Teaching Programming in Higher Education: Analyzing Trends, Technologies, and Pedagogical Approaches Through a Bibliometric Lens

Mariuxi Vinueza-Morales[1], Jorge Rodas-Silva[2], Cristian Vidal-Silva*[3]
Faculty of Sciences and Engineering, Universidad Estatal de Milagro, Milagro, Ecuador[1]
Director of Innovation in Academic Processes, SofTech Research Group, Universidad Estatal de Milagro, Milagro, Ecuador[2]
Facultad de Ingeniería y Negocios, Universidad de Las Américas, Manuel Montt 948, Providencia, 7500975, Santiago, Chile[3]

***Abstract*—In today's information society, developing programming competencies is essential in higher education. Numerous studies have been conducted on effective strategies for fostering these skills. This study performs a bibliometric analysis of research on teaching strategies for programming in higher education, using data from the SCOPUS and Web of Science (WOS) databases between 2014 and 2023. The analysis identifies key trends, influential authors, and collaboration networks in this field. The most effective teaching strategies include project-based learning, flipped classrooms, and collaborative programming. Emerging technologies such as augmented reality and virtual reality are gaining prominence in programming education. Despite the growth of research in this area, challenges remain, such as the lack of longitudinal studies exploring the long-term impact of these methodologies and the need for greater geographic diversity in studies. This paper emphasizes the importance of exploring new technologies and interdisciplinary approaches and fostering international collaborations to enhance programming education. The findings guide researchers and educators on how to optimize programming learning in a global context.**

***Keywords*—Programming; higher education; teaching strategies; bibliometrics**

I. Introduction

In the digital era, programming has become an essential competency in the higher education curriculum [1], especially in Science, Technology, Engineering, and Mathematics (STEM) disciplines [2]. While programming experience in higher education was traditionally confined to technical or engineering fields, today, programming permeates diverse disciplines, recognizing its potential as both a technical tool and a critical thinking skill [3], [4]. Effective acquisition of programming skills requires not only familiarity with syntax and data structures but also deep logical understanding and problem-solving abilities [5].

Teaching and learning strategies for programming in higher education extend beyond simple knowledge transmission techniques. They instill a computational mindset, promote logical thinking, and help students address complex problems systematically [6]. As technology advances, these strategies must evolve to keep pace with the changing demands of the programming field and students' needs, ensuring long-term educational outcomes [7]. For example, Bloom's taxonomy classifies and describes different levels of learning achievement that students can reach [8], [9]. Project-based learning (PBL) and pair programming also encourage collaboration and critical thinking [10], [11]. These strategies emphasize the practical application of knowledge and solving real-world problems, allowing students to consolidate and contextualize what they have learned in realistic settings [12].

As highlighted by Sun et al. [13], adaptability is essential in programming. Tools, languages, and methodologies continuously change and evolve [14]. Therefore, teaching strategies must not only transmit technical knowledge, but also teach students to become autonomous and adaptable learners [15]. This implies fostering skills such as self-learning, curiosity for new technologies, and resilience to overcome challenges [16]. Adaptive learning would cultivate logical and creative thinkers who can innovate and adapt in a constantly changing field like programming education [17]. The extensive literature on programming teaching and learning strategies reflects the growing importance and recognition of these strategies in both educational and professional realms [18]. Researchers, educators, and professionals worldwide have contributed numerous studies, theories, and methodologies, enriching the body of available knowledge [19], [20], [21], [22]. However, the vast amount of information on diverse programming education approaches, along with the fast pace of new developments, presents challenges in staying up to date and identifying the most impactful trends and practices.

The rapid evolution of programming education and the increasing volume of research publications pose a challenge in identifying effective teaching and learning strategies [23]. This research explores publication patterns, the most cited sources, academic collaboration networks, and other relevant aspects to shed light on the current state of research in the programming education at the higher education level. Thus, this document seeks to address the need for a comprehensive understanding of current trends, significant contributors, and prominent research in higher education programming education. Furthermore, our work aims to recognize the most influential authors, leading research centers, and areas that require more attention in programming education.

This study predominantly analyzes data from specific geographic regions, which can limit its generalizability to other educational contexts, particularly in underrepresented or developing countries. Future research should incorporate data from a broader range of regions to provide more globally

*Corresponding authors.

representative insights. This analysis relies exclusively on data from SCOPUS and Web of Science. Although these databases provide extensive coverage of high-quality research, including additional sources, such as local publications or databases not indexed on these platforms, could enhance the study's comprehensiveness.

