are not always correct (Deng & Lin, 2022). This limitation can negatively affect educational applications, potentially undermining the effectiveness of learning exercises. Additionally, research on generating multi-sentence questions that are specifically linked to linear programming (LP) problems is limited.

Developing an AQG system capable of generating long-text questions for analysis remains a challenge, especially for Thai, with its complex structure. Previous research has not provided methods for generating long-text Thai questions with corresponding answers. The proposed system in this study aims to generate accurate Thai questions and answers for analysis and LP modeling without requiring substantial amounts of training data, offering an advantage over generative AI methods. Additionally, an automatic student assistance system has been developed to support learning LP modeling techniques.

LP is a mathematical model used to find optimal solutions based on predefined objectives and conditions. This model is applicable in various business contexts, such as transportation and resource allocation (Ayanian, 2015; Sekhon & Bloom, 2016). A common application of LP is the product-mix problem, where the goal is to maximize profits while managing constraints such as personnel, machines, and materials (Render et al., 2020). However, the effectiveness of the LP model depends on accurately representing the business issue (Martinich, 1997).

In higher education, LP is a key component of the quantitative analysis in business (QBA) course for undergraduate business students. This course teaches students how to analyze problems, define variables, formulate objective functions, and create conditional functions for different business scenarios. Consequently, students practice constructing models with the correct structure.

Building LP models from descriptive text requires strong analytical skills. The complexity of LP problems can make it difficult for students to construct effective models (Pongchairerks, 2017). Additionally, Vibulsukh (1996) highlighted that students often struggle with LP due to an incomplete understanding of methods and procedures, particularly when faced with unfamiliar scenarios. Therefore, for improvement, students need to practice problem analysis, variable specification, objective function creation, and constraint setting across a variety of problems tailored to their knowledge level. Building on these foundations, this study provides two notable contributions to the field.

Development of an automatic question and answer generation system for LP modeling: This study introduces a system specifically designed to generate questions and answers related to LP problems, particularly focusing on business product mix scenarios. The system automates the creation of complex, descriptive questions essential for teaching LP modeling skills, reducing the time and effort required from instructors while ensuring consistent quality in educational assessments.

Enhancement of LP modeling skills through automated learner assistance: The research demonstrates that the system effectively enhances the LP modeling skills of students by providing personalized, adaptive learning experiences. The system tailors questions to align with learning objectives, facilitates continuous skill development, and markedly boosts the academic performance of students by integrating templates and ontology technology, as evidenced by measurable gains in pre-test and post-test scores.

2. MATERIALS AND METHODS

2.1 Structure of the LP model

The LP model comprises three core elements: decision variables, an objective function, and constraint functions. Identifying the key variables is essential for determining the desired outcome of the model. These variables are crucial components in the objective function and conditional functions. For example, let the decision variable X represent the quantity of product j that a company aims to produce to maximize total profit. An objective function is a mathematical expression used to determine the minimum or maximum value, such as maximizing profit or minimizing cost, for the products being produced. This function is typically formulated as a mathematical involving the related variables. A constraint function is an equation or inequality that represents restrictions related to resource requirements or various problem conditions. These functions describe the relationships between different variables within each condition. The number of conditions depends on the difficulty and complexity of the problem being modeled.

In constrained LP models, the goal is to find values for the decision variables that either maximize or minimize the objective function, while simultaneously satisfying all the given constraints. The example below shows how an operational problem can be modeled and analyzed using an optimization LP approach.

***Problem*:** The company specializes in producing ready-made clothing tailored for working women, offering two main product types: shirts and skirts. Each skirt requires 1.5 m of fabric and 2 h of sewing time, with each skirt requiring 2 m of fabric and 1 h of sewing time. The total available work time for tailors is limited to 100 h, and the total fabric allocation for production is 150 m. The company earns a profit of 200 baht per shirt and 220 baht per skirt. The objective is to determine the optimal number of shirts and skirts to maximize the overall profit.

***Decision variables*:** First, the necessary requirement to determine the production quantities for each type of clothing has been identified. The following values will be assigned to represent these variables: X_1 = number of shirts produced, X_2 = number of skirts produced.

***Objective function*:** The goal is to maximize the profit from production. The profit per shirt is 200 baht, while that for a skirt is 220 baht. The total profit, which is denoted as MaxZ, is defined as the objective function.

Max $Z = 200X_1 + 220X_2$, where X_1 and X_2 represent the quantities of the respective products.

***Constraints function*:** The company encounters limitations in terms of the quantity of fabric available for each product. This constraint function will be expressed as follows:

$$1.5X_1 + 2X_2 \leq 150$$

Employing similar reasoning, the restriction on the availability of work time for tailors can be expressed as follows:

$$2X_1 + X_2 \leq 100$$

Finally, considering the impracticality of negative production levels, X_1 and X_2 should be greater than 0. Combining all these conditions yields the following LP model:

$$\text{MaxZ} = 200X_1 + 220X_2$$

Subject to:

$$\begin{aligned} 1.5X_1 + 2X_2 &\leq 150 \\ 2X_1 + X_2 &\leq 100 \\ X_1, X_2 &\geq 0 \end{aligned}$$

This model is referred to as an LP model or linear program, given that the objective function and all constraint functions are linear.

2.2 Problem of generating LP questions

Manually demonstrating LP questions is a challenging task that requires careful attention. The example above illustrates how LP models are developed from business problems using mathematical symbols. However, manually crafting these questions can be a time-consuming task, as they often involve lengthy, interconnected sentences, increasing their complexity. Automated tools that generate questions and provide answers would drastically reduce the workload of teachers. Additionally, these tools would be valuable in the learning process, fostering skill development among students.

2.3 System overview

The overview of the AQG system for learning to create LP models, as illustrated in Figure 1, is designed to support the learning process of LP modeling. This system comprises three main parts:

2.3.1 Question generation (QG) process

This section deals with the automatic generation of linear scheduling problems and corresponding solutions in the production proportional problem domain. These generated problems and answers serve as exercises for the learner and comprise five steps:

Step 1: Collection of questions—This step involves gathering questions related to the product-mix problem from textbooks and websites. A total of 100 questions is collected from various sources, including textbooks and other instructors. Subsequently, the structure, correlation, and format of the problem descriptions are analyzed. The problem description for the product-mix problem includes components such as:

A comprehensive list of all targeted products, an inclusive inventory of raw materials and essential resources, cost or net profit coefficients, the quantity of resources used in the production process, and the quantity of available resources. Additionally, the problem description can be divided into two parts, as illustrated in Figure 2.

First, the content part establishes the connection between resources and products in the given question. This section provides details regarding the producer, the resources involved, the products, the quantity of resources required for production, the quantity of available resources, and the value assigned to each product (profit or cost).

The second part, known as the problem section, identifies the objective of the question, which typically involves either maximizing profit or minimizing costs to determine the best possible solution.

Figure 2 showcases an example of a problem description within the product-mix problem domain, featuring two products: toys A and B. The second sentence in the first paragraph illustrates the resources required for cutting and assembling toy types A and B. The limited resources are cutting and assembling hours, and the unit profit of Toys A and B is mentioned in the third sentence.

