Advancing Automated Exam Generation: Toward Scalable and Adaptive Solutions

Dömsödi Balázs

Doctoral School of Economics and Business Informatics
(Business Informatics Doctoral Program)

Supervisor: Láng Blanka, Ph.D.

Corvinus University of Budapest

Doctoral School of Economics and Business Informatics
(Business Informatics Doctoral Program)

Advancing Automated Exam Generation: Toward Scalable and Adaptive Solutions

Doctoral Dissertation

Dömsödi Balázs

Budapest, 2026

TABLE OF CONTENTS

LIST OF TABLES

LIST OF DIAGRAMS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Blanka Láng, for her continuous guidance and support across many aspects of my academic journey, beginning with my bachelor's programme. I am also deeply thankful to László Kovács for his invaluable assistance in the research projects I participated in, particularly through his statistical expertise, and to Bíborka Fabók for her insightful contribution to the assessment of exam quality. Beyond the academic sphere, I extend my heartfelt appreciation to my fiancée, Zsóka Dombovári, for her unwavering support, to my good friend Márton Juhász for his exceptionally helpful technical advice, and to my family, whose encouragement has been a source of motivation.

1. INTRODUCTION

The manual construction of examination papers has long been recognized as a labor-intensive and cognitively demanding process. Designing assessments that are not only balanced and comprehensive but also aligned with curricular objectives requires a high level of precision and pedagogical judgment. Educators must consider several interrelated dimensions, including difficulty level, question diversity, syllabus coverage, and the cognitive complexity of tasks to ensure that the exam fairly evaluates a broad spectrum of student abilities.

This becomes even more challenging when multiple versions of the same test are needed, such as in the case of large class sizes, or efforts to mitigate academic dishonesty. Generating multiple test versions that maintain consistency in content, difficulty, and structure while also ensuring sufficient variation can quickly become an overwhelming burden for even the most experienced educators.

Given these challenges, there is a growing interest in the development of automated systems capable of generating high-quality assessments. These systems offer the potential not only to carry out the test creation process but also to introduce greater objectivity, consistency, and scalability into assessment practices. By leveraging computational algorithms, such systems can facilitate the automated selection and composition of exam questions, reducing the likelihood of human error, ensuring standardized quality across test versions, and freeing up valuable instructional time that educators can devote to more impactful teaching activities.

This dissertation is situated within this evolving landscape of educational automated assessment generation. Prior to formally commencing doctoral studies, the author joined an ongoing research project dedicated to the development of such a system. The program in question, known as Exercise Generation Algorithm+ (EGAL+), was already supported by a series of preliminary developments and publications by the author's doctoral supervisor (Láng & Kardkovács 2016; Láng 2019; Láng 2020). From the beginning of the doctoral program, the author’s primary role has been to analyze, refine, and advance this tool through rigorous

research and empirical experimentation, thereby contributing to the broader scientific effort to enhance the field of automated educational assessment generation.

The overarching objective of this dissertation is to document the theoretical foundations, research methodology, experimental development, and scholarly contributions that have emerged from the author's engagement with the EGAL+ system. The research presented herein aims to demonstrate both the academic and practical significance of the author's work within the context of current solutions in the field and to offer novel insights and innovations that push the boundaries of what is currently achievable in automated test generation.

A significant portion of the author's doctoral work involved conducting comprehensive literature reviews to map the current state of research in automated assessment systems. An important finding that emerged from this comparative analysis was that most systems prioritize either the generation of questions from raw educational content using machine learning techniques or the optimization of task selection based on predefined constraints like difficulty and curriculum coverage, particularly those employing metaheuristic optimization. Through this analysis, the EGAL+ system was classified among the latter category, focusing on the favorable arrangement and composition of pre-existing tasks to form coherent, balanced assessments.

However, the reviewed systems often lacked flexibility or adaptability and were mainly focused on a narrower range of aspects in terms of composition. In contrast, EGAL+ introduced a novel concept: full pairwise configurability of task relationships through what is termed the Preference Matrix. This matrix allows for the specification of individual preferences at a granular level for how each question should or should not be grouped with every other question. Such a feature enables an unprecedented level of configurability, theoretically allowing any desired composition logic to be defined and operationalized, irrespective of the specific type of assessment or subject matter.

This unique capability distinguishes EGAL+ from other systems in the field. However, the author has also found that this innovation also introduced new challenges. The most pressing among these was scalability. The computational complexity of processing an extensive Preference Matrix increased greatly with the size of the question bank, raising concerns about the feasibility of using EGAL+ in real-world scenarios where hundreds of exams might need to be generated in a short period. In addition, the author noted that relying on users to manually define pairwise preferences could present practical challenges for educators, highlighting the need for further refinement and automation.

Recognizing these limitations, the author undertook a systematic re-engineering of the EGAL+ program, involving the redesign and expansion of core functionalities. The aim was to preserve the innovative strengths of the system, particularly its configurability, while making it more scalable for real-world educational applications. This redesign process forms the core of the experimental research presented in this dissertation and reflects a part of the author's original contribution to the field.

The dissertation also details the incorporation of usability enhancements to simplify the user interface and data input requirements while also exploring the possibilities of integrating machine learning based solutions for not just compiling but also creating the tasks for the educators. The efforts documented in this thesis resulted in multiple international publications, through which the author has contributed to shaping ongoing scholarly discourse in the domain of automated assessment systems.

In summary, this dissertation seeks to:

- Outline the theoretical and methodological foundations underpinning the research described.
- Position the author's work within the broader context of current research on automated assessment generation.

- Present the innovations introduced by the author to the EGAL+ system and explain their scientific significance.
- Discuss the methodological challenges encountered and the solutions devised.
- Highlight the practical implications of this work for future development in the field.

Ultimately, this research aims to bridge the gap between theoretical innovation and practical application in automated assessment systems. By making efforts to transform EGAL+ from a promising experimental prototype into a more robust and scalable tool, while also identifying possible future development directions for similar tools, the research described in this dissertation aspires to make a meaningful contribution to both the academic and practical dimensions of educational technology.

2. DESCRIPTION OF THE RESEARCH QUESTIONS AND OVERVIEW OF THE RESEARCH METHODOLOGY

2.1. Research Questions

This section outlines the central research questions guiding this dissertation. Situated within the broader domain of automated assessment generation systems, this study seeks to investigate the capabilities, limitations, and potential of such technologies in addressing the evolving needs of contemporary educational assessment. While EGAL+ serves as the primary implementation through which these issues are examined, the focus extends beyond this specific system to consider the wider implications of automated assessment generation for educational practice, policy, and future research. The research questions were thus formulated in response to both theoretical concerns and the practical challenges faced by modern educational assessment environments.

1. **What structural and functional limitations exist in current automated exam generation systems?**

This question aims to critically examine the current landscape of automated exam generation technologies by identifying their inherent structural and functional shortcomings. It seeks to uncover systemic constraints that restrict the effectiveness of these systems in diverse educational contexts. By synthesizing insights from existing approaches, this analysis defines the problem space.

2. **How can the trade-off between deep pedagogical parameterization and scalability in EGAL+ be systematically addressed?**

This research question focuses on a central design tension within EGAL+: the balance between rich, fine-grained pedagogical control and the practical need for scalability and efficiency. It investigates how increased complexity in modeling parameters impacts system performance and usability. The question further explores methodological and architectural strategies developed during the research to reconcile this trade-off, aiming to demonstrate how EGAL+ can achieve both pedagogical sophistication and operational viability.

3. **How does EGAL+ perform in real-world educational contexts compared to manual exam compilation in terms of quantitative assessment quality metrics and operational efficiency?**

This question evaluates the practical effectiveness of EGAL+ by benchmarking its performance against traditional, manually constructed exams. It emphasizes empirical comparison using measurable indicators, such as assessment quality, time efficiency, and consistency. The goal is to determine whether EGAL+ can not only match but potentially exceed human performance in exam generation tasks within authentic educational settings. This analysis provides critical evidence regarding the system’s readiness for adoption and its potential impact on educational practice.

4. **Which future research and development directions logically follow for EGAL+, and what do they imply for the field at large?**

This final question is forward-looking, aiming to establish a trajectory for future research and development based on the current findings. It considers both the current limitations of EGAL+ and the broader opportunities for extending its framework. The goal is to articulate a vision for how this work may serve as a foundation for continued innovation in automated assessment and contribute to the evolving landscape of automated educational systems.

2.2. Research Methodology

To address the above research questions, a multi-methodological approach was adopted that integrated literature-based theoretical inquiry, algorithmic analysis, software engineering, and empirical evaluation in real-world educational settings. Each methodological component was selected to align with the specific demands of the research questions and to ensure the robustness of the findings.

The first phase of the research involved a comprehensive and systematic literature review aimed at mapping the current state of automated assessment generation systems and automated educational systems in broader terms. This review included both primary research articles and literature review articles, allowing for a nuanced understanding of the field's methodological foundations, technological trends, and existing gaps. By synthesizing these insights, the author identified where EGAL+ aligns with, diverges from, or extends existing approaches. This domain mapping served as the baseline for subsequent comparative evaluations and the identification of research gaps.

The literature review was particularly focused on categorizing assessment systems according to their methodological core, namely, those that emphasize task generation through machine learning versus those that focus on test composition through algorithmic optimization. EGAL+ was situated within the latter category, prompting a deeper investigation into comparable systems and the unique attributes

of the EGAL+ architecture, including its Preference Matrix and use of the Harmony Search Algorithm (HSA).

Following the literature-based analysis, the methodological focus shifted to the internal evaluation and redesign of EGAL+. A central element of this redesign was a deep methodological analysis of the Harmony Search Algorithm itself, which forms the core optimization technique used in EGAL+ for composing exams based on user-defined preference values. The theoretical properties, advantages, and limitations of the HSA were explored in depth to understand how its behavior influenced the performance, scalability, and flexibility of EGAL+.

After this algorithmic exploration, a comprehensive software engineering audit of the original EGAL+ implementation was conducted. The system was restructured to improve modularity, maintainability, and most of all efficiency. The redesigned version was then subjected to a benchmark comparison with the original version, evaluating improvements in computational performance.

To determine the practical applicability of EGAL+, the final phase involved empirical testing in a real educational setting. The redesigned system was deployed by a university instructor to generate actual exams for live classrooms. The analysis drew on both quantitative and qualitative data. Quantitative data were collected on metrics such as time saved in exam generation, consistency across test versions, and adherence to assessment criteria. Qualitative data were obtained through interviews and surveys with the teacher and students, focusing on perceived usability, effectiveness, and trust in the system. This empirical analysis provided crucial insights into the real-world strengths and limitations of EGAL+, contributing directly to answering the third and fourth research questions.

The final methodological step was the synthesis of findings from the theoretical, technical, and empirical components of the research. This synthesis enabled the identification of specific areas where EGAL+ holds promise for future development and broader application. From the challenges discovered actionable insights emerged. The research methodology flow is visualized in **Figure 1**.

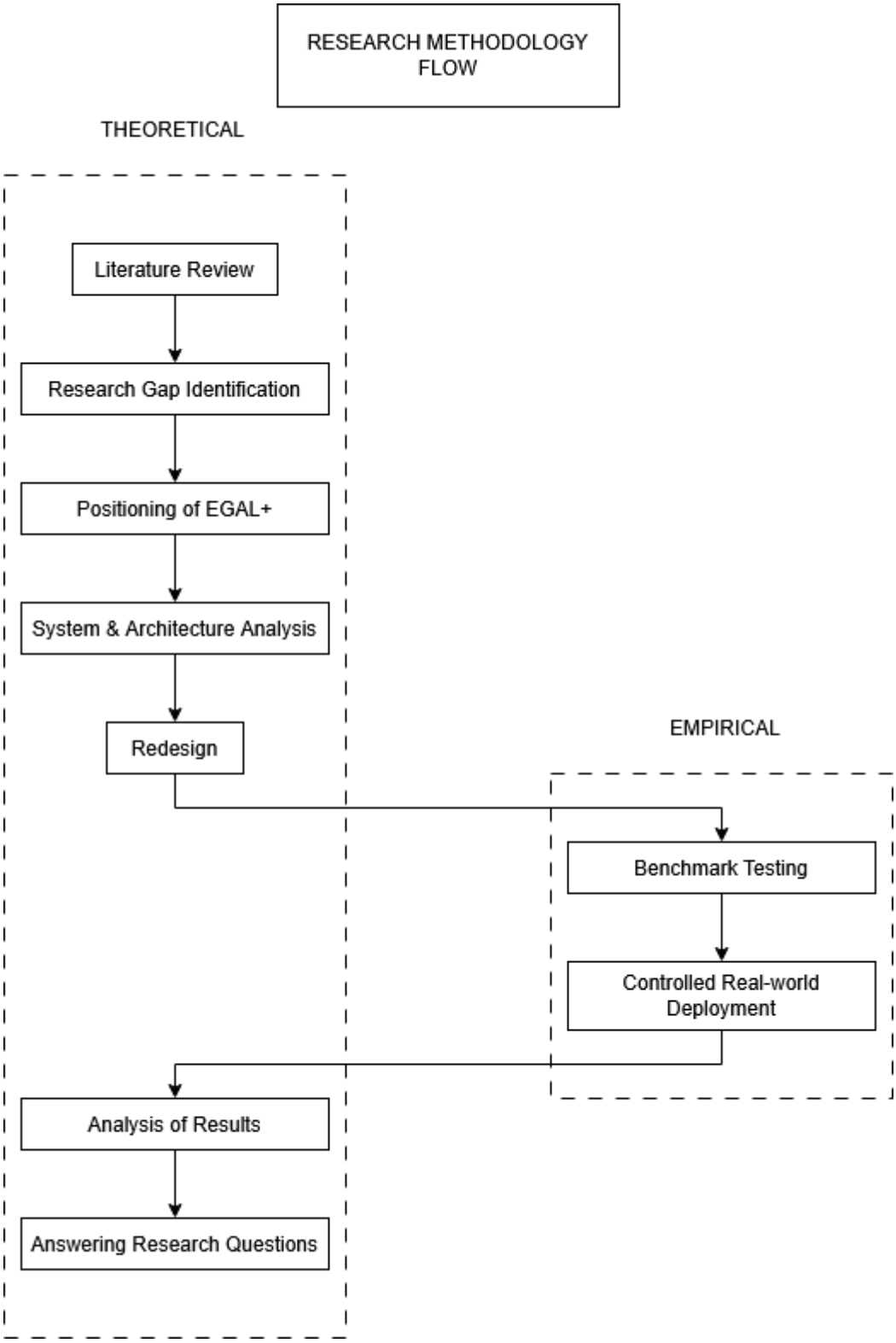

Fig. 1. Research Methodology Flow; Source: Author

3. LITERATURE REVIEW OF AUTOMATION IN EDUCATION WITH AN EMPHASIS ON COMPUTATIONAL ASSESSMENT GENERATION

3.1. Overview

Over the past fifteen years, the field of educational assessment has experienced a transformative shift, largely fueled by the increasing reliance on algorithmic methods to generate test and exam content. This growing interest stems from a widespread recognition of the limitations inherent in manual exam construction. Traditional approaches to assessment creation are often time-consuming, require significant cognitive effort from instructors, and are prone to human error and subjectivity. These limitations can result in inconsistencies in exam quality, bias in question selection, and inefficient use of educator time. These factors can negatively affect the fairness, validity, and reliability of assessments.

In contrast, automated question set generation systems offer a promising alternative. By using computational algorithms to design assessments, educators and institutions can produce high-quality, diverse, and pedagogically sound test materials in a fraction of the time. These automated approaches not only reduce the burden on instructors but also help ensure that assessments are more consistently aligned with curricular standards and targeted learning objectives. This alignment enhances both the effectiveness of assessment and the overall learning experience for students.

This chapter presents a detailed and critical examination of the technological advances and methodological frameworks in algorithmic question set generation, with a focus on developments from the past decade and a half. The discussion evaluates the core algorithmic strategies employed and analyzes how different approaches handle key challenges such as balancing question difficulty, ensuring topic coverage, and generating diverse and non-redundant question sets.

The chapter adopts a systematic approach to the selection and analysis of relevant literature in the domain of automated assessment design. The review

focuses on contributions published over the past 15 years and is based on a corpus of more than 40 publications identified through structured searches conducted using Google Scholar. The search strategy combined keywords related to automated assessment, question generation, metaheuristic optimization, and machine learning, ensuring broad yet targeted coverage of the field. Inclusion criteria emphasized methodological relevance, empirical rigor, and citation frequency to capture both influential and technically substantive works, while publications lacking a clear methodological contribution or falling outside the defined scope were excluded.

The selected studies are organized chronologically to highlight the evolution of the field and are further grouped into thematic categories, including approaches based on metaheuristic optimization algorithms, approaches based on machine learning, and complementary research addressing broader educational challenges. This structured methodology enhances the transparency, reproducibility, and comprehensiveness of the review.

Table 1 contains a breakdown of how many publications were reviewed in each category, showing an emphasis on the broader field to ensure the analysis is not narrowed down to already similar solutions to EGAL+.

Topic	Number of Publications Reviewed
Approaches Based on Metaheuristic Optimization Algorithms	10
Approaches Based on Machine Learning	12
Complementary Research on Broader Educational Challenges	20

Table 1. Breakdown of how many publications were reviewed in each category; Source: Author

The first subchapter of this review focuses on optimization-based methods, particularly those that utilize metaheuristic algorithms. These techniques have proven especially useful in addressing the complex, multi-dimensional nature of assessment design, where competing objectives must be simultaneously satisfied.

Common objectives include maintaining content quality, often informed by educational taxonomies, especially Bloom's taxonomy (Bloom et al., 1956); balancing difficulty, so that exams are appropriately challenging for diverse student populations; ensuring diversity among different test versions to minimize cheating and enhance fairness; maximizing curriculum coverage to ensure that key learning outcomes are addressed and adhering to institutional or instructor-specified constraints, such as question formats, thematic requirements, or length limitations.

Among the most commonly used optimization techniques are metaheuristic optimization algorithms, especially genetic algorithms (GAs), which mimic evolutionary processes through mechanisms such as selection, crossover, and mutation. GAs are particularly well-suited to solving combinatorial problems like test generation, where there are many potential configurations and numerous constraints to consider. However, they are not the only algorithms employed. Researchers have also explored simulated annealing, ant colony optimization, particle swarm optimization, hybrid approaches that combine multiple strategies to improve performance or adaptability, and many other forms of heuristic solutions, including the Harmony Search Algorithm as well.

Another major research direction in the field of automatic test generation involves the emergence of large language models (LLMs) such as GPT-based systems. These models have demonstrated impressive capabilities in generating natural language text, including questions, explanations, and even instructional content. As such, they are increasingly being considered as foundational technologies for the next generation of educational tools.

The literature now includes a growing number of studies exploring the potential of LLMs in automating not just test selection, but the generation of entirely new test items, tailored explanations, or feedback. These models can help bridge the gap between static question banks and adaptive learning environments, creating more interactive and personalized assessment experiences.

However, LLMs, while powerful, are not without limitations. They may generate factually incorrect or pedagogically inappropriate content if not properly guided or reviewed. Their output often lacks transparency, and there remains a need for rigorous validation and quality control mechanisms to ensure that generated content meets educational standards. As such, effective deployment of LLMs in exam generation requires a deep understanding of both their capabilities and constraints, along with robust evaluation frameworks.

Another significant component of this chapter is dedicated to exploring additional automation approaches within the broader context of education. These methods, while not always directly related to exercise or question generation, contribute valuable insights into understanding the diverse needs of educational stakeholders, including institutions, teachers, and students. This section includes a range of computational solutions that address systemic challenges such as scheduling, resource management, and student retention, thereby offering a more comprehensive perspective on how automation can support and enhance educational processes. Furthermore, it enumerates exercise generation approaches that do not fall strictly into the categories of metaheuristics or machine learning, but which nonetheless play a relevant role in the evolving landscape of educational technology. A number of more general studies and reviews are also discussed, shedding light on the requirements, opportunities, and implications of integrating automation within various educational settings.

In the final section of this chapter, the most important aspects of the EGAL+ approach are compared to the reviewed systems on a high level.

3.2. Approaches Based on Metaheuristic Optimization Algorithms

Among the earliest significant contributions of the past fifteen years, Teo et al. (2012) proposed a system for automated exam generation using a multi-constraint genetic algorithm, designed specifically to align with Outcome-Based Education (OBE) requirements. Their work addresses the difficulty many lecturers (especially novice instructors) face when creating comprehensive and high-quality

exam papers, highlighting the limitations of manual question preparation and maintenance. The system, called Auto-Generator Of Examination Questions (AGEQ), focuses on reducing the discrepancy between intended learning outcomes and the content of the generated exam paper. By incorporating targets from Bloom's taxonomy, the system ensures exam questions reflect specified educational objectives, particularly those mandated by Malaysia's Ministry of Higher Education. Although only the cognitive domain is considered, the tool provides a structured method for generating exams that validate conformity to OBE standards, offering support for instructors tasked with delivering outcome-aligned assessments.

Rahim et al. (2017) contributed an automated exam question generator aimed at easing the workload of educators by streamlining the creation of multiple-choice exam questions, with the goal of also supporting any other type of assessment tasks in the future. Their system leverages a genetic algorithm to construct exam sets that align with the six cognitive levels defined in Bloom's Taxonomy, thereby supporting assessments that cater to a range of student learning capabilities from basic recall to more complex cognitive tasks. The prototype was tested using a database of 500 sample questions across two undergraduate computing courses, and experiments involved generating exams with varying chapter selections. Results demonstrated that the generator could produce exam papers with weightages that closely reflected the target cognitive level distribution, with output quality depending on the availability of sufficient questions per Bloom level in the item bank. Although the tool focused specifically on multiple-choice formats, the authors noted its potential scalability to other question types and its utility in quizzes and tests. The study emphasized the burden that manual exam preparation places on instructors, particularly under time constraints, and positioned automation as a practical solution for improving both the efficiency and pedagogical integrity of assessment design.

A year later, Zhou et al. (2018) proposed a novel solution to challenges facing traditional exam creation systems in Chinese education by introducing a

Hybrid Genetic Algorithm (HGA) for automated test generation. Their research was motivated by the limitations of existing item bank systems, which often produced exams that were inefficient, subjective, and lacked adaptability to educational goals. To address these issues, the authors first developed a mathematical model for exam construction based on specific attributes and constraints relevant to high-quality assessments. They then enhanced the standard genetic algorithm by integrating elements of simulated annealing, thereby mitigating common issues like premature convergence and improving the system's ability to search the solution space more effectively. This hybrid approach enabled the generation of exam papers that were both more balanced and better aligned with instructional objectives. The authors verified the model's effectiveness and suggested that it could serve as a foundation for advancing intelligent, scalable, and efficient testing systems within the evolving landscape of digital education in China.

Further innovation came from Ciguené et al. (2019), who tackled the growing need for scalable and individualized assessments, particularly in large or diverse classroom settings, by developing a genetic algorithm-based system that prioritized both fairness and structural differentiation between test versions. Recognizing the challenges of maintaining academic integrity and test equity as class sizes grow, their work introduced a structural metric specifically designed to quantify the distance between different assessments. This metric served as a fitness function within a genetic algorithm framework, guiding the system to generate tests that were structurally distinct from one. By optimizing structural differentiation rather than just content variation, the system aimed to reduce opportunities for answer-sharing among students, all while preserving the validity and fairness of the test content. The researchers demonstrated the feasibility of this approach through preliminary experiments, highlighting its potential to enhance academic assessment by ensuring pedagogical equivalence without sacrificing uniqueness.

Also in 2019, Shanthi et al. advanced the field by designing an automated system for generating multiple-choice exam questions using a genetic algorithm

framework explicitly structured around Bloom's Taxonomy. Their work sought to ease the considerable burden educators face when manually designing assessments, especially under institutional guidelines that demand quality, fairness, and cognitive diversity in exam content. The system was designed to ensure that generated exams included questions spanning all six of Bloom's cognitive levels, from basic tasks to higher-order evaluation, thereby enabling comprehensive assessment of student learning. By incorporating chapter selection features and avoiding the repetition of questions across academic years, the tool provided a level of customization that enhanced its practical relevance for instructors. The fitness function guiding the algorithm was calibrated to enforce proportional representation across Bloom's levels and maintain topic relevance, ensuring that test sets were pedagogically balanced and tailored to varying student abilities. As a web-based, fully automated platform, the system not only improved the speed and efficiency of exam creation but also upheld educational standards by delivering exams that were cognitively diverse and aligned with established instructional goals.

Alam et al. (2020) present a comprehensive review of genetic algorithms (GAs), emphasizing their foundational mechanisms, broad applications, and growing significance within engineering pedagogy. Rooted in evolutionary theory and bio-inspired computation, the article describes GAs as robust, metaheuristic optimization techniques that iteratively evolve candidate solutions through operators such as selection, crossover, and mutation. The authors highlight the algorithm's resilience, adaptability, and capacity to handle complex, ill-structured problems, particularly in educational contexts that demand dynamic modeling and problem-solving strategies. Within engineering education, GAs have been applied to domains like timetable scheduling, control systems, and mathematical modeling, offering students hands-on exposure to optimization techniques in simulated or real-world scenarios. By reviewing both theoretical constructs and implementation methodologies, the study demonstrates the pedagogical value of GAs in fostering analytical thinking and computational experimentation.

Rahim et al. (2020) extended their earlier work by developing a more advanced, intelligent exam question set generator. Their proposed system, known as Automated Exam Question Set Generator (AEQSG), combines Utility-Based Agents (UBA), Learning Agents (LA), Genetic Algorithms, and Bloom's Taxonomy to improve both the efficiency and quality of exam generation. The UBA module can be used to select actions that maximize expected utility at each state and complementing this, the LA component introduces adaptivity by analyzing the outcomes of past exam sets, using methods such as bell curve analysis, and using historical data to enhance future question selections. Bloom's Taxonomy is integrated to automate the cognitive complexity distribution of questions across various levels, ensuring that generated exams align with educational objectives and institutional standards. The genetic algorithm plays a key role in optimizing the selection and arrangement of questions based on the user-defined preferences and learned data patterns. This comprehensive and responsive approach represents a significant step toward fully intelligent, data-informed assessment systems capable of generating high-quality, pedagogically sound exam sets with minimal human intervention.

Wu et al. (2020) introduced an especially innovative method by combining Deep Knowledge Tracing (DKT), genetic algorithms, and dynamic programming. Unlike traditional methods that rely on manually labeled questions, often inefficient and biased, their system used DKT to predict student group performance based on past learning data. This prediction informed the construction of exams that adaptively matched the expected skill levels and score distributions of the target cohort. Two models were introduced: one using dynamic programming and the other using a genetic algorithm, both optimizing performance alignment. By integrating student learning data directly into the generation process, the system could produce high-quality, balanced assessments without manual calibration. Their method marked a significant advancement in personalized assessment design, particularly for large or diverse classrooms where differentiated testing is essential.

Wang et al. (2022) contributed a robust test generation system using a hybrid genetic algorithm designed to meet multi-objective constraints. Their approach integrated advanced evolutionary strategies (such as $\mu + \lambda$ selection, type-preserving subsection crossover, and tabu search-based mutation) as they enhanced the algorithm's ability to avoid premature convergence. The result was a novel and efficient test generation system capable of producing high-quality, constraint-compliant assessments at scale.

Concluding this subchapter, Popescu et al. (2023) developed a system particularly suited to large-scale question banks, where manual selection becomes impractical. Their method enabled users to specify both the number of questions and the topical categories of interest, after which the algorithm applied a category-aware fitness function to ensure contextual alignment. Designed for scalability and speed, the system demonstrated its effectiveness through performance tests, showing how genetic algorithm-based filtering can streamline test generation in complex educational environments where manual selection is no longer feasible.

3.3. Approaches Based on Machine Learning

In 2017 Pilán, Volodina, and Borin developed a comprehensive NLP-based framework for selecting candidate sentences from corpora for use in second language learning exercises, with a particular emphasis on pedagogical suitability. The system, called HitEx, integrates hybrid use of heuristic rules and machine learning to assess sentence quality based on multiple criteria, including complexity and context independence, two factors often overlooked in prior automatic exercise generation efforts. Leveraging Swedish corpora through the Korp infrastructure and lexical resources such as SALDO, the system allows for customizable selection based on user-defined linguistic patterns and difficulty levels. Integration into the online learning platform Lärka made the system accessible for both educators and learners, supporting self-directed learning and exercise generation. An empirical evaluation with teachers and students confirmed that the selected sentences generally met expectations in terms of complexity and independence, and the

difficulty of the resulting exercises aligned well with expectations. Although sample sizes were modest, findings suggest strong potential for improving both human-authored and automated educational content through data-driven sentence selection.

Building on previous foundation, Zanetti et al. (2021) presented an automated method for generating second language learning exercises that uses a combination of machine learning methods and heuristic rules by leveraging parallel corpora derived from movie subtitles, aiming to create authentic, pedagogically useful content. Their approach focuses on reconstructing sentences using bilingual sentence pairs, which offers a dual benefit: the learner engages with syntactic structures in the target language while also relying on the accompanying translation for disambiguation. By prioritizing sentence alignment precision over recall and filtering out ambiguous or overly short examples, the system seeks to ensure linguistic clarity and learner relevance. Notably, the authors addressed the difficulty of providing automated feedback for open-ended exercises by using the parallel translation to constrain acceptable responses. Evaluation across language pairs Italian-English, English-Swedish, and Swedish-Italian revealed promising alignment accuracy, suggesting the method's potential scalability. However, the predominance of short, simple sentences highlighted the need to adapt the system for more advanced learners in future iterations.