A. Research Questions

This project is a quasi-experimental quantitative study designed to answer the following research questions:

- RQ1: What are the predominant trends in programming teaching and learning strategies in higher education? This article addresses this question by analyzing research from SCOPUS and WOS databases published between 2014 and 2023, highlighting frequently cited approaches such as project-based learning, flipped classrooms, and collaborative programming. The analysis focuses on the evolution of these strategies and their impact on programming competency development, acknowledging that relevant works in non-indexed conferences and journals may further complement the identified trend.
- RQ2: Who are the most influential authors, and what are the seminal publications shaping the field of programming education? This article responds by identifying the most cited authors and works from SCOPUS and WOS between 2014 and 2023. The analysis considers author networks, citation impact, and recurring themes in key publications, recognizing that other influential contributors may also emerge from non-indexed sources.
- RQ3: What journals and institutions make the most significant contributions to programming education research? This article explores this question by examining the journals and institutions with the highest SCOPUS and WOS production and citation metrics between 2014 and 2023. The review highlights institutions consistently contributing to shaping discourse on programming education, acknowledging valuable works published in non-indexed platforms.

II. Background

Computational thinking plays a critical role in the modern digital era, providing individuals with essential problem-solving skills that transcend the boundaries of computer science and programming, as noted by Lu et al. [24]. Based on principles from computer science, mathematics, and logic [25], this approach enables individuals to break down complex problems, recognize patterns, and design algorithmic solutions [26]. As highlighted by Shen et al. [27], computational thinking significantly influences daily life by helping individuals make informed decisions and solve problems efficiently. Whether optimizing daily routines or critically evaluating online information, this skill set empowers individuals to navigate the complexities of the digital world [28]. Furthermore, when incorporated into educational curricula, it fosters essential competencies such as logical reasoning and creativity, preparing students for future challenges [29]. Algorithm 1 presents an algorithm describing the steps to master new topics through computational thinking [30].

Algorithm 1 Procedure for Mastering a New Topic

Require: Topic is identified
Ensure: Relevant materials are gathered
  Obtain an initial understanding
  Define the boundaries of the topic
  Search for appropriate resources
  Develop a structured learning approach
  Set criteria for successful learning
  while *learning not achieved* **do**
    Refine the selected resources
    Reevaluate and explore the materials
    Experiment with the information
    Implement newly acquired knowledge ▷ If feasible
    Share or explain learned concepts ▷ If feasible
  end while

Beyond everyday applications, the benefits of computational thinking extend to a wide range of disciplines [25], [31]. Shin et al. [32] demonstrate how this competency enhances scientific research by helping scientists analyze complex data, simulate experiments, and develop better models to understand the natural world. Computational thinking supports research in fields such as biology, physics, and social sciences, improving decision-making by providing powerful tools for data-driven insights [33]. With the growing demand for digital literacy in the workforce, computational thinking equips individuals with the necessary tools to thrive in an evolving job market [34]. This competency empowers individuals not only as students or professionals but also as active participants in a technology-driven society.

A. Programming Competencies in Higher Education

Programming competencies hold critical importance in higher education, particularly within computer science and across Science, Technology, Engineering, Arts, and Mathematics (STEAM) disciplines [35]. The ability to think algorithmically, solve complex problems systematically, and develop automated solutions through coding has become an indispensable skill set for students, regardless of their field of study [36]. Developing programming competencies in higher education equips students with fundamental skills that go beyond coding:

1) Algorithmic thinking: This allows students to break down intricate problems into smaller, manageable tasks while constructing logical sequences of steps to address them [24].

2) Problem-solving through programming: Programming fosters creativity and resilience, pushing students to iteratively refine their solutions until achieving the most effective outcome [37].

3) Abstract thinking: Students can conceptualize real-world problems as abstract models, facilitating a deeper understanding of complex phenomena across various disciplines [38].

4) Automation and efficiency: Programming enables students to streamline repetitive tasks, enhancing both productivity and efficiency [39].

The impact of programming competencies transcends computer science, offering substantial benefits in STEAM disciplines [35], [40]. These skills contribute to both the technical and creative dimensions of each field.

- Science: Programming is a powerful tool for analyzing large datasets, modeling complex systems, and simulating experiments. In fields like biology, physics, and chemistry, the ability to automate data processing and perform statistical analyses enhances the precision and speed of scientific discoveries.
- Technology: Programming drives a deeper understanding of technological systems, empowering students to innovate and develop new software tools.
- Engineering: Programming is indispensable in engineering, whether used to model and simulate physical systems or optimize design processes. Coding equips students with tools that improve accuracy and efficiency in the mechanical, civil and electrical engineering disciplines.
- Art: In the arts, programming opens new avenues for digital creativity.
- Mathematics: Programming enhances mathematical problem-solving by allowing the simulation of mathematical models and solving large-scale calculations that would otherwise be impossible through manual methods.

III. Methodology and Research Design

The increasing volume of academic publications and the proliferation of research streams can make it challenging for researchers to stay updated within a specific field. Systematic literature reviews synthesize available scientific information and help identify areas of uncertainty where further investigation is required [41]. Typically, literature review research addresses a single scientific database, which limits the ability to gain a comprehensive view of knowledge and trends within a specific domain. Therefore, some authors argue for the necessity of using multiple databases [42]. In this context, the data for the present study were extracted from the Web of Science (WOS) and SCOPUS databases.