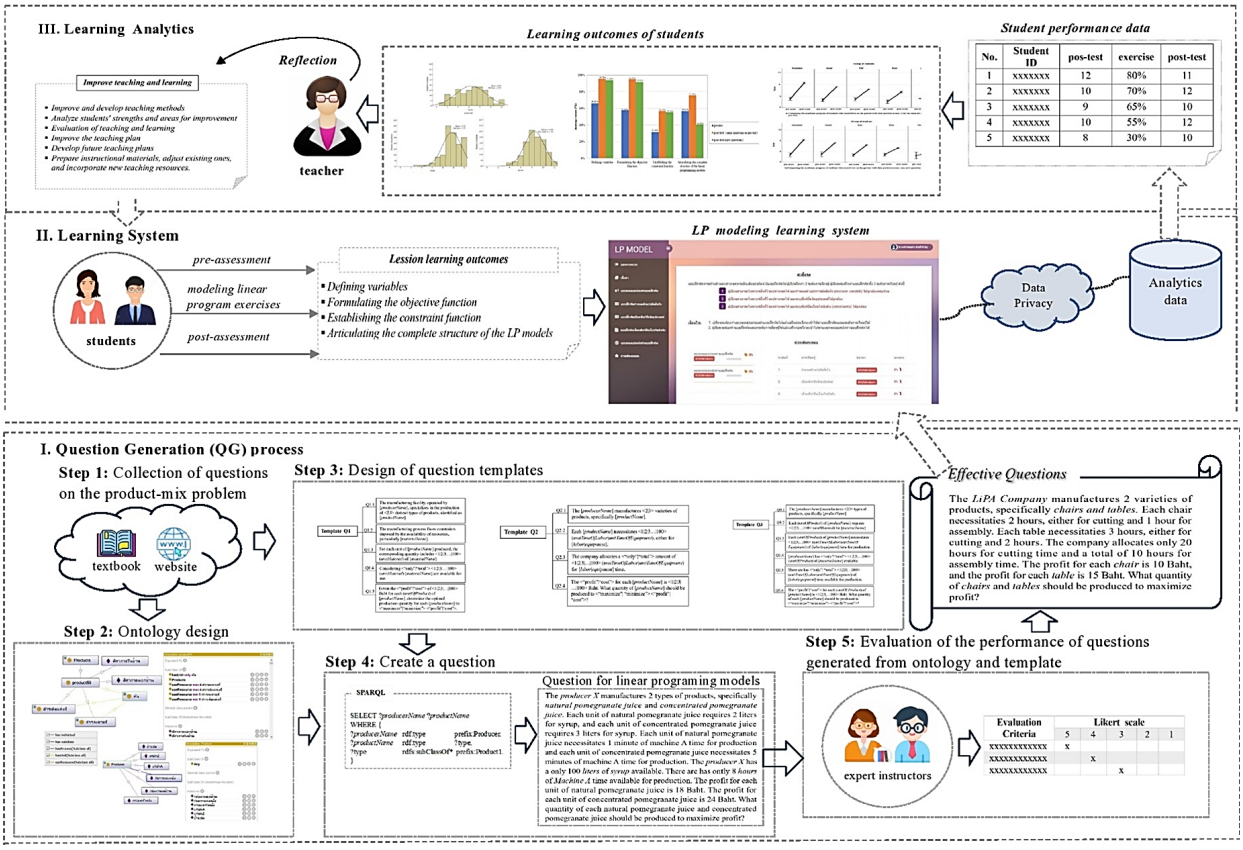
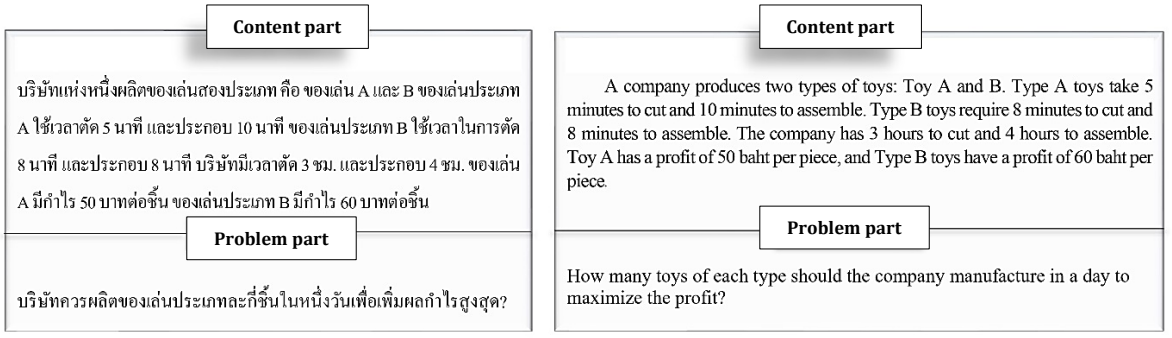


Figure 1. Overview of the AQG system for learning to create LP models
Note: The system comprises three parts: I. Question generation (QG) process, II. Learning system, and III. Learning analytics



a) Problem description of product-mix problem in Thai

b) English translation of the Thai text with problem description of product-mix

Figure 2. Problem description of the product-mix problem

Step 2: Ontology design—An ontology is created to structure the knowledge and categorize the questions systematically. As shown in Figure 3, the ontology design begins with the creation of the base class Thing. Next, four subclasses related to components in the content part (producer, product, resource, and classifier) are added. These classes are then populated with entities and their respective subclasses, enabling a more comprehensive representation of the production proportioning problem. Three types of relationships exist between the classes in the ontology:

The hierarchy of relationships (is-a hierarchy) represents the class structure in which subclasses inherit properties and characteristics from their parent classes. For example, Product1 and Product2, as subclasses, inherit attributes from the Product class.

The representation as class member relationships (instance-of) signifies how individual entities serve as instances or members of specific classes. For example, a table could be an instance of the Product2 class, and Conston Company could be an instance of the Producer class. The details of the classes and relationships for each type are as follows:

The Producer class represents the names of product manufacturers, with instances such as Thai Butter Company and Silly Nut Company.

The Products class has a subclass called group products, which includes products that use similar resources, such as the Product1 and Product2 classes. Instances of this class include both abstract words representing product categories, such as Product A and Product B, as well as actual products found in the real world, such as table and chair.

The Resource class pertains to the resources used in the production of products and is divided into three subclasses: Equipment, Labor, and RawMaterial.

Within the Equipment class, further subclasses are denoted by abstract terms related to machines, such as MachinesX and MachinesY. In contrast, the EqPhysical class encompasses real-world types of machines, including tire-cutting machines and sewing machine.

The Labor class is further categorized into subclasses such as tailors and carpenters. The RawMaterial class is divided into two subclasses: RawAbstract and RawPhysical. The RawAbstract class includes subclasses with abstract terms related to materials, such as rawmaterialA and rawmaterialB. The RawPhysical class comprises subclasses representing physical materials, such as pine and mahogany.

The Classifier class serves as a classification system designed to quantify time and quantities of products or production resources. This system encompasses two subclasses: ClassifierAbstract and ClassifierPhysical. The ClassifierAbstract class contains specific data instances and abstract terms related to units and types. In contrast, the ClassifierPhysical class is further categorized into four subclasses: countable, measure, time, and weight classes. These subclasses offer sample data in units applicable to products or resources, such as pound, foot, inch, kilogram, minutes, hours, and pieces.

Object properties denote relationships between two individuals or instances of a class. The definitions of these object properties are outlined as follows:

hasProcess: This property defines the relationships between the classes Products, Labor, and Equipment, representing the production process of a product. Additionally, this property encompasses various processes, including cutting, sewing, finishing, rolling, packing, weaving, and manufacturing, whether executed by labor or machine.

hasUnit: This property establishes relationships between the classes Products, Resource, and Classifier, indicating the units associated with products and resources.

useResource: This property highlights the relationships between the classes Products and RawMaterial, specifying the raw material used in the manufacturing of a product.

useTime: This property defines the connections between the classes Products, Labor, and Equipment, focusing on determining the required time for the manufacturing process.

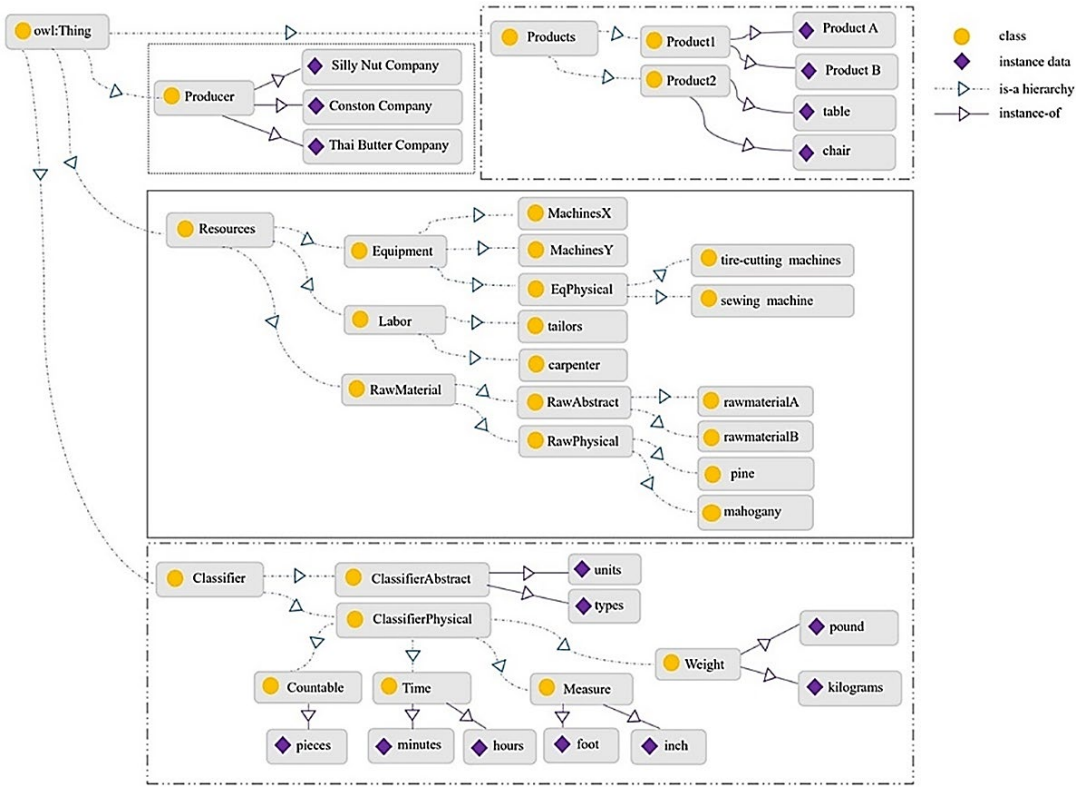


Figure 3. Graph depicting some of the ontology class hierarchies and their relationships with class members

Step 3: Design of question templates—Templates for questions are designed to standardize the question generation process. In the templates, square brackets [ ] represent instances of classes defined in the domain ontology, which will be populated when generating questions. Angle brackets < > signify a variable that can hold either a string or a number, with values randomly selected for each question. The values in curly brackets { } can be retrieved and displayed multiple times, with each value concatenated by the word 'and.' Parentheses ( ) represent object properties. For instance, "hasUnit" denotes the unit quantity of a product or production

resource, which will be represented as instance data in the 'Classifier' class.