In the same year, another line of research by Mehta and Smetannikov (2021) explores a novel approach to generating fill-in-the-blank exercises for English language learning by framing the task as a sequence labeling problem. Departing from earlier corpus-based methods that primarily rely on collocations or frequency statistics, the authors employ Natural Language Processing (NLP) techniques to identify suitable blank positions within sentences, focusing specifically on verb conjugation. Their method involves classifying each token in a sentence as either a potential blank or not, allowing for a more context-sensitive and pedagogically targeted exercise generation process. While their current work limits scope to verbs, the underlying framework is adaptable and could be extended to cover broader

grammatical structures. The authors position this contribution as a foundation for further research in automated grammar exercise generation, offering both a technical baseline and a practical tool to support scalable, individualized English language instruction.

Meanwhile, parallel efforts were underway by Pandraju and Mahalingam (2021) who addressed a critical gap in the field of Automated Question Generation (AQG) research by proposing a model capable of generating questions from both textual and tabular data sources. While most AQG systems traditionally overlook structured formats like tables, thereby missing key information in domains such as policy documentation or financial reporting, the authors emphasized the necessity of integrating these data types for comprehensive question generation. To support this goal, they adapted machine learning based methodologies for answer-aware question generation. Experimental results demonstrated the model's capability to generate high-quality questions, although these results were based on automated evaluation, and the authors acknowledged the need for human validation in future work. Their approach represents a significant step forward in the development of AQG systems that can effectively utilize all informational formats within source documents, thereby enhancing the flexibility and completeness of automated assessment tools.

Next year Rao et al. (2022) introduced a supervised machine learning approach to exam generation, focusing on aligning test content with individual skill profiles. Their system employed classifiers such as Naive Bayes, Random Forest, and Decision Trees to predict areas of strength. Based on these predictions, the algorithm selected questions from a question bank that best matched each subject’s known skills, allowing for a more personalized and relevant assessment. This tailored approach not only improved evaluation accuracy but also provided educators with clearer insights into specific competencies. A user-friendly graphical interface complemented the system, offering visual feedback and enabling both subjects and evaluators to better understand skill alignment.

Attention also turned toward domain-specific applications as Chung, Hsiao, and Lin (2022) attempted to develop a solution for creating effective programming practice questions by developing an Artificial Intelligence (AI)-assisted Programming Question Generation (PQG) model. Recognizing that instructors often struggle to manually produce sufficient and pedagogically diverse questions, the authors proposed a novel approach that combines Local Knowledge Graphs (LKG) with Abstract Syntax Trees (AST) to construct semantic networks of programming knowledge. This hybrid method enables the automated generation of contextually relevant and syntactically accurate programming questions. The system analyzes the structural and conceptual elements of code to create new assessment items aligned with learners' cognitive progression. Evaluations by a group of experienced instructors revealed high satisfaction with the quality, relevance, and utility of the generated questions, indicating the model's strong potential to enhance programming education. By automating question creation, the proposed approach reduces instructor workload while enriching learning resources, marking a significant step forward in intelligent educational support systems.

With the rise of generative AI, 2023 saw an increasing focus on the integration of language models into real-world educational settings.

Alnefaie et al. (2023) addressed the absence of a comprehensive Quranic question-and-answer (Q&A) dataset by exploring the use of AQG tools to construct such a corpus. Aiming to support the future development of Islamic QA systems, the authors treated several AQG web services, namely ExploreAI, Cathoven, Questgen, and Lumos Learning, as black-box tools and systematically evaluated their performance in generating questions from English Quran translations. The study highlights that while all tools were capable of producing large volumes of Q&A pairs, their output quality varied across metrics such as syntactic and semantic correctness, ambiguity, and answerability. Among the tools, Cathoven emerged as the most effective, delivering superior results in most evaluative dimensions. Using these generators, the researchers compiled a dataset of 40,585 Q&A pairs derived from the Sarwar and Yusuf Ali translations of the Quran. Their findings emphasize

the potential of AQG technology to facilitate scalable dataset creation, while also acknowledging the importance of carefully assessing tool performance.

Bachiri and Mouncif (2023) proposed a comprehensive AI-driven system to automate the generation of multilingual multiple-choice questions for MOOCs, addressing the ongoing challenge of producing high-quality, scalable assessments on platforms like Open edX and Moodle. Their solution integrates machine learning and Natural Language Processing techniques across a modular pipeline. The system is capable of managing complex input texts and producing questions in multiple languages, with quality control mechanisms in place through post-processing and retranslation. Evaluations, including comparisons between human- and machine-generated questions, demonstrated that the AI-produced items performed comparably in terms of assessing learner competency and question quality. While the system currently focuses on vocabulary-based multiple-choice questions (MCQs), the authors emphasized its extensibility to other question types and noted the potential for personalizing difficulty and integrating learner performance data. Their research positions the tool as a practical advancement in pedagogical engineering, offering scalable, data-informed assessment design in digital learning environments.

Continuing to explore the pedagogical value of generative AI tools in real learning contexts Al-Obaydi, Pikhart, and Klimova (2023) explored the intersection between emerging AI technologies and foundational learning theories by examining ChatGPT's potential as a language learning tool. Through a qualitative study involving university students tasked with writing assignments using ChatGPT, the authors assessed whether interactions with the chatbot aligned with accepted definitions of learning and contributed meaningfully to language development. While ChatGPT demonstrated some support for learner engagement and motivation, the study found that it lacked critical components traditionally associated with effective pedagogy, such as sustained teacher-student interaction. Moreover, the system appeared to emphasize content knowledge over linguistic practice, offering limited support for active language production unless paired with

other teaching strategies. The research raised important concerns around ethical use, academic integrity, and over-reliance on generative tools, especially in writing tasks where superficial engagement could hinder deeper learning. Despite these limitations, the study acknowledged ChatGPT's value as a supplementary tool in foreign language education, particularly when guided by educators and integrated thoughtfully into a broader pedagogical framework.

Still in 2023 Tran et al. explored the use of LLMs, specifically GPT-3 and GPT-4, to generate isomorphic MCQs tailored for computing courses, aiming to alleviate the burden of manual question design faced by instructors. Acknowledging the limitations of reused or template-based MCQs, the study assessed whether modern LLMs could create high-quality questions from a given question stem by generating a correct answer and plausible distractors. Through a series of studies using question stems from the Canterbury Question Bank and an introductory C programming course, the authors found that GPT-4 significantly outperformed GPT-3, correctly generating answers 78.5% of the time compared to 30–36% accuracy by GPT-3. Notably, the study introduced a structured prompt engineering approach to improve output quality and examined the impact of question discrimination scores on model performance, though no significant correlation was observed. The findings highlight GPT-4's potential as a classroom tool for drafting quiz content in real time, with implications for improving assessment personalization. While the models are not yet fully reliable, the work demonstrates meaningful progress toward integrating LLMs into assessment workflows and sets the stage for future studies comparing LLM-generated and instructor-authored MCQs in live educational settings.

Further building on the rise of generative AI, Olney (2023) investigated the capabilities of LLMs in generating MCQs by directly comparing fine-tuned models to human-authored textbook questions. In a controlled evaluation using aligned content from a college science textbook, the study assessed outputs from an augmented Macaw model and Bing Chat against human benchmarks across seven criteria. Results showed that both LLMs matched human performance on six out of

seven metrics. Interestingly, the two models exhibited distinct failure modes: Macaw often repeated answer choices, while Bing Chat occasionally failed to include the correct answer in the options. After excluding flawed items, Macaw's output quality aligned closely with human-authored questions, whereas Bing Chat's performance improved but remained below the human standard. The study highlighted the surprising maturity of LLMs in MCQ generation and identified directions for improving their reliability. Despite limitations related to topic specificity and task simplification, the findings highlight the growing viability of LLMs as tools for scalable and high-quality question generation in educational settings.

3.4. Complementary Research on Broader Educational Challenges

While the primary focus of this thesis is on the automatic generation of questions and exercises, it is important to recognize that numerous computational approaches have been developed to address other critical challenges within educational environments. Although these methods may not directly involve assessment content generation, they tackle related problems such as scheduling, timetabling, student retention, and resource allocation, which are important issues that reflect the broader needs of institutions, educators, and learners. These approaches often employ sophisticated algorithms and data-driven techniques that reveal valuable insights into the operational and pedagogical aspects of education. By examining such solutions, we gain a deeper understanding of the educational ecosystem, and the state-of-the-art methods used to optimize its various components. This broader perspective can inform the design of exercise generation systems by aligning them more closely with real-world institutional constraints and user needs.

Additionally, this section also enumerates some related literature reviews with the goal of better understanding the needs and implications of automation in education, and any notable exercise generation approaches that do not fall under the categories of heavily metaheuristic or machine learning based approaches.

As another example of advanced exercise generation tools, Farida et al. (2011) presented the ODALA+ approach as an advancement of their earlier Ontology Driven Auto-evaluation Learning Approach (ODALA), aiming to enhance the personalization of learning activities through adaptive exercise generation and learner profile modeling. Building on previous research in ontological teaching domain modeling and automated learner evaluation, ODALA+ incorporates additional stages that use evaluation outcomes to construct learner profiles and dynamically generate suitable exercises. The system, integrated into the WebSIELA platform, operates by executing conditional generation scenarios composed of reusable generation and elimination primitives tailored to each learner's evolving profile. Evaluation of the prototype involved ten university instructors conducting multiple sessions and reviewing exercise appropriateness, yielding a 91.3% satisfaction rate for exercise generation and 70% for evaluation processes, leading to an overall satisfaction score of approximately 80.65%. Notably, ODALA+ adopts a "glass box" evaluation paradigm, making the assessment logic transparent and traceable, an improvement over earlier "black box" models. The study underscores the system's potential in delivering individualized learning paths.

Deriving from exercise generation, Al-Betar, Khader, and Zaman (2012) address the challenge of university course timetabling by introducing a hybrid Harmony Search Algorithm, a memetic computing approach designed to navigate the complexity of large combinatorial optimization problems. Recognizing the limitations of classical methods in exploring vast search spaces with numerous local optima, the authors leverage the strengths of population-based global search strategies while incorporating local refinement techniques. Their proposed hybrid enhances the traditional Harmony Search Algorithm by integrating a hill climbing mechanism for improved local exploitation and adopting a global-best strategy from particle swarm optimization to accelerate convergence. Tested across eleven benchmark datasets of varying sizes, the Harmony Search Algorithm achieved optimal solutions for all small instances and delivered competitive, often superior, results on medium and large-scale datasets. By balancing exploration and

exploitation within a flexible metaheuristic framework, the study demonstrates the potential of hybrid algorithms to solve the university course timetabling problem more effectively. This work contributes not only to the optimization literature but also offers practical implications for academic institutions seeking efficient and high-quality scheduling solutions.

As educational optimization systems matured, interest also grew in how to represent and respond to learner characteristics. Chrysafiadi and Virvou (2013) conducted a comprehensive literature review on student modeling techniques developed over the prior decade, aiming to answer three central questions: what aspects of the student should be modeled, how these characteristics are modeled, and how the resulting model can be applied to enable adaptivity in educational software. The review spans various systems including intelligent tutoring systems (ITS), e-learning platforms, educational games, and mobile learning environments, emphasizing the growing demand for personalization in computer-based learning. The authors identified that the most commonly modeled student attributes include knowledge levels, learning preferences and styles, misconceptions, cognitive and affective states, and meta-cognitive abilities. Overlay models emerged as the dominant technique for representing students' knowledge mastery, while perturbation models were used to detect misconceptions. Stereotyping was frequently employed to capture learning styles, and a significant rise was noted in affective modeling using Bayesian networks and fuzzy logic to address uncertainty. Furthermore, the review highlighted a growing trend toward hybrid and ontology-based models, allowing more abstract, reusable, and comprehensive representations of learners. The findings provide valuable guidance for developers and researchers designing adaptive educational systems, underscoring the importance of flexible, multi-dimensional, and data-driven approaches to student modeling.

In the same year, the rapid expansion of Massive Open Online Courses (MOOCs) prompted deeper analysis of learner engagement and Adamopoulos (2013) investigated the issue of student retention, one of the most persistent challenges facing MOOCs, by conducting an interdisciplinary analysis that

integrates econometric, text mining, opinion mining, and machine learning methods. Using a large dataset of user-generated course reviews, the study identifies and quantifies the impact of various factors, including course content, instructor quality, platform design, and institutional reputation, on learners' likelihood of completing online courses. Notably, instructor effectiveness emerged as the strongest predictor of course retention, while elements such as peer assessment and the promise of certification also played positive roles. The research adopts the Grounded Theory Method (GTM) in a quantitative context, allowing for exploration of causal relationships from student feedback. By synthesizing diverse analytical approaches, the study not only advances theoretical understanding of MOOC dynamics but also provides actionable insights for course designers, educational platforms, and policymakers seeking to enhance the effectiveness and sustainability of online learning environments.

Getting back to automated generation of high-quality assessments, Almeida et al. (2013a) tackled the challenge of generating mathematics exercises in the context of Portugal's shift toward Bologna-aligned higher education, which emphasized student autonomy and reduced contact hours. To support this transformation, they introduced PASSAROLA, a sophisticated yet accessible exercise generation system designed to create randomized math problems alongside intelligent feedback mechanisms. Unlike systems limited to simple answer formats, PASSAROLA is capable of handling more complex types through LaTeX-based templates and integration with tools like Maxima and Perl, enabling deep mathematical expression, evaluation, and error detection. The authors underscored the importance of designing "well-posed" exercises that not only ensure controlled difficulty but also anticipate common student errors. A notable feature of their approach lies in the system's capacity to differentiate mistakes by producing feedback specific to each type of error, thereby enhancing the pedagogical value of automated assessment. Their work highlights that effective exercise generation goes beyond randomization, requiring careful selection of problem parameters and

leveraging external computational tools to support both expressiveness and precision in student evaluation.

In a complementary effort to their earlier work, Almeida et al. (2013b) introduced a more in-depth presentation of PASSAROLA, describing its capability of supporting diverse academic disciplines and complex answer formats in more detail. The system was designed to aid lecturers in creating rich, structured exercises without requiring a background in computer science. PASSAROLA distinguishes itself from traditional tools by supporting heterogeneous objects, such as graphs, music scores, and source code in expressing exercises. Exercises created with PASSAROLA can include not only questions and answers but also step-by-step solutions, explanations, and automated assessment processes. The system's templating capabilities and use of reusable components allow for scalable exercise creation while facilitating user contributions, making it suitable for both local and web-based environments. By enabling control over complexity, supporting error-specific feedback, and simplifying integration of domain-specific knowledge, the system advances the goal of individualized and self-regulated learning. Ultimately, the authors positioned PASSAROLA as a flexible, open-source alternative to proprietary assessment systems, capable of adapting to a broad spectrum of educational contexts.

In the following year Malafeev (2014) presents a method for the automatic generation of open cloze exercises modeled on the widely recognized Cambridge English exams, targeting the needs of English language learners and educators in computer-assisted language learning environments. The system, called Exercise Maker, allows users to input arbitrary English texts, such as news articles or reviews, to generate grammar- and vocabulary-focused cloze exercises, thereby enabling more flexible and contextually relevant teaching materials. Notably, the method supports difficulty adjustment to align with learners' proficiency levels, addressing a key limitation in static textbook resources. Through three evaluation experiments, the system was shown to produce exercises nearly indistinguishable from those found in authentic Cambridge tests, with experienced instructors

reporting high realism and utility. Malafeev emphasizes the pedagogical value of targeting grammatical and lexicogrammatical constructs, which are often problematic for learners from structurally different first languages. The study demonstrates how automated tools can bridge the gap between individualized instruction and scalable content creation, enhancing learner engagement while supporting the rigorous demands of standardized test preparation.

As adaptive learning improved in the general context of education, Sukstrienwong (2017) introduced a genetic algorithm-based approach for forming heterogeneous student groups by balancing learning styles and academic attributes, with the aim of enhancing equity and collaboration in cooperative learning environments. Recognizing the complexity and time-consuming nature of manually constructing such balanced groups, the study leverages Index of Learning Styles (ILS) alongside students' academic data, such as their Grade Point Average and prior coursework, to create groupings that are both diverse and pedagogically sound. The proposed system, implemented as a web-based application called Genetic Algorithm for Forming Student Groups (GAFSG), encodes students' profiles as multidimensional vectors and applies a genetic algorithm to minimize intergroup disparities using a fitness function based on Euclidean distance. A case study conducted in an undergraduate computer science course demonstrated that algorithmically generated groups outperformed self-selected ones in terms of balance and composition, offering a more equitable foundation for group-based assignments. This research highlights the potential of evolutionary algorithms to assist educators in managing increasingly complex classroom dynamics and fostering effective peer learning.

And just one year later of Sukstrienwong's student group forming solution, a notable study was presented by Hnida, Khalidi Idrissi, and Bennani (2018) which proposed a novel method for addressing the problem of automatic sequencing of instructional units in virtual learning environments by introducing an adapted Harmony Search Algorithm designed to automatically compose such elements tailored to individual learners. Recognizing the limitations of "one-size-fits-all"

content delivery, particularly the risk of cognitive overload and learner disengagement, the authors aimed to dynamically generate personalized learning sequences that align both with student proficiency and the structural coherence of subject matter. Their algorithm models the sequencing task as a constrained optimization problem, where each solution vector represents a possible instructional path, optimized through iterative improvisation inspired by musical harmony creation. Two core constraints guided the sequencing logic: ensuring a progression from simpler to more complex concepts and maintaining conceptual coherence based on pre-assessed learner knowledge. Experimental results suggested the system could generate viable, individualized courseware in real time without significant performance loss, even as complexity scaled. The study highlights the potential of metaheuristic approaches like HSA in advancing adaptive e-learning technologies and envisions future integration within platforms such as Moodle, to further assess pedagogical alignment from both student and instructor perspectives.

Meanwhile, evolutionary algorithms continued to prove effective in managing institutional constraints, particularly in scheduling tasks as Chen et al. (2021) developed an improved genetic algorithm to address the course scheduling in universities, where growing enrollment numbers and limited resources have made manual scheduling both inefficient and unsustainable. Traditional scheduling methods, while employing various optimization algorithms like hill climbing or simulated annealing, often fall short in handling the complexity and dynamic nature of institutional needs. In response, the authors implemented a genetic algorithm-based system to automate the scheduling process, considering both hard constraints (e.g., room and teacher conflicts) and soft constraints (e.g., teacher or student preferences). The system allows for manual overrides, ensuring adaptability in real-world academic environments. Experimental results showed that the proposed algorithm significantly enhanced the speed and accuracy of schedule generation, offering a scalable and cost-effective solution to reduce the workload of administrative staff. Their findings highlight the potential of genetic approaches in

satisfying the increasing demand for personalized and efficient course scheduling in higher education.

Still in the context of institutional logistics, Ngo et al. (2021) addressed the complex challenge of examination timetabling by proposing a multi-objective optimization model that balances logistical, institutional, and student-centric constraints using a GA. Recognizing exam scheduling as an NP-hard problem with increasing complexity in large-scale academic environments, the authors developed an approach capable of managing over 2,500 students while optimizing room utilization, exam spacing, and resource efficiency. The model incorporates a mix of hard constraints (such as room availability and seating capacity) and soft objectives (such as minimizing student stress through reasonable exam spacing). By combining greedy initialization with genetic search, the system ensures feasible initial solutions and guides optimization towards ideal outcomes without requiring exact preferences from decision-makers. Although challenges remain, particularly in calibrating the relative importance of competing objectives, the study presents a robust and scalable solution for institutional exam scheduling. The authors suggest that future improvements could focus on parallel computing and refined scaling strategies to enhance both algorithmic performance and decision-maker usability.

Further improving on educational content generation Eryiğit et al. (2021) explored the challenges of teaching morphologically rich languages and proposed a gamified solution to facilitate the learning of complex grammatical structures, using Turkish as a case study. Recognizing that grammar in such languages is often embedded within individual words, posing difficulties for both learners and educators in generating meaningful exercises, the authors developed a mobile application to automatically generate morphology-focused tasks. Their gamified learning environment gradually introduced morphological components and their interactions, aiming to enhance both engagement and comprehension. Over a three-week study with international students enrolled in an introductory Turkish course, the researchers collected data through questionnaires, e-journals, and interviews. Results demonstrated strong student approval, with participants reporting high

levels of perceived efficacy, enjoyment, engagement, and overall satisfaction with the learning experience. The study suggests that the integration of gamification with automated morphological exercise generation not only supports more effective learning but also fills a gap in the pedagogical resources available for teaching complex grammar in morphologically rich languages.

Staying in the field of exercise generation, Basse et al. (2021) introduced an ontology-based system designed to automate the generation of Structured Query Language (SQL) exercises, addressing the significant time burden typically associated with creating and grading practice questions in database education. Recognizing that SQL learners benefit from extensive hands-on practice to internalize syntax and resolve semantic errors, the authors proposed a tool that allows educators to generate questions by simply inputting a database schema, from which the system produces a variety of exercises sorted by difficulty level. This not only streamlines instructional design but also enables learners to interact with the system independently, receiving targeted feedback that supports iterative learning. The underlying ontology helps ensure that the queries generated are both pedagogically relevant and structurally coherent. By shifting the role of the instructor from manual content creation to schema definition, the system facilitates scalable and personalized SQL practice while promoting deeper learner engagement with database concepts.

As a higher-level overview of such solutions, Mulla and Gharpure (2023) provide a thorough review of AQG methodologies, situating them within a broad spectrum ranging from educational to conversational systems. Their survey categorizes AQG methodologies into three principal domains: standalone question generation, visual question generation, and conversational question generation. The paper offers a detailed classification of existing approaches while also mapping the landscape of publicly available datasets suited to each. Importantly, the authors highlight the growing complexity of AQG tasks, particularly as models expand beyond textual input to incorporate multimodal data.

As educational AI continued to evolve and with the public release of ChatGPT in late 2022, 2023 brought an increased emphasis on machine learning based educational technologies and their integration into diverse learning modalities.

Abu Khurma, Ali, and Hashem (2023) offer a critical, experience-based reflection on the adoption of ChatGPT in the United Arab Emirates education system. Framed around their firsthand engagement as educators at the Emirates College for Advanced Education, the authors chart their evolving perceptions from initial skepticism to active exploration of ChatGPT's role in supporting student learning. The paper emphasizes both the potential and risks of integrating AI tools into classrooms, particularly highlighting applications in research support, writing development, and exam preparation. Central to their analysis is the argument that equitable access and responsible use must be prioritized to avoid exacerbating existing disparities in educational outcomes. The authors propose practical safeguards, including diversified assessment formats, teacher training, and AI-content detection tools, while underscoring the importance of policy frameworks to guide ethical implementation.

In the same year, Ngo (2023) conducted an investigation into university students' perceptions of ChatGPT, exploring its perceived benefits, limitations, and potential role in academic contexts. Through surveys and interviews with over 200 participants, the study found a generally favorable view of ChatGPT as a learning tool, particularly for its accessibility, multilingual input capability, and utility in saving time and enhancing writing clarity. Students acknowledged its value in offering personalized feedback and serving as a supplementary tutor, echoing emerging literature on AI-assisted learning. However, concerns were also raised regarding the reliability of generated information, improper citation practices, and the system's limitations in handling idiomatic language and complex subject matter. The study underscored the risk of over-reliance on AI tools and the potential for academic misconduct, suggesting mitigation strategies such as integrating usage guidelines, promoting verification of outputs, and reinforcing academic integrity.

Still in 2023 Heilala, Shibani, and Gomes de Freitas examined how emerging technologies, particularly AI and LLMs, can reshape pedagogy. The study advocates for a future-oriented, integrative educational model that combines lifelong learning principles with real-time, adaptive systems, aiming to prepare students for complex, technology-driven environments. By incorporating AI into curriculum design and emphasizing collaborative, problem-based learning, the authors propose a framework that cultivates essential 21st-century skills, including critical thinking, adaptability, and innovation. Their discussion also highlights the growing relevance of concepts like pantagogy, suggesting a pedagogical approach that evolves alongside learner needs and technological change. While their analysis points to the immense potential of AI-enhanced education, it also acknowledges challenges like ethical implementation, promoting overdependence, or even the possibility of replacing human roles. Ultimately, the study highlights the need for collaboration and sustainable educational practices that align with the dynamic nature of industry and society.

Further investigating this topic, Kasneci et al. (2023) offer a balanced and forward-looking analysis of the opportunities and limitations presented by large language models (LLMs) like ChatGPT in educational contexts. Framing these tools as both transformative and potentially problematic, the authors examine the dual perspectives of students and educators, illustrating how LLMs can enhance learning through content generation, personalized feedback, adaptive assessments, and inclusive pedagogical support. At the same time, they caution that successful integration demands new competencies, particularly in digital literacy, critical thinking, and ethical reasoning. The paper highlights the utility of LLMs across educational levels, from elementary writing assistance to advanced research support at the university level, as well as their potential in professional training and remote collaboration. However, challenges such as algorithmic bias, interpretability, and risk of misuse are emphasized as crucial concerns. Kasneci et al. argue that educators must proactively establish pedagogical frameworks and oversight mechanisms that preserve academic integrity while leveraging the

adaptive strengths of LLMs. Their commentary closes with a call for responsible, context-aware use, stressing that LLMs, when used with care and transparency, can serve as powerful tools to democratize learning and prepare students for the complex digital landscape ahead.

And concluding the year's wave of early reflections included in this section of the literature review, Lo (2023) conducted a review to assess the early educational implications of ChatGPT during the first three months following its release. Analyzing 50 academic articles, the study offers a comprehensive view of ChatGPT's potential, performance across disciplines, and emerging concerns. The findings reveal that ChatGPT excels in certain domains such as economics, performs adequately in programming, but struggles in areas like mathematics, raising questions about its domain-specific reliability. Despite these inconsistencies, ChatGPT shows strong potential as both an instructional assistant and as a virtual tutor that can support flipped classrooms and student collaboration. However, the review also underscores serious challenges. ChatGPT can generate misleading or inaccurate information, especially in specialized or time-sensitive contexts due to limitations in its training data. A major concern is its facilitation of academic dishonesty, particularly through its ability to bypass traditional plagiarism detectors, potentially undermining fair assessment. Lo emphasizes that these risks necessitate immediate institutional action, including redesigning assessments (e.g., incorporating oral or multimedia tasks) and revising academic integrity policies. Importantly, the review calls for robust instructor training and student education to ensure ethical, critical, and effective use of generative AI. Lo concludes that while ChatGPT offers transformative opportunities in education, its responsible integration requires systemic policy updates, pedagogical innovation, and a clear focus on maintaining academic standards.

3.5. High Level Comparison of EGAL+ with the Reviewed Systems

The above literature review has shown that in contemporary educational environments, particularly within large-scale and digital learning contexts, the

demand for advanced systems capable of automating question and exam generation has grown substantially. Traditional methods of manually preparing, selecting, and organizing exam content have become increasingly impractical and burdensome, especially for educators managing diverse classrooms and ever-expanding curricula. These manual approaches often suffer from inconsistencies, limited scalability, and a lack of alignment between the intended learning outcomes.

To address the growing demands of producing sufficient volumes of high-quality, pedagogically sound, and cognitively diverse questions, a task that not only consumes significant time but also requires deep expertise in instructional design, ensuring fairness and content balance, the vision for an ideal automated question generation and exam compilation system must extend well beyond simple randomization or rule-based selection. Such a system should represent a comprehensive, intelligent framework capable of supporting a wide range of assessment tasks, from multiple-choice questions to more complex, open-ended exercises, with the capacity to evolve in support of new assessment formats in the future. It should allow for the generation of exercises that span the full spectrum of cognitive skills, from basic recall to higher-order analytical reasoning, while also ensuring that each assessment aligns with the specific instructional goals and learner expectations, and ideally providing automated personalized student feedback as well.

The expectations and goals seen in the literature review describe a system, that must be driven by a robust optimization engine, likely built upon a metaheuristic framework that avoids premature convergence and is capable of effectively searching the solution space for the most suitable combinations of questions. At its core, the system should incorporate a well-designed fitness function that evaluates candidate exams based on multiple objectives, including topic coverage, proportional representation of chapters, and controlled difficulty levels. This function should be adaptable to user-defined preferences as well as learned data patterns, enabling the system to intelligently select questions. By doing so, the system would produce high-quality, balanced assessments without the need

for manual calibration, while ensuring that each exam is pedagogically aligned and structurally valid.

Scalability is a critical expectation as well according to the reviewed literature. Instructors should be able to generate individualized assessments for large student groups with minimal effort, leveraging user-friendly interfaces that allow for intuitive control over parameters such as topic selection, number of questions, and difficulty levels, which not only improves the relevance and accuracy of the assessments but also reduces opportunities for answer-sharing, preserving test integrity and academic honesty. Consequently, it would be beneficial if such a system was integrated seamlessly with widespread online learning platforms.

The generation of correct answers and plausible distractors also seems to be a common demand, as well as the ability to pull content from diverse data sources, which further expands the versatility of such a system, making it adaptable to various instructional formats and academic disciplines.

Ultimately, based on the reviewed literature, an ideal automated question and exam generator system should bridge the gap between assessments in individualized tutoring and large-scale educational content deployment.

While the comprehensive literature review outlines the ambitions and multifaceted demands placed upon next-generation automated assessment systems, EGAL+ is positioned as a focused response to a specific, yet critical subset of those challenges. It is not designed to be the all-encompassing, ideal system that autonomously handles the full pipeline from question generation to personalized feedback. Rather, the importance of identifying a well-defined research gap within this expansive field and delivering a meaningful contribution was recognized when designing EGAL+, and its goals were defined accordingly.

