Received February 4, 2022, accepted February 28, 2022, date of publication March 8, 2022, date of current version March 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3157609

Computing Education Research Compiled: Keyword Trends, Building Blocks, Creators, and Dissemination

MIKKO APIOLA[1], MOHAMMED SAQR[2], SONSOLES LÓPEZ-PERNAS[3], (Graduate Student Member, IEEE), AND MATTI TEDRE[2]

[1]Department of Computing, University of Turku, 20500 Turku, Finland
[2]School of Computing, University of Eastern Finland, 80101 Joensuu, Finland
[3]Departamento de Sistemas Informáticos, ETSI Sistemas Informáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Mikko Apiola (mikko-ville.apiola@utu.fi)

ABSTRACT The need for organized computing education efforts dates back to the 1950s. Since then, computing education research (CER) has evolved and matured from its early initiatives and separation from mathematics education into a respectable research specialization of its own. In recent years, a number of meta-research papers, reviews, and scientometric studies have built overviews of CER from various perspectives. This paper continues that approach by offering new perspectives on the past and present state of CER: analyses of influential papers throughout the years, of the theoretical backgrounds of CER, of the institutions and authors who create CER, and finally of the top publication venues and their citation practices. The results reveal influential contributions from early curriculum guidelines to rigorous empirical research of today, the prominence of computer programming as a topic of research, evolving patterns of learning-theory usage, the dominance of high-income countries and a cluster of 52 elite institutions, and issues regarding citation practices within the central venues of dissemination.

INDEX TERMS Computer science education, computing education research, computing education, scientometrics, science mapping, review.

I. INTRODUCTION

In the past 20–25 years, computing education research (CER) has matured from being an interest of teachers in colleges and universities, from all walks of computing research, into a respectable research specialization of its own [1], [2]. In the past two decades, the field has seen all the signs of an established field emerge: there are journals specialized in CER, research conferences, research centers, the field's own unique conceptual frameworks, seminal publications, and professorial appointments explicitly dedicated to the field [1]. But over the many decades that have passed, from CER's early beginnings as informal meetings to today's flourishing research area, the field has seen massive changes. Trends come and go, groups form and dissolve, expectations and concerns of computing change, and technologies rise and fall with the hype cycle. In order to understand the present-day issues of CER and to appreciate the field's current debates, it is important to recognize its history and evolution. By building a narrative of the past, it is possible to see relationships between societies and people; to build a sense of identity; and to weave a narrative of authors, institutions, and their networks, as well as where communities have come from, and potentially reveal clues as to where CER is headed.

This article complements previous reviews, overviews, and meta-research of CER by offering four unique scientometric perspectives of the past and present of CER. First, the article provides a historical overview of the influential publications and dominant topics and themes of research over the past six decades of CER. Second, it investigates the co-citation metrics of CER publications in order to identify the common building blocks of CER and gain insights into the foundational studies that CER builds upon. Third, it analyzes the creators of CER: influential authors, institutions, and networks of collaboration. Finally, because a crucial part of science is dissemination, where the central venues act as gatekeepers

The associate editor coordinating the review of this manuscript and approving it for publication was John Mitchell.

to control what is valued and accepted and what is not, the article identifies the most influential publication venues of CER and analyzes their citation practices. The study relies on Scopus metadata from CER publications imported from the most influential dedicated publication venues of CER, as well as CER publications identified by a common keyword search. The article concludes with a discussion and reflection on the findings.

A. A BRIEF HISTORY OF CER

There is not a single birth date for computing as a discipline, as it gradually separated itself from a number of existing academic fields [3]. Similarly, there is no single moment when computing education separated from courses on numerical mathematics, electrical engineering, and natural sciences, among others [4], [5]. The mass production of stored-program computers required large numbers of programmers and, as early as 1954, a conference on computing education acknowledged the need for organized computing education efforts [6]. By the early 1960s, organizations like DPMA (Data Processing Management Association) and IFIP (International Federation for Information Processing) presented their own curricula, private companies' own training programs flourished, and computing education had become institutionalized in a large number of universities [7].