In this study, three distinct question types were generated for the product-mix problem. These question types are referred to as Template Q1, Template Q2, and Template Q3. Each template is formulated as follows (translated from Thai to English):

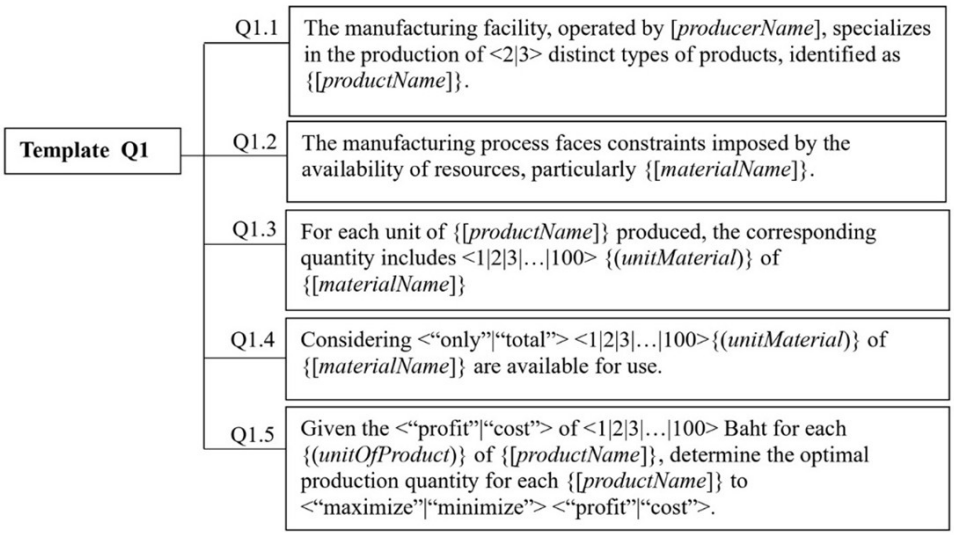


Figure 4. Product-mix problem sub-templates of template Q1 for raw material usage

Template Q1: This template focuses on questions related to products that rely solely on raw materials in their production. The template comprises five sub-templates (Q1.1–Q1.5), as shown in Figure 4. The manufacturing facility of the specified producer produces various product types, with production constrained by the availability of raw materials. Each product requires a specific quantity of raw materials per unit produced. The template considers the available quantities of raw materials and aims to determine the optimal production quantity for each product, either to maximize profit or minimize costs, based on the cost or profit per unit.

Template Q2: This template focuses on products produced using specific labor or equipment in the production process. The template comprises four sub-templates (Q2.1–Q2.4), as shown in Figure 5. This template asks for the names of 2 or 3 different product types produced by the company, the labor or equipment time required for each product, and the total labor or equipment time available. Additionally, the template requests for the calculation of the optimal production quantity for each product, aiming to maximize profit or minimize cost for each product, based on the profit or cost per unit.

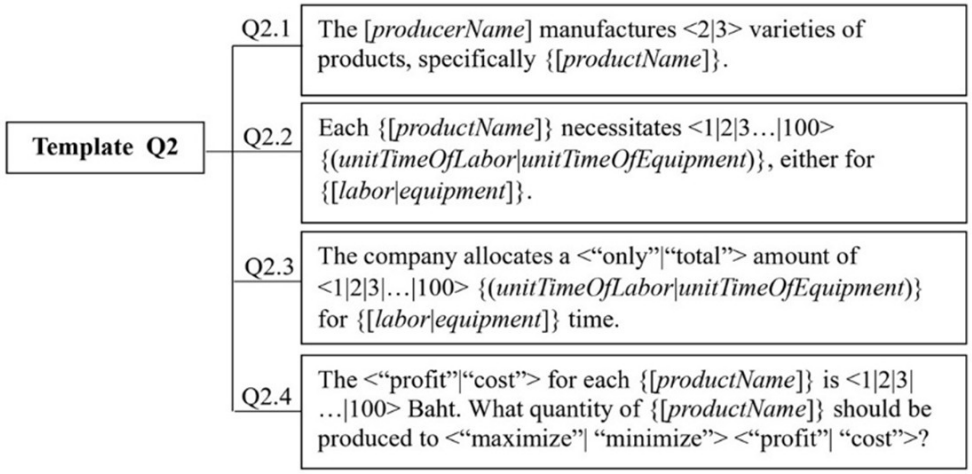


Figure 5. Product-mix problem sub-templates of template Q2 for labor and equipment usage

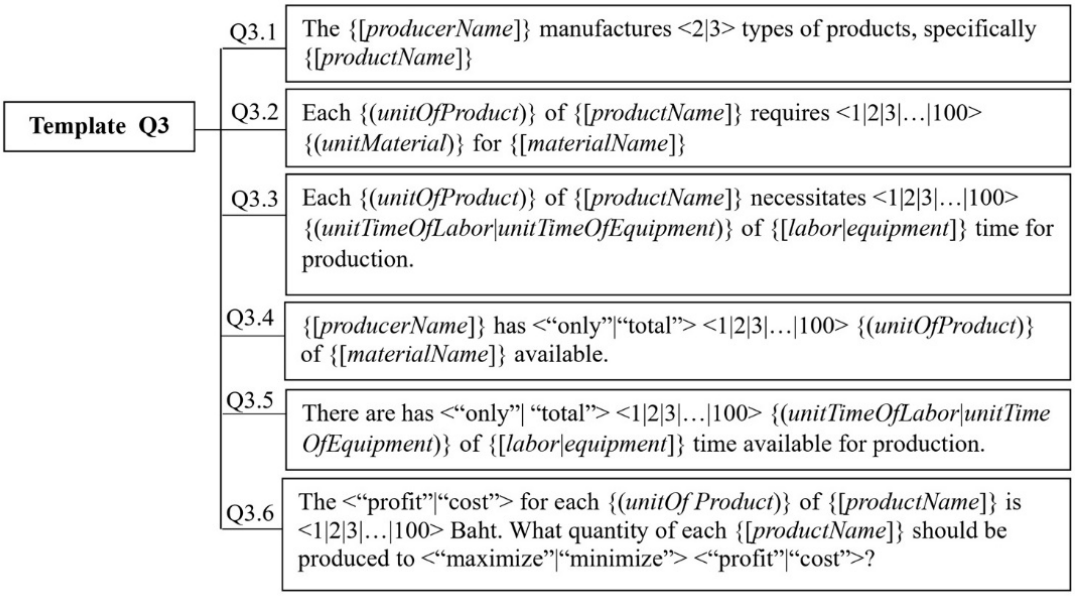


Figure 6. Product-mix problem sub-templates of template Q3 on product, material, and labor/equipment time allocation

Template Q3: This template focuses on the products used, the materials, and the time for labor or equipment in the production process. The template comprises six sub-templates (Q3.1–Q3.6), as shown in Figure 6. This template specifies that the producer manufactures 2 or 3 types of products and requests for their names. The template details the material and labor or equipment time required for each unit and the total material and labor or equipment time available. Finally, this template requests for the calculation of the optimal quantity of each product, either aiming to maximize profit or minimize cost, based on the profit or cost per unit.

Step 4: Create a question—Using the ontology and templates from Step 3, specific questions are created using the Resource Description Framework (**RDF**) and SPARQL. RDF structures knowledge into classes with properties, forming subject–predicate–object triples to describe ontologies. These components enable knowledge extraction through SPARQL.

First, questions for template Q1 are created. Sub-template Q1.1 uses [*producerName*] and [*productName*] as instances of the Producer and Product classes, respectively. These entities are linked through RDF properties and extracted using a SPARQL query (Figure 7) to form sentence S1.1. Similarly, sub-template Q1.2 retrieves [*materialName*] using another SPARQL query (Figure 8) to generate sentence S1.2.