Using both WOS and SCOPUS for the bibliometric analysis of teaching and learning strategies for higher education programming ensures a comprehensive and high quality coverage of the relevant academic literature. Both databases are renowned for their extensive indexes of peer-reviewed journals, conference proceedings, and other scholarly outputs across disciplines, including computer science and education [43]. The international scope of these databases [44] ensures a global perspective on teaching methodologies, crucial for understanding the varied approaches to programming education in higher education. Both WOS and SCOPUS have significant prestige and are recognized for improving the robustness of analysis by cross-referencing data, ensuring completeness, and minimizing potential bias in the literature review [45]. Thus, the choice of SCOPUS and WOS allows for a comprehensive examination of publication patterns in the field of programming education. To achieve this, records from both sources were merged into a single dataset. Table I summarizes the search criteria, while Fig. 1 shows the publication trends over the period of time (2014-2023).

In terms of coverage, WOS encompasses a broader range than SCOPUS, with 1,464 records compared to 361. After comparing and removing duplicates, the process identified 1,697 documents related to teaching and learning strategies in programming education, revealing that approximately 11% of the documents overlap or share similarities.

A. Evaluation Instruments

As highlighted by Chen et al. [46] and Trinidad et al. [47], analyzing publication trends provides researchers with a deeper understanding of the scientific landscape, allowing them to identify emerging knowledge areas and contribute to the advancement of their fields. To support this type of analysis, the bibliometric study used the Bibliometrix platform using the R programming language. Bibliometrix is an open-source toolset that offers flexibility by integrating with various statistical packages. This adaptability makes Bibliometrix particularly valuable for exploring evolving research areas and uncovering new trends in programming education strategies.

One of the key strengths of Bibliometrix is its free access and ease of use through a web-based interface, which encapsulates its core capabilities and establishes a framework for real-time data analysis [48]. For instance, Biblioshiny, a module of Bibliometrix, enables users to perform relevant bibliometric and visual analyses through an interactive web interface [49]. This tool facilitates identifying connections between changes in scientific production, citations, author collaborations, and other essential bibliometric indicators, especially in programming education. Consequently, Bibliometrix is a robust and accessible tool for the scientific community, providing an effective means to explore and evaluate academic literature.

IV. Results

This section presents the results of the bibliometric analysis, highlighting the most relevant authors, important institutions, influential documents, and the keywords with the highest relevance in the analyzed articles.

A. Most Relevant Authors

The bibliometric analysis identifies the most prominent authors in the field. Table II presents a list of authors with notable contributions and the number of citations they have received. Among them, Li Yong, Liu Yonggang, Liu Yan, Liu Yuan, and Yong Wang stand out. These researchers were identified by comparing SCOPUS and WOS records using metrics such as the H-index and citation count, which measure scientific performance in the field [50]. The authors in Table II have made significant advancements in programming education through detailed studies on effective teaching and learning strategies. Their work includes innovative methodologies and practices with long-term impacts on students' learning outcomes.

By examining the H-index and citation count, we can evaluate the influence and impact of these researchers in promoting programming education. This analysis provides valuable insights into the key contributors driving the field and underscores the importance of developing effective teaching and learning strategies.

TABLE I. NON-PHARMACOLOGICAL INTERVENTION STRATEGIES BASED ON CONTROLLED EXERCISE AND NUTRITIONAL EDUCATION

Criterion	Values
Time Span	2014 – 2023
Date of Query	June 2023
Document Types	Journal articles and Conference proceedings
Journal and Conference Types	Any type
Search Fields	Title, Abstract, and Keywords
Search Terms	Programming AND Universities AND Strategies AND (Learning AND Teaching) – in English
Records	WOS: 1464; SCOPUS: 361
Total Records	1,697

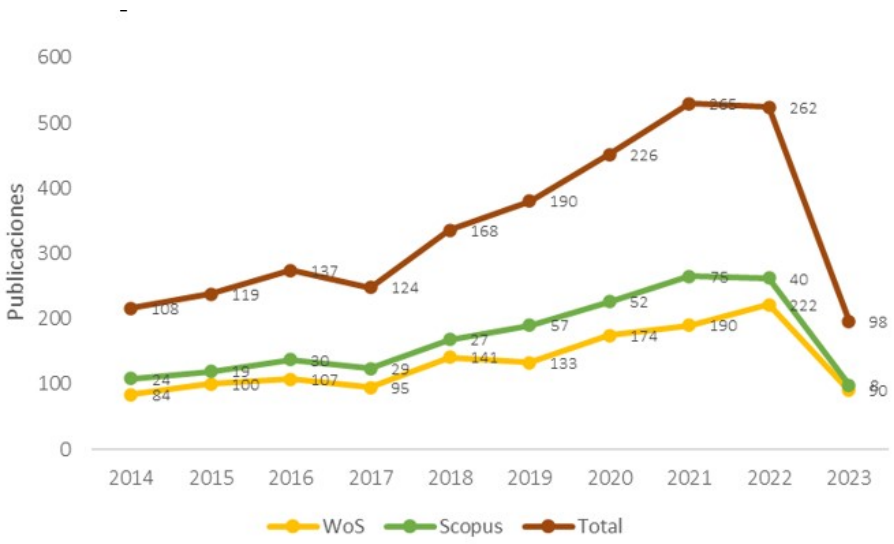

Fig. 1. Total number of publications in WOS and SCOPUS, 2015–2023.