The design of EGAL+ is grounded in the observation that, once a sufficient pool of questions or tasks exists, the quality of an assessment is exclusively

determined by how well these elements are selected and compiled. Are all relevant topics represented? Is there adequate variation in question types and cognitive depth? Are the exams equal in difficulty, ensuring fairness across instances? In response to these concerns, EGAL+ introduces a fine-grained preference-based compilation mechanism through the use of a Preference Matrix, a structure capable of encoding pairwise preferences between all questions within the question bank and additionally including individually assignable difficulty values for each element as well.

This framework enables instructors to articulate nuanced compilation goals, such as preferring one question over another in any given setup, even when both relate to the same concept or chapter, or when balancing different cognitive levels across an exam, or based on any other consideration. This approach supports a level of precision in exam compilation that is capable of expressing virtually any concept the instructor may seek to express.

Additionally, to its capability of fine-tuned compilation, EGAL+'s other foundational principle is to maintain strict difficulty equivalence across all generated exams within a cohort. This is operationalized by ensuring that the total difficulty score, assigned manually or algorithmically to each question, remains consistent across different versions of the exam. Such an approach guarantees structural fairness, particularly in environments where multiple variants of an assessment are needed, for example to mitigate academic dishonesty.

To tackle these challenges, a multi-objective fitness function is designed to guide the compilation process of EGAL+. Specifically, the system strives to maximize both the overall adherence to the expressed preferences in the Preference Matrix and the diversity between exam instances within the same cohort, ensuring that each version of the assessment is meaningfully distinct. This dual-objective optimization is carried out while strictly enforcing equal total difficulty across all exams, a hard constraint that is never compromised.

To maintain both precision and scalability, EGAL+ is built upon the Harmony Search Algorithm, a metaheuristic optimization method inspired by the improvisation process of musical harmony. This algorithm is commonly used in the reviewed educational automation solutions, and particularly well-suited for navigating large, complex solution spaces efficiently, making it possible to compile high-quality exams from question banks containing even thousands of entries without compromising performance or precision. EGAL+ thus aspires to offer a solution that is not only computationally robust but also highly parametrizable, allowing users to finely control the structure and content of generated exams.

Looking toward the future, EGAL+ is designed with extensibility in mind. Planned features include mechanisms for evolving question difficulty ratings based on empirical student performance, allowing the system to adapt and improve over time by refining its internal values of question challenge levels. Furthermore, seamless integration with widespread Learning Management Systems (LMS) such as Moodle and the development of a user-friendly interface based on stakeholder feedback are envisioned as essential steps in bringing EGAL+ from research to practice. However, these are aspirational directions; the immediate focus of this thesis is centered on the computational methods that make scalable, fine-tuned exam compilation possible.

It is important to emphasize what EGAL+ does not aim to solve. The generation of questions from raw instructional content, often approached through generative AI and large language models, falls outside the core scope of EGAL+. While there exists a technical bridge for integration via external Application Programming Interface (API) calls to such models, and while these tools can offer valuable support for question authoring, the literature review underscored that human oversight remains indispensable in the current state of the art. Given the ongoing need for expert validation of AI-generated content and no foreseeable sign of the possibility of omitting it, and the already saturated focus on this domain in current research, this study of EGAL+ deliberately directed its contributions to the

assessment compilation problem, where the field remains comparatively underexplored and in need of scalable, high-precision solutions.

In summary, EGAL+ does not attempt to replicate or replace the full functionality of an idealized end-to-end automated assessment system. Instead, it offers a targeted, high-performance solution to the pressing problem of high-quality exam composition, a problem that lies at the heart of fair and scalable educational evaluation.

The following chapter provides a detailed description of the Harmony Search Algorithm, the core methodology behind EGAL+. Subsequent sections of the thesis will delve into the detailed implementation of the methods that build up this solution.

4. HARMONY SEARCH ALGORITHM: THE CORE METHODOLOGY OF EGAL+ FOR EXAM COMPILATION

4.1. Defining Metaheuristic Algorithms

The Harmony Search Algorithm belongs to the broader class of metaheuristic algorithms, which necessitates an understanding of what metaheuristics fundamentally are before delving into the specific characteristics and mechanics of the Harmony Search Algorithm itself.

The field of optimization has always been a central area of research in computational mathematics and computer science. A primary focus within this domain has been the development of algorithms designed to identify optimal or near-optimal solutions to a wide range of optimization problems. Traditionally, these algorithms have relied heavily on numerical techniques rooted in gradient-based optimum search methods. Such classical methods typically involve the use of iterative improvement that seeks a candidate solution by following the direction of the steepest descent or some variant thereof. These gradient-based methods often

operate within a localized neighborhood of an initial starting point, thereby incrementally refining the solution.

While these simpler optimization strategies have proven effective in relatively less complex models, particularly where the objective function is smooth, convex, and exhibits a unique global optimum, they show considerable limitations when applied to real-world optimization scenarios. In practice, many real-life problems are characterized by nonlinearity, discontinuity, multimodality, and high-dimensional search spaces, which render traditional methods inadequate or inefficient. One of the primary drawbacks of gradient-based techniques is their inherent sensitivity to initial conditions; when the objective function possesses multiple local optima, the solution obtained may be heavily dependent on the choice of the starting point. As a result, these methods often converge to a local rather than a global optimum.

Moreover, the reliance on information for choosing the best starting point in the process of optimum search can pose significant challenges when the objective function is noisy or exhibits discontinuities or multiple peaks. In such cases, gradient-based methods may become unstable or computationally expensive, and in some instances, completely infeasible. Furthermore, more traditional optimization methods often demand substantial computational resources, especially in high-dimensional spaces, due to the need for repeated evaluations and storage of derivative information. These computational bottlenecks have prompted researchers to explore alternative optimization paradigms that do not depend on such information and are more robust in handling complex, multimodal problems.

One such class of alternative approaches that has garnered significant attention is metaheuristic algorithms. Metaheuristics are high-level, problem-independent algorithmic frameworks that employ a combination of deterministic rules and stochastic processes to explore the search space and identify optimal solutions. A defining characteristic of metaheuristics is their inspiration from natural processes. These include biological evolution, social and behavioral

patterns, and principles derived from physical sciences. By mimicking these processes, metaheuristic algorithms are able to escape local optima and perform a more global search of the solution space, often leading to better performance on complex optimization problems.

The concept of metaheuristics lacks a clear, universally agreed-upon definition, and its interpretation continues to be a topic of discussion. One way to define the term might be:

> A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms. The term is also used to refer to a problem-specific implementation of a heuristic optimization algorithm according to the guidelines expressed in such a framework. (Sörensen & Glover, 2016, p. 4)

Over the years, numerous metaheuristic algorithms have been proposed and refined to address the computational challenges posed by traditional methods. The robustness and flexibility of metaheuristic-based approaches have made them particularly attractive for solving complex engineering and scientific optimization problems where classical methods often fail. In numerous applications, metaheuristic algorithms have successfully addressed issues such as multimodality and the need for global search capabilities, thereby overcoming many of the deficiencies inherent in conventional numerical optimization techniques.

Despite their successes, however, no single metaheuristic algorithm universally outperforms all others across all problem domains. Consequently, ongoing research continues to explore the development of new, more powerful metaheuristic algorithms. These efforts frequently involve drawing analogies from a broader spectrum of natural and artificial systems. The ultimate goal is to design algorithms that are not only capable of efficiently navigating complex search spaces but also adaptable to the dynamic and uncertain nature of real-world optimization problems.

4.2. Some Notable Metaheuristic Algorithms

To reinforce the points discussed in the previous subchapter and to provide a deeper understanding of how metaheuristic algorithms draw inspiration from natural processes, this section presents a series of illustrative examples. These examples highlight some of the most influential, foundational, and widely adopted metaheuristic algorithms. Each algorithm is briefly explained, with particular attention paid to the natural phenomena or biological behaviors they simulate, in order to demonstrate how these analogies contribute to the effectiveness and versatility of such methods in solving complex optimization problems.

Evolutionary programming, introduced by Fogel et al. in 1966, represents metaheuristic algorithm designed to evolve solutions for complex problems, particularly those involving prediction. In its original formulation, evolutionary programming focused on evolving finite state machines, which operate through state transition tables, which define how the system moves from one state to another based on input symbols from a specific alphabet. The evolutionary process in this context involves applying uniform random mutations to the elements. These mutations serve as the primary means of variation, introducing changes in how the finite state machines behave in response to different inputs. Notably, the algorithm relies heavily on two core evolutionary operators: mutation and selection. The selection itself in evolutionary programming is carried out using a stochastic tournament mechanism. In this process, a subset of individuals is chosen at random, and the best-performing ones, based on a predefined fitness criterion, are selected to propagate their traits to the next generation. This probabilistic approach allows for a balance between exploration of new solutions and exploitation of existing high-performing individuals.

The Genetic Algorithm (GA), initially described by Holland (1975), and subsequently refined and expanded by numerous researchers, is a metaheuristic algorithm inspired by the principles of natural selection and evolutionary biology. Drawing from the Darwinian concept of survival of the fittest, GAs simulate the

process of natural evolution to identify optimal or near-optimal solutions to complex problems. Fundamentally, the GA operates through the iterative application of three primary genetic operators: selection (often referred to as reproduction), crossover, and mutation. The selection phase embodies the survival of the fittest principle by preferentially choosing individuals with higher fitness to pass their genetic information to the next generation. Crossover, or recombination, involves the exchange of genetic material between pairs of parent solutions to produce offspring, thereby introducing new genetic combinations into the population. Mutation, on the other hand, introduces stochastic variability by randomly altering individual bits within a solution, thus ensuring genetic diversity and preventing premature convergence to local optima. One of the most distinctive features of GAs, setting them apart from other metaheuristic techniques is their population-based approach. Rather than exploring the solution space one point at a time, GAs evaluate multiple candidate solutions simultaneously in each generation. This parallelism enables a broader and more diversified exploration of the search space, enhancing the algorithm's ability to escape local optima and increasing the likelihood of identifying globally optimal solutions.

Due to these advantageous characteristics, Genetic Algorithms have been extensively applied in various domains of optimization, and as discussed in the Literature Review chapter of this thesis, GAs have gained particular popularity in addressing optimization challenges within the educational domain as well.

Tabu search, a metaheuristic optimization technique proposed by Glover (1977), can be broadly characterized as an enhanced form of gradient-descent search that incorporates adaptive memory structures. Unlike traditional local search methods, which may become trapped in local optima, tabu search employs a memory-based strategy to navigate more effectively through the solution space. This memory is implemented through a structure known as the tabu list, which records a finite history of previously visited solutions or moves, as well as certain other configurations deemed undesirable or counterproductive to revisit. The effectiveness of tabu search is significantly influenced by the definition of a state

within the problem context, the neighborhood structure surrounding that state, and the length or capacity of the tabu list. These are key design parameters that must be carefully calibrated depending on the nature and complexity of the optimization problem. In addition to the core memory-based restrictions, other supplementary mechanisms, such as aspiration criteria and diversification strategies are often incorporated to enhance performance and flexibility. Aspiration criteria serve as an exception to the tabu constraints allowing the algorithm to override its tabu list and allow a move when all available new states are already in the tabu list. This ensures that high-quality solutions are not excluded merely because they may become inaccessible. Diversification, on the other hand, introduces a controlled element of randomness into the search process. While tabu search is fundamentally deterministic, diversification helps to mitigate premature convergence by periodically redirecting the search toward unexplored regions of the solution space. If progress stalls, indicating that the search is converging without yielding improved solutions, a diversification strategy may trigger a reset or redirection based on random or heuristic-driven criteria.

Simulated annealing, suggested by Kirkpatrick et al. (1983), extends the principles of the Monte Carlo method to enable the exploration of equilibrium and non-equilibrium states in complex n-body systems. This approach draws inspiration from the physical process of annealing observed in materials science, particularly in the way that liquids solidify or metals undergo recrystallization during controlled cooling. In a typical annealing procedure, a material is initially heated to a high temperature, resulting in a highly disordered state. It is then cooled gradually, allowing the system to remain close to thermodynamic equilibrium throughout the process. As the temperature steadily decreases, the system incrementally transitions into more ordered configurations, ideally converging toward minimal energy. This controlled descent toward lower energy can be conceptualized as an adiabatic pathway to the system's most stable configuration. However, if the starting temperature is insufficiently high or if the cooling rate is excessively rapid, the system may fail to reach this optimal state, and stabilization may happen in states

that are energetically suboptimal but locally minimal, effectively trapping the system in a non-global minimum of the energy landscape.

Ant Colony Optimization (ACO) is a biologically inspired metaheuristic developed by Dorigo et al. (1991), motivated by the decentralized problem-solving capabilities of real ant colonies. Despite having minimal individual intelligence, ants collectively exhibit highly organized behaviors, such as discovering the shortest paths between their nest and food sources. This emergent coordination arises from indirect communication via pheromone trails, a mechanism whereby ants deposit a chemical substance along their paths, which in turn influences the routing decisions of other ants. Over time, frequently used paths accumulate more pheromone, making them increasingly attractive and reinforcing their selection through a positive feedback loop. ACO translates this natural behavior into a computational framework for solving complex combinatorial optimization problems. In the algorithm, multiple artificial agents, modeled as ants, build solutions by traversing a graph that represents the problem space (such as a network of cities in the Traveling Salesman Problem). Each agent constructs a solution incrementally, guided by a combination of artificial pheromone trails (reflecting past collective experience) and heuristic information (such as proximity or cost). As in real ant colonies, the most successful solutions influence future search behavior by receiving higher pheromone reinforcement. To avoid premature convergence on suboptimal paths, ACO incorporates mechanisms such as pheromone evaporation, which gradually reduces the influence of older trails, and tabu restrictions that prevent agents from revisiting recently chosen elements within a single solution. Although the individual agents operate with only local information and simple rules, their collective behavior produces a powerful and adaptive search process. The algorithm's strength lies in its capacity to balance exploration of new regions in the solution space with the exploitation of high-quality solutions discovered so far. This makes ACO particularly effective for problems that are difficult to solve using traditional deterministic methods.

Having outlined the foundations and underlying mechanisms of metaheuristic optimization, and illustrated through key examples, the next subchapter builds upon this groundwork by providing a detailed description of the Harmony Search Algorithm.

4.3. Comprehensive Description of the Harmony Search Algorithm

The Harmony Search Algorithm represents a sophisticated metaheuristic optimization technique that was first conceptualized and formally introduced by Geem et al. in 2001. As detailed previously in this section, in the broader field of metaheuristics, many algorithmic frameworks are inspired by processes rooted in natural phenomena, and in this tradition, the development of the Harmony Search Algorithm marks a departure from exclusively natural inspirations, instead leveraging an artificial yet deeply human process: musical harmony. As a novel conceptual foundation for optimization, music, particularly the pursuit of harmony in musical performance, serves as a compelling metaphor and model for optimization. Harmony, in its musical sense, is the artful combination of sounds that are aesthetically pleasing to the human ear. The process of achieving such harmony, especially in improvisational settings like jazz ensembles, involves the careful and iterative selection of musical notes by different instruments, aiming to achieve a sonically satisfying collective output. This pursuit closely mirrors the iterative refinement process inherent in many heuristic optimization techniques, wherein candidate solutions are continuously evaluated and improved to reach an optimal or near-optimal outcome.

In this context, iteratively selecting a pitch from the allowable tonal range of musical instruments in an ensemble, forms what can be described as a collective "harmony vector." If the resultant harmony is perceived as musically successful, the experience is not lost. Instead, it is retained within the cognitive framework of each musician, effectively enriching their internal repertoire and increasing the likelihood that a similarly harmonious combination will emerge in future improvisations. This dynamic illustrates a form of experiential learning and

adaptive memory utilization within a creative system. A parallel process occurs in the domain of computational optimization where each decision variable can be viewed as analogous to a musician, initially selecting a value from within its defined bounds or permissible range. Together, the values assigned to each variable construct a complete solution vector, representing a candidate solution to the optimization problem at hand. If the aggregation of these variable values results in a favorable outcome, judged according to the objective function, the algorithm records or reinforces this experience. Much like a musician remembering a successful note choice, the algorithm retains this valuable solution pattern in memory, thereby increasing the probability of generating similarly high-quality solutions in subsequent iterations (Lee & Geem, 2005).

The analogy between the processes of engineering optimization and musical improvisation is visualized in **Figure 2.**:

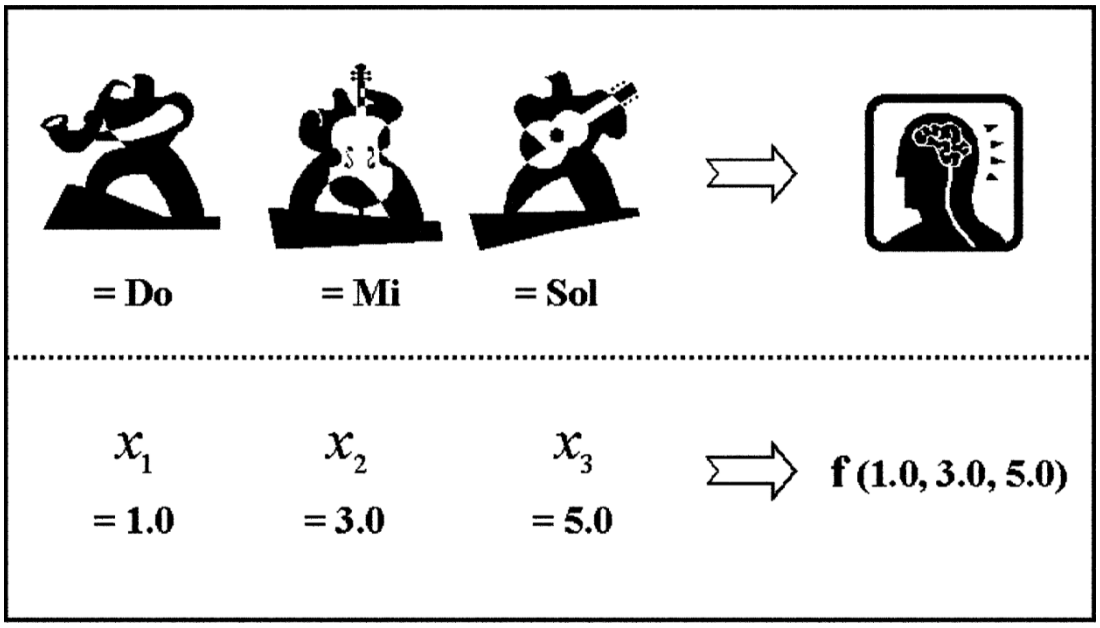

Fig. 2. Parallel between engineering optimization and musical improvisation; Source: Lee & Geem (2005)

As described by Geem et al. (2001), from a theoretical standpoint, musical harmony has been the subject of intellectual inquiry since antiquity. The ancient

Greek mathematician and philosopher Pythagoras (circa 582-497 BC) was among the earliest to investigate the relationships between sound frequencies and perceived consonance, laying the groundwork for the mathematical treatment of harmony. Later, in the 18th century, French theorist Jean-Philippe Rameau (1683-1764) formalized many foundational aspects of classical harmony theory. In more recent history, scholars such as musicologist James Tirro have meticulously chronicled the evolution of musical traditions like American jazz, further illustrating the depth and richness of musical harmony as a cultural and analytical subject. The analogy between musical performance and optimization is not merely superficial. In both domains, the aim is to identify the best possible outcome: in music, this is the attainment of an aesthetically pleasing harmony; in optimization, it is the identification of a global optimum, characterized by minimal cost, maximal efficiency, or some other performance criterion as determined by an objective function. In music, aesthetic judgment emerges from the simultaneous interplay of notes and rhythms produced by individual instruments, analogous to how the evaluation of an objective function arises from the interaction of multiple decision variables. Just as musicians refine their performance through repeated practice and real-time adaptation, optimization algorithms iteratively refine candidate solutions in pursuit of improved performance metrics.

According to the detailed description of the Harmony Search Algorithm by Lee & Geem (2005), the procedure begins with an essential initialization phase where both the optimization problem and the algorithm's governing parameters are defined. This step sets the foundation for the entire process by specifying the decision variables, the size and structure of the search space, and the key algorithmic controls.

The abbreviation $\boldsymbol{N}$ is used to define the total **Number of Decision Variables**, and $\boldsymbol{X_i}$ to denote the **Decision Variables** themselves. The **Harmony Memory** (**HM**) is a matrix-like structure that stores a set of potential **Solution Vectors** ($\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3} \ldots \boldsymbol{X_n}$), each vector representing a candidate solution discovered throughout the optimization process. The number of **Solution Vectors**

that **HM** can hold is defined by the **Harmony Memory Size (HMS)**. Additionally, the probabilistic parameters **Harmony Memory Consideration Rate (HMCR)** and **Pitch Adjusting Rate (PAR)** are established as well, which respectively control how often new candidate solutions are drawn from existing memory and how often random mutation is applied on individual **Decision Variables**. The **Maximum Number of Iterations** (or improvisations) is often denoted as $\mathbf{N_i}$, which determines how long the algorithm runs, and serves as a primary stopping criterion, but other termination criteria can also be introduced to further tune the optimization process.

Following the setting of the above parameters, the algorithm generates an initial **Harmony Memory** filled with **Solution Vectors** comprising randomly selected values for each **Decision Variable**, drawn from their permissible ranges. These initial **Solution Vectors** serve as the starting point of the search process. Each one is evaluated using a **Fitness Function**, denoted $\boldsymbol{f(x)}$, which the algorithm attempts to optimize, typically by minimizing or maximizing the function value based on the nature of the problem.

Once the initial memory has been populated, the algorithm proceeds to the improvisation phase. In this stage, a new **Solution Vector** is synthesized through a procedure designed to mimic the improvisational process in musical composition. The new **Solution Vector** is constructed by determining the value of each of its **Decision Variables** according to a set of probabilistic rules based on **HMCR** and **PAR**. First the **HMCR** specifies the likelihood that a value for a **Decision Variable** is selected from the existing **HM**. This allows the algorithm to exploit previously identified high-quality solutions by reusing their components. If the value is not selected from the memory, an event that occurs with a probability of **1-HMCR**, then the algorithm turns to random selection, introducing diversity into the solution space by generating a completely new value for that variable. Once a value is chosen, the selected value is either slightly modified with the probability of **PAR** before being added to the new **Solution Vector** or is added without modification, thereby exploring the neighborhood of the current solution for potentially better

alternatives. The interplay of these rules ensures a balance between exploration and exploitation, which is vital for effective optimization.

After the new **Solution Vector** has been generated, it undergoes evaluation using the same fitness function $\boldsymbol{f(x)}$ employed during initialization. The goal of this evaluation is to determine whether the new **Solution Vector** improves upon the solutions already stored in the **HM**. If the fitness of the new **Solution Vector** is superior to that of the worst performing currently in **HM**, then the new **Solution Vector** replaces the worst one. This memory update mechanism ensures that the overall quality of the solutions in **HM** gradually improves over time, guiding the algorithm toward increasingly optimal results. Each replacement reflects the algorithm's capacity to retain only the best solutions encountered throughout its search.

The optimization process continues iteratively, with new **Solution Vectors** being generated, evaluated, and possibly stored in **HM**, until the predefined termination conditions are satisfied. This stopping criterion may be based on reaching **Ni** number of iterations, beyond which the algorithm ceases to operate. Alternatively, the algorithm may halt if it identifies a solution that fully satisfies all constraints imposed by the problem, indicating that a sufficiently optimal or feasible solution has been found or if there has not been significant improvement for a given number of iterations.

The improvisation process of the Harmony Search Algorithm is shown in a flow chart format in **Figure 3.**:

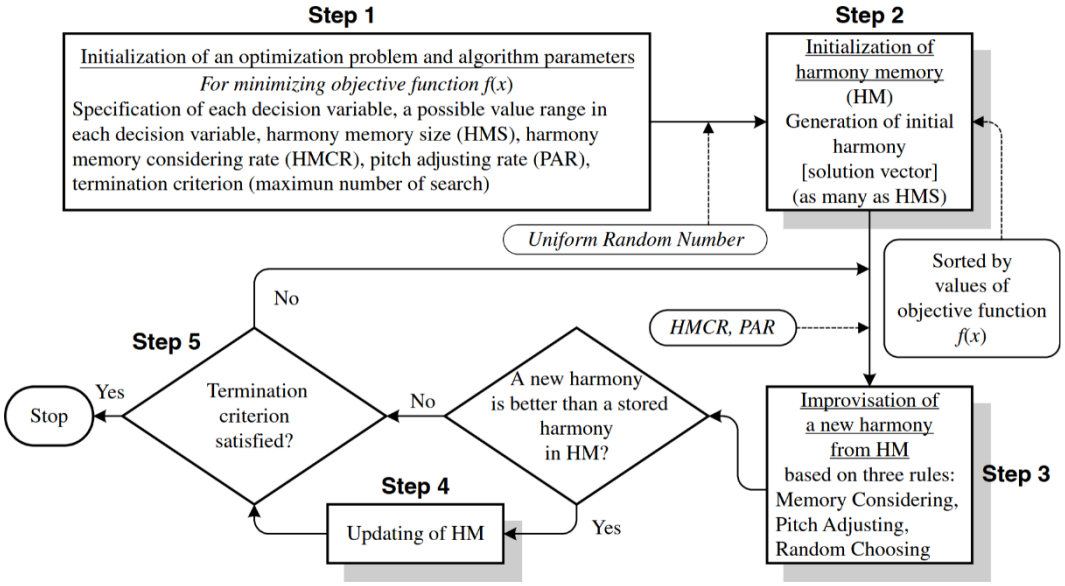

Fig. 3. The improvisation process of the Harmony Search Algorithm; Source: Lee & Geem (2005)

In 2004, Lee and Geem conducted a benchmarking study on the Harmony Search Algorithm applied to structural optimization problems involving continuous sizing variables. They evaluated the algorithm's performance using a series of truss design examples, including large-scale structures subjected to multiple loading conditions. The HSA was benchmarked against conventional gradient-based optimization methods and Genetic Algorithms. The results showed that HSA consistently produced optimal or near-optimal solutions and outperformed both traditional gradient-based methods and Genetic Algorithms. This superior performance is likely due to HSA's approach of generating new solution vectors by taking into account all existing **Solution Vectors**, guided by the **HMCR**, while in contrast, Genetic Algorithms typically rely on only two parent solutions to create new ones.

Yang (2009) also examined the HSA through the lens of fundamental metaheuristic principles, comparing its mechanisms to those found in other algorithms, elaborating on how HSA effectively balances intensification and diversification, a key reason behind its success as a metaheuristic. In HSA, diversification is driven by two distinct mechanisms: randomization and pitch

adjustment. Randomization introduces entirely new solutions, similar to approaches used in other metaheuristics, helping the algorithm broadly explore the solution space, and the diversification mechanism governed by the **PAR** parameter can slightly alter an existing solution within a narrow range, acting as a local refinement strategy. It not only encourages exploration near promising areas but also supports maintaining diversity in the population. The interplay between these components, **HMCR**, **PAR**, and randomization, results in a finely balanced search process that contributes significantly to the performance of HSA compared to other algorithms. Beyond its strategic design, the HSA also stands out for its simplicity and ease of implementation. It tends to be less sensitive to parameter settings, reducing the need for meticulous tuning to achieve high-quality results. Moreover, as a population-based method, HSA is naturally suited for parallelization, allowing multiple **Solution Vectors** to evolve simultaneously. This parallelism, combined with controlled elitism and a well-calibrated balance of exploration and exploitation, underpins its high efficiency. The HSA's strategic design and flexibility are central to its continued success and popularity across a wide range of optimization problems.

Manjarres et al. (2013) identified a broad spectrum of domains where the HSA has been effectively applied, including engineering, construction, telecommunications, robotics, energy and healthcare. The algorithm has consistently shown strong performance in generating near-optimal solutions, as validated through extensive simulation-based experiments and corroborated by rigorous statistical hypothesis testing. Given the rapid growth in research activity surrounding Harmony Search, it is anticipated that its computational advantages will extend into a range of emerging disciplines. These include, but are not limited to business intelligence, forensic science, crime analytics, smart grid management genomics and renewable energy, many of which are integral to the evolving landscape of Big Data. Additionally, this expansion is likely to inspire the development of novel HSA variants and functionalities tailored to the unique requirements of these new application areas. The increasing interest in HS also

signals promising avenues for future research. From a computational standpoint, enhancing memory efficiency and improving algorithmic speed are key challenges. From a theoretical perspective, the development of easy-to-use implementations of HSA is particularly important, since practitioners often prefer algorithms that can be deployed with minimal manual tuning. The study has also shown that despite the existence of various HSA variants proposed in the literature, many of these derivations fail to integrate feedback from actual end users. Instead, they are often based on theoretical assumptions that may not align with practical needs. To ensure the long-term viability and real-world applicability of HSA, greater collaboration between academic researchers and domain-specific practitioners is essential. Such interdisciplinary engagement will help ensure that algorithmic developments are responsive to real-world challenges, thereby reinforcing HSA as a robust tool for decision-making in complex optimization scenarios.

The next chapter elaborates on the detailed implementation of the mechanics and functions of EGAL+, illustrating how it enables a new level of flexibility and precision in automated assessment design, building on the foundation of the Harmony Search Algorithm.

5. THE IMPLEMENTATION OF EGAL+

5.1. Theoretical Overview

EGAL+ is an advanced computational framework developed to automate the construction of task sequences from question banks for educational assessment purposes. Its primary function is to ensure that the exams generated are not only varied, but also fair by maintaining a given difficulty level across all produced task sequences, while also being aligned with specific grouping requirements defined by the user, and to do all that in a scalable way. The complete source code for EGAL+ has been made publicly accessible for research and practical use at GitHub under the following URL: https://github.com/balazs-domsodi/EGALPP.