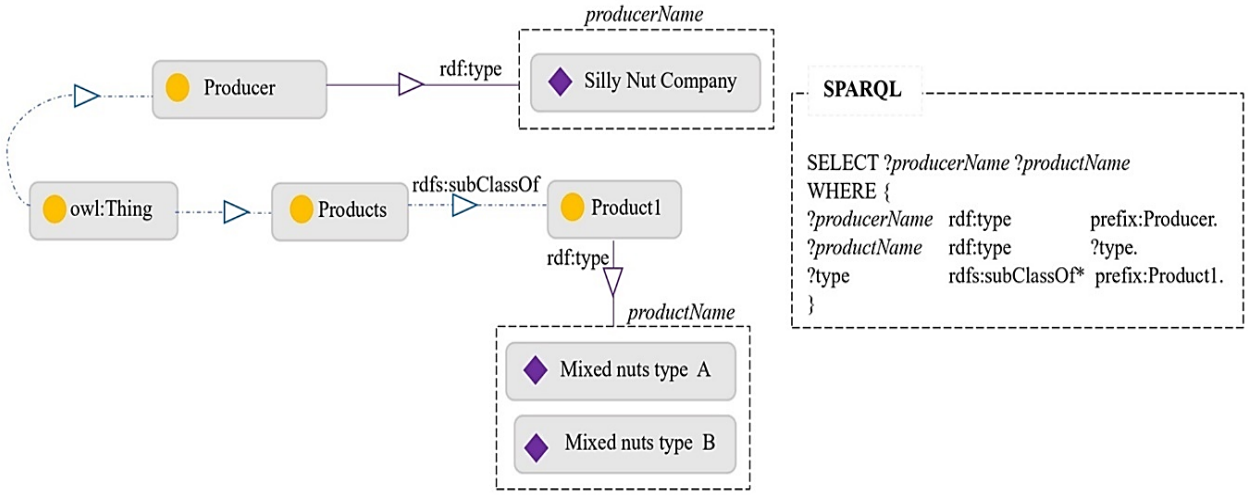


Figure 7. Ontology & SPARQL for producer name and product name

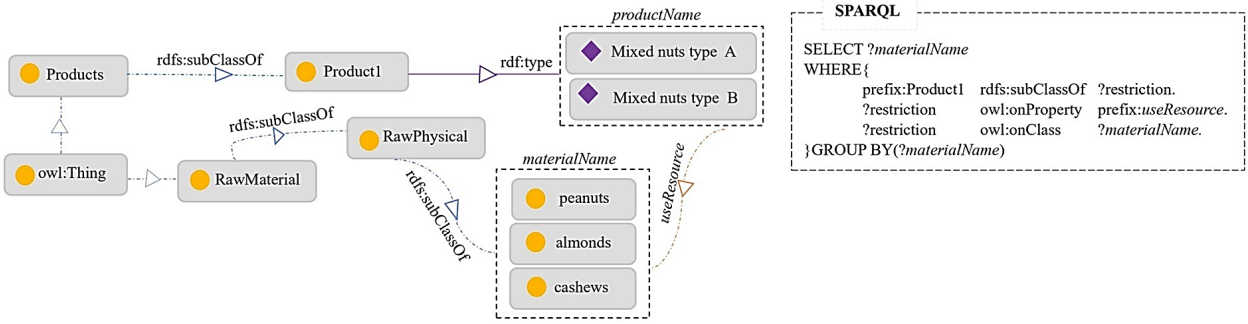


Figure 8. Ontology & SPARQL for raw material name

For sub-template Q1.3, the quantity of raw materials used per product unit is specified, with the material unit (*unitMaterial*) selected from Figure 9. Sentence S1.3 is created for Q1.3. Sentence S1.4 of sub-template Q1.4 details the amount of material required per product unit, while sentence S1.5 of sub-template Q1.5 expresses the value in terms of profit or cost to determine the most appropriate total value. The product unit (*unitOfProduct*) is also selected from Figure 9. Example result sentences (S1.1–S1.5) are shown in Figure 10.

Additionally, example sentences for each sub-template (Q2.1–Q2.4) and result sentences (S2.1–S2.4) of template Q2 are provided, as shown in Figure 11. These result sentences are obtained from data extracted from the ontology (Figures 7–9). Similarly, sample sentences (S3.1–S3.6) for each sub-template (Q3.1–Q3.6) of template Q3 are presented using the same methods specified in templates Q1 and Q2, as shown in Figure 12.

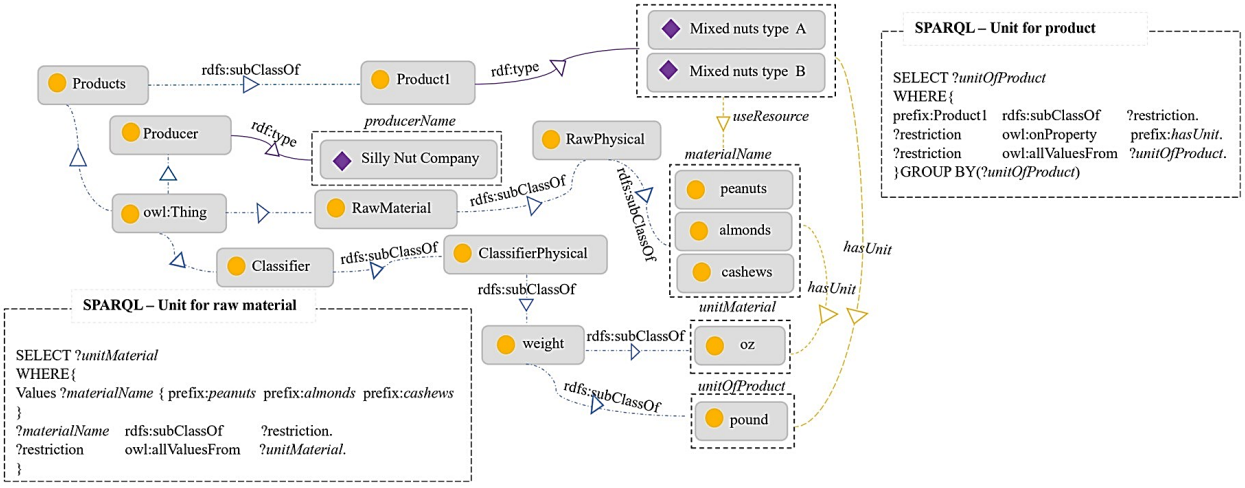


Figure 9. Ontology & SPARQL for unit of raw material and unit of product

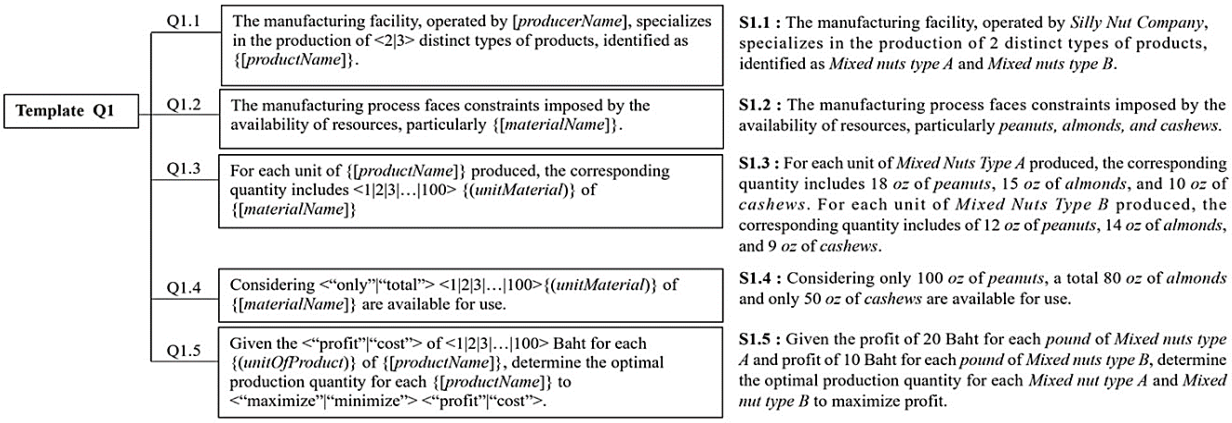


Figure 10. Structure of template Q1 with sub-templates and resulting sentences

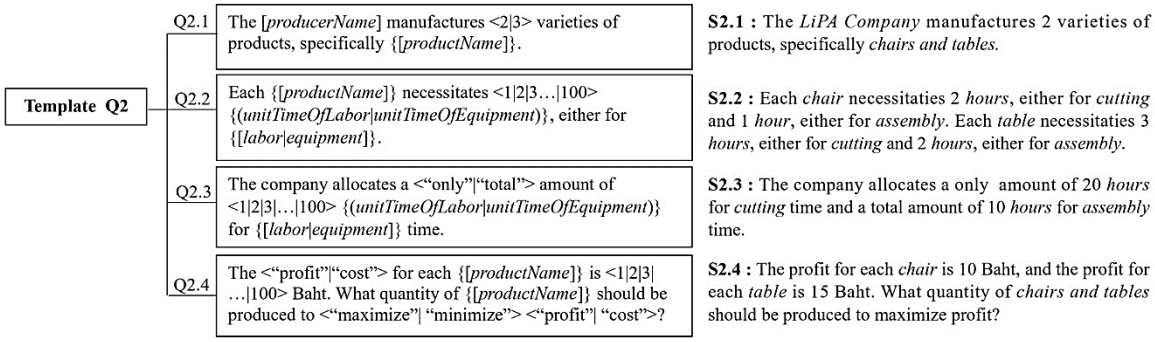


Figure 11. Structure of template Q2 with sub-templates and resulting sentences

Step 5: Evaluation of questions—The questions generated from the ontology and template were evaluated on the following three aspects: the accuracy of the questions and answers, the consistency of the content, and the appropriateness of the difficulty level. This evaluation was conducted by three expert instructors who teach LP courses. They used a five-point Likert scale (Jamieson, 2004) to express their opinions. The evaluation results indicated that the questions and answers generated by the system were accurate, the content was consistent, and the overall difficulty level was moderate, making them well-suited for use with students.