TABLE II. TOP AUTHORS IN PROGRAMMING EDUCATION RESEARCH IN HIGHER EDUCATION

Authors	WOS Citations	WOS H-index	SCOPUS Citations	SCOPUS H-index	Publications
Li Yong	2058	23	7591	42	11
Liu Yonggang	29	3	682	15	11
Liu Yan	2072	22	9997	50	11
Liu Yuan	1151	18	938	16	11
Yong Wang	2124	26	8192	50	11

B. Key Institutions

The bibliometric review identified the most prominent institutions in the field based on their significant publication output. Table III lists the top ten universities, their country of origin, and the number of related publications. These results corroborate the findings of Apiola et al. [51] and Perez and Garcia [52], highlighting that universities play a crucial role in publishing research and disseminating experiences related to programming education in higher education. Their contributions emphasize the importance of integrating strategies that help students develop critical thinking and problem-solving skills, ultimately improving their academic and professional development.

TABLE III. TOP INSTITUTIONS CONTRIBUTING TO PROGRAMMING EDUCATIONAL RESEARCH

Institution	Publications	Country
University of California, San Francisco	74	USA
University of Toronto	69	Canada
University of Colorado Boulder	59	USA
University of Michigan	59	USA
University of Sydney	43	Australia
University of Calgary	36	Canada
McGill University	35	Canada
University of Pennsylvania	35	USA
National University of Singapore	34	Singapore
University of Minnesota	34	USA

C. Influential Documents

Identifying key documents begins by determining how each document connects to internal research references (endogenous references) and external sources (exogenous references) found in academic databases such as SCOPUS and WOS. Duque and Duque Oliva [53] explain that the average number of citations is calculated by dividing internal references by the time elapsed since the document's initial publication. This approach offers a method to assess the influence and acceptance of research over time.

When analyzing influential documents, our findings show that the study by Sung et al. [54] has the highest number of citations, with a total of 620 and an average of 77.5 per year. McLaughlin et al. [55] follows closely with 603

citations and an average of 60.3 per year, while Bers et al. [56] has a total of 378 citations and an annual average of 37.8. These three authors stand out due to the high citation counts of their respective articles, which focus on topics related to programming education strategies. Table IV lists top cited documents in programming education research.

D. Keywords with the Highest Relevance

During the analysis, we extracted the terminologies that had the most significant impact across the reviewed documents. Using the Biblioshiny tool, we created a graphical representation or "word cloud" (Fig. 2) to visualize the most prominent terms identified in this study. As explained by Alsalem et al. [57], this method allows scholars to compare different sections and visually identify the most significant terms, highlighted in bold. The word cloud emphasizes key terms such as "learning", "students", "education", "teaching", and "programming", which are closely aligned with the study's focus on programming education for undergraduate students. This approach explores the interaction between pedagogical approaches, technology, and programming in educational settings while analyzing how these elements influence students' ability to acquire programming skills.

Using a TreeMap further illustrates the clustering of potential keywords in the research articles, as discussed by Secinaro et al. [58]. Table V presents the top ten keywords by frequency and percentage of appearance in the analyzed documents. The analysis shows that 41% of the most relevant terms include words such as "education", "students", "teaching", "learning", and "programming", which underscores the importance of this research topic within the field of scientific publications.

V. Discussion

This study provides a comprehensive overview of current trends in teaching and learning programming through a bibliometric analysis of the SCOPUS and WOS databases. The results highlight several key areas that require further exploration and themes that have been consistently addressed in existing research.

A. Current Trends

One of the main trends identified in the analysis is the emphasis on project-based learning and flipped classrooms. These strategies have proven to be effective in enhancing problem-solving skills and fostering greater engagement by allowing students to apply learned concepts to real-world problems [54]. In addition, there has been a growing adoption of collaborative programming techniques such as pair programming, which improve code quality and overall student performance [55]. Regarding emerging topics, there is increasing interest in using immersive and simulation technologies such as augmented reality and virtual reality to teach programming [56].

B. Limitations of Current Research

Despite advances in programming education, several challenges and limitations persist. One of the key limitations identified is the lack of longitudinal studies that measure the long-term impact of different programming teaching strategies. Most of the analyzed studies focus on short-term outcomes, such as grades or students' performance in individual courses, but more research is needed to explore how these strategies influence long-term professional development and career outcomes [59]. Another limitation is the lack of diversity in the educational contexts studied. Most of the research has been conducted in developed countries, particularly in the United States and China, which may not reflect the educational realities of other regions. Furthermore, more research is needed on programming education in non-formal settings and self-learning environments, as many students learn programming independently or through online platforms outside traditional classrooms [60].