In its processes for assessment generation, EGAL+ builds task sequences from a user-supplied repository of tasks, called the question bank. Each individual task within this repository is annotated with the following key attributes:

- **Textual content**, representing the task itself.
- **Difficulty rating**, measured on a scale ranging from 1 to 5.
- **Coexistence preference** vector, which expresses the desirability of pairing each task with all other tasks in the bank. This vector has a length equal to the total number of tasks in the question bank minus one (excluding itself), and each value lies on a 0-10 scale. A score of 0 explicitly forbids the inclusion of two tasks within the same sequence, whereas higher scores signify stronger preferences for those tasks to appear together. All the coexistence preference vectors together are denoted as the **Preference Matrix**, which is a unique and novel feature of EGAL+ in the landscape of automated assessment generation, enabling more precise grouping definition for tasks in sequences than ever before in a scalable manner.

An example of a **Preference Matrix** can be seen in Table 2.

	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
Task 1	-	2	8	5	9
Task 2	2	-	10	3	4
Task 3	8	10	-	1	6
Task 4	5	3	1	-	10
Task 5	9	4	6	10	-

Table 2. An example of a Preference Matrix; Source: Author

Each task with these attributes in the question bank represents a **Decision Variable** for the Harmony Search Algorithm, which may or may not be included in a task sequence created during the generation process. Normalization of metrics was considered, but it was concluded that genuine comparability of question banks requires empirical calibration by topic experts rather than statistical rescaling.

The EGAL+ system is not inherently constrained by the type of questions it can support, it only processes and optimizes representations of tasks. Consequently, whether a task takes the form of multiple-choice, essay, grouping, or fill-in-the-blank, or any other form, is immaterial to the algorithm itself.

The only practical requirement is that each task can be encoded as a string of characters, which serves as the unit of representation within the system. Even in more complex scenarios, these strings could act as references to external resources or files, preserving compatibility with the algorithmic framework. In this sense, EGAL+ remains fundamentally content-agnostic, with its optimization procedures entirely decoupled from the semantic nature or format of the underlying tasks.

In addition to providing the question bank, users must specify several other generation parameters as well, defining the characteristics of the exam to be created:

- **The number of tasks to be included in each task sequence**, defining the length of the **Solution Vectors** for the Harmony Search Algorithm, and the task sequences representing the **Solution Vectors** themselves.
- **The total number of task sequences required for the exam**, which represents the **Harmony Memory Size (HMS)**, and where the task sequences together form the **Harmony Memory (HM)** itself for the Harmony Search Algorithm. The **HM** consists of the task sequences, which contain the indexes of the chosen tasks (the **Decision Variables**)

- **Desired overall difficulty**, representing the exact desired sum of contained task difficulties in all generated task sequences, with low, medium and high options to choose from, for which the program automatically suggests the exact values that can meet all generation criteria, based on the other input arguments.

To find the selectable desired overall difficulty options, the program performs a structured search to identify three feasible target sums of task difficulties that correspond to the low, medium, and high difficulty categories. These target values are selected to be as distinct as possible while still allowing the construction of a sufficient number of valid sequences that satisfy coexistence constraints. This way, EGAL+ ensures that within a single exam, all task sequences assigned to the same difficulty category possess identical cumulative difficulty scores.

To make the process of finding the three distant achievable cumulative difficulty values that adhere to all of the constraints as well computationally feasible, one of the defining features of EGAL+ is the integration of difficulty-level determination with the initial generation of sequences. Instead of separating these steps into independent phases, the program simultaneously explores possible difficulty configurations while constructing valid sequences and classifying them according to their difficulty totals. This integrated method eliminates unnecessary repetition and ensures that the output population of sequences is immediately ready for refinement by the Harmony Search Algorithm. See the exact code implementation of this in the following subchapter.

It is also worth mentioning that the task sequence-building process itself is iterative. Rather than attempting to generate full initial sequences in a single step by random allocation, which would increase the likelihood of producing invalid or redundant combinations, tasks are added one by one to each sequence and validated after each step. The index of each chosen task is recorded during this process, enabling compact storage and more efficient memory usage. This incremental

strategy significantly accelerates task sequence construction and reduces computational overhead.

After providing all other input arguments, the user selects a target difficulty category and EGAL+ transitions directly to the Harmony Search Algorithm based optimization phase. In this stage, the initial population of sequences, already compliant with strict coexistence and difficulty constraints, is enhanced to improve the overall quality of the exam. Sequence quality is evaluated according to two principal criteria which define the multi-objective fitness function $\boldsymbol{f(x)}$:

- **Coexistence preference adherence,** which reflects the extent to which the included tasks collectively exhibit high coexistence preference scores**.**
- **Distinctiveness,** which measures how different a sequence is compared to others in the same population.

Sequences that display stronger pairwise preferences and exhibit greater uniqueness relative to other sequences are prioritized as higher quality within the final output set. The inclusion of prohibited tasks together according to the Preference Matrix, or any deviation from the target cumulative difficulty of the task sequence is not allowed under any circumstances.

The determination of sequence quality within EGAL+ relies on a formally defined fitness function that integrates the two principal components of coexistence preference adherence and distinctiveness. Both components are quantitatively measured and then combined to produce a single fitness value for each candidate sequence.

Coexistence preferences are systematically represented in the Preference Matrix, denoted as $\boldsymbol{CP}$, where each element $\boldsymbol{CP[a][b]}$ specifies the desirability of including task $\boldsymbol{a}$ together with $\boldsymbol{b}$ task in the same sequence. For a given task sequence $\boldsymbol{p}$, which is a vector of $\boldsymbol{ex\_length}$ number of task indexes, the total

coexistence score $\boldsymbol{F_{coexistence}(p)}$ is calculated by summing the pairwise preference values of all unique task pairs within that sequence as defined by formula (1).

$$F_{coexistence}(p) = \sum_{i=0}^{ex\_length\ -1} \sum_{j=i+1}^{ex\_length} CP[p[j]][p[i]] \qquad (1)$$

This formula ensures that every pair of tasks contributes to the overall measure of how well the sequence aligns with the predefined coexistence expectations.

While coexistence focuses on internal harmony within a single sequence, distinctiveness emphasizes the external differentiation between sequences. The relative distinctiveness of two task sequences in the population is defined as the number of tasks in which they differ. For a particular $\boldsymbol{p}$ task sequence, its distinctiveness score $\boldsymbol{F_{distinctiveness}(p)}$ is obtained by summing its pairwise differences with all other task sequences in the current population, excluding any other sequence it intends to replace.

This metric ensures that the overall population avoids redundancy by favoring sequences that contribute to new combinations rather than merely replicating existing ones.

The fitness function of a $\boldsymbol{p}$ task sequence integrates both the coexistence score and distinctiveness score into a single quantitative measure by addition as defined by formula (2).

$$F_{fitness}(p) = F_{distinctiveness}(p) + F_{diversity}(p) \qquad (2)$$

Although various mathematical operations could theoretically be used to merge the coexistence score and distinctiveness score components, addition was deliberately chosen during EGAL+'s development. This decision reflects the assumption that coexistence and diversity represent independent yet equally significant aspects of quality. The chosen formula for merging and the relative weighting between these two components can be determined subjectively by the

instructor, based on their specific concepts for the given assessment creation. Currently, EGAL+ treats both components as equally weighted; however, future versions may include options allowing users to specify custom ratios and merge operations to better reflect their pedagogical goals.

Once the fitness function for all task sequences has been initially calculated, the progressive enhancement of the average quality of the population begins, while preserving the required constraints.

In each iteration, EGAL+ produces a candidate task sequence using one of three probabilistic approaches based on the Harmony Search Algorithm:

- **Generation from scratch:** A completely new sequence is assembled by randomly selecting tasks while adhering to coexistence and difficulty constraints. This happens with the probability of **1 - HMCR**.
- **Modification of an existing sequence:** An existing sequence is copied, and one task is substituted with a random alternative from the question bank, provided that the replacement maintains validity. This happens with the probability of **HMCR * PAR**.
- **Direct copying:** Occasionally, a high-quality sequence is retained unchanged to preserve beneficial characteristics in the population. This happens with the probability of **HMCR * (1 – PAR)**.

The candidate sequence is then compared to the lowest-quality sequence currently stored in the **Harmony Memory** (the population). However, even if the candidate task sequence is better in terms of fitness than the lowest-quality sequence, the potential substitution is not determined solely by the candidate's individual fitness score. Because the distinctiveness component of the fitness function depends on pairwise differences among all sequences, replacing a single sequence may alter the fitness values of the entire population. Therefore, EGAL+ evaluates whether substituting the lowest-ranked sequence with the candidate

would yield an overall improvement in population quality rather than merely benefiting the candidate in isolation. Only if the replacement leads to a net increase in total population fitness is the substitution accepted. If no suitable replacements can be identified within a given time frame, the candidate sequence is discarded, and the population remains unchanged for that iteration.

The optimization process continues iteratively under two termination conditions:

- **Generation limit:** If the number of iterations reaches a predefined $\mathbf{N_i}$ maximum, the process stops even if improvements remain possible.
- **Convergence criterion:** If the population's average fitness does not improve beyond a predefined threshold over a given number of consecutive iterations, the process stops.

The final output consists of a population of the required number of task sequences that each contain the required number of tasks and satisfy all strict coexistence constraints, maintain the requested difficulty target, and exhibit enhanced quality as measured by both coexistence alignment and distinctiveness.

One of the most notable strengths of EGAL+ is its ability to manage extremely large question banks, which may contain several thousand individual tasks, while still generating high-quality and constraint-compliant task sequences for even hundreds of students within just a few seconds on standard, widely available computer systems. This impressive level of computational performance is the result of a highly optimized internal architecture that has been specifically designed to minimize unnecessary processing steps, reduce memory overhead, and maintain strong algorithmic efficiency even as input sizes and complexity grow.

Because it can reliably perform rapid exam generation on such a large scale, EGAL+ offers particular advantages in educational contexts where extensive subject matter must be tested across large student populations. Examples of such

settings include online learning platforms as well as large undergraduate courses that require frequent examinations covering broad and detailed curricular areas.

5.2. Member Variables and Processes

Shortly after joining the research focusing on the development of EGAL+, the author of this thesis has rebuilt the entire program from the ground up, which was originally written in PHP, in C++, substantially refining almost all of its processes and extending its capabilities to achieve the high level of optimization now integral to the program.

C++ is a general-purpose programming language that extends the capabilities of C, functioning as a superset basically. In addition to the features inherited from C, it introduces mechanisms for defining new types in a flexible and efficient way. These mechanisms allow developers to break complex applications into manageable components by creating types that closely reflect the program's conceptual structure. When applied effectively, this approach results in code that is shorter, clearer, and easier to maintain. Unlike some higher-level languages, C++ does not provide built-in high-level abstractions such as native matrix types or strings with extensive built-in functions. Instead, developers are expected to define such abstractions themselves (or use the standard library on some occasions, denoted as the std namespace), since creating both general-purpose and domain-specific types is a core aspect of C++ programming. A key principle of C++ design is to avoid features that impose runtime or memory overhead when they are not used. At the heart of this design are user-defined classes. Despite offering advanced abstraction capabilities, C++ maintains C's low-level efficiency and direct hardware access. This ensures that custom types can be implemented with minimal performance cost. The language's minimalist yet powerful design makes it suitable for virtually all computing platforms and places high performance and fine-grained control at the center of its philosophy (Stroustrup, 2013).

C++ was chosen specifically by the author of this thesis to be the source language of EGAL+ because of its high support for breaking down complex

applications through data abstraction, without incurring any unnecessary overhead, and its focus on performance, and as such being an ideal choice for an optimization problem. The following is a high level description of the class developed for this purpose, detailing its member variables and elaborating on its processes of exam generation, which were designed to implement the optimization theory of EGAL+ efficiently and in a scalable way.

Member variables:

- **std::vector<std::string> task_contents;** This variable represents the complete collection of the textual contents of each individual task. Each task is a distinct unit of content that can appear within a generated exam. Conceptually, this vector forms the part of the question bank which consists of the tasks themselves. Its length equals the size of the question bank.
- **std::vector<unsigned char> task_difficulty_values;** Each task in the question bank is associated with a numerical measure that expresses how challenging it is relative to the others. This vector associates each task with a quantitative difficulty rating on a 1-5 scale. Its length equals the length of the **task_contents** member variable and the size of the question bank.
- **std::vector<std::vector<unsigned char>> coexistence_preferences;** This structure encodes the degree to which tasks can or should appear together within the same sequence. Each pair of tasks has an associated score that reflects their mutual compatibility, consequently this member variable directly implements the Preference Matrix, a symmetric matrix containing the pairwise coexistence preferences between tasks on a 0-10 scale, where 0 prohibits joint selection. Since the compatibility between task A and task B is identical to that between task B and task A, storing the full matrix would be

redundant, because each off-diagonal value would be duplicated, so to save space and simplify storage, only one half of the symmetrical Preference Matrix is stored. That means that the length of the outer vector equals the size of the question bank, and the length of each inner vector equals its index number (i.e., row N contains N values, each corresponding to the coexistence value with tasks 0 to N-1).

- **unsigned int exercise_length;** This variable defines how many tasks will appear in each completed task sequence. It defines the dimensionality of each **Solution Vector** in the Harmony Search Algorithm. The higher values this takes, the more it increases the complexity of the optimization process, as both coexistence validation and distinctiveness computation scale with this value.
- **unsigned int population_size;** This variable represents the number of task sequences (i.e., the **Harmony Memory Size** or **HMS**) to be maintained simultaneously during the optimization process. It directly influences the computational cost of each optimization iteration, since distinctiveness calculations are pairwise over the population.
- **std::map<unsigned int, std::vector<std::pair<std::vector<unsigned int>, double>>> population_options;** This is a complex structure that stores candidate populations indexed by their associated difficulty category, making the **unsigned int** difficulties the keys in this context. Each value for these keys is a vector of pairs where the **std::vector<unsigned int>** represents a single task sequence by storing the indexes of its tasks, and the **double** represents the quality value of that sequence, reflecting the fitness function (coexistence + distinctiveness). That way each group corresponds to a specific difficulty sum and contains all sequences that match

that exact difficulty total. It functions as the central repository of potential solutions during the generation process, making the candidate population associated with the chosen difficulty the **Harmony Memory** or **HM** in the Harmony Search Algorithm.

- **unsigned char number_of_options_goal;** This value expresses a target for how many distinct difficulty categories should be represented in the **population_options**. This is by default set to 3, but it can be lower if no three distinct population options can be created with a unique difficulty, based on the input arguments. In the future it may also hold a greater number if the user interface will allow defining a custom **number_of_options_goal** in a subsequent version of the program.
- **unsigned int difficulty_difference_goal_in_options;** This variable specifies the required spacing between adjacent cumulative difficulty keys in the **population_options**. Its goal is to prevent difficulty categories from being too similar and helps that they are meaningfully distinct rather than just marginally different. When time constraints are met in the initial generation process, it is adaptively reduced, relaxing the difficulty partitioning until feasible population options can be built.

Below, it is described in detail how EGAL+ employs its methods to generate exams with an unprecedented degree of parameterization flexibility, combined with a highly efficient optimization process.

The execution begins by loading task data from an external question bank file into the vectors **task_contents**, **task_difficulty_values**, and **coexistence_preferences**. Once these structures are initialized, they serve as the immutable foundation for generating and refining the population of candidate exams.

The first process is the calculation of the first set of potential cumulative difficulty values. To do this, the algorithm computes the sums of the lowest and highest **exercise_length** number of elements in **task_difficulty_values**. From these values, it derives the initial **difficulty_difference_goal_in_options** by equally dividing this distance to **number_of_options_goal** parts, which expresses the targeted spacing between difficulty categories. This way the **difficulty_difference_goal_in_options** variable starts from the highest possible value based on the **exercise_length** and the **task_difficulty_values**.

Then comes the sequence construction method which uses these variables along with the user-defined **exercise_length** to create sequences of task indices, each sequence representing a part of a possible solution. A single candidate sequence is built by randomly selecting indices that point into the question bank, effectively resolving **task_contents**, **task_difficulty_values**, and **coexistence_preferences**, while ensuring that the size of the sequence matches **exercise_length**, that no index repeats, and that all chosen indices are mutually valid according to **coexistence_preferences**. Each attempt to add a new index involves checking all existing members of the sequence against the candidate index, using the corresponding matrix entries in **coexistence_preferences** to determine compatibility. If an attempt fails for too much time (time frame hard coded as of now, can be made into a user customizable parameter), the process flags the sequence as unfinishable, discards it, and restarts with a clean new vector. The method continues to iterate until a fully valid sequence is found, which is then sorted by its indices and returned.

The population generation loop repeatedly calls the sequence construction method described above and obtains the sum of **task_difficulty_values** for the returned sequence and inserts it into the **population_options** under its corresponding difficulty key if it already exists, and if there is no corresponding difficulty key yet, it creates a new key associated with that cumulative difficulty and inserts the sequence under that one. If the key already exists, the algorithm ensures that duplicate sequences are not stored by discarding the returned sequence.

As task sequences in **population_options** accumulate, the program searches for those difficulty keys that have reached a frequency threshold defined by **population_size** multiplied by a currently hard coded, but optionally user definable >=1 value, determining how many proven available variations are expected above **population_size** to ensure space for future modifications during enhancement. Keys that meet or exceed this threshold are considered strong enough to form a difficulty category.

The algorithm attempts to collect a list of such categories where each successive key differs from the previous one by at least **difficulty_difference_goal_in_options**, ensuring a spread of difficulty levels rather than tightly clustered ones. If the number of collected categories reaches **number_of_options_goal**, the program filters **population_options** so that only entries corresponding to these selected keys remain, and each surviving vector is resized to **population_size** to standardize its length. If sufficient categories cannot be gathered within the given time limit constraint (again hard coded currently but can be changed into a user customizable parameter), the algorithm progressively reduces **difficulty_difference_goal_in_options** and, if necessary, **number_of_options_goal** to relax its criteria.

Once **population_options** has been filled and the user has selected a specific difficulty, the program focuses on that difficulty exclusively. It removes all entries from **population_options** except the one whose key matches the chosen difficulty. Each task sequence stored under that key then receives a computed fitness score. This score is based first on internal compatibility: for each pair of task indices within a sequence, the program retrieves the corresponding value from **coexistence_preferences** and accumulates it. Second, the fitness function accounts for distinctiveness relative to the other sequences already stored under the same difficulty key by computing the set difference between the current sequence and each other sequences and incorporating the number of differing elements into the score. These fitness values are stored directly in the second component of each pair within the vector mapped by **population_options**. At this point, the population

consists entirely of candidate exams with the same total difficulty value, standardized task sequence lengths of **exercise_length**, and initial fitness values ready for the Harmony Search Algorithm based refinement.

The implementation of the Harmony Search Algorithm seeks to increase the overall average fitness of the population stored in **population_options** while maintaining its size as **population_size**. The algorithm iteratively generates new candidate sequences using a process controlled by parameters that mimic the Harmony Search Algorithm.

At each iteration, a random value (**HMCR**) determines whether a new candidate will be drawn from the existing population (memory consideration) or will be freshly generated by calling the sequence construction method used for initial population generation as well with the addition of difficulty checking.

When a task sequence is derived from the existing population, it may undergo a pitch adjustment step (with the probability of **PAR**), in which one randomly chosen task index is replaced with a new index that has the same difficulty (ensuring the total difficulty remains constant as dictated by **task_difficulty_values**) and that remains compatible with all other tasks according to **coexistence_preferences**. This substitution attempt is bounded again by a hard coded but optionally user definable time limit, ensuring the program does not stall while searching for valid replacements.

Each new candidate is assigned a fitness score using the same method described earlier. The program then identifies the weakest member in the current population by scanning **population_options** and comparing their stored fitness values. If the new candidate's fitness exceeds that of the weakest member, the program evaluates whether replacing that member improves the population's average fitness (since the replacement of a task sequence may change the fitness values of other sequences as well because of the pairwise relation in distinctiveness). If it does, the weakest member is replaced, and all fitness values are recomputed to maintain consistency.

The algorithm tracks progress by comparing the current average fitness to the previous generation's average; if improvements fall below a small threshold epsilon (hardcoded, but optionally user definable) for too many consecutive generations the loop terminates early. Otherwise, the process continues until a specified generation limit is reached.

At completion, the **population_options** structure holds a refined exam abstraction which is characterized by high coexistence compatibility and mutual distinctiveness between task sequences, and that also exactly meet the user's difficulty requirement. A flowchart of the implementation is shown in **Figure 4**.

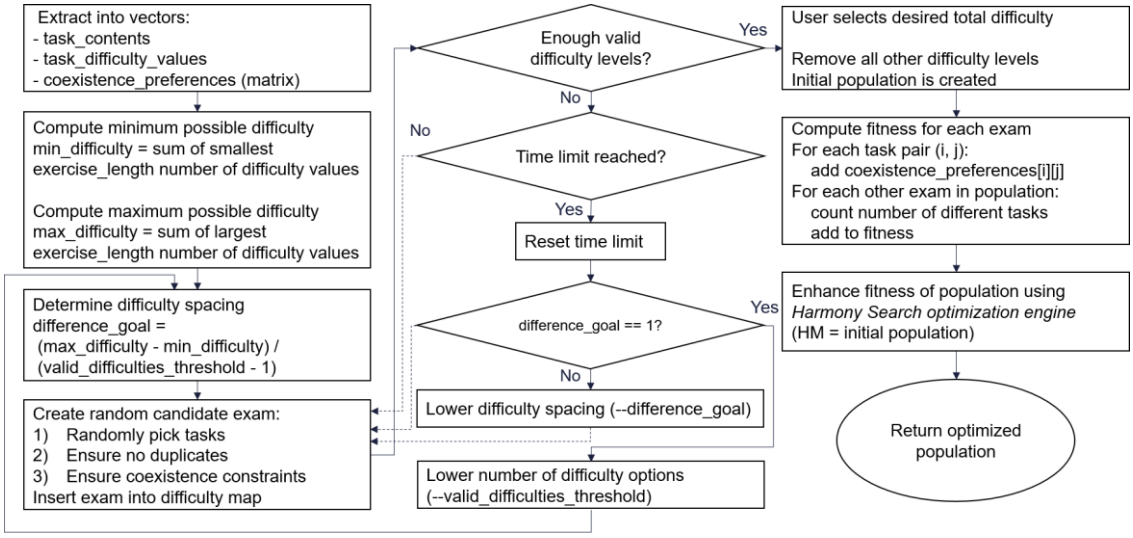

Fig. 4. Flowchart of the implementation of EGAL+; Source: Author

As discussed in the detailed explanation above, the program-level implementation of EGAL+, as designed and developed by the author of this thesis, demonstrates a strong emphasis on achieving high computational efficiency while maintaining fine-grained parameterization options, enabling users and researchers to customize the program's internal behavior with great precision and also keep high scalability. This level of optimization in compiling constraint bound task sequences paired with the high level of configuration possibilities is one of the central contributions of this thesis in advancing the practical capabilities of automatic assessment generation systems.

However, while the technical design focused heavily on optimizing algorithmic performance, the user level handling of parameterization was deliberately left open and for now most parameters are hard coded placeholders. For instance, the setting of key optimization parameters like the Harmony Memory Consideration Rate (HMCR), Pitch Adjustment Rate (PAR), and the numerous timing constraints, as well as the weighting and operation between the coexistence and distinctiveness components of the fitness function, and the variables regarding the convergence related stopping criteria, are all dependent on what user preferences may arise regarding the exam generation and how much time are they willing to sacrifice for what measure of enhancement to be expected.

This raises an important question: should users be granted full access to all parameters for maximum flexibility, or would a more streamlined interface with preset configurations be more beneficial for most use cases? If the latter, on what basis should these presets be defined? By use case, domain, experience level, or some other factor? Additionally, simplifying the overall process of input configuration can significantly improve the accessibility and adoption of the system, particularly among non-technical users or educators who may not be familiar with the underlying algorithmic mechanics, but to find a well-suited method for defining all the variables in an automatic way can be a research project on its own in the future.

For now, this thesis provides an advanced system that is proven to work and ready to tackle the generation process itself, given the input parameters can be set by the user to support the concept of the exam.

The following subchapter provides a detailed overview of the current user-level operation of the program, including how users interact with the system, what configurations are available, and what steps are involved in generating assessments using the existing interface.

5.3. User Level Operation

The program is currently operable via a command line user interface (UI), and its use requires the user to provide several input parameters. Some of these parameters can be specified directly through the command line and others by creating an external input file (referred to as the Question Bank).

Prior to generation, the following necessary data must be defined in an external .txt file (Question Bank) to be read by the program.

- **Task contents:** textual definition of the individual tasks.
- **Task difficulty values:** numerical values indicating task difficulty (on a scale from 1 to 5).
- **Preference Matrix (coexistence preferences):** pairwise values expressing the preference for tasks to appear together in task sequences (on a scale from 0 to 10, where higher values indicate stronger preference, and 0 denotes a prohibited coexistence).

The external Question Bank file must be structured line by line, where each line represents the attributes of a single task. These attributes should be separated by tab characters. The first entry on each line corresponds to the textual content of the task, the second to its difficulty value, and the final entry contains the coexistence preference values, separated by semicolons, corresponding to the task's joint appearance preference with all preceding tasks.

For the first task, represented on the first line of the file, the coexistence preference value serves merely as a placeholder, as there are no preceding tasks to reference. Consequently, its value is arbitrary and may range from 0 to 255. The presence of this placeholder ensures that the indexing of the coexistence preference values is consistent with the indexing of other task attributes. Without it, the first task would lack this attribute due to the absence of prior tasks with which to compute a joint preference.

Optionally answer options can be included in the Question Bank as well for each task, which must follow the coexistence preference values, separated by a tab character in each task line, and the correct answer must be the first one of all the answer options, which are again separated by tab characters.

Answer options have no effect on the generation and optimization process, but if they are specified, the program is able to include them in the output, ready to be used in further processing of the generated exam.

A representative example of the Question Bank input file structure can be seen in **Figure 5**.

```
1    ExampleTaskText1    1    255
2    ExampleTaskText2    1    8
3    ExampleTaskText3    2    7;6
4    ExampleTaskText4    4    5;4;3
5    ExampleTaskText5    4    2;1;0;9
6    ExampleTaskText6    5    10;7;5;4;9
7    ExampleTaskText7    1    8;6;4;7;3;2
8    ExampleTaskText8    2    3;6;7;5;5;6;8
9    ExampleTaskText9    1    9;3;4;2;3;4;5;6
10   ExampleTaskText10   2    4;3;2;1;3;4;5;2;1
```

Fig. 5. Example content excerpt of a Question Bank; Source: Author

Prior to the exam generation, the following variables must be specified via the program's command-line interface:

- **Exercise length:** the desired length of the task sequences to be generated. This value must lie between 0 and the total number of lines (i.e., tasks) in the Question Bank file.
- **Population size:** the number of task sequences to be generated, also referred to as the Harmony Memory Size (HMS). This variable must be assigned a value between 0 and UINT_MAX as defined in C++.

At a certain stage during program execution, specifically, after the creation of the initial population and before the commencement of the Harmony Search Algorithm-based optimization, an overall target difficulty level must be selected from the available options provided by the program.

The output of the program consists of the specified number of task sequences, each containing the desired number of tasks, and constructed such that no prohibited task pairs appear together, while the total difficulty of each sequence matches the value specified by the user. Furthermore, all sequences are subject to HSA based optimization process, which was specifically designed to achieve the highest level of quality attainable by the program within the constraints of the available time frame, as determined by the multi-objective fitness function.

The exams produced by the program are automatically structured using the widely recognized GIFT format, a standard developed for the efficient creation and exchange of quiz content. This format ensures seamless compatibility with the Moodle Learning Management System (LMS), allowing educators to import and deploy the generated exams directly into their Moodle courses without the need for additional formatting or manual adjustments.

This section also demonstrates that the program currently requires only the minimal set of inputs from the user, which are sufficient for generating the exam. The fine-tuning of all other variables regarding the optimization process, as discussed in the previous subchapter, is considered a potential subject for future research, building upon the already functioning robust and scalable system proposed by this thesis. This future work should entail a systematic investigation into how optimization parameters can be automatically adjusted based on input sizes, structural complexity, and data relationships, and the possibilities of dynamic adjustment strategies, where parameters are recalibrated in real time based on intermediate computational results or convergence patterns.

Presented below is a step-by-step example that illustrates the user-level interaction with the program during the process of generating exams. This example

is intended to provide a clear and practical understanding of how users operate the system, from initial input to the final output.

First the user needs to choose one from the main operations of the program as shown in **Figure 6**.