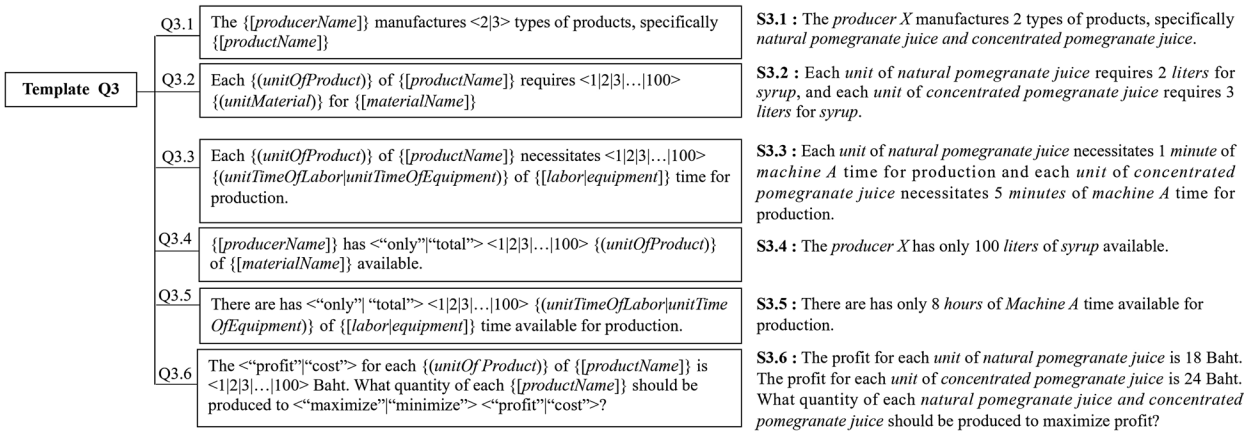


Figure 12. Structure of template Q3 with sub-templates and resulting sentences

2.3.2 Learning system

The second stage continued with the design and development of a web application-based LP modeling learning system. This system, grounded in Gagné's learning theory (Gagné, 1985), comprises three key components: pre-assessment, modeling linear program exercises, and post-assessment.

Pre-assessment: Students complete a pre-test to assess their knowledge before starting exercises. This multiple-choice test, generated by a system, contains 12 questions. Each correct answer earns one point, and students can view their scores after completing the test.

Modeling linear program exercises: After the pre-test, students work through exercises on a web-based application, which is divided into three levels: defining decision variables, formulating the objective function, and writing constraint functions. For each level, students answer system-generated questions. If an answer is incorrect, the system provides a warning. After two incorrect attempts, hints are provided to guide the students. Scores are tracked based on correct and incorrect answers. Students must answer correctly before proceeding to the next exercise.

Post-assessment: After scoring 100% on all exercises, students take a post-test with 24 multiple-choice questions. The first 12 questions introduce new challenges, while the last 12 revisit pre-test problems. Scores are displayed similarly to the pre-test.

2.3.3 Learning analytics

In the learning analysis process, an assessment of the pre-test and post-test was conducted to ensure the validity of the content. Three experts assessed 12 questions to evaluate the alignment between the test and the intended behavioral learning objectives. The expert review confirmed the alignment among the problems, questions, and objectives. System performance and learning achievement were assessed by comparing pre-test and post-test scores, employing statistical analyses (average, SD, and t-test). The average normalized gain method (Hake, 1998) measured learning progress across the following three levels: low gain ($g < 0.3$), medium gain ($0.3 \leq g < 0.7$), and high gain ($g \geq 0.7$).

Student achievement test scores revealed four lesson learning outcomes: identifying decision variables, formulating the objective function, identifying and formulating constraints, and writing the complete

structure of LP models. Additionally, students were classified into five groups based on their scores: *excellent* (10–12 points or 80%–100%), *good* (9 points or 70%–79%), *fair* (8 points or 60%–67%), *poor* (6–7 points or 50%–59%), and *very poor* (0–5 points or 0–49%). This classification was determined using a fixed criteria method for grading in the QBA course by calculating the proportion of the score range compared to 100% based on the received scores of the students.

3. RESULTS AND DISCUSSION

The efficiency of a question generation system, which uses an ontology-based knowledge base and question templates, was evaluated with production allocation as a case study. The question templates were found to adapt flexibly to information from the ontology-based knowledge base. Three experts assessed the generated questions and solutions, rating them as moderately appropriate in terms of accuracy ($\bar{X}$ = 3.31, SD = 0.58), content relevance ($\bar{X}$ = 3.41, SD = 0.71), and difficulty level according to the learners' knowledge ($\bar{X}$ = 3.03, SD = 0.70). This system shows potential for application in other domains to create LP models for presenting business problems.

In this research, the efficiency of the learning management process (E_1) and the efficiency of the learning outcomes (E_2) are analyzed (Brahmawong, 2013). The analysis of E_2 was based on the average scores and percentages from the learners' exercises in the developed system. Additionally, the efficiency of E_2 was assessed using achievement results from two sets of post-exercise tests. The results showed that the efficiency of process E_1 was 89.66, and the efficiency of E_2 was 80.81 when using the same test as before the exercises, which is higher than the standard criterion (80/80). However, when a new test was used, the efficiency of learning outcomes (E_2) dropped to 70.45, which is below the set criterion. These research results indicate that the exercises may be easier than the tests. Learners must practice linear programming model construction skills with a variety of problems beyond those provided by the developed system.

3.1 Comparison of pre-test and post-test among students in different groups

Figure 13 shows the experimental learning results of 132 students, presenting the pre-test and post-test outcomes. The post-test included the same questions as the pre-test (**have seen**) and new questions (**never seen**). The pre-test results indicated that most students scored very poorly, with 50 students (37.88%) in this category. Only 16 students (12.12%) achieved an excellent score, while the numbers scoring good, fair, and poor were relatively close.

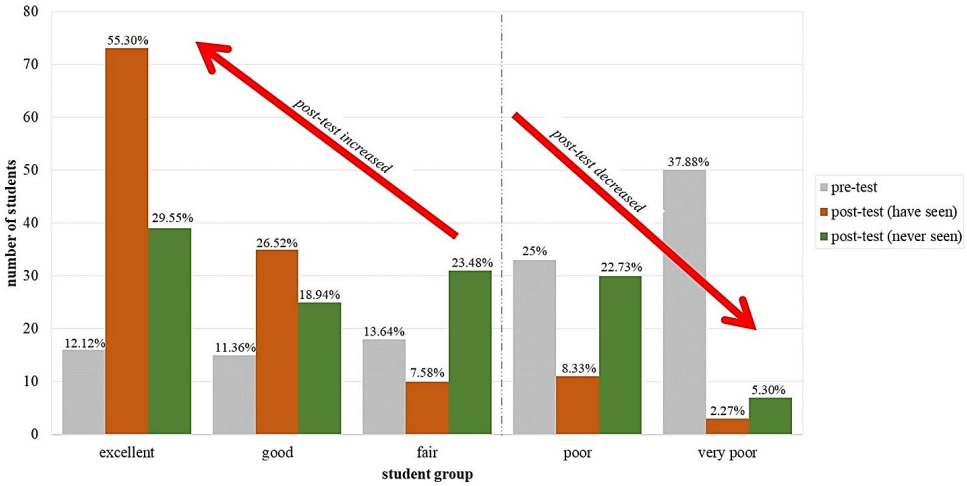


Figure 13. Comparing the percentage of student scores in five groups based on their pre-test and post-test scores

Comparison results of the pre-test and post-test scores revealed that the number of students who scored excellent and good increased in both post-tests. However, the increase in students scoring excellent and good in the post-test (never seen) was less than in the post-test (have seen). Simultaneously, the number of students scoring fair notably decreased in the post-test (have seen) but increased in the post-test (never seen). This change indicates that many students improved from fair to excellent and good in the post-test (have seen), implying a deeper understanding. Meanwhile, some students still struggled to fully grasp the material in the post-test (never seen) and thus scored fair. Additionally, the number of students who scored poor and very poor decreased in both post-tests, with a greater decrease observed in the post-test (have seen) compared to the post-test (never seen).

Figure 14 illustrates the learning progress of 50 students who initially scored very low on the pre-test. The figure presents three groups: the pre-test scores, post-test (have seen) scores, and post-test (never seen). The results demonstrate a considerable improvement in post-test scores compared to pre-test scores, regardless of whether the students encountered familiar or new questions. This improvement implies that the learning activities implemented in this study effectively enhanced the abilities of students, particularly in problem analysis and modeling. Even students with initially very low scores drastically improved their performance after participating in the learning activities.