C. Implications for Future Research

This study provides several implications for future research. First, more research is needed on the use of emerging technologies such as artificial intelligence and machine learning in programming education. These technologies have the potential to personalize teaching and provide immediate feedback to students, which could significantly improve learning outcomes [61]. Second, there is a need to further explore interdisciplinary approaches to teaching programming. Integrating programming with other disciplines has proven to be effective in improving student understanding and motivation, but more research is needed to understand how these approaches can be optimally designed and implemented [62]. In addition, more research is required on the barriers students face when learning to program, particularly in disadvantaged contexts. Programming can be a difficult skill to acquire, and many students struggle with abstract concepts and the complex syntax of programming languages. Understanding these barriers and how to overcome them is crucial to ensuring that all students have the opportunity to develop programming skills [63].

D. Threats to Validity

As with any study, this bibliometric analysis has limitations that may affect the validity of the results. The following threats to validity should be considered:

- Selection bias: The use of the SCOPUS and Web of Science databases, although they include a large volume of high-quality research, may have excluded important studies from other databases not considered in this analysis. This raises the possibility of selection bias, as some relevant studies published in non-indexed academic journals or platforms may not have been included.
- Temporal bias: The analysis focused on publications from 2014 to 2023, which means that any research conducted before this period was excluded. Although this time range captures the most recent trends, it may omit foundational studies or pioneering approaches in programming education that still influence the field. This temporal bias could limit the understanding of the complete evolution of programming teaching methodologies over time.
- Variability in bibliometric indicators: The quality and impact of a research paper were evaluated using bibliometric metrics such as the citation count and the

TABLE IV. TOP CITED DOCUMENTS IN PROGRAMMING EDUCATIONAL RESEARCH

Author	Journal	Total Citations (TC)	Average Citations per Year
Sung et al. (2016)	Computers & Education	620	77.5
McLaughlin et al. (2013)	Academic Medicine	603	60.3
Bers et al. (2014)	Computers & Education	378	37.8
Douzas et al. (2018)	Information Sciences	350	58.33
Chiu-Lin and Gwo-Jen (2016)	Computers & Education	319	39.88

Fig. 2. Word cloud highlighting the most frequent terms in the analyzed articles.

TABLE V. TOP TEN KEYWORDS BY FREQUENCY AND PERCENTAGE OF APPEARANCE

Keyword	Frequency	Percentage
Learning	1127	2.50%
Programming	911	2.02%
Students	898	1.99%
Teaching	594	1.32%
Education	500	1.11%
University	345	0.76%
Computer	339	0.75%
Course	332	0.74%
Based	299	0.66%
Strategies	274	0.61%

H-index. However, these metrics may not fully capture the qualitative importance of a study. Factors such as the subject area, the type of publication (journal versus conference), and the cultural context of the authors can influence the visibility and impact of a study. The metrics used do not always reflect the real-world impact in the classroom or the practical implementation of the pedagogical strategies discussed.

E. Future Research

Future research should explore longitudinal studies to evaluate the long-term impact of programming teaching methodologies. These studies are crucial to understanding how educational strategies influence skill retention and professional outcomes over long periods. While this study provides a comprehensive bibliometric analysis of programming teaching strategies, it does not evaluate the direct learning outcomes of these methodologies. Future research should incorporate empirical assessments to validate the effectiveness of the identified strategy in improving students' programming competencies.

VI. CONCLUSIONS

This bibliometric analysis provides a detailed insight into current trends and key contributors in research on programming education in higher education. The findings reveal that programming continues to be a growing area of interest, with a significant number of studies published in recent years, particularly in countries like the United States and China. Pedagogical strategies such as project-based learning, flipped classrooms, and collaborative programming have emerged as effective approaches to improving student programming skills.

The analysis also identified several challenges and areas that require further research. Among these is the need for more longitudinal studies that explore the long-term impact of different programming teaching strategies, as well as greater diversity in the educational contexts studied. Furthermore, it is essential to explore new technologies, such as artificial intelligence, in the context of programming education, as well as to integrate more interdisciplinary approaches that can enhance students' understanding and motivation to learn.

Another important conclusion is that while research in this

field has grown significantly, disparities remain in terms of international collaboration and access to resources. Developing countries could benefit from greater support and collaboration with institutions from more developed nations to ensure that advances in programming education are accessible to all.

In terms of contribution, this article not only identifies the key trends and most influential contributors, but also provides a roadmap for future research in programming education. We hope that the findings of this study will assist educators, researchers, and policymakers in developing more effective and equitable approaches to programming education in higher education.

REFERENCES

[1] V. Basilotta-Gómez-Pablos, M. Matarranz, L.-A. Casado-Aranda, and A. Otto, "Teachers' digital competencies in higher education: a systematic literature review," *International Journal of Educational Technology in Higher Education*, vol. 19, no. 1, p. 8, Feb 2022. [Online]. Available: https://doi.org/10.1186/s41239-021-00312-8

[2] P. Abichandani, V. Sivakumar, D. Lobo, C. Iaboni, and P. Shekhar, "Internet-of-things curriculum, pedagogy, and assessment for stem education: A review of literature," *IEEE Access*, vol. 10, pp. 38 351–38 369, 2022.