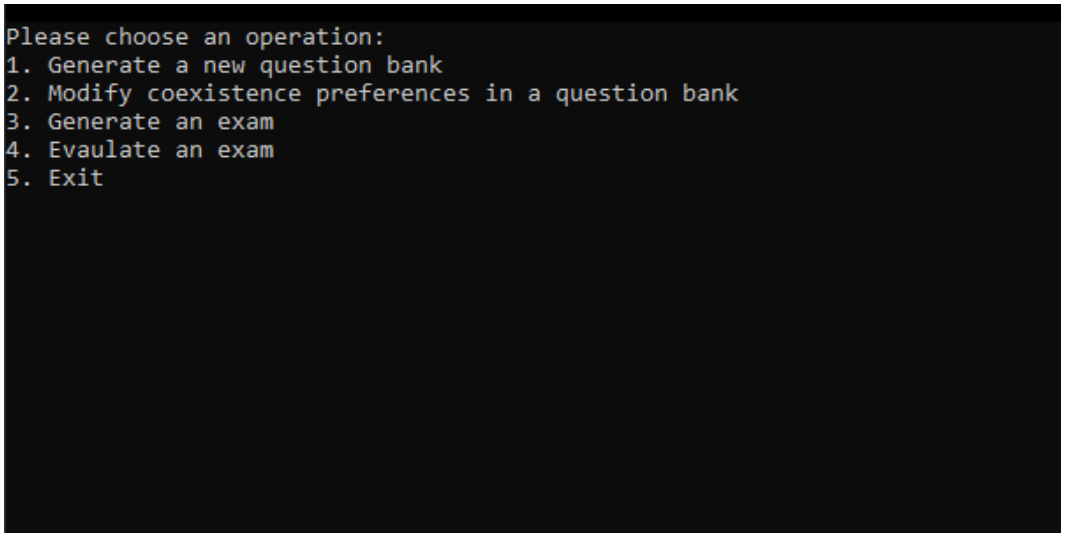

Fig. 6. The main operations as represented in the UI; Source: Author

To generate and optimize task sequences, the user must select the third option, "Generate an exam", by entering value 3. After the operation is chosen, the program asks for the name of the Question Bank file to be used in the process, which in the current implementation needs must be located in a folder named databank next to the executable file. The prompt is shown on **Figure 7**.

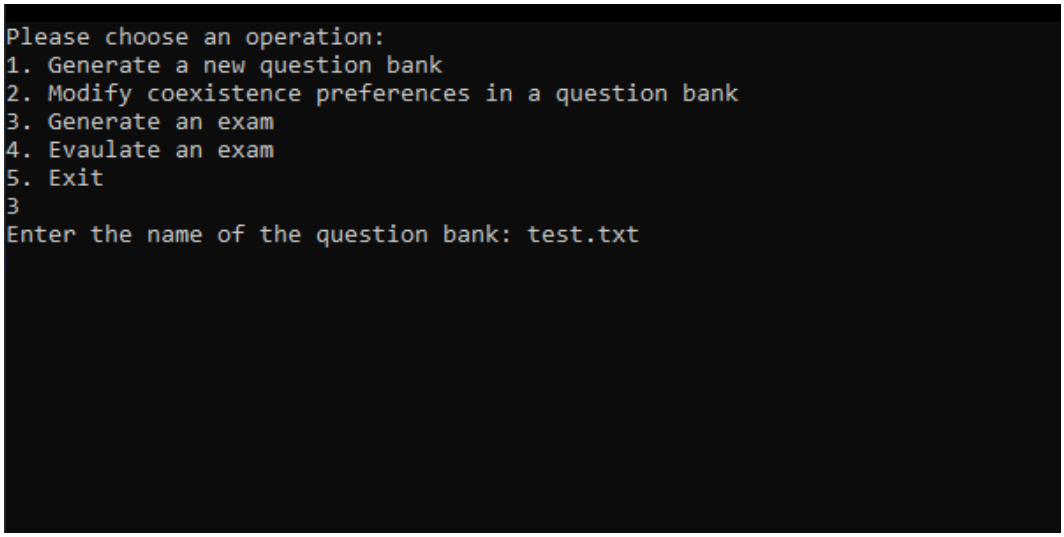

Fig. 7. Defining the Question Bank file in the UI; Source: Author

Next, the user is prompted to specify the desired exercise length, and the population size, respectively defining how many tasks the sequences should be compiled of and the total number of sequences, as can be seen in **Figure 8**.

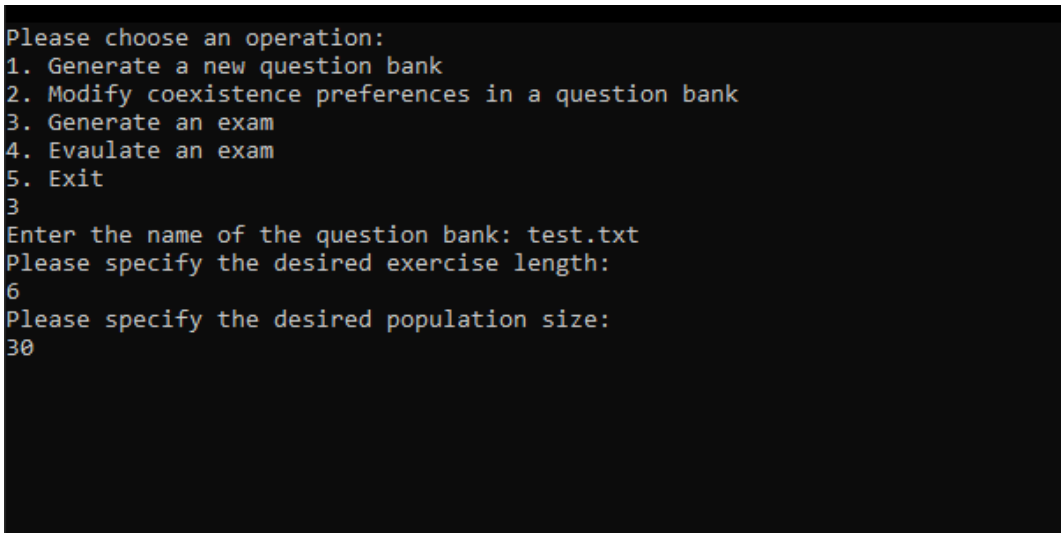

Fig. 8. Defining the exercise length and population size on the UI; Source: Author

Immediately after setting the values of exercise length and population size, the program looks for three achievable target difficulty options for the exercises to be generated (calculated by summarizing all the included tasks in an exercise) and lets the user choose one. The prompt of the difficulty options is shown in **Figure 9**.

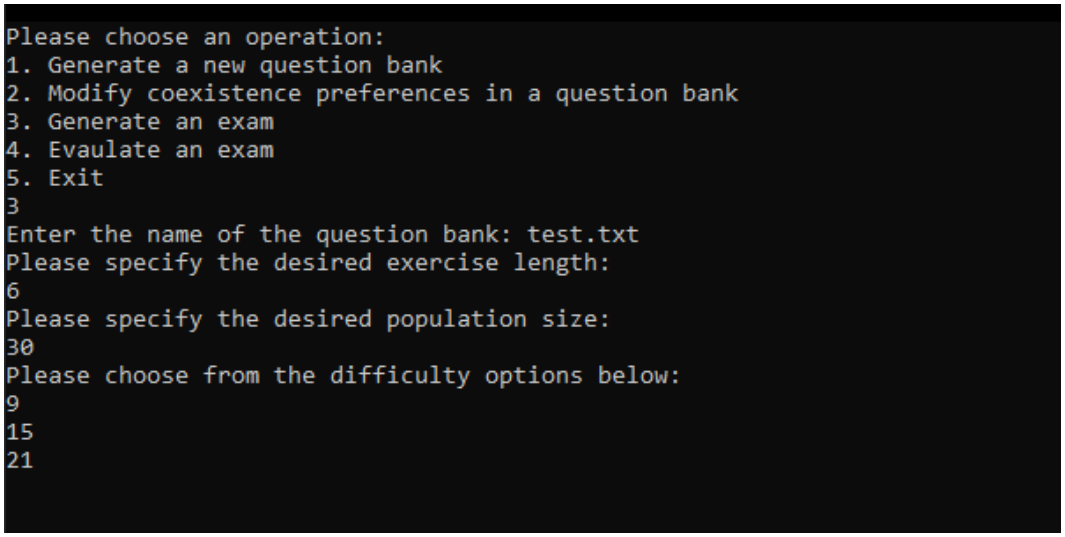

Fig. 9. Choosing a target difficulty value in the UI; Source: Author

When the target difficulty is chosen, the program executes the enhancement process and generates the output. The resulting UI can be seen in **Figure 10**.

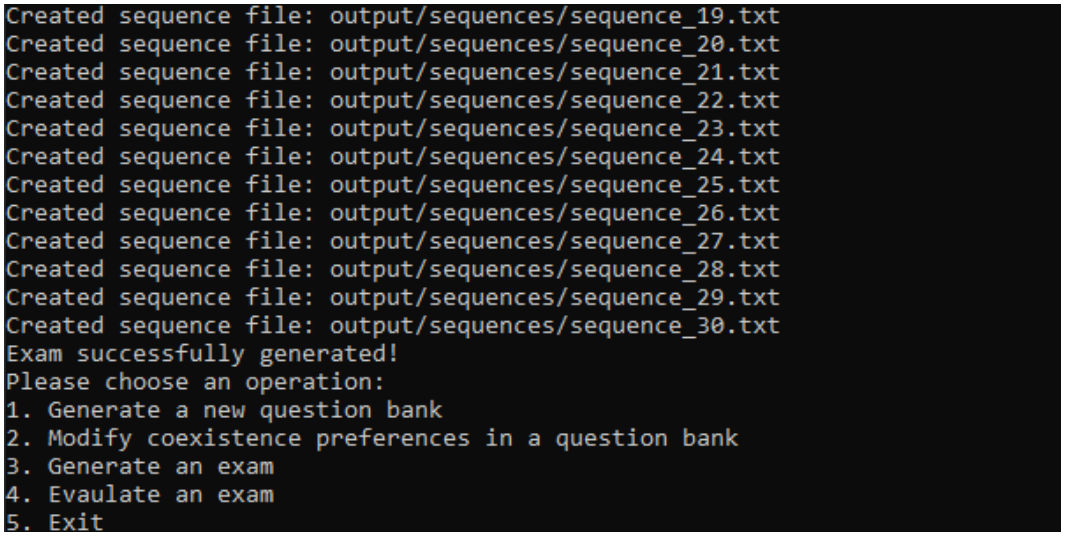

Fig. 10. Successful generation UI feedback; Source: Author

The output sequences are formatted in GIFT format as shown in **Figure 11**.

```
ExampleTaskText3 {
    =Task3CorrectAnswer
    ~Task3InorrectAnswer1
    ~Task3InorrectAnswer2
    ~Task3InorrectAnswer3
}

ExampleTaskText6 {
    =Task6CorrectAnswer
    ~Task6InorrectAnswer1
}

ExampleTaskText8 {
    =Task8CorrectAnswer
    ~Task8InorrectAnswer1
}

ExampleTaskText14 {
    =Task14CorrectAnswer
    ~Task14InorrectAnswer1
    ~Task14InorrectAnswer2
```

Fig. 11. Example excerpt of a task sequence output; Source: Author

The resulting output is ready to be imported into Moodle or to be further processed in any other way compatible with the standardized GIFT format. It is worth noting that the answer options, while useful for generating multiple choice questions (as seen on **Figure 11**), are completely optional and essay questions can also be generated just as easily by simply not providing any answer options.

In addition to the core functionalities of EGAL+, specifically, the compilation and optimization of task sequences, the program includes three other supplementary operations as well: "Generate a new question bank," "Modify coexistence preferences in a question bank," and "Evaluate an exam" (as can be seen in **Figure 6**). These auxiliary components, while not essential to the principal mechanisms of task sequence compilation and optimization, were incorporated into the system to support extended research objectives and to enhance the program's testability within real-world educational contexts.

The operation titled "Generate a new question bank" enables the program to autonomously construct a repository of tasks derived from unstructured textual input. This is accomplished through an external AI-based mechanism accessed by API, that processes raw text and populates the question bank accordingly. Notably, this functionality contributed directly to one of the author's research publications associated with EGAL+, an outcome that will be discussed in a subsequent chapter.

The other two operations, "Modify coexistence preferences in a question bank" and "Evaluate an exam", were primarily designed to facilitate experimental validation in authentic educational settings. More specifically, the "Modify coexistence preferences in a question bank" operation provides a somewhat streamlined editing interface for adjusting coexistence preferences between tasks, and the "Evaluate an exam" operation allows for the automatic measurement of an assessment's quality by applying the EGAL+ fitness function to any properly formatted GIFT input file.

To modify coexistence preferences in a Question Bank, the program first requests the name of the file to be used, which must be located in a folder named databank as described earlier. It then prompts the user to define groups of questions, entered as individual numbers or numeric ranges separated by commas, for example "2,5,6-10,12". For each group, the program reads the Question Bank line by line, checking the coexistence values associated with each question. If two questions belong to the same group, their coexistence value is raised to the value provided by the user, while preserving any higher existing values. After all groups have been defined, the program examines every pair of groups that do not share any questions and requests a joint coexistence preference for them as well, updating the coexistence values between questions in the two groups in the same manner. Each modification is written to a temporary file, which then replaces the original, ensuring the question bank remains consistent after every change.

The operation of modifying coexistence preferences in a Question Bank is shown in **Figure 12**.

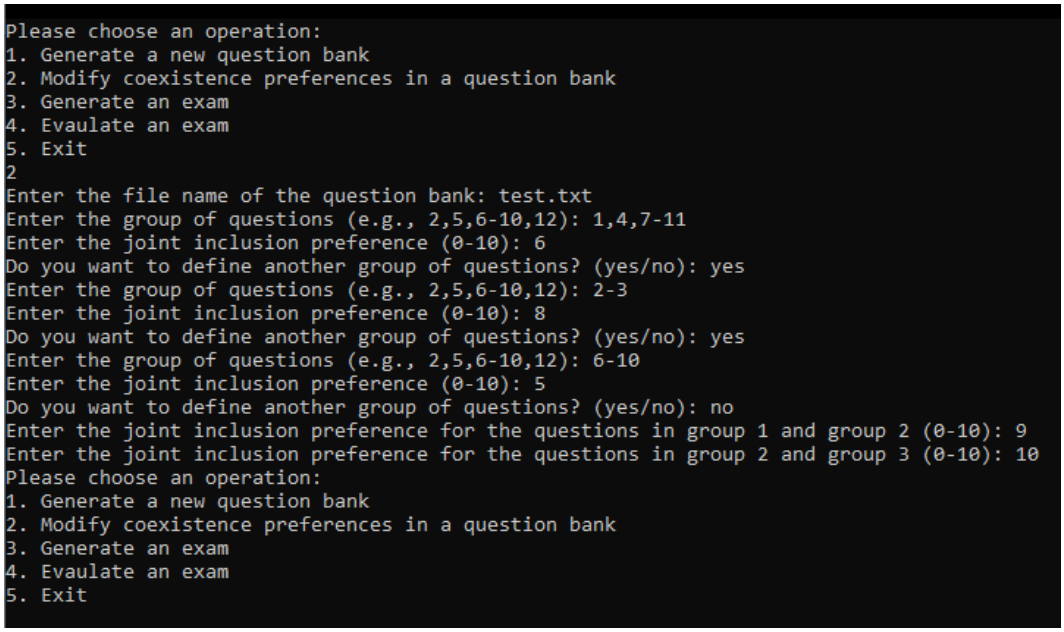

Fig. 12. The operation of modifying coexistence preferences in a Question Bank; Source: Author

A more detailed explanation of the in-development feature for modifying coexistence preferences in a Question Bank can be found in Appendix 2. It is important to note that the current interface represents an early approach to providing these tools and is to be further developed in the future.

The insights drawn from testing in a real-world educational setting will be elaborated upon in a later chapter focused on empirical evaluation.

Concluding the chapter outlining the theoretical foundations, implementation specifics, and user-level operations of EGAL+, the next chapter will focus on the performance benchmarking of the program. These measurements are of particular importance, as the principal contribution of EGAL+ lies on the one hand in its capacity to carry out optimization processes involving such highly detailed and complex parameters, and on the other hand in its scalability while working with such detailed interconnected parameters, a capability that significantly distinguishes this program from existing approaches in the domain.

6. BENCHMARKING THE SCALABILITY OF EGAL+

6.1. System Environment and Evaluation Prerequisites

As previously discussed in this thesis, upon joining the research focused on the development of EGAL+, the author undertook a comprehensive reconstruction of the program's logic. This redevelopment was implemented entirely from the ground up, resulting in significant improvements in both structure and performance. In transitioning from the original implementation, written in PHP, to a more robust version coded in C++, the author introduced several new solutions designed to enhance the program’s optimization capabilities and overall robustness.

Since during the comprehensive literature review, no existing assessment generator software was identified that offered a level of parameterization comparable to that of the pairwise Preference Matrix utilized in EGAL+, rather than benchmarking the revised version of EGAL+ against an entirely different program, the author opted to conduct performance evaluations by comparing it to the earlier iteration of EGAL+ that existed prior to the author’s involvement in the research and exhibited limited scalability. As previously mentioned, what truly sets EGAL+ apart from other assessment generation tools is its ability for deep, Preference Matrix-based parameterization. This feature allows for an exceptional level of customization and precision in configuring the generation process, which was not yet showcased in comparable systems. However, for this advanced level of configurability to have practical significance, it was imperative to ensure that the program could also scale effectively in terms of performance.

Given that the logic guiding the fitness value optimization functions remained unchanged in terms of the resulting magnitude of fitness enhancement between versions, the primary focus of the comparative evaluation discussed in this section was the time required for execution for the previous and the new method. A detailed analysis of the program’s effectiveness in improving fitness values will be provided in a subsequent chapter, which explores EGAL+ in the context of an actual university-level examination scenario.

The evaluation methodology of this benchmarking involved executing both the previous and the newly improved versions of the program multiple times using identical input parameters, followed by a comparative analysis of the results based on key performance indicators. Prior to conducting these tests, it was necessary to establish appropriate parameter values and determine the number of iterations for each scenario. For the initial comparison, the author selected parameter settings corresponding to the maximum limits permitted by the earlier version of the program, which contained hardcoded constraints. This baseline was used to determine whether the modifications introduced by the author yielded measurable improvements under controlled and previously validated conditions.

Following this initial scenario, the hardcoded limitations of the earlier version were progressively relaxed by the author by modifying its source code, and the parameters were incrementally scaled across four other stages. These stages culminated in a final scenario involving a question bank of 1,000 items, from which exercises consisting of 50 tasks were generated for a simulated student cohort of 500. This setup represents a large-scale, yet still practically feasible assessment environment.

All performance data were collected on a laptop equipped with an Intel Core i7-9750H 2.6 GHz CPU and 16.0 GB of RAM, a hardware configuration that can be considered widely available for consumer use nowadays and thus suitable for generalizability. For each test scenario, both the previous and improved versions of the program were executed separately and consecutively using the same input data. The average execution time and standard deviation were recorded and analyzed to assess performance improvements.

As previously noted, EGAL+ utilizes the Harmony Search Algorithm, a metaheuristic optimization technique that incorporates elements of stochasticity. Because of the inherent randomness in this algorithmic approach, it is recommended to perform multiple runs for any given test case to obtain reliable and generalizable results. In alignment with this principle, each of the five test

scenarios was executed 30 times. During these tests, coexistence preference values were randomly assigned with non-zero entries, task difficulty levels were distributed randomly on a scale from 1 to 5, and sequence-based cumulative difficulty targets were also selected at random.

It is important to highlight that the advanced target difficulty search mechanism, one of the more sophisticated features of the program, was not present in any form in the earlier version and was entirely invented and introduced by the author. This novel component, which addresses a complex search problem in the context of assessment generation, is detailed in a previous chapter dedicated to the technical implementation of EGAL+. The new version of EGAL+ is executing this search process as well within the same time constraints as the original version, thereby further demonstrating its enhanced computational efficiency.

Additionally, it should be noted that the previous version of the software imposed several constraints, not only in terms of question bank size, exercise length, and population size, but also in the permissible number of zero entries within the Preference Matrix. These limitations have been completely removed in the revised version, further expanding the program's flexibility and applicability to real-world educational scenarios.

The original PHP implementation can be found here: https://github.com/balazs-domsodi-h53osf/EGALplus

In the following subchapter, a detailed presentation of the data used and gathered in this comparative analysis is provided. The findings are supported by visualizations such as charts and graphs, accompanied by clear explanations to aid interpretation and understanding of the results.

6.2. Data Presentation and Visual Analysis of Comparative Findings

In the original version of the program, several hard-coded limitations were in place: the size of the question bank was capped at 50 items, individual task

sequences could not exceed 25 tasks, and the population size for algorithmic generation was restricted to 100. Consequently, the initial test cases were designed and parameterized in accordance with these constraints.

Following the removal of these hard-coded limits, a set of test cases was developed to reflect a range of incrementally increasing parameter values. This gradual scaling was intended to demonstrate the system's ability to handle increasingly complex and demanding assessment scenarios. The final benchmark case in this series expanded the parameters significantly, allowing a question bank size of up to 1000, exercise lengths of up to 50 questions, and a population size of up to 500 individuals. These benchmark values are summarized in **Table 3**.

Test case number	Size of question bank	Exercise length	Population size
1st	50	25	100
2nd	75	25	100
3rd	100	50	200
4th	500	50	350
5th	1000	50	500

Table 3. Parameters for the benchmarking test cases; Source: Author

Each configuration from the test series was executed under both the previous and the improved versions of the system, with execution times recorded in seconds. The performance was evaluated by comparing not only the total execution time but also the average and standard deviation of run times. The resulting data, presented in **Table 4**, illustrate the improvement in performance with the updated version, affirming the improved method's scalability and efficiency.

Test case number	Previous method		Improved method		Improvements	
	Avg. execution time (seconds)	*Standard deviation (seconds)*	*Avg. execution time (seconds)*	*Standard deviation (seconds)*	*Change in avg. execution time*	*Change in standard deviation*
1st	8.356	0.376	1.460	0.057	-82.528%	-84.840%
2nd	9.282	0.348	2.344	0.034	-74.747%	-90.230%
3rd	52.727	2.824	4.843	0.067	-90.815%	-97.628%
4th	1256.94	100.02	5.1	0.66	-99.594%	-99.340%
5th	6047.28	311.34	12.18	0.06	-99.799%	-99.981%

Table 4. Comparative benchmarking results in seconds; Source: Author

The execution time of the first three cases are visualized in **Diagram 1**.

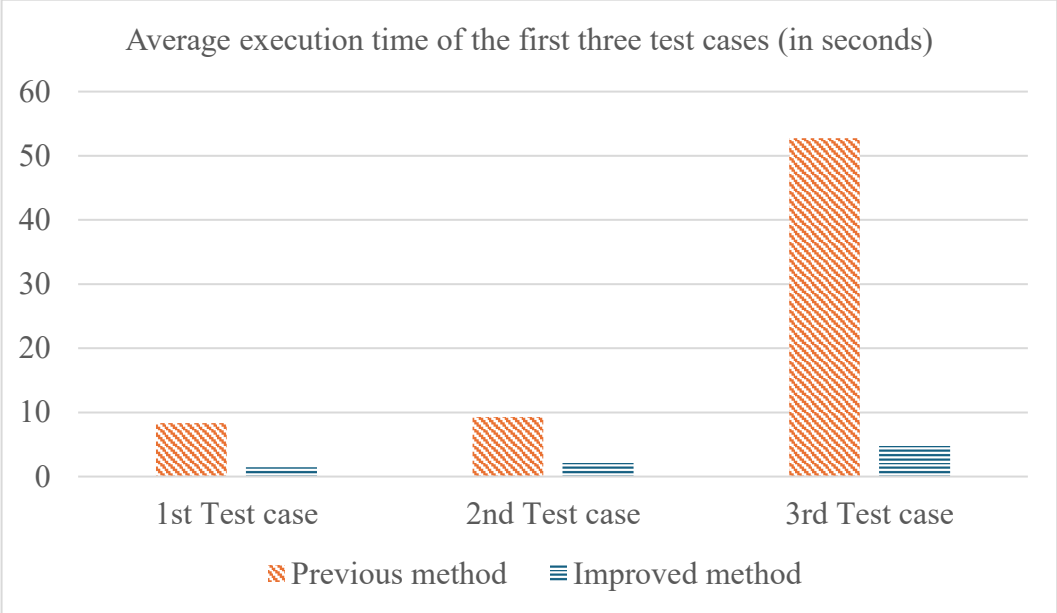

Diagram 1. Average execution time of the first three test cases (in seconds); Source: Author

The empirical results indicate that the enhancements introduced in the newer version lead to significant performance gains, even under the original, more limited parameter settings. In the first two test cases, where execution times remained under ten seconds for both the legacy and the updated methods, the performance improvements, while present, are less impactful in terms of user

experience. However, these cases still reflect better consistency and lower variability in the improved system's results.

More dramatic improvements emerge in the third test case. While the original method required nearly a full minute to return results, the enhanced system reduced the user waiting time to under three seconds, which represents a substantial leap in efficiency.

The fourth test case further emphasizes the improved system's performance. Here, the execution time of the enhanced method represents just 0.406% of the original method's time. Notably, the original version required nearly 21 minutes to complete the task, whereas the improved version executed it in under 6 seconds. This striking reduction highlights not only the efficiency of the new system but also the exponential growth in execution time associated with the original method as problem complexity increases. The enhanced system's performance remains nearly constant by comparison, showcasing its scalability and stability.

The fifth and final test case was specifically designed to reflect a realistic use-case scenario that justifies the enhancements introduced in this research. Under this configuration, the original method required over 1.5 hours to generate results. In contrast, the improved version completed the task in only 12 seconds, equating just 0.201% of the previous runtime. This substantial reduction exemplifies the real-world applicability and transformative impact of the improvements made to the system.

In addition to the previous comparisons, the growth patterns of the execution time associated with both methods in each test cases were visually represented in **Diagram 2**. This visualization clearly demonstrates the steep upward curve of the original method's execution time as complexity increases, in contrast to the relatively flat and consistent performance of the improved system.

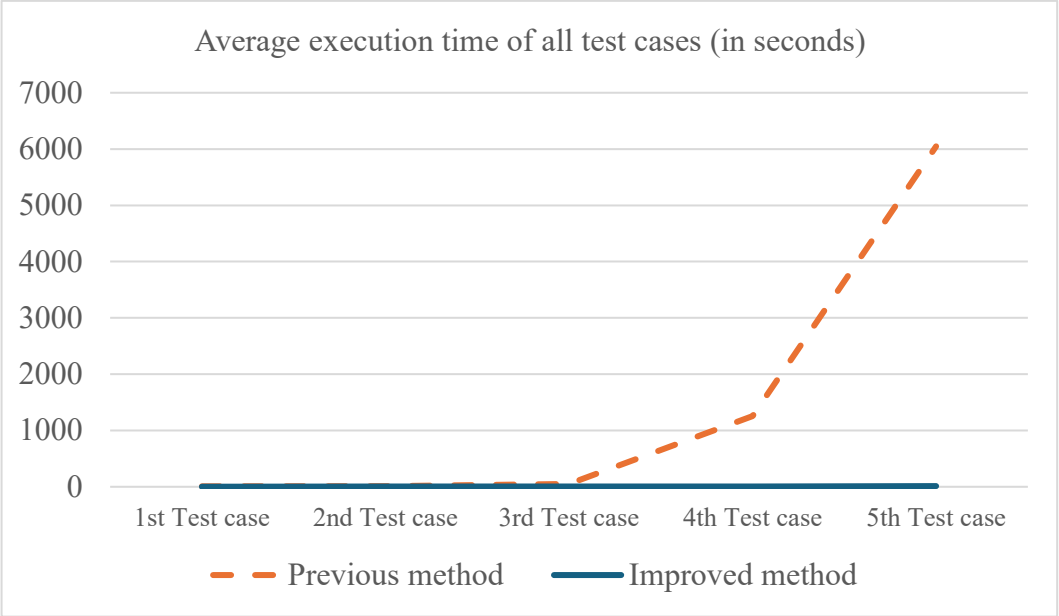

Diagram 2. Average execution time of all test cases (in seconds); Source: Author

The results of these test cases collectively underscore the significant advancements introduced by the improved EGAL+ system. This implementation fills the gap of performance and scalability regarding a previously unseen level of configurability in automatic assessment generation.

Having established EGAL+ as a robust and scalable solution for automatic exam generation, the thesis has outlined its underlying methodology based on metaheuristics, detailed its technical implementation, and validated its performance through empirical analysis.

In the subsequent chapter, the research will be extended further by exploring a complementary enhancement to EGAL+: the automated population of the question bank from raw textual input. While this feature is not directly related to the compilation and optimization phases of task sequence generation, it addresses a key aspect of system usability and forms the basis of one of the author's peer-reviewed publications.

7. INTEGRATION OF AI AND METAHEURISTICS IN EDUCATIONAL SOFTWARE: A HYBRID APPROACH TO EXAM GENERATION

7.1. Theoretical grounding

As discussed in the literature review, assessment generation algorithms can be categorized according to their methodological approach, most commonly falling into two broad groups: those that utilize heuristic optimization procedures, and those that are based on machine learning techniques. Within this framework, these systems can be further classified based on whether they emphasize the strategic composition of tasks for an assessment or concentrate on the generation of the tasks themselves. The prior is generally associated with heuristic optimization procedures, while the latter is more aligned with machine learning techniques.

The EGAL+ system can be situated within the heuristic-based category focusing on the composition of pre-existing tasks, rather than generating novel items from scratch. However, one significant limitation of EGAL+ lies in its reliance on a large and diverse question bank. The effectiveness and robustness of the program are highly contingent upon access to a well-structured and comprehensive repository of questions, which allows the system to leverage its full potential.

Despite the strengths of EGAL+ in configuring and optimizing task selection, the issue of constructing and parameterizing such a question bank still holds challenges in several key areas. To address these to some extent, the author of this thesis proposed a preliminary exploration into automatic question generation from textual knowledge sources, such as textbooks. This approach aims to mitigate the dependency on precompiled question banks and bring the system closer to practical implementation in real-world educational environments.

Accordingly, this chapter presents a hybrid methodology that integrates metaheuristic techniques with LLMs for the purpose of first generating the tasks and then compiling them into assessments as well. This description builds upon the

author's prior work (Láng & Dömsödi 2024) and represents a step toward bridging the gap between theoretical system design and practical application.