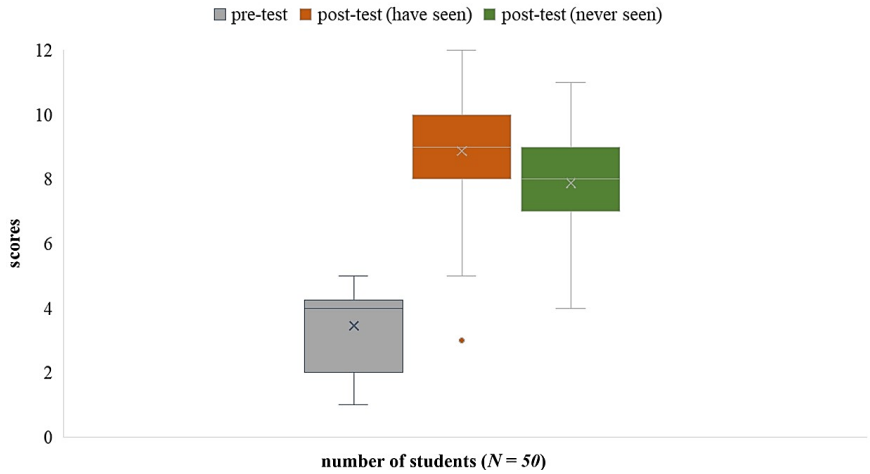


Figure 14. Learning progress of 50 students who initially scored very low on the pre-test

3.2 Individual achievement before and after learning

When analyzing the pre-test scores of students, the mean was 6.38, with 67 students (50.76%) scoring above average and 65 students (49.24%) scoring below average. The post-test (have seen) scores exhibited an increased mean of 9.70, with 73 students (55.30%) scoring above average and 59 students (44.70%) scoring below average. Similarly, in the evaluation of post-test (never seen) scores, the average increased to 8.45, with 64 students (48.48%) scoring above average and 68 students (51.52%) scoring below average.

Figure 15 compares the score distributions before and after completion of the exercises. After completion, the average scores increased for the original and new questions, indicating an improvement in student skills. The average score for the original questions (9.70) was higher than for the new questions (8.45), but both were still higher than the pre-test average score (6.38). The post-test score distributions for both sets of questions were narrower compared to the pre-test, indicating that most students had notably similar scores after completion of the exercises, which reveals a substantial improvement in skills. Furthermore, regardless of whether the original or new questions were used, the scores increased clearly and consistently in the post-test. This experiment concludes that completing exercises effectively improves the skills of students, regardless of whether original or new questions are used for assessment.

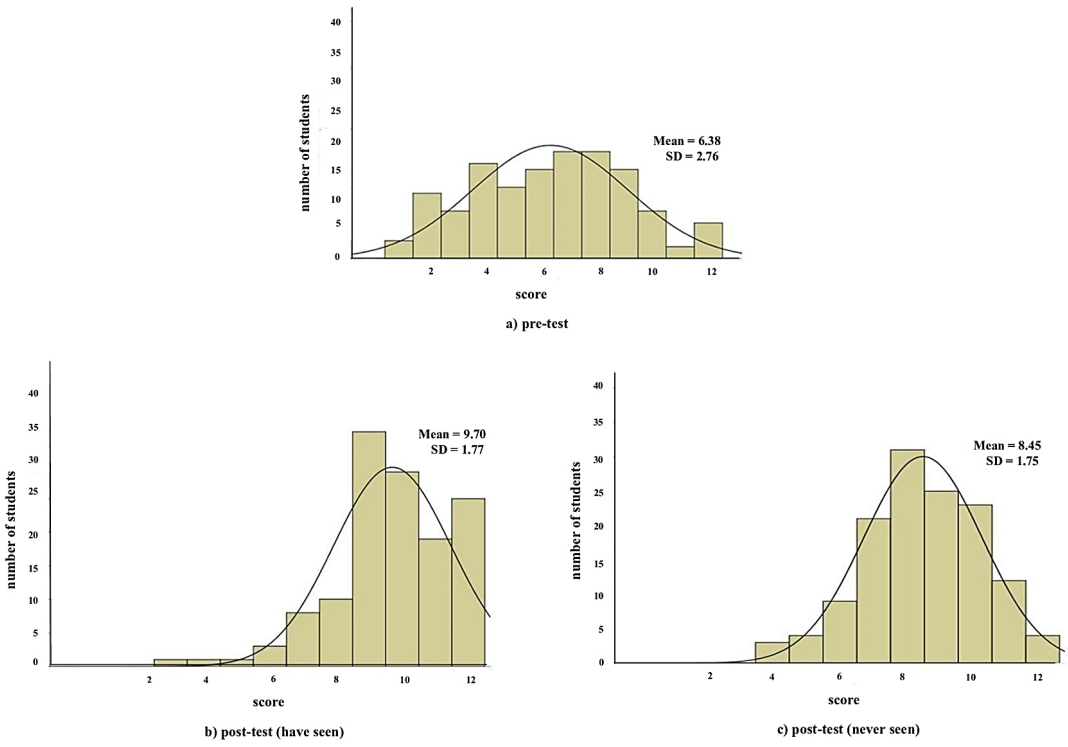


Figure 15. Comparison of score distribution before and after the exercise. (a) pre-test score distribution, (b) post-test score distribution (have seen), and (c) post-test score distribution (never seen)

In this study, t-test statistics were used to compare the average scores before and after the learning arrangement for the sample. Table 1 shows that the average scores on the post-test (have seen) (9.70) and with new questions (8.45) were higher than the pre-test average score (6.38), and this difference was statistically significant at the 0.05 level. Notably, the AQG system for learning linear deterministic modeling has contributed to the increased learning achievement of students.

Table 1. Summary of pre-test and post-test score percentages and t-test results (n = 132)

	Mean	SD	t
Pair 1 pre-test score	6.38 (53.17%)	2.76	14.94*
post-test (have seen) scores	9.70 (80.83%)	1.77	
Pair 2 pre-test score	6.38 (53.17%)	2.76	8.95*
post-test (never seen) scores	8.45 (70.42%)	1.75	

* $p < .05$

3.3 Analyzing learning progress

Initially, the sample students revealed an average classroom achievement score of 53.17 before engaging in exercises across all three learning levels and undergoing two test sets. Data analysis using the normalized gain technique revealed that when students took the post-test (have seen), the entire class displayed an average learning progress of 0.59 (g = 0.59). This finding indicates that the entire class made moderate progress, demonstrating an increase in learning achievement of 59%. When evaluating the results of the exercises using a test with new questions, learners in the entire class achieved an average learning progress of 0.37 (g = 0.37). Overall, the progress of learners consistently remained at a moderate level, exhibiting a 37% increase.

Figure 16 shows a comparison of test results before and after learning, categorized by the normalized gain in learning progress. The results indicate that when students took the post-test with familiar questions, most students (42.42%) exhibited moderate learning progress. Only 20.45% of students exhibited low progress, while 37.12% achieved high progress, indicating substantial improvement for many. However, when new questions were introduced in the post-test, the majority of students (46.21%) demonstrated moderate progress. The proportion of students with low progress increased to 43.94%, while those with high progress decreased to 9.85%.

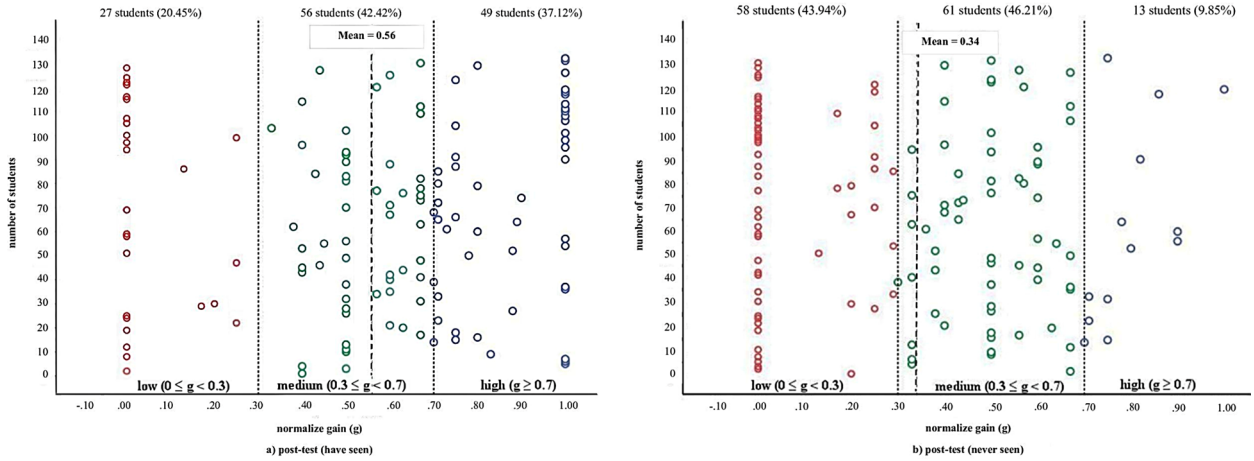


Figure 16. Comparison of pre-test and post-test results, categorized into groups based on normalized gain

The figure may indicate that the new questions were extremely challenging or posed difficulties in applying knowledge to new situations. Overall, the findings indicate that students effectively improved their knowledge with familiar questions, but applying knowledge to new questions remains a challenge for some groups.