[3] M. González-Sanmamed, A. Sangrá, A. Souto-Seijo, and I. Estévez Blanco, "Ecologías de aprendizaje en la era digital: desafíos para la educación superior," *PUBLICACIONES*, vol. 48, no. 1, pp. 25–45, 2018.

[4] J. Jiménez-Toledo, C. Collazos, and O. Revelo-Sánchez, "Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura," *TecnoLógicas*, vol. 22, pp. 83–117, 2019.

[5] Y.-T. Lin, M. K.-C. Yeh, and S.-R. Tan, "Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations," *IEEE Transactions on Education*, vol. 65, no. 4, pp. 617–627, 2022.

[6] P. Compañ-Rosique, R. Satorre-Cuerda, F. Llorens-Largo, and R. Molina-Carmona, "Enseñando a programar: un camino directo para desarrollar el pensamiento computacional," *Revista de Educación a Distancia (RED)*, vol. 46, 2015.

[7] F. A. Adamopoulos, *Learning Programming, Student Motivation*. Cham: Springer International Publishing, 2020, pp. 1058–1067. [Online]. Available: https://doi.org/10.1007/978-3-030-10576-1_182

[8] S. Masapanta-Carrion and J. Velázquez-Iturbide, "A systematic review of the use of bloom's taxonomy in computer science education," in *SIGCSE '18*. Association for Computing Machinery, 2018, pp. 441–446.

[9] B. Bloom, *Taxonomy of Educational Objectives*. Longman, 1956.

[10] A. Younis, R. Sunderraman, M. Metzler, and A. Bourgeois, "Developing parallel programming and soft skills: A project-based learning approach," *Journal of Parallel and Distributed Computing*, vol. 158, pp. 151–163, 2021.

[11] N. Shin, J. Bowers, J. Krajcik, and D. Damelin, "Promoting computational thinking through project-based learning," *Disciplinary and Interdisciplinary Science Education Research*, vol. 3, no. 1, p. 7, 2021.

[12] A. Bawamohiddin and R. Razali, "Problem-based learning for programming education," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, p. 2035, 2017.

[13] Q. Sun, J. Wu, and K. Liu, "Toward understanding students' learning performance in an object-oriented programming course: The perspective of program quality," *IEEE Access*, vol. 8, pp. 37 505–37 517, 2020.

[14] E. Merelli, N. Paoletti, and L. Tesei, "Adaptability checking in complex systems," *Science of Computer Programming*, vol. 115-116, pp. 23–46, 2016.

[15] Q. Cheng, D. Benton, and A. Quinn, "Building a motivating and autonomy environment to support adaptive learning," in *2021 IEEE Frontiers in Education Conference (FIE)*, 2021, pp. 1–7.

[16] B. Vesin, K. Mangaroska, and M. Giannakos, "Learning in smart environments: user-centered design and analytics of an adaptive learning system," *Smart Learning Environments*, vol. 5, 2018.

[17] J. Qadir, K.-L. A. Yau, M. Ali Imran, and A. Al-Fuqaha, "Engineering education, moving into 2020s : Essential competencies for effective 21st century electrical & computer engineers," in *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9.

[18] M. Thuné and A. Eckerdal, "Analysis of students' learning of computer programming in a computer laboratory context," *European Journal of Engineering Education*, vol. 44, pp. 1–18, 2018.

[19] B. Xie, D. Loksa, G. Nelson, M. Davidson, D. Dong, H. Kwik, A. Hui Tan, L. Hwa, M. Li, and A. Ko, "A theory of instruction for introductory programming skills," *Computer Science Education*, vol. 29, no. 2-3, pp. 205–253, 2019.

[20] L. Silva, A. J. Mendes, and A. Gomes, "Computer-supported collaborative learning in programming education: A systematic literature review," in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, pp. 1086–1095.

[21] G. Liargkovas, A. Papadopoulou, Z. Kotti, and D. Spinellis, "Software engineering education knowledge versus industrial needs," *IEEE Transactions on Education*, vol. 65, no. 3, pp. 419–427, 2022.

[22] A. Yusuf and N. M. Noor, "Research trends on learning computer programming with program animation: A systematic mapping study," *Computer Applications in Engineering Education*, vol. 31, no. 6, pp. 1552–1582, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22659

[23] C.-S. Cheah, "factors-contributing-to-the-difficulties-in-teaching-and-learning-of-computer-programming-a-literature-review," *Contemporary Educational Technology*, vol. 12, p. ep272, 05 2020.