7.2. Generative AI Integration for Autonomous Task Generation

While the enhancements introduced by the author have substantially improved the scalability and functionality of EGAL+, as detailed throughout this thesis, they have also highlighted another significant operational challenge that was also part of the program form the initial design: the creation, classification, and maintenance of a sufficiently large and diverse question bank. This challenge poses a serious constraint, as the manual development of such a resource often exceeds the time, effort, and expertise that instructors can reasonably allocate, particularly in resource-limited educational environments.

Consequently, among others, one of the key barriers to the effective deployment of EGAL+ lies in the need for extensive textual task sets derived from source materials such as textbooks. To address this specific limitation to some extent, the author proposed that an automated task generation module powered by artificial intelligence should be integrated into the system. This AI-based module would enable the automatic transformation of knowledge materials from educational texts into diverse assessment items. Once generated, these tasks could be seamlessly incorporated into the system's existing workflows, enabling the dynamic compilation of task sequences and bringing the system closer to offering a fully automated solution for exam creation.

Automated question generation has emerged as a prominent application in the field of natural language processing. Such AQG systems are capable of autonomously producing questions from textual or visual content, typically oriented around specific subject matter or conceptual frameworks. These systems have gained considerable traction in educational technology, and their increasing adoption can be attributed to their adaptability, scalability, and overall effectiveness.

Many of the AQG systems can be generally categorized into two main types: closed-domain and open-domain. Closed-domain AQG systems operate within a narrowly defined subject area, such as medicine or literature, leveraging domain-specific ontologies to ensure the relevance and accuracy of generated questions. In contrast, open-domain AQG systems are designed to function across a broad range of subject areas, relying on more generalized ontologies and linguistic models. Given the broad applicability of EGAL+ and its aim to support a wide range of academic disciplines, the open-domain approach was deemed more appropriate for this research.

Comparative research has evaluated the performance of various LLMs, including but not limited to GPT-3 and GPT-4. These studies assessed factors such as content relevance, grammatical accuracy, and alignment with educational objectives. It was found that some models consistently outperform others in certain areas, while in different use cases, the same systems may exhibit weaknesses (Alnefaie et al., 2023; Tran et al., 2023).

Based on these findings and recognizing the potential benefits of diversity in accessible tools, future iterations of EGAL+ may support the integration of multiple AQG platforms. This would offer users the flexibility to choose among different APIs based on their specific content needs, domain focus, or institutional constraints. But then again this also raises the problem of assisting the user, especially if they may not be an expert in the field, in choosing the best tool for the given scenario or to even automate that decision as well.

For this research it was chosen, that the goal is to demonstrate the feasibility of this hybrid approach of heuristics and LLMs, and not to carry out the research regarding the optimized selection of the best available AI tool for the given use case, so PrepAI, a popular tool accessible at https://prepai.io, was selected for initial integration into the EGAL+ system. PrepAI meets every selection criterion previously outlined, including support for open-domain content, multiple input formats (such as PDF documents), and the ability to generate well-structured tasks

for easy programmatic processing. Its integration serves as a practical illustration of how generative AI can be effectively combined with metaheuristic optimization algorithms to enable the high-volume generation of quality assessment content.

7.3. Practical Implementation

To support seamless functionality between EGAL+ and PrepAI, the integration was designed to operate directly within the EGAL+ user interface. This integration is facilitated through PrepAI's API, enabling users to generate questions without leaving the EGAL+ application environment. In **Figure 4** of this thesis, this operation is represented as "Generate a new question bank."

The integration also ensures that the output file produced by PrepAI is immediately compatible with EGAL+ for further processing and task sequencing. Each entry in the resulting question bank file includes the generated question text, a difficulty rating (ranging from 1 to 2, as assigned by PrepAI), and the corresponding multiple-choice options in the expected formatting. While users are allowed to manually overwrite the difficulty rating, the initial values reflect PrepAI's internal scoring.

During the exam generation process, the user is prompted to input several parameters, like a name for the new question bank, a topic keyword to guide the AI's content focus, the source document (typically a textbook in PDF format), the total number of tasks to be generated for the question bank, and the specific page range within the input file to be considered. Thanks to PrepAI's robust capabilities, the system can process hundreds of pages and generate up to a thousand questions from a single document, making it a powerful asset when used in conjunction with EGAL+, which is well prepared to work with inputs of this magnitude.

To illustrate the effectiveness of this AI-assisted workflow, a demonstration was conducted using PrepAI to generate a 1,000-item question bank from the first 200 pages of Myths and Legends of Ancient Greece and Rome by E. M. Berens (2007).

The topic was selected for its broad inclusion in general education, ensuring the relevance and interpretability of the generated questions across diverse curricula. Although the coexistence preferences applied in this demonstration were selected arbitrarily and not by subject matter experts, they do not impact the system's operational effectiveness.

To simulate more complex difficulty handling within EGAL+, the initial difficulty ratings provided by PrepAI (either 1 or 2) were mapped onto a broader 1-5 scale. Questions initially rated as 1 were randomly reassigned a value within the range of 1-2, and those rated as 2 were reassigned values within the range of 3-5. With these settings 100 task sequences were generated, each containing 30 tasks, and the "medium" difficulty setting was selected from the three difficulty options offered by EGAL+.

The hybrid question generation and task sequencing process produced impressive performance outcomes. Although the author does not claim subject-matter expertise in classical literature, all generated questions were validated in terms of their factual accuracy relative to the source material. As they were based directly on content present in the text, the questions can reasonably be considered valid. The Harmony Search Algorithm implemented within EGAL+ completed the optimization of all 100 sequences in under one second, operating under the same computational parameters detailed in the benchmarking chapter. Furthermore, the system's internal quality enhancement mechanisms improved task sequence fitness by approximately 1.3% over the initial population, despite the fact that this initial population was already fully compliant with all specified constraints. These results highlight the combined power of EGAL+ and generative AI tools in producing high-quality, curriculum-aligned assessments with minimal manual intervention.

For further transparency and reproducibility, the following materials were made publicly available:

- **Generated Question Bank:** https://github.com/balazs-domsodi/EGALPP/blob/main/databank/greece_and_rome.txt

- **Generated Task Sequences:** https://github.com/balazs-domsodi/EGALPP/blob/main/output/tasks.txt

  (Note: These are not formatted in GIFT, instead questions are separated by colons. The GIFT formatting of the output was not yet implemented at the time of this supplementary research.)

- **Initial Population Task Indices:** https://github.com/balazs-domsodi/EGALPP/blob/main/output/initial.txt

- **Enhanced Population Task Indices:** https://github.com/balazs-domsodi/EGALPP/blob/main/output/enhanced.txt

  (Note: The initial and enhanced population files include the aggregated difficulty value at the beginning of the file, and the corresponding fitness scores at the end of each line.)

It is important to note that the use of the PrepAI module in EGAL+ requires users to possess an active PrepAI account. Authentication credentials must be provided via two text files (client_id.txt and client_secret.txt) located in the root directory of the EGAL+ application.

In summary, this chapter demonstrates a highly efficient and scalable method for educators to automate the transformation of textbook content into structured question banks using EGAL+. While the current limitations of generative AI warrant expert review of all AI-generated question sets, this workflow represents a significant step toward minimizing manual workload and transforming EGAL+ from a research prototype into a practical, real-world educational tool. Though not a complete solution to all implementation challenges, this development significantly advances the system's viability and practicality for adoption in assessment-driven learning environments.

8. A CASE STUDY IN LIVE UNIVERSITY EXAM ENVIRONMENT

8.1. Case Study Settings and Overview

From the latter half of the year 2024, the author of this thesis actively participated in a comprehensive empirical research project designed to evaluate the practical functionality and real-world applicability of the EGAL+ system. This investigation marked a significant transition from prior theoretical analyses of EGAL+ to an applied experimental context (Láng, Kovács, & Dömsödi, 2026).

Specifically, the study was aimed at assessing the extent to which the theoretical advantages previously attributed to EGAL+ could be applied within the operational constraints and dynamics of actual university examination settings. The overarching objective was twofold: first, to validate the theoretical findings concerning the EGAL+ system through direct empirical observation; and second, to explore its potential for broader implementation across assessment settings in educational institutions.

The research was structured as a comparative field study embedded within authentic university examination scenarios. Two cohorts of students, matched in size and selected through random assignment, served as the subjects of this investigation. One group was administered conventional examination materials manually composed by academic staff, while the second group received examinations that had been algorithmically generated using the EGAL+ program. This structure allowed for a controlled comparison between traditional assessment methods and those enabled by EGAL+.

The methodological framework employed in the study integrated both quantitative and qualitative data collection strategies. Quantitative analysis focused on several key performance indicators, including student achievement scores, the measured quality of examination sets, and the amount of time faculty members expended in exam preparation. Meanwhile, qualitative insights were obtained

through structured feedback solicited from both instructors and students, addressing their perceptions of exam clarity, fairness, and overall satisfaction.

The findings from this empirical study were both conclusive and promising. Statistical analysis revealed that the utilization of EGAL+ significantly decreased the amount of time required by educators to develop examination materials, without producing any significant effect on student performance outcomes. Furthermore, when evaluated against objective quality metrics, such as the alignment of examination tasks with instructional intentions, the diversity and distribution of tasks, and the uniformity of exam difficulty as perceived by subject-matter experts, EGAL+ was found to consistently generate exams of superior quality compared to those manually created.

To support the transparency and interpretability of these findings, a detailed account of the statistical methodologies applied in the case study is provided in Appendix 1 of this dissertation. Appendix 1 offers a comprehensive overview of the analytical procedures, data processing steps, and evaluation criteria that underpin the reported results, thereby enabling reproducibility and facilitating critical assessment of the study's methodological rigor. It is important to note that while Appendix 1 serves to contextualize and substantiate the outcomes presented here, the statistical analyses themselves were conducted independently and are included solely for explanatory purposes.

These outcomes collectively support the assumption that EGAL+ is not only a viable tool for reducing the workload associated with examination preparation, but also an effective mechanism for enhancing the overall quality of assessments. Importantly, no negative consequences were observed in the deployment of EGAL+, either in terms of student achievement or user satisfaction. Consequently, the study affirms the system's potential for broader adoption in educational contexts, particularly in institutions seeking to optimize efficiency without compromising assessment quality or fairness.

8.2. Detailed Methodology

The study was conducted at Corvinus University of Budapest during the autumn semester of 2024, specifically in the months of November and December, aligning with the institution's scheduled quarterly examination period.

The digital learning management system Moodle served as the platform through which the exams were administered, ensuring consistency in the testing environment across all student groups.

The experimental procedure was implemented across four distinct student groups, each situated in a separate classroom and assessed in different time slots to prevent information exchange and ensure independent participation. A total of 107 undergraduate students, all of whom were enrolled in the fifth semester of a seven-semester Bachelor of Science program in Business Information Systems, were included in the study. The assessments examined within this research were theoretical examinations derived from the Business Intelligence lecture series, which forms a core component of the curriculum.

The empirical study focuses on a single course, which can be viewed as a cohort-based cluster sample drawn from the broader population of students enrolled in the same program at the institution. Under stable admission criteria, selection procedures, and program requirements, successive cohorts are generated through a consistent mechanism, supporting the assumption that observed differences are largely random rather than systematic. A more detailed statistical justification of this sampling approach is provided in Appendix 1.

At the same time, cohort-specific effects (e.g., demographic shifts or external influences) may limit representativeness, and findings may not be directly generalized beyond the given program and institution.

Importantly, the measured efficiency improvements of the EGAL+ approach are not dependent on the specific content of the question bank. This

suggests broader applicability, particularly in course settings characterized by large student numbers and extensive, relatively stable question pools. Future work should therefore focus on expanding data collection across multiple courses and institutions, with targeted selection of such course types to further validate and generalize the findings.

Two separate quarterly exams constituted the subject of the case study. The first exam carried a maximum score of 20 points, while the second allowed students to earn up to 30 points. Each exam was structured to begin with a multiple-choice section, consisting of 10 questions, each valued at one point. These questions offered four response options, only one of which was correct. The multiple-choice section was time-limited to 10 minutes, and students were required to correctly answer at least six questions in order to achieve a passing grade. The content of these MCQs focused on fundamental concepts taught in the course, and students were provided with a predefined list of examinable topics to support their preparation.

Following the multiple-choice section, students were required to respond to open-ended essay questions. In the first exam, students addressed two essay questions, each worth five points, within a 20-minute timeframe. In the second exam, the number of essay questions increased to four, again valued at five points each, with a total allocated time of 30 minutes. These questions were designed to evaluate students' conceptual understanding and their ability to articulate deeper analytical connections within the theoretical content of the course.

To support both the manually created and the algorithmically generated exams, the instructor developed extensive question banks, characterized as follows:

- 138 multiple-choice questions for the first quarterly exam
- 46 essay questions for the first quarterly exam
- 83 multiple-choice questions for the second quarterly exam
- 64 essay questions for the second quarterly exam

These question pools were compiled from two main sources: previously used exam questions and newly generated questions created with the aid of the PrepAI question-generation module, which is integrated within the EGAL+ system as described in a previous chapter of this thesis.

The PrepAI module extracted potential question content from the PDF version of the course's designated textbook. All questions generated by PrepAI underwent a rigorous manual validation process, during which those deemed ambiguous, overly simplistic, irrelevant, or repetitive were discarded.

In preparation for exam generation using EGAL+, the instructor separately populated the Preference Matrix for each quarterly exam to ensure alignment with the thematic structure of the course. The coexistence preference values, used to regulate the likelihood that certain questions would appear together in a test, were manually defined using EGAL+'s built-in coexistence editor tools.

The Business Intelligence course was intentionally selected as the focus of the experiment due to the relative temporal stability of its theoretical content. Given that initializing the Preference Matrix in the programs current state is still a time-intensive process, the course's slow-changing nature ensured that this work would yield long-term benefits, avoiding the need for repeated setup in subsequent semesters. The theoretical scope of the exams emphasized broad concepts and mathematical foundations related to business intelligence and data analysis, as opposed to software-specific or implementation-based knowledge, which is assessed separately through a practical examination component.

In this case it was advantageous that the foundational principles underlying data analytics and machine learning, including the treatment of large language models within the context of natural language processing, remain conceptually consistent over time (Chang et al., 2024).

The student sample was divided into two groups through random assignment. One group, consisting of 53 students, received exams that were

manually constructed by the instructor in accordance with their pedagogical preferences. The remaining 54 students were administered exams generated using EGAL+. Notably, the EGAL+ system enabled the creation of individualized exam sets for each student with no additional effort required from the instructor, a feature that exemplifies one of the tool's practical advantages, as it is an effective mitigation of cheating by answer sharing. Students were unaware of the method by which their exams were created, thereby eliminating any potential bias in their perceptions or performance.

Upon completion of each quarterly exam, students were immediately prompted to rate the perceived difficulty of both the multiple-choice and essay sections on a scale of 1 to 10. At no point were students informed whether their exam sets had been generated manually or by EGAL+, maintaining the blind nature of the study. For comparative purposes, students were also asked to rate the difficulty of their corresponding practical exams using the same scale.

In both the case of EGAL+ generated and the case of manually created exams, the structure and timing of the exams were held constant to ensure validity in comparisons. The time allocations were as follows:

- Multiple-choice section of the first exam: 10 minutes
- Essay section of the first exam: 20 minutes
- Multiple-choice section of the second exam: 10 minutes
- Essay section of the second exam: 30 minutes

The grading methodology also remained consistent across both groups. Moodle's automated grading functionality was employed for the multiple-choice sections, which required a direct match between students' responses and pre-validated correct answers. For the essay responses, grading was conducted manually by instructors using a standardized evaluation guideline. This guideline was distributed to all participants involved in the grading process to promote reliability and consistency in evaluation.

8.3. Results

The results presented in this chapter are based on the empirical evaluation of the EGAL+ system within a controlled case study environment. The findings are discussed at a high level in this chapter, focusing on the most relevant comparative outcomes between manually constructed examinations and those generated with the assistance of EGAL+. A more detailed description of the statistical procedures, including the underlying assumptions, test selection, and computation steps, is provided in Appendix 1 of this dissertation, which serves as a methodological supplement to support reproducibility and enable a more rigorous examination of the analytical framework employed in this study.

The analysis of the case study's results commenced with an investigation into the homogeneity of distribution patterns for three key exam quality metrics: coexistence values, question diversity, and overall fitness scores (which is the sum of the prior two components, as described in a previous chapter of this thesis detailing the implementation of EGAL+). These distributions were compared across exams created manually and those generated with the use of the EGAL+ system, with statistical tests applied to determine whether significant differences existed at commonly accepted significance levels.

The results revealed that, in nearly all cases, the hypothesis of homogeneity in the distributions could be rejected, suggesting statistically significant differences between the EGAL+ generated and the manually constructed exams. One exception to this trend was observed in the coexistence value distributions of the first quarterly essay exams. In this particular instance, the coexistence value distributions for both the EGAL+ generated and the manually created exams were statistically indistinguishable, indicating that EGAL+ did not yield a significant advantage in this metric for that specific case. But nevertheless, the use of EGAL+ still showed significant increase in the diversity component, thus generating exams in better overall quality.

A more nuanced situation emerged in the analysis of the first quarterly multiple-choice exams, where the test results were inconclusive: the significance of the difference between coexistence distributions varied depending on the selected significance threshold. Thus, no definitive statement could be made regarding distributional divergence of the coexistence values in this case. Still in that case the diversity and the overall fitness of the exams were unquestionably higher in the case of the ones generated with EGAL+.

Importantly, in all other cases, where statistically significant differences were observed, the results consistently favored the EGAL+ generated exams. Specifically, the analysis indicated superior performance with respect to all the assessed quality dimensions in those cases.

Summarizing the exam quality metrics, the superiority of the EGAL+ generated exams compared to manually created ones is substantiated by the fact that in terms of overall fitness scores, the generated tests showcased significantly higher results in all cases, as well in terms of the diversity component, and only in two cases regarding the coexistence component were similar results produced.

While it can be noted that it is theoretically possible to match the output quality of EGAL+ through meticulous manual compilation, particularly if ample time and effort are invested, the process would be substantially more resource intensive. The complexity of managing coexistence preferences, ensuring diverse yet pedagogically coherent content, and maintaining balance in task difficulty significantly raises the cost, in terms of instructor time and cognitive load, when attempting to replicate the optimization routines that EGAL+ executes automatically. A comparison in scalability with other assessment generator tools in similar regards could also be a valid investigation, however no such tool was found which allowed for such complex parameterization between tasks, which is the cornerstone added value of EGAL+.

Further analysis also considered subjective indicators of exam difficulty, based on students' post-exam evaluations. These ratings were gathered immediately

following the completion of both quarterly assessments. Students were asked to rate the difficulty of both the multiple-choice and essay components on a standardized scale from 1 to 10. These subjective evaluations revealed a noteworthy pattern: perceived difficulty was relatively consistent across both EGAL+ generated and manually compiled exams. That is, if a student considered one type of exam to be difficult or easy, they tended to perceive the other in a similar way. This suggests that EGAL+ does not introduce unintended shifts in student perception or experience of difficulty, thereby reinforcing the pedagogical neutrality of the system with respect to student experience.

In addition, performance data showed that student achievement scores did not differ significantly between the EGAL+ generated and manually created exam groups at any standard level of statistical significance. This outcome is of particular importance, as it demonstrates that the use of EGAL+ generated exam materials does not impair or artificially inflate student performance outcomes, further validating the reliability of the program.

A qualitative interview with the course leader offered further support for EGAL+ as a valuable instructional tool. According to the instructor, the implementation of EGAL+ significantly reduced the time required for exam preparation: from approximately 2.5 hours for manual exam construction to as little as 15 minutes when using EGAL+. It should be noted that if further exams were conducted in subsequent semester from the same materials, the generation process is likely to drop to mere seconds, as no initial parameterization would be required. This substantial reduction in time investment reflects the tool's capacity to streamline assessment design without compromising quality or fairness.

In conclusion, the results of this study underscore the considerable potential of the EGAL+ system in educational assessment contexts. By improving exam quality across multiple objective dimensions, reducing instructor workload, maintaining stable student performance, and preserving subjective perceptions of fairness and difficulty, EGAL+ emerges as a powerful and practical innovation.

Additionally, its algorithmic structure introduces safeguards that can limit predictability and reduce opportunities for academic dishonesty, further supporting its value as a robust tool for modern educational environments.

It is important to note that the proposed algorithm explicitly optimizes the same metrics used in its evaluation. Consequently, any claims of superior performance along these dimensions are, to some extent, expected and may be considered tautological. To provide a more robust and independent validation of exam quality, a third-party subject matter expert conducted a blind review using criteria that are not directly aligned with the algorithm's optimization objectives. The results of this evaluation are presented in Appendix 3.

9. CONCLUSIONS

9.1. Summary of Key Findings

This dissertation is set out to address challenges associated with the scalable, high-quality generation of educational assessments through the design, refinement, and evaluation of the EGAL+ system. From the beginning of the related research process, the problem was clearly articulated: manual exam creation is both time-consuming and cognitively demanding, particularly when assessment quality, fairness, and multiple-version consistency are non-negotiable requirements.

In this landscape, EGAL+ was envisioned and developed as a solution focused not on generating assessment content per se, but on compiling high-quality, balanced, and customizable exams from existing question banks through metaheuristic optimization.

The literature review established the theoretical and technological foundations upon which this work was built. Existing systems in the field were found to either prioritize content generation, often through large language models and AI-based techniques, or compositional optimization. Among composition-focused systems, EGAL+ introduced a distinctive innovation: the Preference

Matrix, which allows for pairwise control over task inclusion and coexistence relationships, an unparalleled feature in the current landscape. This design choice elevated EGAL+ into a system capable of expressing complex pedagogical intentions with a granularity that no other identified tool offered.

However, such expressivity came at a computational cost. The core contribution of the thesis, therefore, was to redesign EGAL+ to overcome the scalability bottlenecks introduced by its own innovations. The original PHP-based implementation was refactored entirely in C++, enabling a dramatic increase in computational performance and extending the range of feasible use cases from small-scale assessments to large-cohort deployments. This redesign maintained strict optimization criteria: maximizing adherence to user-defined preferences while ensuring task diversity and difficulty equivalence across multiple versions.

Empirical validation through benchmarking confirmed the performance gains of the new implementation, with execution time reductions exceeding 99% in high-scale scenarios. Importantly, these performance improvements were achieved without sacrificing optimization quality. New features such as the advanced difficulty balancing mechanism, entirely conceived and implemented by the author, further enhanced the tool's practical utility.

To extend EGAL+'s capabilities, a hybridized methodology integrating generative AI with metaheuristic optimization was developed and prototyped. By linking EGAL+ to a third-party task generation API, the system demonstrated a credible path toward reducing reliance on precompiled question banks. While this hybrid model does not eliminate the need for human oversight, it offers a scalable foundation for future research into semi or fully automated assessment ecosystems.

The practical effectiveness of EGAL+ was empirically validated in a real university examination setting. Tests generated with EGAL+ consistently outperformed manually composed exams across several dimensions of quality, including structural diversity, alignment with preference constraints, and overall fitness. The collaborating instructor reported significant reductions in exam

preparation time, while students perceived the difficulty and fairness of EGAL+-generated exams as comparable to traditional assessments. No significant differences were found in student achievement scores, confirming the pedagogical neutrality of the system and validating its reliability for real-world educational use.

In summary, EGAL+ represents a significant and original contribution to the field of automated exam composition. It not only responds to a clearly defined research gap but also charts a plausible path toward future developments in the integration of automation into educational workflows.

This thesis additionally highlighted and underscored the authors significant contributions in advancing the broader domain of automated assessment generation through studies related to EGAL+.

9.2. Research Questions Revisited and Answered

1. **What structural and functional limitations exist in current automated exam generation systems?**

EGAL+ addresses a critical and underexplored research gap in automated exam generation by introducing a generalizable framework for scalable, high-precision exam compilation that is capable of expressing virtually any examination concept through its granular Preference Matrix.

Unlike prior work, EGAL+ recognizes that the true determinant of assessment quality lies in the principled selection and arrangement of tasks to meet nuanced pedagogical objectives. The Preference Matrix encodes pairwise coexistence preferences between tasks, alongside the assignable difficulty values, enabling instructors to articulate complex, multidimensional goals, such as balancing cognitive levels, controlling topical coverage, or favoring subtle variations in question type with precision.

This expressive structure, combined with the strict enforcement of total difficulty equivalence across multiple exam variants, ensures structural fairness

while promoting maximum diversity between instances, thereby helping to mitigate academic dishonesty. Through a multi-objective optimization process using the Harmony Search Algorithm, EGAL+ translates these encoded preferences into balanced, distinct, and pedagogically aligned exams, demonstrating that a preference-based model of this granularity can serve as a general-purpose solution for a wide range of educational contexts and exam design philosophies, a capability not previously realized.

2. **How can the trade-off between deep pedagogical parameterization and scalability in EGAL+ be systematically addressed?**

At the time the author began work on EGAL+, several technical and practical factors constrained the system's performance, scalability, and pedagogical flexibility, prompting targeted developments to address each shortcoming. The following are the relevant limitations identified and addressed by the author:

- **Constraints in the Question Bank**

  Earlier limitations on Question Bank and Preference Matrix arguments were eliminated, increasing flexibility and enabling more complex pedagogical intentions to be encoded without artificial constraints.

- **Target difficulty search**

  The author introduced an entirely new advanced target difficulty search mechanism, maintaining precise control over difficulty distribution while keeping the execution time of its improved version.

- **Computational inefficiency of the original implementation**

  The author completely rebuilt EGAL+ in C++, a language selected for its high performance, efficient data abstraction, and minimal overhead. By redesigning every process of the program form the ground up, the author removed all major scalability bottlenecks, transforming EGAL+ from an experimental development concept to an exam generation system applicable for mass deployments.

- **Dependence on precompiled Question Banks**

  The author prototyped a hybrid system integrating generative AI with EGAL+'s metaheuristic optimization, providing a scalable pathway toward semi-automated or fully automated exam generation by minimizing human intervention in the generation of the tasks as well.

Through a meticulous literature review of the domain, the author identified and prioritized the aspects of EGAL+ that should be developed first to transition it from an experimental concept to a real-world applicable solution. These advancements not only resolved the original system's bottlenecks but also positioned EGAL+ as a foundation for future research in automated and semi-automated assessment ecosystems.

3. **How does EGAL+ perform in real-world educational contexts compared to manual exam compilation in terms of quantitative assessment quality metrics and operational efficiency?**

The empirical evaluation conducted as part of this dissertation confirms that EGAL+ became ready for deployment in authentic educational contexts after the integration of the author's work. The system was successfully used to generate exams for real university cohorts, with demonstrated advantages over traditional methods in terms of exam quality and reduced preparation time for instructors.

Key findings include:

- **Substantial reduction in instructor workload (from hours to minutes).**
- **Superior fitness, coexistence and diversity scores of algorithmically generated exams.**
- **Optimization for maximizing diversity in exams mitigates cheating by reducing opportunities for answer copying.**
- **No significant difference in student performance outcomes.**

- **Stable student perceptions of fairness and difficulty.**
- **Reproducible, high-quality output across multiple exam types.**

However, EGAL+ still requires expert oversight in preparing the Question Bank and defining configuration parameters. Its effectiveness remains contingent upon domain expertise and access to appropriate content. Nonetheless, these limitations do not diminish its applicability, they point to areas of development.

4. **Which future research and development directions logically follow for EGAL+, and what do they imply for the field at large?**

The work presented in this dissertation not only advances the technical development of EGAL+ but also highlights several broader trajectories for innovation in the field of automated assessment generation systems.

EGAL+ shows that it's possible to put such fine-grained customization into practice. This suggests that future systems should be able to represent more complex teaching goals. To achieve this, they will need optimization engines that can handle greater variety and interfaces that accept rich input.

Another major finding from this dissertation is that pedagogical sophistication must be matched by scalability. EGAL+'s reimplementation shows that systems with such granularity in parameterization can support large cohorts and multiple test versions. Future work should treat computational performance as foundational, with educational theorists and computer scientists collaborating.

Integrating generative AI into EGAL+ for question creation reveals the promise of hybrid systems that combine content generation, metaheuristic composition, and analytics. Rather than monolithic platforms, modular architectures can enable end-to-end educational ecosystems.

However, it is important to note that the expansion of a question bank through generative artificial intelligence still requires systematic human

professional revision, as current AI systems exhibit well-documented limitations in reliability, validity, and pedagogical alignment. Large language models generate outputs based on probabilistic patterns rather than verified knowledge structures, which makes them inherently susceptible to "hallucinations," i.e., producing factually incorrect yet plausible content (Ji et al., 2023). In the context of assessment design, such inaccuracies can undermine both the validity and reliability of test items, potentially leading to the dissemination of misconceptions among learners.

In addition to factual errors, AI-generated educational content is prone to redundancy, bias, and misalignment with intended learning outcomes. Studies on the use of generative AI in higher education indicate that outputs may reflect biases embedded in training data, lack appropriate difficulty calibration, and fail to meet disciplinary or pedagogical standards (Kasneci et al., 2023; Zawacki-Richter et al., 2019). For question bank development, this creates risks such as ambiguous phrasing, or tasks that do not adequately assess the targeted cognitive level.

Consequently, human expert oversight remains an indispensable component of quality assurance. Human reviewers are required to verify factual correctness, ensure pedagogical appropriateness, and evaluate ethical and contextual suitability. Current research consistently supports hybrid human–AI workflows as the most reliable approach to mitigating errors and maintaining academic standards in AI-assisted content generation (Kasneci et al., 2023).