3.4 Learning progress in terms of learning achievement by content

This study focused on students' learning achievements in writing LP models, assessing four learning outcomes: identifying decision variables (**variables**), formulating the objective function (**objective function**), identifying and formulating constraints (**constraints**), and writing the complete structure of LP models (**models**). Figure 17 shows the observed learning progression across each content area by comparing pre-test and post-test results. The data reveal substantial improvements, especially in the abilities of students to work with variables and objective functions. After completing the exercises and taking the post-test (have seen), students achieved the highest scores in variables (average score of 95.70%) and the objective function (average score of 94.95%) compared to other areas. Even on the post-test (never seen), students

still performed best in variables (average score of 93.69%) and the objective function (average score of 91.92%). These results indicate a notable increase in scores across all topics, indicating that the exercise practice developed in this study positively impacted students' understanding and performance.

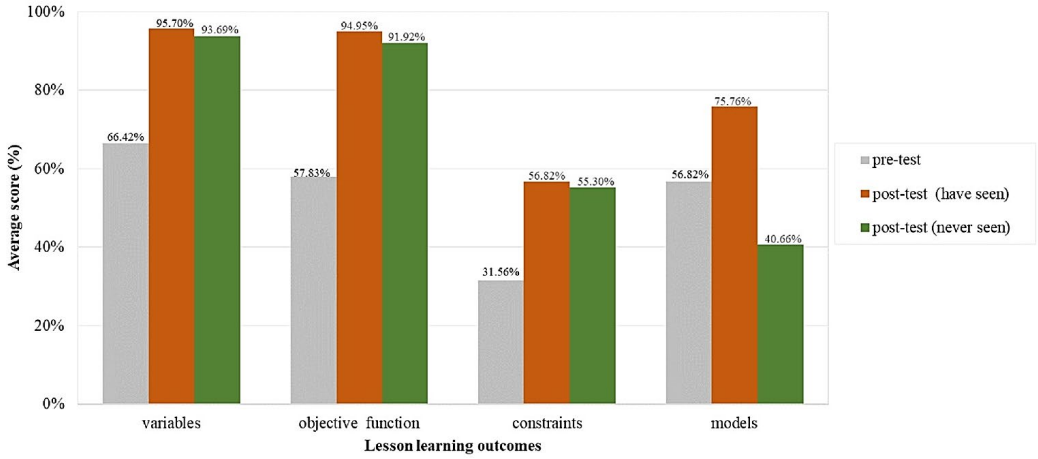


Figure 17. Learning progression in students' scores across each content area, measured by comparing pre-test and post-test results

Figure 18 shows that in the pre-test, most students scored less than 3 points in all types of questions. Some students revealed slight improvement in certain content areas, although the changes were not statistically significant. After completing the exercises and taking the post-test (have seen), more students scored higher in all content areas, with a noticeable increase in the number of students scoring the maximum of 3 points. This finding demonstrates that the learning helped students improved their ability to answer questions. In the post-test (never seen), students performed better on questions related to variables and objective function, with many achieving the highest score of 3 points. For questions on constraints and models, a substantial number of students also scored highly. This finding indicates that the exercises developed in this research effectively enhanced students' ability to understand and answer questions regarding formulating LP models, whether with familiar and new questions.

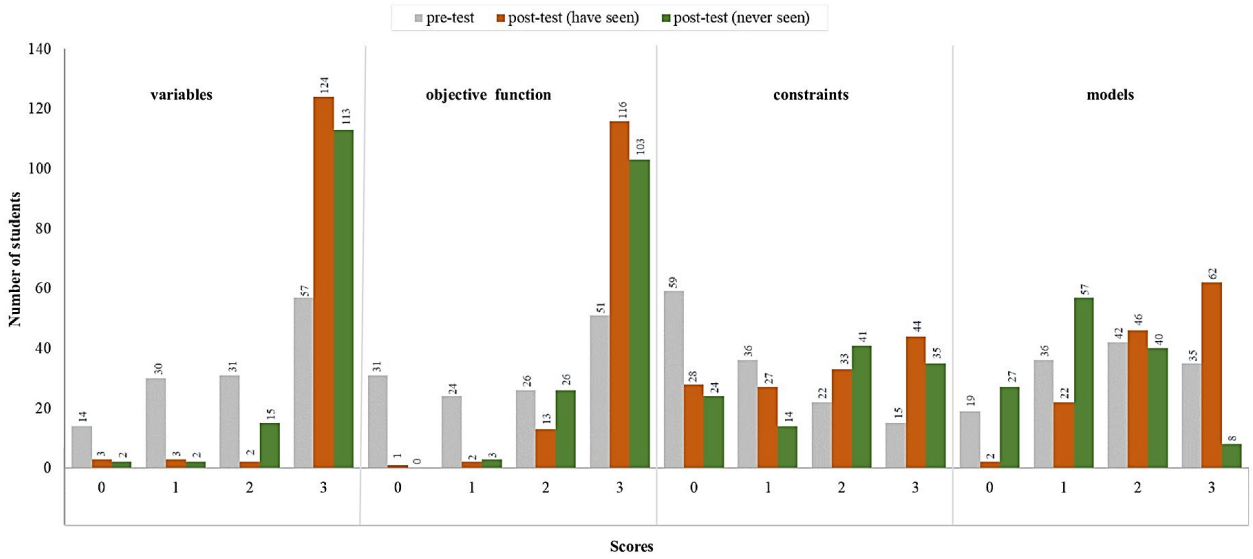


Figure 18. Distribution of students' scores across four content areas—defining variables, identifying the objective function and constraints, and writing complete models during the pre-test, post-test (have seen), and post-test (never seen)

Figure 19 illustrates the learning progress of students in formulating LP models by comparing post-test (have seen) results and post-test (never seen) results. The comparison is realized in accordance with the lesson learning outcomes and the normalized gain of the students. Notably, after students practiced the exercises and took the post-test (have seen), a notable increase in learning progress was observed in all content areas, especially in variables and objective function, where many students showed the greatest improvement.

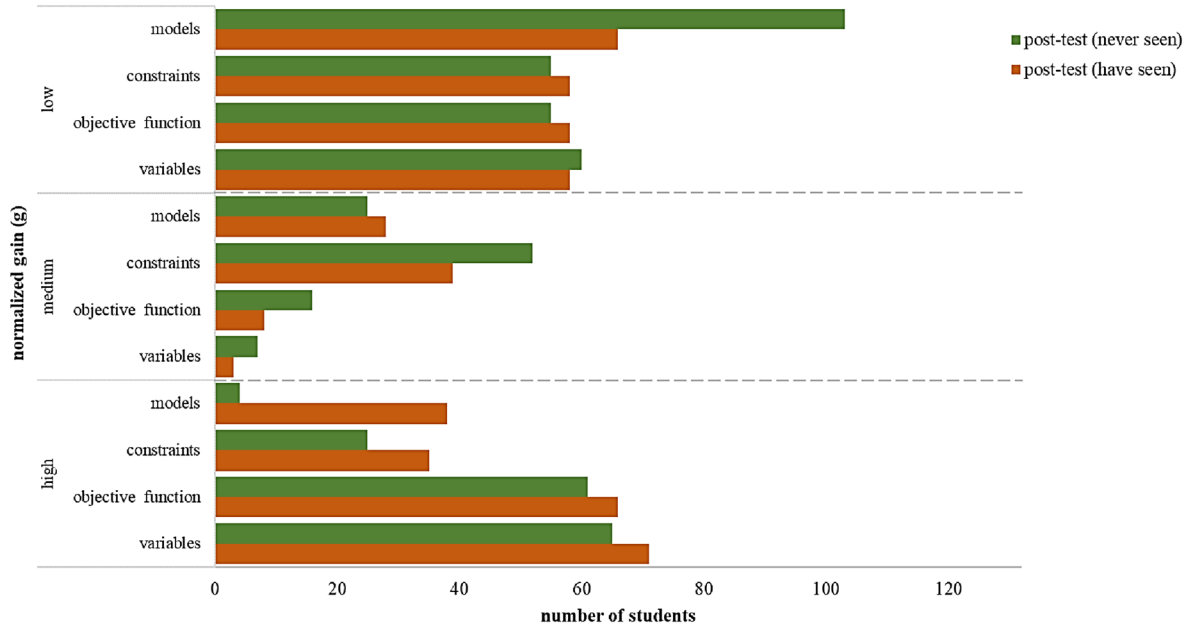


Figure 19. Comparison of learning progress of students in LP modeling using the normalized gain method at three levels of progression (low, medium, and high) across different content areas, including defining variables, identifying objective functions and constraints, and writing complete models, based on post-test results (have seen and never seen)

For the content on constraint functions and the complete structure of LP models, most students demonstrated limited learning progress. When analyzing the learning progress by content area after students took the post-test (never seen), students exhibited progress in all learning outcomes across all levels. However, some students scored full marks before and after completing the exercises. Based on the analysis of the level of learning progress, this group showed no change in progress and was therefore categorized with students demonstrating low learning progress, resulting in an inflated number of students in the low progress category. This experiment reveals that students can enhance their knowledge and skills in formulating LP models after practice, particularly in variables and objective function. However, other content areas were less developed, and the incorrect grouping of students with low progress influenced the results.