[24] C. Lu, R. Macdonald, B. Odell, V. Kokhan, C. Demmans Epp, and M. Cutumisu, "A scoping review of computational thinking assessments in higher education," *Journal of Computing in Higher Education*, vol. 34, no. 2, pp. 416–461, Aug 2022. [Online]. Available: https://doi.org/10.1007/s12528-021-09305-y

[25] Y. Li, A. H. Schoenfeld, A. A. diSessa, A. C. Graesser, L. C. Benson, L. D. English, and R. A. Duschl, "Computational thinking is more about thinking than computing," pp. 1–18, 2020.

[26] K. Srinivasa, M. Kurni, and K. Saritha, "Computational thinking," in *Learning, Teaching, and Assessment Methods for Contemporary Learners: Pedagogy for the Digital Generation*. Springer, 2022, pp. 117–146.

[27] J. Shen, G. Chen, L. Barth-Cohen, S. Jiang, and M. Eltoukhy, "Connecting computational thinking in everyday reasoning and programming for elementary school students," *Journal of Research on Technology in Education*, vol. 54, no. 2, pp. 205–225, 2022.

[28] K. Kanaki and M. Kalogiannakis, "Assessing algorithmic thinking skills in relation to age in early childhood stem education," *Education Sciences*, vol. 12, no. 6, p. 380, 2022.

[29] K. Kwon, A. T. Ottenbreit-Leftwich, T. A. Brush, M. Jeon, and G. Yan, "Integration of problem-based learning in elementary computer science education: effects on computational thinking and attitudes," *Educational Technology Research and Development*, vol. 69, pp. 2761–2787, 2021.

[30] C. Vidal-Silva, J. Cárdenas-Cobo, M. Tupac-Yupanqui, J. Serrano-Malebrán, and A. Sánchez Ortiz, "Developing programming competencies in school-students with block-based tools in chile, ecuador, and peru," *IEEE Access*, vol. 12, pp. 118 924–118 936, 2024.

[31] R. P. Lai, "Beyond programming: A computer-based assessment of computational thinking competency," *ACM Transactions on Computing Education (TOCE)*, vol. 22, no. 2, pp. 1–27, 2021.

[32] N. Shin, J. Bowers, S. Roderick, C. McIntyre, A. L. Stephens, E. Eidin, J. Krajcik, and D. Damelin, "A framework for supporting systems thinking and computational thinking through constructing models," *Instructional Science*, vol. 50, no. 6, pp. 933–960, 2022.

[33] D. Helbing, S. Mahajan, R. H. Fricker, A. Musso, C. I. Hausladen, C. Carissimo, D. Carpentras, E. Stockinger, J. A. Sanchez-Vaquerizo, J. C. Yang *et al.*, "Democracy by design: Perspectives for digitally assisted, participatory upgrades of society," *Journal of Computational Science*, vol. 71, p. 102061, 2023.

[34] A. Yadav and U. Berthelsen, *Computational Thinking in Education: A Pedagogical Perspective*. Routledge, 2021. [Online]. Available: https://books.google.cl/books?id=2H9kzgEACAAJ

[35] J.-A. Marín-Marín, A.-J. Moreno-Guerrero, P. Dúo-Terrón, and J. López-Belmonte, "Steam in education: a bibliometric analysis of performance and co-words in web of science," *International Journal of STEM Education*, vol. 8, no. 1, p. 41, Jun 2021. [Online]. Available: https://doi.org/10.1186/s40594-021-00296-x

[36] A. Melro, G. Tarling, T. Fujita, and J. K. Staarman, "What else can be learned when coding? a configurative literature review of learning opportunities through computational thinking," *Journal of Educational Computing Research*, vol. 61, no. 4, pp. 901–924, 2023. [Online]. Available: https://doi.org/10.1177/07356331221133822

[37] Z. Ju, "Computational thinking through programming: a meta-analysis of collaborative versus solo problem solving," Masters by Research, Faculty of Arts and Social Sciences, Sydney School of Education and Social Work, University of Sydney, 2024. [Online]. Available: https://hdl.handle.net/2123/32647

[38] Y. Qian and I. Choi, "Tracing the essence: ways to develop abstraction in computational thinking," *Educational technology research and development*, vol. 71, no. 3, pp. 1055–1078, Jun 2023. [Online]. Available: https://doi.org/10.1007/s11423-022-10182-0

[39] N. Selwyn, T. Hillman, A. Bergviken-Rensfeldt, and C. Perrotta, "Making sense of the digital automation of education," *Postdigital Science and Education*, vol. 5, no. 1, pp. 1–14, Jan 2023. [Online]. Available: https://doi.org/10.1007/s42438-022-00362-9

[40] P. Dúo-Terrón, "Analysis of scratch software in scientific production for 20 years: Programming in education to develop computational thinking and steam disciplines," *Education Sciences*, vol. 13, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2227-7102/13/4/404

[41] R. Briner and D. Denyer, "Systematic review and evidence synthesis as a practice and scholarship tool," pp. 112–129, 2012.

[42] S. Echchakoui, "Why and how to merge scopus and web of science during bibliometric analysis: the case of sales force literature from 1912 to 2019," *Journal of Marketing Analytics*, vol. 8, 2020.