Although ongoing advances in model architecture and evaluation techniques may reduce these limitations, there is no clear consensus on when (or whether) AI systems will achieve a level of reliability that would justify the removal of human oversight. Therefore, at the present stage of technological development, excluding human professional revision would pose significant methodological and educational risks.

Creating Preference Matrices and setting other parameters for EGAL+ mostly remains manual for now, however automation offers a promising path

toward context-aware configuration, especially considering machine learning models that can interpret curriculum structures, prior assessment data, or learning outcomes. A complementary avenue for more accessible and intuitive configuration can be the development of an advanced graphical user interface.

Finally, EGAL+'s LMS integration shows the importance of interoperability. It signals that future systems should be designed for embeddedness within broader educational ecosystems as well, not as isolated applications.

10. LIST OF REFERENCES

Abd Rahim, T. N. T. et al. (2017). Automated exam question generator using genetic algorithm. In: 2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Miri, Malaysia, 2017, pp. 12-17 https://doi.org/10.1109/IC3E.2017.8409231

Abd Rahim, T. N. T. et al. (2020). Automated Exam Question Set Generator Using Utility Based Agent and Learning Agent. International Journal of Machine Learning and Computing, 10(1), pp. 164-169. https://doi.org/10.18178/ijmlc.2020.10.1.914

Abu Khurma, O., Ali, N., & Hashem, R. (2023). Critical Reflections on ChatGPT in UAE Education: Navigating Equity and Governance for Safe and Effective Use. International Journal of Emerging Technologies in Learning (iJET), 18(14), pp. 188-199. https://doi.org/10.3991/ijet.v18i14.40935

Adamopoulos, P. (2013). What makes a great MOOC? An interdisciplinary analysis of student retention in online courses. In Thirty Fourth International Conference on Information Systems, pp. 1-21.

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2020). Genetic Algorithm: Reviews, Implementations, and Applications. International Journal of Engineering Pedagogy (iJEP), 10(6), pp. 57-77. https://doi.org/10.3991/ijep.v10i6.14567

Al-Betar, M. A., Khader, A. T. & Zaman, M. (2012). University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 42(5), pp. 664-681. https://doi.org/10.1109/TSMCC.2011.2174356

Almeida, J. J., Araujo, I., Brito, I., Carvalho, N., Machado, G. J., Pereira, R. M. S., & Smirnov, G. (2013a). Math exercise generation and smart assessment. 2013 8th Iberian Conference on Information Systems & Technologies (CISTI), pp. 1-6.

Almeida, J. J., Araujo, I., Brito, I., Carvalho, N., Machado, G. J., Pereira, R. M. S., & Smirnov, G. (2013b). PASSAROLA: High-order exercise generation system. 2013 8th Iberian Conference on Information Systems & Technologies (CISTI), pp. 1-5.

Alnefaie, S.S.M., Atwell, E. & Alsalka, M.A. (2023) Using Automatic Question Generation Web Services Tools to Build a Quran Question-and-Answer Dataset. International Journal on Islamic Applications in Computer Science And Technology, 11 (2), pp. 1-12.

Bachiri, Y.-A., & Mouncif, H. (2023). Artificial Intelligence System in Aid of Pedagogical Engineering for Knowledge Assessment on MOOC Platforms: Open EdX and Moodle. International Journal of Emerging Technologies in Learning (iJET), 18(05), pp. 144-160. https://doi.org/10.3991/ijet.v18i05.36589

Bakoyannis, G. (2020). *Nonparametric tests for transition probabilities in nonhomogeneous Markov processes*. Journal of Nonparametric Statistics, 32(1), 131–156. https://doi.org/10.1080/10485252.2019.1705298

Basse, A., Diatta, B., & Ouya, S. (2021). Ontology-Based System for Automatic SQL Exercises Generation. Internet of Things, Infrastructures and Mobile Applications, pp. 738-749. https://doi.org/10.1007/978-3-030-49932-7_69

Bayoud, H. A. (2021). *Tests of normality: New test and comparative study. Communications in Statistics-Simulation and Computation*, 50(12), 4442–4463. https://doi.org/10.1080/03610918.2019.1643883

Békés, G., & Kézdi, G. (2021). *Data analysis for business, economics, and policy*. Cambridge University Press.

Berens, E. M. (2007). Myths and Legends of Ancient Greece and Rome

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., S. Yu, P., Yang, Q., Xie, X. (2024). A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 15(3), pp. 1-45. https://doi.org/10.1145/3641289

Chen, X., Yue, X.-G., Li, R. Y. M., Zhumadillayeva, A., & Liu, R. (2021). Design and Application of an Improved Genetic Algorithm to a Class Scheduling System. International Journal of Emerging Technologies in Learning (iJET), 16(01), pp. 44-59. https://doi.org/10.3991/ijet.v16i01.18225

Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. Expert Systems with Applications, 40(11), pp. 4715-4729. https://doi.org/10.1016/j.eswa.2013.02.007

Chung, C. Y., Hsiao, I. H., & Lin, Y. L. (2022). AI-assisted programming question generation: Constructing semantic networks of programming knowledge by local knowledge graph and abstract syntax tree. Journal of Research on Technology in Education, 55(1), pp. 94-110. https://doi.org/10.1080/15391523.2022.2123872

Ciguené, R., Joiron, C., & Dequen, G. (2019). Automatically generating assessment tests within higher education context thanks to genetic approach. In E. G. Talbi & A. Nakib (Eds.), Bioinspired heuristics for optimization. studies in computational intelligence (Vol. 774), pp. 269-282. https://doi.org/10.1007/978-3-319-95104-1_17

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Routledge. https://doi.org/10.4324/9780203771587

Dorigo, M., Colorni, A., & Maniezzo, V. (1991). Distributed Optimization by Ant Colonies. Proceedings of the First European Conference on Artificial Life, pp. 134-142.

Eryiğit, G., Bektaş, F., Ali, U., & Dereli, B. (2021). Gamification of complex morphology learning: the case of Turkish. Computer Assisted Language Learning, 36(8), 1421-1449. https://doi.org/10.1080/09588221.2021.1996396

Farida, B. D., Malik, S. M., Catherine, C., & Pierre Jean, C. (2011). Adaptive Exercises Generation Using an Automated Evaluation and a Domain Ontology: The ODALA+ Approach. International Journal of Emerging Technologies in Learning (iJET), 6(2), pp. 4-10. https://doi.org/10.3991/ijet.v6i2.1562

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). Artificial Intelligence Through Simulated Evolution

Frey, B. B. (Ed.). (2018). *The SAGE encyclopedia of educational research, measurement, and evaluation*. Sage Publications. https://psych.wisc.edu/Brauer/BrauerLab/wp-content/uploads/2014/04/Murrar-Brauer-2018-MM-ANOVA.pdf

Geem, Z. W., Kim, J. H., & Loganathan G. V. (2001) A New Heuristic Optimization Algorithm: Harmony Search. SIMULATION. 76(2), pp. 60-68. https://doi.org/10.1177/003754970107600201

Glover, F. (1977). Heuristics for integer programming using surrogate constraints, Decision Sciences, 8(1), pp. 156-166. https://doi.org/10.1111/j.1540-5915.1977.tb01074.x

Habeb Al-Obaydi, L., Pikhart, M., & Klimova, B. (2023). ChatGPT and the General Concepts of Education: Can Artificial Intelligence-Driven Chatbots Support the Process of Language Learning?. International Journal of Emerging Technologies in Learning (iJET), 18(21), pp. 39-50. https://doi.org/10.3991/ijet.v18i21.42593

Heilala, J., Shibani, A., & Gomes de Freitas, A. (2023). The Requirements for Heutagogical Attunement within STEAM Education. International Journal of Emerging Technologies in Learning (iJET), 18(16), pp. 19-35. https://doi.org/10.3991/ijet.v18i16.42313

Hnida, M., Khalidi Idrissi, M., & Bennani, S. (2018). Automatic Composition of Instructional Units in Virtual Learning Environments. International Journal of Emerging Technologies in Learning (iJET), 13(06), pp. 86-100. https://doi.org/10.3991/ijet.v13i06.8107

Holland, J. H. (1975) Adaptation in Natural and Artificial Systems

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., … Fung, P. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys, 55*(12), 1–38. https://doi.org/10.1145/3571730

Kasneci, E., Seßler, K., Küchemann, S. et al. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and individual differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., … Kasneci, G. (2023). ChatGPT for good? On opportunities and

challenges of large language models for education. *Learning and Individual Differences, 103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kirkpatrick, S., Gelatt, C., & Vecchi, M. (1983). Optimization by Simulated Annealing, Science, 220(4598), pp. 671-680. http://doi.org/10.1126/science.220.4598.671

Láng, B. & Kardkovács, T. Zs. (2016). Solving exercise generation problems by diversity oriented meta-heuristics, In: Shuang, Cang; Yan, Wang (eds.) SKIMA: 2016 10th International Conference on Software, Knowledge, Information Management & Applications: University of Information Technology, December 15- 17 2016, Chengdu, China, pp. 49-54, https://doi.org/10.1109/SKIMA.2016.7916196

Láng, B. (2019). Solving Exercise Generation Problems Using the Improved EGAL Metaheuristic Algorithm. SEFBIS Journal, 13, pp. 23-31.

Láng, B. (2020). Solving Exercise Generation Problems Using the Improved EGAL Metaheuristic Algorithm with Precedence Constraints. In: Auer, M., Hortsch, H., Sethakul, P. (eds) The Impact of the 4th Industrial Revolution on Engineering Education. ICL 2019. Advances in Intelligent Systems and Computing, 1135. https://doi.org/10.1007/978-3-030-40271-6_56

Lee, K. S., Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. Computers & Structures, 82(9–10), pp. 781-798. https://doi.org/10.1016/j.compstruc.2004.01.002

Lee, K. S., Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering, 194(36–38), pp. 3902-3933. https://doi.org/10.1016/j.cma.2004.09.007

Lo, C. K. (2023). What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. Education Sciences, 13(4), 410. https://doi.org/10.3390/educsci13040410

Long, J. S., & Trivedi, P. K. (1992). Some specification tests for the linear regression model. Sociological Methods & Research, 21(2), 161–204.

Malafeev, A. (2014). Language Exercise Generation: Emulating Cambridge Open Cloze. International Journal of Conceptual Structures and Smart Applications (IJCSSA), 2(2), pp. 20-35. https://doi.org/10.4018/IJCSSA.2014070102

Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M. N., Salcedo-Sanz, S., & Geem, Z. W. (2013). A survey on applications of the harmony search algorithm. Engineering Applications of Artificial Intelligence, 26(8), pp. 1818-1831. https://doi.org/10.1016/j.engappai.2013.05.008

Marozzi, M. (2013). *Nonparametric simultaneous tests for location and scale testing: A comparison of several methods*. Communications in Statistics-Simulation and Computation, 42(6), 1298–1317. https://doi.org/10.1080/03610918.2012.665546

Mehta, S., & Smetannikov, I. (2021). Finding the Blank with Sequence Labeling for English Learning. Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System, pp. 191-195. https://doi.org/10.1145/3437802.3437834

Monge, M. (2023). *Two-sample Kolmogorov-Smirnov tests as causality tests: A narrative of Latin American inflation from 2020 to 2022*. Revista Chilena de Economía y Sociedad, 7(1), 68–78. https://rches.utem.cl/?p=2374

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

Ngo, S. T., Jaafar, J. B., Aziz, I. A., Nguyen, G. H., & Bui, A. N. (2021). Genetic Algorithm for Solving Multi-Objective Optimization in Examination Timetabling Problem. International Journal of Emerging Technologies in Learning (iJET), 16(11), pp. 4-24. https://doi.org/10.3991/ijet.v16i11.21017

Ngo, T. T. A. (2023). The Perception by University Students of the Use of ChatGPT in Education. International Journal of Emerging Technologies in Learning (iJET), 18(17), pp. 4-19. https://doi.org/10.3991/ijet.v18i17.39019

Olney, A. M. (2023). Generating multiple choice questions from a textbook: LLMs match human performance on most metrics. Workshop on Empowering Education with LLMs - the Next-Gen Interface and Content Generation at the AIED'23 Conference

Pandraju, S., & Mahalingam, S. G. (2021). Answer-Aware Question Generation from Tabular and Textual Data using T5. International Journal of Emerging Technologies in Learning (iJET), 16(18), pp. 256-267. https://doi.org/10.3991/ijet.v16i18.25121

Pilán, I., Volodina, E., & Borin, L. (2017). Candidate sentence selection for language learning exercises: from a comprehensive framework to an empirical evaluation. arXiv preprint, arXiv:1706.03530. https://doi.org/10.48550/arXiv.1706.03530

Popescu, D.A., Stanciu, G.C., Nijloveanu, D. (2023). Application of Genetic Algorithm in the Generation of Exam Tests. Balas, In: V.E., Jain, L.C., Balas, M.M., Baleanu, D. (eds) Soft Computing Applications. SOFA 2020. Advances in Intelligent Systems and Computing, 1438. https://doi.org/10.1007/978-3-031-23636-5_12

Rao, P., Kiranmai, T., Samhitha, E., Shiva, R., & Kusuma, S. (2022). A Survey on Automated Assessment Questions Generation System Using Supervised Algorithms. In International Journal for Research in Applied Science and

Engineering Technology, 10, pp. 1675-1677. https://doi.org/10.22214/ijraset.2022.47700

Schröer, G., & Trenkler, D. (1995). *Exact and randomization distributions of Kolmogorov-Smirnov tests two or three samples*. Computational Statistics & Data Analysis, 20(2), 185–202. https://doi.org/10.1016/0167-9473(94)00040-P

Shanthi, B. S. A., Harshitha, L. J. R., & Manasa, K. (2019). Automated exam question generator using genetic algorithm. International Research Journal of Engineering and Technology (IRJET), pp. 1687-1691.

Sörensen, K., Sevaux, M., & Glover, F. (2018). A History of Metaheuristics. In: Handbook of Heuristics, pp. 791-808. https://doi.org/10.1007/978-3-319-07124-4_4

Stroustrup, B. (2013). The C++ programming language

Sukstrienwong, A. (2017). A Genetic-algorithm Approach for Balancing Learning Styles and Academic Attributes in Heterogeneous Grouping of Students. International Journal of Emerging Technologies in Learning (iJET), 12(03), pp. 4-25. https://doi.org/10.3991/ijet.v12i03.5803

Teo, N. H. I., Bakar, N. A., & Karim, S. (2012). Designing GA-based auto-generator of examination questions. 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, pp. 60-64. https://doi.org/10.1109/EMS.2012.69

Tran, A., Angelikas, K., Rama, E., Okechukwu, C., Smith, D. H., & MacNeil, S. (2023). Generating Multiple Choice Questions for Computing Courses Using Large Language Models. 2023 IEEE Frontiers in Education Conference (FIE), pp. 1-8. https://doi.org/10.1109/FIE58773.2023.10342898

Viehmann, T. (2021). *Numerically more stable computation of the p-values for the two-sample Kolmogorov-Smirnov test* (arXiv:2102.0803). arXiv. https://doi.org/10.48550/arXiv.2102.08037

Wang, Y., & Wang, X. (2022). Test Paper Automatic Generating Method Based on Hybrid Genetic Algorithm. In Genetic and Evolutionary Computing, pp. 599-609. https://doi.org/10.1007/978-981-16-8430-2_54

Wooldridge, J. M. (2016). *Introductory econometrics: A modern approach*. Nelson Education.

Wu, Z., He, T., Mao, C., & Huang, C. (2020). Exam paper generation based on performance prediction of student group. Information Sciences, 532, pp. 72-90. https://doi.org/10.1016/j.ins.2020.04.043

Yang, X. S. (2009). Harmony Search as a Metaheuristic Algorithm. In: Music-Inspired Harmony Search Algorithm: Theory and Applications (Editor Geem, Z. W.), Studies in Computational Intelligence, 191, pp. 1-14. https://doi.org/10.1007/978-3-642-00185-7

Zanetti, A., Volodina, E., & Graën, J. (2021). Automatic Generation of Exercises for Second Language Learning from Parallel Corpus Data. International Journal of TESOL Studies, 3(2), pp. 55-70. https://doi.org/10.46451/ijts.2021.06.05

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators?. *International Journal of Educational Technology in Higher Education, 16*(1), 39. https://doi.org/10.1186/s41239-019-0171-0

Zhou, C., Lin, L., & Shuai, P. (2018). Design of auto-generating examination paper algorithm based on hybrid genetic algorithm. 2018 IEEE 3rd

International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 563-567. https://doi.org/10.1109/ICCCBDA.2018.8386579

11. AUTHOR'S PUBLICATIONS ON THE TOPIC

Láng, B., & Dömsödi, B. (2022). Development of the Improved Exercise Generation Metaheuristic Algorithm EGAL+ for End Users. *International Journal of Emerging Technologies in Learning (iJET)*, *17*(11), pp. 210–224. https://doi.org/10.3991/ijet.v17i11.28099

Dömsödi, B., & Láng, B. (2022). Improvements of the EGAL+ metaheuristic algorithm for solving exercise generation problems with a significantly larger number of elements. In: OGIK'2022: Országos Gazdaságinformatikai Konferencia, pp. 55-56.

Dömsödi, B., & Láng, B. (2023). Exploring Hybrid Solutions in Educational Software: The Integration of AI and Metaheuristics. In: OGIK'2023: Országos Gazdaságinformatikai Konferencia, pp. 15.

Láng, B., & Dömsödi, B. (2024). Integration of AI and Metaheuristics in Educational Software: A Hybrid Approach to Exercise Generation. *International Journal of Emerging Technologies in Learning (iJET)*, *19*(06), pp. 38–51. https://doi.org/10.3991/ijet.v19i06.49829

Dömsödi, B., Láng, B., & Kovács, L. (2024). Integrating AI and Metaheuristics in Educational Software: A Case Study in Live University Exam Settings. In: OGIK'2024: XX. Országos Gazdaságinformatikai Konferencia: Az előadások összefoglalói, pp. 53.

Dömsödi, B., Láng, B., & Kovács, L. (2025, November 13). Innovative educational systems through the fusion of metaheuristics and AI [Conference presentation]. Day of Hungarian Science, Budapest University of Economics and Business, section: Interdisciplinary approaches to addressing the opportunities and

challenges posed by digitalization and artificial intelligence. https://webapi.uni-bge.hu/api/v1/files/display/documents/bueb-section-brochure-2025.pdf#page=6

Láng, B., Kovács, L., & Dömsödi, B. (2026). AI-Enhanced Exam Generator Program: A Case Study in Live University Exam Settings. *Central-European Journal of New Technologies in Research, Education and Practice*, *8*(1), 1–24. https://doi.org/10.36427/CEJNTREP.8.1.12403

12. APPENDIX 1: STATISTICAL ANALYSES OF CASE STUDY

12.1. Disclaimer of Authorship

The material presented in this appendix is included to provide essential methodological background for the statistical analysis underlying the case study discussed in this dissertation. While the forthcoming publication arising from this case study is a joint work of three authors, it is important to clearly distinguish the individual contributions.

The author of this dissertation, who is also a co-author of the forthcoming publication, was primarily responsible for the development of the algorithm employed in the study, as well as for designing and implementing the solutions that rendered the program suitable for application in the given case study. However, the author did not participate in the statistical analysis of the data collected.

The statistical analysis presented in this appendix was conducted entirely by co-author László Kovács. Accordingly, the contents of this appendix are not the original work of the dissertation author and should be fully attributed to László Kovács.

This appendix is included for the purpose of transparently documenting the analytical methods applied in the case study, thereby supporting the interpretation of the results related to the author's developments and ensuring the reproducibility of the findings, as discussed in the associated manuscript (Láng, Kovács, & Dömsödi, 2026).

12.2. Research Context and Research Design

The experiment was conducted at Corvinus University of Budapest during the fall semester of 2024, specifically in November and December, when quarterly exams were held. Students used the Moodle platform for the exams. The tests were administered to four groups of students across four classrooms at different time

slots. Altogether, 107 BSc students enrolled in the fifth semester of a seven-semester business information systems program participated.

A single year cohort of students enrolled in the same business information systems program may be treated as a cluster sample from the population of students enrolled in the program in the same institution. Cluster sampling can be considered appropriate when naturally occurring groups (clusters) exhibit internal heterogeneity while remaining comparable across clusters. The validity of using this cohort as a representative sample of the broader student population rests on the assumption of structural equivalence across cohorts.

Specifically, if admission criteria, selection procedures, and program requirements remain stable over time, then each cohort is generated through the same underlying selection mechanism. This consistency implies that each yearly intake approximates a realization from the same population distribution. Consequently, differences between cohorts are expected to be random rather than systematic. Hence statistical methods and tests can be applied to draw conclusions on the population of unobserved students as well.

However, it is important to acknowledge limitations. External factors such as demographic trends, policy changes, or shifts in labor market attractiveness of the program could introduce cohort effects, potentially undermining representativeness. Furthermore, this cluster sampling method does not make the results generalizable to students of other programs and other institutions. Therefore, the authors aim to tackle this limitation in their future work, extending the test of EGAL+ to several programs in multiple universities.

The exams included in the research were theoretical tests from the business intelligence lectures. The first quarterly exam was worth a total of 20 points, while the second was worth 30 points. Both exams featured an initial section comprising 10 MCQs, each worth one point, with four answer options per question, of which exactly one was correct. Students had 10 minutes to answer these 10 questions. To pass the exam, students needed to answer at least six questions correctly; otherwise,

the exam resulted in a failing grade. These 10 MCQs covered the fundamental concepts of the course, for which students received a predefined list in advance for their preparation.

Following the multiple-choice section, students tackled essay questions. In the first quarterly exam, there were two essay questions, each worth five points, to be answered within 20 minutes. These questions aimed to test the students' understanding of the deeper relationships within the course material. In the second quarterly exam, students answered four essay questions, each worth five points, within a 30-minute timeframe.

In Moodle, the instructor prepared the following question banks:

- Multiple-choice questions for the first quarterly exam (138 questions);
- Essay questions for the first quarterly exam (46 questions);
- MCQs for the second quarterly exam (83 questions);
- Essay questions for the second quarterly exam (64 questions).

These question banks were partially compiled from questions from previous exams and partially generated using the PrepAI question-generation tool from a PDF version of the textbook assigned as mandatory reading for the course. Questions generated by PrepAI were manually reviewed to filter out overly simple, ambiguous, repetitive, or irrelevant ones.

The preference matrix for the two quarterly exams was filled separately by the instructor to align with the thematic areas of the course. The material covered in the first quarter encompasses the role of business intelligence in enterprises, its technological solutions, and analytical tools, focusing on the following topics:

- Decision support;
- Data analysis environments in enterprises;
- Data warehouses;
- Data types;

- Data quality issues;
- Data standards;
- Online analytical processing; dimensions and fact data;
- Data visualization;
- Big data management.

Within these topics, the co-occurrence of questions is generally not preferred (preference value of 0). However, when a topic is sufficiently broad, slight preferences (values of 1–3) may be permitted. For instance, the topic of data analysis environments in enterprises is quite broad, encompassing various analytical tools applied across corporate functions (hence the preference value of 3 within the topic). Conversely, questions on dimensions and fact data categorization involve only two definitions addressed in different contexts (hence, a preference value of 0 within the topic).

Between different topics, preferences are generally fully allowed (a preference value of 10), with exceptions in specific cases. For example, data quality issues are often managed using data standards, and addressing these issues is essential during data warehouse construction. Therefore, there are overlaps in the concepts across these topics, so their joint occurrence is preferred less. Data warehouses are part of the enterprise data analysis environment, so these topics are also less preferred together. The data type of variables determines how these variables and their relationships are visualized, which connects to the logic of classifying dimensions and fact data. Consequently, these three topics are also not maximally preferred together. **Table 5** presents the preference matrix, reflecting these principles.

	Data analysis environments	Data standards	Data quality issues	Data types	Data visualization	Big data	Online analytical processing	Decision support	Data warehouses
Data analysis environments	3	10	10	10	10	10	10	10	5
Data standards	10	2	4	10	10	10	10	10	3
Data quality issues	10	4	1	10	10	10	10	10	3
Data types	10	10	10	1	4	10	6	10	10
Data visualization	10	10	10	4	1	10	8	10	10
Big data	10	10	10	10	10	0	10	10	10
Online analytical processing	10	10	10	6	8	10	0	10	10
Decision support	10	10	10	10	10	10	10	2	10
Data warehouses	5	3	3	10	10	10	10	10	1

Table 5. Preference matrix for the exam in the first quarter; Source: Láng, Kovács, & Dömsödi (2026)

The material for the second-quarter exam covers selected chapters on data mining and machine learning:

- Cross-Industry Standard Process for Data Mining (CRISP) data mining process
- General framework of machine learning
- Supervised machine learning
- Unsupervised machine learning
- Natural language processing

The determination of preference values within the topics follows the same principles as for the first-quarter exam, based on the broadness of the topic. For example, questions within the CRISP topic exclusively focus on the various steps of the CRISP-DM process, conceptually constituting a single element; therefore,

the within-topic co-occurrence preference is set to 0. Conversely, the topic of natural language processing spans a wide range of areas, from simple tokenization to sentiment analysis and topic modeling, allowing for a higher within-topic preference value of 4. The preference matrix, designed based on these principles, is presented in **Table 6**.

	CRISP	Supervised machine learning	General machine learning	Unsupervised machine learning	Natural language processing
CRISP	0	10	10	10	10
Supervised machine learning	10	1	5	4	8
General machine learning	10	5	3	5	10
Unsupervised machine learning	10	4	5	1	8
Natural language processing	10	8	10	8	4

Table 6. Preference matrix for the second-quarter exam; Source: Láng, Kovács, & Dömsödi (2026)

This business intelligence course was chosen as the subject of our study because we required material that changes only minimally over time, given that populating the preference matrix is a time-intensive task. The benefits of using the EGAL+ software are most evident when this work needs to be done only once instead of being repeated each semester. The theoretical content of the two quarterly exams focuses on the general concepts and mathematical principles of business intelligence and data analysis rather than the specific workings and software implementations of algorithms. The latter areas are assessed during the course's practical exam. The general principles of data analysis and machine learning are considered temporally stable. Even incorporating large language models into the curriculum (under the natural language processing topic) required no drastic revisions to the theoretical content, as these models can be treated as supervised learning algorithms conceptually (Chang et al., 2024). Additionally, this course is

mandatory and taken by a large number of students every year, allowing the instructor to benefit from the software's advantages over several years.

Of the 107 students who took the exams, 53 were randomly assigned to receive traditionally prepared exams. In these cases, the instructor manually designed the exams to reflect their preferences. The remaining 54 students were given test sets generated using EGAL+. For these students, each received a unique test set, as generating individualized sets using the tool required no additional effort from the instructor. Students were not informed before the exam about how their tasks were generated. At the end of both quarterly exams, immediately after completing their exercises, students rated the difficulty of the multiple-choice and essay sections of the theoretical exam on a scale from 1 to 10, unaware of whether their test sets were EGAL+ generated or traditionally prepared. The difficulty of the practical exams was also rated on a scale from 1 to 10 by the students for reference.

The traditional and EGAL+-generated exams were identical in all aspects except for the generation method:

- Both were created from the same question bank;
- The exams were designed to have equivalent overall difficulty, with individual questions rated on a 1–5 difficulty scale. Summing up these difficulty ratings by individual exams yielded the same result for every student, regardless of whether they received traditional or EGAL+-generated exams. The total difficulties were as follows:
    - First-quarter multiple-choice questions: 28;
    - First-quarter essay questions: 7;
    - Second-quarter MCQs: 28;
    - Second-quarter essay questions: 14;
- Students were also allotted the same amount of time in all cases:
    - First-quarter MCQs: 10 minutes;
    - First-quarter essay questions: 20 minutes;

- Second-quarter MCQs: 10 minutes;
- Second-quarter essay questions: 30 minutes;

The sole difference was that, in the second case, the EGAL+ software was used to select questions based on the preference matrix, while in the first case, the instructor manually selected questions, which are as follows:

- MCQs were organized into 10 separate sub-question banks based on difficulty and topic. Students received one randomly selected question from each sub-bank in both multiple-choice sections;
- Essay questions were organized into two sub-banks for the first-quarter exam and four sub-banks for the second-quarter exam, also based on difficulty and topic. Students received one randomly selected question from the appropriate sub-bank for each essay section;

Grading was conducted identically for both groups. The Moodle system automatically evaluated the MCQs by checking if the students' responses matched the pre-determined correct answers. For the essay questions, instructors utilized a standardized grading guide provided to all grading instructors. An example of this guide is provided in the appendices. Notably, the uniqueness of the exams effectively eliminated the possibility of academic dishonesty through the online sharing of answers among students.

12.3. Data Collection

The experiment was conducted at Corvinus University of Budapest during the fall semester of 2024, specifically in November and December, when quarterly exams were held. Students used the Moodle platform for the exams. The tests were administered to four groups of students across four classrooms at different time slots. Altogether, 107 BSc students enrolled in the fifth semester of a seven-semester business information systems program participated.

In the research context defined in Subsection 3.3., the following variables were collected in three separate datasets on the students who took the exams. In all the variables, the following indices are applied:

- i: student id = $\{1,2, \dots ,107\}$;
- j: exam number = $\{1st\ quarter, 2nd\ quarter\} = \{1,2\}$;
- k: exam type = $\{EGAL+, Manual\} = \{E, M\}$;

First, the objective qualities of the individual student exams were evaluated using the components of formula (2):

- CMC_{ijk}: coexistence value of the multiple-choice exam for student i of type k in quarter j;
- CE_{ijk}: coexistence value of the essay exam for student i of type k in quarter j;
- DMC_{ijk}: diversity value of the multiple-choice exam for student i of type k in quarter j;
- DE_{ijk}: diversity value of the essay exam for student i of type k in quarter j;
- FMC_{ijk}: fitness value of the multiple-choice exam for student i of type k in quarter j;
- FE_{ijk}: fitness value of the essay exam for student i of type k in quarter j;

The second dataset contains the subjective student perceptions of the exams, which was assessed by an anonymous student survey where students were asked to judge the difficulty of each exam item on a scale of 1–5, where 1 means very easy and 5 means very difficult. This survey data cannot be connected to any of our data from other sources as they are completely anonymous. Therefore, the k index cannot be defined for these survey variables.