3.5 Discussion

Based on the analysis of students' pre-learning and post-learning achievements, most learners initially had very poor knowledge, below the threshold. However, after engaging in linear programming modeling exercises with varying difficulty levels, along with a large number of questions generated by the automated generation system, students demonstrated substantial improvement. Their post-test scores were statistically higher than their pre-test scores at the 0.05 level, reaching a fairly good level of academic achievement. The exercises were well-suited for students with moderate to low scores but may have been too easy for high-achieving students. Moreover, students showed a better understanding of writing, defining variables, and writing objective functions compared to other content areas.

4. CONCLUSION

In this research, questions were automatically generated using an ontology knowledge base combined with question templates. Expert evaluations confirmed the accuracy of the generated questions and answers, with content designed to be at a moderate difficulty level, suitable for students with medium-level knowledge. This approach could be extended to other domains to create linear programming models. When tested in an exercise system developed by our team, the approach showed improved academic outcomes, especially for learners with weak to moderate performance. Future research should focus on creating exercises that address a range of problems suitable for high-achieving learners. Additionally, developing a hybrid approach (combining templates and generative AI) could enhance question flexibility and reduce the reliance on fixed question formats.

ACKNOWLEDGMENT

This work was financially supported by Faculty of Science, Silpakorn University, grant number SRIF-JRG-2563-07.

REFERENCES

Ayanian, T. (2015). *Health human resource planning of physiotherapists and occupational therapists in Newfoundland and Labrador* [Master's thesis, University of Toronto]. University of Toronto. TSpace. http://hdl.handle.net/1807/69582

Brahmawong, C. (2013). Developmental testing of media and instructional package. *Silpakorn Educational Research Journal*, *5*(1), 7–20. [in Thai]

Chotirat, S., & Meesad, P. (2021). Part-of-Speech tagging enhancement to natural language processing for Thai Wh-question classification with deep learning. *Heliyon*, *7*(10), Article e08216. https://doi.org/10.1016/j.heliyon.2021.e08216

Chotirat, S., & Meesad, P. (2022). Automatic question and answer generation from Thai sentences. In P. Meesad, S. Sodsee, W. Jitsakul, & S. Tangwannawit (Eds.), *Proceedings of the 18th International Conference on Computing and Information Technology (IC2IT 2022)* (pp. 163–172). Springer. https://doi.org/10.1007/978-3-030-99948-3_16

Das, B., Majumder, M., Phadikar, S., & Sekh, A. A. (2021). Automatic question generation and answer assessment: A survey. *Research and Practice in Technology Enhanced Learning*, *16*, Article 5. https://doi.org/10.1186/s41039-021-00151-1

Deng, J., & Lin, Y. (2022). The benefits and challenges of ChatGPT: An overview. *Frontiers in Computing and Intelligent Systems*, *2*(2), 81–83. https://doi.org/10.54097/fcis.v2i2.4465

Fung, Y.-C., Lee, L.-K., & Chui, K. T. (2023). An automatic question generator for Chinese comprehension. *Inventions*, *8*(1), Article 31. https://doi.org/10.3390/inventions8010031

Gagné, R. (1985). *The conditions of learning and theory of instruction* (4th ed.). Holt, Rinehart and Winston.

Hake, R. R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, *66*(1), 64–74. https://doi.org/10.1119/1.18809

Jamieson, S. (2004). Likert scales: How to (ab)use them. *Medical Education*, *38*(12), 1217–1218. https://doi.org/10.1111/j.1365-2929.2004.02012.x

Kumar, V., Ramakrishnan, G., & Li, Y.-F. (2019). Putting the horse before the cart: A generator-evaluator framework for questions generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (pp. 812–821). Association for Computational Linguistics. https://doi.org/10.18653/v1/K19-1076

Kurdi, G., Leo, J., Parsia, B., Sattler, U., & Al-Emari, S. (2020). A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, *30*, 121–204. https://doi.org/10.1007/s40593-019-00186-y

Kusuma, S. F., Siahaan, D. O., & Fatichah, C. (2020). Automatic question generation in education domain based on ontology. In *Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (pp. 251–256). IEEE. https://doi.org/10.1109/CENIM51130.2020.9297991

Kusuma, S. F., Siahaan, D. O., & Fatichah, C. (2022). Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowledge-Based Systems*, *249*, Article 108906. https://doi.org/10.1016/j.knosys.2022.108906

Martinich, J. S. (1997). *Production and operations management: An applied modern approach*. John Wiley & Sons.

Maurya, K. K., & Desarkar, M. S. (2020). Learning to distract: A hierarchical multi-decoder network for automated generation of long distractors for multiple-choice questions for reading comprehension. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM'20)* (pp. 1115–1124). Association for Computing Machinery. https://doi.org/10.1145/3340531.3411997

Mulla, N., & Gharpure, P. (2023). Automatic question generation: A review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence, 12*(1), 1–32. https://doi.org/10.1007/s13748-023-00295-9

Panchal, P., Thakkar, J., Pillai, V., & Patil, S. (2021). Automatic question generation and evaluation. *Journal of University of Shanghai for Science and Technology*, *23*(5), 751–761. http://doi.org/10.51201/JUSST/21/05203

Phakmongkol, P., & Vateekul, P. (2021). Enhance text-to-text transfer transformer with generated questions for Thai question answering. *Applied Sciences, 11*(21), Article 10267. https://doi.org/10.3390/app112110267

Phatthiyaphaibun, W., Chaovavanich, K., Polpanumas, C., Suriyawongkul, A., Lowphansirikul, L., Chormai, P., Limkonchotiwat, P., Suntorntip, T., & Udomcharoenchaikit, C. (2023). PyThaiNLP: Thai natural language processing in Python. In L. Tan, D. Milajevs, G. Chauhan, J. Gwinnup, & E. Rippeth (Eds.), *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)* (pp. 25–36). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.nlposs-1.4

Pongchairerks, P. (2017). *Linear programming*. Danex Intercorporation.

Render, B., Stair, R. M., Jr., Hanna, M. E., & Hale, T. S. (2020). *Quantitative analysis for management* (13th ed.). Pearson.

Ruangchutiphophan, P., Saetia, C., Seneewong Na Ayutthaya, T., & Chalothorn, T. (2023). Thai knowledge-augmented language model adaptation (ThaiKALA). In *Proceedings of the 2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)* (pp. 1–6). IEEE. https://doi.org/10.1109/iSAI-NLP60301.2023.10355001

Sekhon, R., & Bloom, R. (2016). *Applied finite mathematics*. Lumen Learning.

Soni, S., Kumar, P., & Saha, A. (2019, March 14–15). *Automatic question generation: A systematic review* [Conference session]. International Conference on Advances in Engineering, Science, Management & Technology (ICAESMT), Dehradun, India. http://dx.doi.org/10.2139/ssrn.3403926

Steuer, T., Filighera, A., Tregel, T., & Miede, A. (2022). Educational automatic question generation improves reading comprehension in non-native speakers: A learner-centric case study. *Frontiers in Artificial Intelligence*, *5*, Article 900304. https://doi.org/10.3389/frai.2022.900304

Susanti, Y., Tokunaga, T., Nishikawa, H., & Obari, H. (2018). Automatic distractor generation for multiple-choice English vocabulary questions. *Research and Practice in Technology Enhanced Learning*, *13*, Article 15. https://doi.org/10.1186/s41039-018-0082-z

Van Campenhout, R., Brown, N., Jerome, B., Dittel, J. S., & Johnson, B. G. (2021). Toward effective courseware at scale: Investigating automatically generated questions as formative practice. In *Proceedings of the Eighth ACM Conference on Learning @ Scale (L@S '21)* (pp. 295–298). ACM. https://doi.org/10.1145/3430895.3460162

Vibulsukh, W. (1996). *A diagnosis of mathematics learning deficiencies on linear programming of Huachiew Chalermprakiet University students* [Unpublished master's thesis]. Kasetsart University. [in Thai]

Wiwatbutsiri, N., Suchato, A., Punyabukkana, P., & Tuaycharoen, N. (2022). Question generation in the Thai language using MT5. In *Proceedings of the 2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1–6). IEEE. https://doi.org/10.1109/JCSSE54890.2022.9836271

Zhu, F., Laosen, N., Laosen, K., Paripremkul, K., Nanthaamornphong, A., Ng, S.-K., & Bressan, S. (2022). A comparative empirical evaluation of neural language models for Thai question-answering. In *Proceedings of the 2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)* (pp. 120–123). IEEE. https://doi.org/10.1109/ITC-CSCC55581.2022.9894948

Zou, B., Li, P., Pan, L., & Aw, A. T. (2022). Automatic true/false question generation for educational purpose. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)* (pp. 61–70). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.bea-1.10