[43] R. Pranckutė, "Web of science (wos) and scopus: The titans of bibliographic information in today's academic world," *Publications*, vol. 9, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2304-6775/9/1/12

[44] A. Valente, M. Holanda, A. M. Mariano, R. Furuta, and D. Da Silva, "Analysis of academic databases for literature review in the computer science education field," in *2022 IEEE Frontiers in Education Conference (FIE)*, 2022, pp. 1–7.

[45] J. Zhu and W. Liu, "A tale of two databases: the use of web of science and scopus in academic papers," *Scientometrics*, vol. 123, 2020.

[46] X. Chen, D. Zou, H. Xie, and F. L. Wang, "Past, present, and future of smart learning: a topic-based bibliometric analysis," *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, p. 2, Jan 2021. [Online]. Available: https://doi.org/10.1186/s41239-020-00239-6

[47] M. Trinidad, M. Ruiz, and A. Calderón, "A bibliometric analysis of gamification research," *IEEE Access*, vol. 9, pp. 46 505–46 544, 2021.

[48] Z. Li, G. Wang, J. Lu, D. G. Broo, D. Kiritsis, and Y. Yan, "Bibliometric analysis of model-based systems engineering: Past, current, and future," *IEEE Transactions on Engineering Management*, vol. 71, pp. 2475–2492, 2024.

[49] J. Moral-Munoz, E. Herrera-Viedma, A. Espejo, and M. Cobo, "Software tools for conducting bibliometric analysis in science: An up-to-date review," *El Profesional de la Información*, vol. 29, 01 2020.

[50] J. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[51] M. Apiola, S. López-Pernas, M. Saqr, A. Pears, M. Daniels, L. Malmi, and M. Tedre, "From a national meeting to an international conference: A scientometric case study of a finnish computing education conference," *IEEE Access*, vol. 10, pp. 66 576–66 588, 2022.

[52] M. Perez and P. Garcia, "Tracing participation beyond computing careers: How women reflect on their experiences in computing programs," *ACM Trans. Comput. Educ.*, vol. 23, no. 2, apr 2023. [Online]. Available: https://doi.org/10.1145/3582564

[53] P. Duque and E. J. Duque Oliva, "Tendencias emergentes en la literatura sobre el compromiso del cliente: un análisis bibliométrico," *Estudios Gerenciales*, vol. 38, no. 162, pp. 120–132, mar. 2022.

[54] Y.-T. Sung, K.-E. Chang, and T.-C. Liu, "The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis," *Computers & Education*, vol. 94, pp. 252–275, 2016.

[55] J. McLaughlin, M. Roth, D. Glatt, N. Gharkholonarehe, C. Davidson, L. Griffin, D. Esserman, and R. Mumper, "The flipped classroom: A course redesign to foster learning and engagement in a health professions school," *Academic medicine : journal of the Association of American Medical Colleges*, vol. 89, 11 2013.

[56] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Computers & Education*, vol. 72, pp. 145–157, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131513003059

[57] M. A. Alsalem, A. H. Alamoodi, O. S. Albahri, K. A. Dawood, R. T. Mohammed, A. Alnoor, A. A. Zaidan, A. S. Albahri, B. B. Zaidan, F. M. Jumaah, and J. R. Al-Obaidi, "Multi-criteria decision-making for coronavirus disease 2019 applications: a theoretical analysis review," *Artificial Intelligence Review*, vol. 55, pp. 057–067, 08 2022.

[58] S. Secinaro, V. Brescia, D. Calandra, and P. Biancone, "Employing bibliometric analysis to identify suitable business models for electric cars," *Journal of Cleaner Production*, vol. 264, p. 121503, 2020.

[59] L. Chiu-Lin and H. Gwo-Jen, "A self-regulated flipped classroom approach to improving students' learning performance in a mathematics course," *Computers & Education*, vol. 100, pp. 126–140, 2016.

[60] Y. Chen, Y. Li, R. Narayan, A. Subramanian, and X. Xie, "Gene expression inference with deep learning," *Bioinformatics*, vol. 32, no. 12, pp. 1832–1839, 02 2016. [Online]. Available: https://doi.org/10.1093/bioinformatics/btw074

[61] J.-M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools," *Computers and Education*, vol. 97, pp. 129 – 141, 2016.

[62] C. Carraccio, R. Englander, E. Van Melle, O. ten Cate, J. Lockyer, M.-K. Chan, J. Frank, and L. Snell, "Advancing competency-based medical education: A charter for clinician-educators," *Academic medicine : journal of the Association of American Medical Colleges*, vol. 91, 12 2015.

[63] H. B. Shapiro, C. H. Lee, N. E. Wyman Roth, K. Li, M. Çetinkaya Rundel, and D. A. Canelas, "Understanding the massive open online course (mooc) student experience: An examination of attitudes, motivations, and barriers," *Computers & Education*, vol. 110, pp. 35–50, 2017.