- PMC_{ij}: difficulty perception from 1 to 5 of the multiple-choice exam for student i in quarter j;

- PE_{ij}: difficulty perception from 1 to 5 of the essay exam for student i in quarter j;
- PP_{ij}: difficulty perception from 1 to 5 of the practical exam for student i in quarter j;

For the survey, 45 responses were received in quarter 1 and 39 in quarter 2. Thus, we have that $n_1 = 45$ and $n_2 = 39$.

To assess how the exam generation types (EGAL+ or Manual) affected the student scores, we obtained a third dataset containing the student scores of each exam, the binary variable indicating whether they took an EGAL+ or a Manual exam, and two control variables for individual student's abilities. Note that we do not need to define the k index for these variables as the exam type is expressed by a separate binary variable in this dataset.

- SMC_{ij}: performance score of the multiple-choice exam for student i in quarter j
- SE_{ij}: performance score of the essay exam for student i in quarter j
- $ST_{ij} = SMC_{ij} + SE_{ij}$: total performance score of the exam for student i in quarter j
- $Type_{ij}$: 1 if student i took an EGAL exam in quarter j, 0 if they took a manual exam
- $Retake_{ij}$: 1 if student i has failed the course in a previous semester, 0 otherwise; the value is the same $\forall j$
- $Time_{ij}$: in a two-week timeframe before the exam, how many times did student i access the course materials on Moodle in quarter j

The control variable $Time_{ij}$ had the limitation that it could not register if more students were preparing for the exam together using the same device. Therefore, this variable likely underestimated the preparation time of some students. However, as course material download was prohibited on Moodle, the

variable was not distorted by students preparing offline using their downloaded materials.

In addition to tabular data collection, the business intelligence course leader was interviewed regarding the workload associated with manual and EGAL+-supported exam generation. The course leader has held this position since 2021 and has supervised the exam generation process for four academic years (2021/22, 2022/23, 2023/24, and 2024/25). They maintain a question bank consisting of 138 MCQs and 46 essay questions for the exam in the first quarter, as well as 83 MCQs and 38 essay questions for the exam in the second quarter. During the academic years from 2021/22 to 2023/24, the course leader manually created sub-question banks from these questions according to the principles outlined in section 3.4. Since these exams are administered during seminars, students take the exams in four different time slots; consequently, the course leader needed to create different versions of these sub-question banks for each seminar to minimize redundancy arising from randomly selected questions. According to the course leader, preparing the exams takes an average of 2.5 hours each year. However, by generating individual exams separately for each student with the support of EGAL+, this time is reduced to approximately 15 minutes. Furthermore, the course leader noted that controlling the sub-question banks for coexistence, diversity, and difficulty is a manual and somewhat subjective process, resulting in exams that may be less effective in these respects compared to those generated by EGAL+, where these aspects are controlled directly and objectively through the algorithm's fitness function.

12.4. Statistical Methodology

The experiment was conducted at Corvinus University of Budapest during the fall semester of 2024, specifically

First, the objective qualities of the individual student exams were compared through the homogeneity of their distributions. We expected that the qualities of the EGAL+ and manual exams were not significantly different. If they were, we

anticipated that the EGAL+ type would show a significantly larger proportion of exams with higher quality in coexistence, diversity, and overall fitness. The homogeneity of the exam quality measure distributions among the EGAL+ and manual types was tested using the two-sample Kolmogorov–Smirnov (KS) test, as proposed by Monge (2023), Viehmann (2021), and Bakoyannis (2020). The KS test statistic was applied to each element of the variable set $X = \{CMC, CE, DMC, DE, FMC, FE\}$ as follows:

$$D = \sup_{x} \left| F_{XEj}(u) - F_{XMj}(u) \right|$$

where F_{XEj} and F_{XMj} are empirical distribution functions of a given variable in the set X for the E (EGAL+) and M (Manual) types, respectively, in quarter j. The observed samples had sizes of $n_E = 55$ and $n_M = 52$ respectively for the EGAL+ (E) and manual (M) types. The sample sizes were adequate to achieve sufficient statistical power for the tests (Marozzi, 2013). The test statistic and the p-values based on the Kolmogorov distribution for each comparison were obtained via the ks.test function in the R language, using the algorithm proposed by Schröer & Trenkler (1995) and applying numerical improvements as suggested by Viehmann (2021).

Subjective student preconceptions of the exams were compared by examining whether the distribution of ratings for the multiple-choice and essay exams was independent of the ratings for the practical exams. Since the practical exams did not utilize EGAL+ in any way, it could be assumed that if the subjective ratings of the practical and the other two exam types show a strong level of association, then student perceptions are not influenced by EGAL+, as they rated exams with and without EGAL+ similarly.

The level of association between the two variables in set $Z = \{PMC, PE\}$ and PP was measured by the χ^2 statistic and Cramér's V as proposed by Cohen (2013) and Frey (2018). The following formulas were calculated for each variable in Z separately:

$$\chi^2_{Z,PP} = \sum_{l=1}^{5}\sum_{m=1}^{5} \frac{f^2_{Z_l,PP_m}}{f_{Z_l} \cdot f_{PP_m}} - 1 \quad (3)$$

$$V_{Z,PP} = \sqrt{\frac{\chi^2_{Z,PP}}{n \cdot \min\{\max(l) - 1\,, \max(m) - 1\}}} \quad (4)$$

In the formulas, f_{Z_l,PP_m} is the joint frequency of observations that take value l in the variable currently examined from set Z and simultaneously take the value m in the PP, while f_{Z_l}, and f_{PP_m} are the marginal frequencies for the values l and m in variables $Z = \{PMC, PE\}$and PP, respectively. As variables PMC, PE, PP are all measured on a scale from l to 5, both l and m range from 1 to 5. Therefore, the $\min\{\max(l) - 1\,, \max(m) - 1\}$ part in the $V_{Z,PP}$ formula provided by Frey (2018) simplifies to $5 - 1$, and n denotes the total number of observations. As the $V_{Z,PP}$ measures are calculated for both quarters 1 and 2, we have $n_1 = 45$ and $n_2 = 39$, respectively.

The effect of the exam generation types (EGAL+ or manual) on student scores was modeled using ordinary least squares (OLS) regression. Each variable in the set $Y = \{SMC, SE, ST\}$ was applied as a target variable in a multivariate regression with $Type, Reatake, Time$ being the feature variables of the models. Our regression model is defined as follows.

$$Y_{ij} = c_j + \alpha_j Type_{ij} + \beta_j Retake_{ij} + \gamma_j Time_{ij} + e_{ij} \quad (5)$$

With this model, we can capture the effects of the $Type, Reatake, Time$ variables on the three different exam scores separately for each quarter j. The main variable of interest is $Type$, and it was assumed that its α_j coefficients are not significantly different from 0 in any of the fitted regressions, as this would indicate that the exam generation type has no statistically significant effect on the exam scores of students. The role of the $Reatake, Time$ variables are to control for individual student abilities and efforts taken while preparing for the exams. Moreover, it was assumed that students who needed to retake the course would have

significantly lower scores, while students who spent more time preparing for the exams were expected to achieve significantly higher scores. These phenomena should not distort the effect of the $Type$ variable on exam scores due to the random classification of students into EGAL+ and manual groups. However, including these variables in our models eliminated any possible confounding effects from the α_j coefficients caused by different individual student characteristics (Békés & Kézdi, 2021).

The e_{ij} residual term of the models was assumed to be normally distributed, homoscedastic, and serially uncorrelated. Homoscedasticity of the residuals was tested using the White and Breusch–Pagan tests, while serial correlation was examined via the Breusch–Godfrey test, as suggested by Wooldridge (2016). Further, the normality of the residuals was checked using the Jarque–Bera test (Bayoud, 2021). Testing whether our linear model specification is adequate was done using Ramsey's RESET test, as proposed by Long & Trivedi (1992) and Wooldridge (2016).

12.5. Results

First, the homogeneity of the coexistence, diversity, and overall fitness score distributions was examined between the manual and EGAL+ groups using the KS test. **Table 7** presents the test statistics and p-values derived from the Kolmogorov distribution.

Quality Type	Exam	D Statistic	p-value	Significance
Fitness	Quarter 1 - Essay	1.00000	0.00000	Yes on all levels
Diversity	Quarter 1 - Essay	1.00000	0.00000	Yes on all levels
Coexistence	Quarter 1 - Essay	0.13131	0.78692	No on all levels
Fitness	Quarter 1 - Multiple-choice	1.00000	0.00000	Yes on all levels
Diversity	Quarter 1 - Multiple-choice	1.00000	0.00000	Yes on all levels
Coexistence	Quarter 1 - Multiple-choice	0.30553	0.01430	Yes on 5%, but not on 1%
Fitness	Quarter 2 - Essay	0.98000	0.00000	Yes on all levels
Diversity	Quarter 2 - Essay	0.98000	0.00000	Yes on all levels
Coexistence	Quarter 2 - Essay	0.61091	0.00000	Yes on all levels
Fitness	Quarter 2 - Multiple-choice	0.78431	0.00000	Yes on all levels
Diversity	Quarter 2 - Multiple-choice	0.74510	0.00000	Yes on all levels
Coexistence	Quarter 2 - Multiple-choice	0.51408	0.00000	Yes on all levels

Table 7. Test statistics and p-values derived from the Kolmogorov distribution; Source: Láng, Kovács, & Dömsödi (2026)

Our results indicate that the null hypothesis of distribution homogeneity can be rejected at all common significance levels for nearly every exam, except for the coexistence levels of the first quarter's essay exams. In this latter case, we can state that at all levels there are no significant differences in the coexistence distribution of the manual and EGAL+ exams. However, for the first quarter's multiple-choice exam, we cannot make a straightforward decision regarding significant differences in the coexistence distribution between the manual and EGAL+ exams, as the decision depends on the significance level.

Figure 13 examines the empirical density functions of these objective exam quality measures. We found that in every case where these distributions are significantly different at all levels, the difference favors the EGAL+ exams. The EGAL+ values are concentrated at the higher end of the distribution compared to the manual exam values.

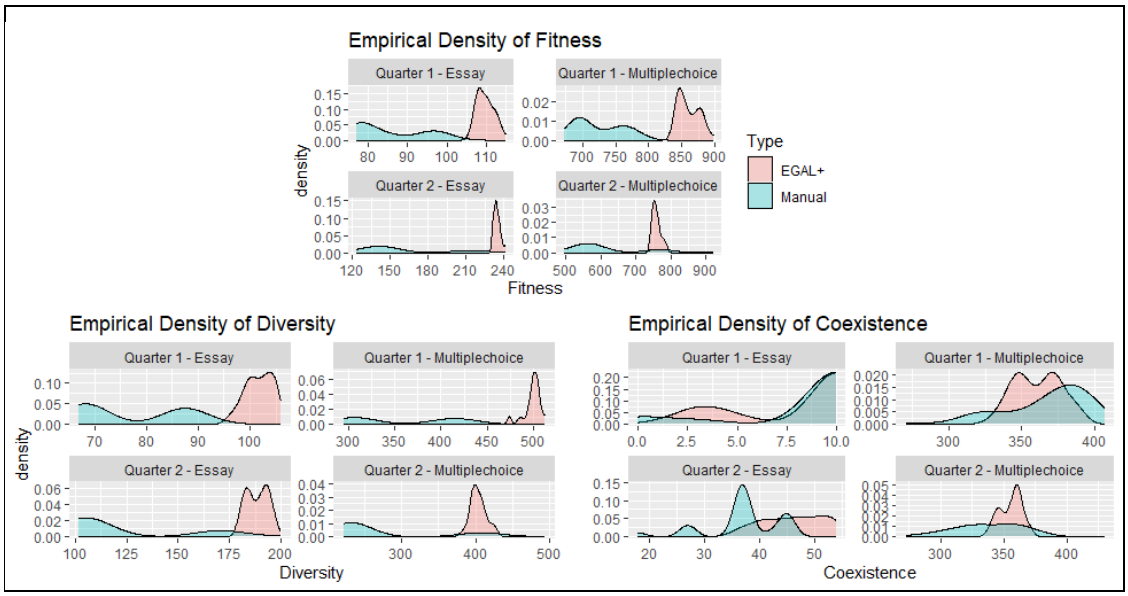

Fig. 13. Empirical density functions of fitness, diversity, and coexistence; Source: Láng, Kovács, & Dömsödi (2026)

The only case where the manual exam's objective qualities show concentration at the higher end of the distribution compared to the EGAL+ exams is in the two cases where the differences in the distribution are not considered significant at all levels (the coexistence values of the first quarter's multiple-choice and essay exams). This is further confirmed by the fact that the overall fitness of the EGAL+ exams is significantly higher in these cases. Overall, it can be concluded that the objective qualities of the exams are significantly higher for the EGAL+ compared to the Manual exams in all aspects, except for the first quarter's coexistence levels, where the distributions of EGAL+ exams are not significantly different from those of the manual exams at all levels.

Cramér's V coefficients measuring the consistency of subjective student perceptions between the multiple-choice, essay, and practical exams are provided in **Table 8**.

Cramér's V	Quarter 1	Quarter 2
	Practical	Practical
Essay	0.657	0.710
Multiple-choice	0.690	0.717

Table 8. Association results of subjective student opinions; Source: Láng, Kovács, & Dömsödi (2026)

The V coefficients indicate that the consistency between the EGAL+-affected essay and multiple-choice exams and the practical exam, which was unaffected by EGAL+, is around 0.7, representing the boundary between strong and moderate consistency (Frey, 2018). Overall, it can be concluded that subjective student performance is relatively consistent between the EGAL+-affected and unaffected exams. This indicates that if a student perceives an exam unaffected by EGAL+ as easy or difficult, they generally tend to perceive the EGAL+-affected exam similarly. This tendency is evident even for the first quarter's essay and practical exams, where the consistency is at its lowest (0.657). The stacked column chart in **Figure 14** shows that students who rated the essay exam (affected by EGAL+) as high or low in difficulty were also given a high/low score for the practical exam (unaffected by EGAL+) similarly in large proportions.

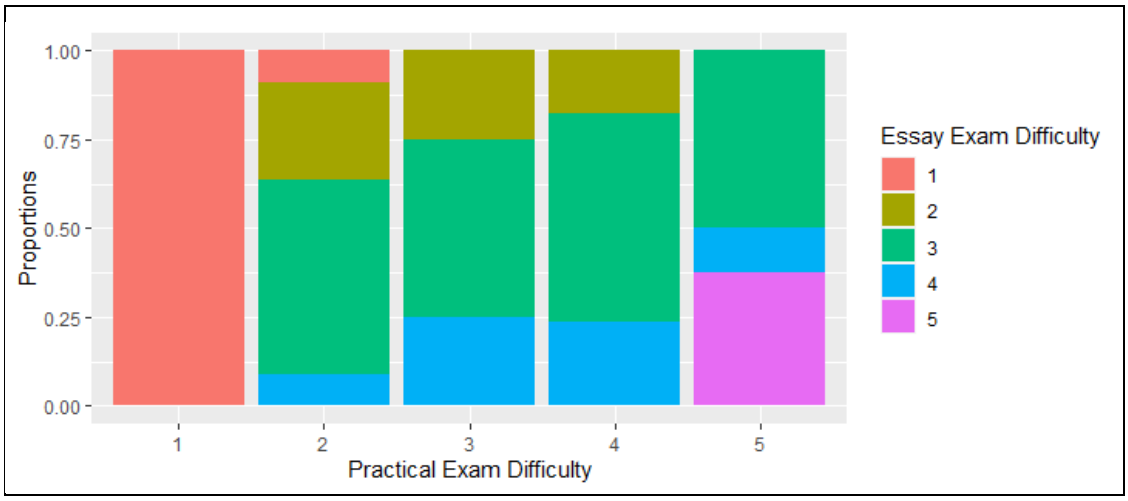

Fig. 14. Consistency between practical and exam difficulty ratings in the first quarter; Source: Láng, Kovács, & Dömsödi (2026)

The tendency illustrated in **Figure 14** is consistent across the remaining three pairs and can be considered even stronger as Cramér's V is higher in these cases.

Estimated OLS regression coefficients for the models defined in Equation 5 are presented in **Table 9**.

	Multiple-choice - Quarter 1		Essay – Quarter 1		Multiple-choice - Quarter 2		Essay – Quarter 2	
	Coefficient	**p-value**	**Coefficient**	**p-value**	**Coefficient**	**p-value**	**Coefficient**	**p-value**
Intercept	6.470	0.00%	4.988	0.00%	7.970	0.00%	14.658	0.00%
Type = EGAL+	0.125	75.99%	0.685	43.39%	-0.157	70.56%	1.207	22.77%
Retake = Yes	-1.143	19.25%	-3.118	9.63%	-0.761	38.77%	-4.086	5.59%
Time	0.017	6.35%	0.043	2.84%	0.020	16.70%	0.023	51.24%
R^2	8.9%		13.3%		4.0%		8.1%	
Global F-test (p-value)	0.1172		0.0284		0.4552		0.1461	
White test (p-value)	0.8430		0.4963		0.9584		0.9410	
Breusch-Pagan (p-value)	0.6275		0.1467		0.7658		0.7819	
Breusch-Godfrey (p-value)	0.8858		0.4046		0.5734		0.7962	
Jarque-Bera (p-value)	0.1077		0.5211		0.2082		0.3904	
RESET (p-value)	0.8503		0.8365		0.5655		0.3935	

Table 9. OLS regression summary; Source: Láng, Kovács, & Dömsödi (2026)

Examining the coefficient estimates and the p-values of their partial t-tests, we concluded that none of the α_j coefficients are significant at any common significance levels, indicating that EGAL+ does not influence student performance scores, even if we control for individual student abilities and motivation with the $Retake$ and Time variables.

Overall, the models exhibit weak in-sample fit and mostly insignificant population fit with R^2 values below 10% and global F-test p-values exceeding all common significance levels, suggesting that model explanatory power is not significantly different from 0 in most cases (Wooldridge, 2016). The only exception is the model where the target variable is the essay performance score in the first quarter, where we have significant explanatory power at the 5% level but not at 1%. The R^2 shows a moderate in-sample fit of 13.3% explanatory power. Therefore,

this model alone demonstrates some weak-moderate explanatory power and significant feature effects on some common significance levels. Here, it can be stated that we have the $Time$, and $Retake$ variables have some significant effect on student performance. $Time$ is significant at 5% and $Retake$ is significant only at 10%. However, the exam type (EGAL+ or manual) does not show a significant level of student performance scores even in this best-performing model, confirming it does not influence student performance at all.

Based on their coefficients, we can say that if we take two students with the same exam type (EGAL+ or manual) and retake (yes or no) values, then the student who opened the study materials on Moodle one more time is expected to score 0.043 points more on the essay exam in the first quarter. For the retake coefficient, we can interpret that if we take two students with the same exam type (EGAL+ or manual) and the same time spent studying on Moodle, then the student who already failed the course before is expected to score 3.118 points lower on the essay exam in the first quarter. The $Time$ variable has a similar effect on the multiple-choice scores in the first quarter as well, although only at a 5% level, while the $Retake$ variable is significant only at the 10% level. However, in these cases, the model itself is not significant at any common significance levels based on the p-values from the F-tests.

All four models are diagnostically appropriate. The e_{ij} residuals are homoscedastic and normally distributed at all common significance levels based on the p-values of the White + Breusch–Pagan, Breusch–Godfrey, and Jarque–Bera tests, respectively. Ramsey's RESET test indicates that model specification is appropriate at all common significance levels, with p-values above 10% in all models. This suggests that no nonlinear terms are needed to model student performance.

13. APPENDIX 2: APPLICATION OF COEXISTENCE PREFERENCE MODIFICATION

13.1. Overview

A precise understanding of the coexistence preference modification process can be most effectively conveyed through an example grounded in a widely understandable domain. For this purpose, a question bank in basic mathematics is considered.

The structure of the question bank adheres strictly to the required format: each row begins with the question text, followed by a difficulty value, then a sequence of coexistence preferences (with respect to previously listed tasks), and finally the answer options, with the correct answer appearing first. The constructed question bank is shown in **Figure 15**.

```
2+2=?   1   10  4   3   5
5+3=?   1   2       8   7   6
9-4=?   1   2;2 5   6   3
3*3=?   2   8;8;8   9   6   12
8/2=?   2   8;8;8;2 4   2   6
x+2=5   3   6;6;6;7;7   3   2   4
2x=6    3   6;6;6;7;7;2 3   2   6
1/2+1/2=?   4   4;4;4;6;6;7;7   1   1/2 2
3/4-1/4=?   4   4;4;4;6;6;7;7;2 1/2 1/4 3/4
A shop sells 3 apples for 6 dollars. What is the price of 1 apple?  5   3;3;3;5;5;6;6;7;7   2   1   3
```

Fig. 15. Question Bank for demonstration purposes; Source: Author

In this configuration, coexistence values have been deliberately structured to achieve two simultaneous goals: first, to discourage multiple questions of the same type from appearing together, and second, to encourage the inclusion of different conceptual categories within a single exam. This is accomplished by assigning low coexistence values (typically 2) between tasks of the same type, such as the two addition problems or the two equation problems. In contrast, higher values (6-8) are assigned between tasks belonging to different categories, such as arithmetic and algebra. Moderate values (3-5) are used when the conceptual

distance is smaller but still meaningful, such as between arithmetic and word problems.

The underlying reasoning is pedagogical. If two tasks assess nearly identical skills, their coexistence is discouraged in order to avoid redundancy and to maximize coverage of the curriculum. Conversely, tasks that represent different cognitive processes are encouraged to co-occur, thereby producing balanced and comprehensive exams. In effect, the coexistence matrix encodes a refined grouping strategy: instead of explicitly forbidding duplicates, it probabilistically discourages them while promoting diversity.

13.2. Demonstration of the Modification Process

In the first step, a group is defined consisting of the basic arithmetic tasks (tasks 1–3). The following input is provided:

- Group of questions: 1-3
- Joint inclusion preference: 1

Upon execution, the program iterates through the selected lines and examines their coexistence values with respect to each other. Whenever a value exceeds the specified threshold (1), it is reduced to 1. As a result, the coexistence values among these three tasks become uniformly low. The effect of this modification is that it becomes highly unlikely for more than one basic arithmetic question to appear in the same exam. This is desirable because it enforces diversity at the most fundamental level, ensuring that simple operations do not dominate the assessment.

In the second step, a group is defined for the algebraic equation tasks (tasks 6–7), with the following input:

- Group of questions: 6-7
- Joint inclusion preference: 1

The same mechanism is applied: coexistence values between these two tasks are reduced accordingly.

In the third step, a group is defined for the fraction tasks (tasks 8–9):

- Group of questions: 8-9
- Joint inclusion preference: 1

Again, coexistence values are capped at 1 within this group.

After these intra-group modifications, the program proceeds to handle inter-group relationships. Since the defined groups do not overlap, the user is prompted to specify coexistence preferences between groups. Consider the following input:

- Between group 1 (tasks 1–3) and group 2 (tasks 6–7): 6
- Between group 1 (tasks 1–3) and group 3 (tasks 8–9): 5
- Between group 2 (tasks 6–7) and group 3 (tasks 8–9): 7

For each pair of groups, the program scans the relevant lines and updates coexistence values between tasks belonging to different groups, again applying a capping mechanism. If any value exceeds the specified threshold, it is reduced accordingly.

The resulting question bank after the aforementioned modifications can be seen in **Figure 16**.

```
2+2=?   1   1   4
5+3=?   1   1   8
9-4=?   1   1;1 5
3*3=?   2   8;8;8   9   6   12
8/2=?   2   8;8;8;2 4   2   6
x+2=5   3   6;6;6;7;7   3
2x=6    3   6;6;6;7;7;1 3
1/2+1/2=?   4   4;4;4;6;6;7;7   1
3/4-1/4=?   4   4;4;4;6;6;7;7;1 1/2
A shop sells 3 apples for 6 dollars. What is the price of 1 apple?  5   3;3;3;5;5;6;6;7;7   2   1   3
```

Fig. 16. Question Bank after modifications for demonstration purposes; Source: Author

By assigning moderately high coexistence values between different conceptual groups, the system actively encourages their joint inclusion. For instance, arithmetic and algebra are allowed to co-occur with relatively high probability, ensuring that exams contain both foundational and intermediate-level tasks.

This two-stage modification process, first within groups, then between groups, achieves a refined balance. Redundancy within categories is minimized, while diversity across categories is preserved and even encouraged. The resulting coexistence structure does not enforce rigid rules but instead shapes the solution space in a probabilistic manner, guiding the exam generation algorithm toward desirable outcomes.

13.3. Known Limitations and Future Directions

As an early implementation certain limitations remain. The reliance on manual group definition may become impractical for large-scale question banks, as the absence of semantic understanding means that relationships must be inferred and encoded by the user rather than automatically detected.

Future improvements could address these limitations by introducing automated grouping techniques based on similarity measures or machine learning approaches that could reduce the burden on the user. Moreover, incorporating data-driven methods, such as analyzing student performance, could allow coexistence preferences to evolve dynamically. Visualization tools could provide insight into the coexistence matrix, facilitating more informed and efficient modifications, and a graphical user interface may significantly improve the tool’s accessibility.

14. APPENDIX 3: INDEPENDENT EXPERT EVALUATION OF EXAM QUALITY

14.1. Context and Methodology of the Blind Expert Review

The following section presents a blind review conducted by an independent subject matter expert teaching the same course at Corvinus University of Budapest. As part of the blind review protocol, the evaluator was not informed about the underlying differences between the two sets of exam tasks. However, references in the review to "individually assigned task sets" may be interpreted as corresponding to exams generated by the EGAL+ system. The review is presented in full, with the sole modification being its translation from Hungarian into English by the author of this dissertation.

14.2. Blind Expert Review

Regarding the essay questions in the first midterm exam, I often found the individually assigned task sets to lack sufficiently practice-oriented questions. While several sets did include practically focused items, in multiple cases the two theoretical questions merely required the definition of a single concept each. These can be answered by students through memorization alone. However, given that the course itself is strongly practice-oriented, it would be important to ensure that each student encounters questions that assess genuine understanding, requiring not only recall, but also the ability to recognize and apply learned material in the context of a specific business problem. In contrast, the three randomly assigned essay task sets each contain at least one, and typically two, questions that evaluate student knowledge through practical examples.

With respect to the multiple-choice questions in the first midterm exam, both sets of tasks are of high quality: they comprehensively cover the course material, and each set includes questions based on practical examples. An advantage of the individually assigned question sets is that there were very few instances of repetitive question types (e.g., multiple true/false items, several

questions related to data visualization, or multiple items addressing the same type of variable within a single set). As such, this collection of task sets could provide a more varied and balanced assessment during the midterm exams.

My observations regarding the essay questions in the second midterm exam are similar to those made for the first: in some instances, the individually assigned task sets also lacked practice-oriented questions. At the same time, it is evident that a significantly higher proportion of these sets include one or even two open-ended questions based on practical examples. The four questions posed within each individual set are diverse and cover the relevant topics and therefore appear sufficient for assessing students' knowledge. The three randomly assigned sets, however, include six questions, among which some address the same topic multiple times, potentially leading to an overly detailed assessment of a student's proficiency in a given area within the constraints of a midterm exam.

In terms of the quality of the multiple-choice task sets in the second midterm exam, it is most difficult to distinguish between "good" and "very good." The questions are varied, of high standard, and cover all topics, thereby encompassing the full course material. Although the individually assigned question sets may feel somewhat less unique, since it appears that certain questions recur across different sets, they would still provide a more diverse assessment compared to repeatedly assigning the same three task sets to different groups of students.

14.3. Interpretation of the Expert Review

The blind expert evaluation provides a nuanced perspective on the comparative quality of EGAL+-generated and manually constructed exam sets. Overall, the review does not indicate a clear and systematic superiority of one approach over the other in terms of general quality. Both types of exam sets are described as high-quality, comprehensive, and capable of covering the relevant course material effectively.

That said, certain tendencies can be identified. The EGAL+-generated (i.e., “individually assigned”) task sets appear to offer greater diversity in question types and reduce redundancy within a single exam. This is particularly evident in the multiple-choice sections, where the expert highlights a lower incidence of repetitive question formats and a broader coverage of different assessment angles. Such variation may contribute to a more balanced and comprehensive evaluation of student knowledge.

On the other hand, the manually constructed exam sets, particularly those based on a smaller number of pre-defined variants, demonstrate strengths in ensuring the inclusion of practice-oriented, application-focused questions, especially in essay formats. In some instances, the EGAL+-generated sets were noted to rely more heavily on theoretically oriented questions that could be addressed through memorization, although this issue was not pervasive and appeared to diminish in later exam iterations. Also, it is important to note that such tendencies can be handled through the reassignment of values in the Preference Matrix.

Importantly, the expert review suggests that neither approach introduces substantial deficiencies in exam quality. Instead, the differences observed are primarily related to emphasis: EGAL+ tends to enhance variability and reduce repetition, while traditional exam design may provide more consistent integration of applied, practice-oriented assessment elements. Taken together, these findings indicate that EGAL+-generated exams achieve a level of quality comparable to that of manually created exams, without clear evidence of systematic inferiority.