As computing programs mushroomed in colleges and universities through the 1960s, increasing numbers of computing educators wished to advance the art of computing education. Conferences were an important venue for airing ideas about computing education through the 1960s: among numerous others, IFIP's TC-3 (established in 1963) and Association for Computing Machinery (ACM)'s SIGCSE (Special Interest Group in Computer Science Education, established in 1968) both organized their first conferences in 1970. Empirical results were published in journals like International Journal of Man-Machine Studies, the Computer Journal, Computers & Education, and Communications of the ACM. Along with countless course descriptions, new learning tools, experience reports, and ideas for how to best teach computing topics, there were also scattered examples of empirical research on computing education [8], [9]. From the 1980s and 1990s on, there was a growing interest in studies on learning programming languages, on how learning programming teaches "generic" skills, and on the psychology of programming [e.g., the Psychology of Programming Interest Group (PPIG) and the Empirical Studies of Programmers (ESP)], to mention a few [9], [10]. At the same time there was a growing feeling that "swap meets" for sharing one's pedagogical ideas, useful assignments, or lecture props [11] do not suffice for building a respectable body of knowledge on computing education.

A number of shifts started around the year 2000. From the early 2000s on, computing symposia started to place increasingly stringent restrictions on the methodological quality of the papers they published [1], [12], several capacity-building initiatives aimed at training computing education researchers for disciplinary based education research [13], and a few landmark works re-defined the computing education research landscape (e.g., [14]). As ACM's prime journal on computing education was re-branded from JERIC (Journal of Educational Resources in Computing) to TOCE (Transactions on Computing Education), it announced a shift away from engineering or technology articles [1, pp. 64, 80], and new conferences like ICER (the International Computing Education Research Conference) started to expect methodologically rigorous, theory-informed studies.

B. RELATED RESEARCH

Over the years, the CER field has seen a number of analyses, reviews, and meta-research on CER publications. One of the earliest was that of Valentine [15], who classified SIGCSE Technical Symposium papers as *Marco Polo, Tools, Experimental, Nifty, Philosophy, and John Henry*. Valentine's research soon sparked a number of other efforts to analyze CER papers (e.g., [11], [14], [16]–[18]). Meta-analyses of papers published in single CER venues include analyses of the ICER (International Computing Education Research) conference [19], [20], the ACE (Australian Computing Education) conference [17], [21], the ITiCSE (Innovation and Technology in Computer Science Education) conference [22], [23], Koli Calling conference [24], ICALT Conference on Advanced Learning Technologies [25], a comparison of research in Koli Calling and Informatics in Education journal [26] and the NACCQ (National Advisory Committee on Computing Qualifications) conference [27]. Several analyses and reviews have focused on papers on computer programming [8], [28]–[30], and in K-12 education [31], [32]. One of the most prolific analysts, Simon, developed a classification system to categorize CER into four dimensions: context (the subject matter), theme, scope, and nature (an experience report, empirical analysis or experiment) [1], [17]. For instance, a recent analysis showed that today's ACE conference has more research papers than experience reports—especially research studies on computer programming [21]. The extensive work of Simon is well summarized in [1].

Reviews focusing on theory use in CER have started to appear. A study [33] reviewed 72 papers from ICER (2005-2010), and found that 57% of papers used a theoretical basis (a theory, model, or framework), with the most common theories being Bloom's taxonomy, cognitive apprenticeship, cognitive load theory, schemata theory, self-efficacy theory, situated learning, structure of the observed learning outcome (SOLO) taxonomy, and threshold concepts [33]. Researchers went on to analyze TOCE and Computer Science Education (CSE) publications ($n = 308$) [34], and found that 51% of papers had a theoretical base, mostly drawn from computing, psychology, and education. In recent work [35], the target was theoretical constructs (TCs) with a specific origin in CER, with findings showing common examples of TC's such as proximal flow, learning edge momentum, and others, but still an overall modest use of TCs with no prevailing theoretical works that are broadly applied [35]. The results do point towards gradual maturation of CER through building its own theoretical constructs and in this way claiming its independence [35]. A recent review on learning theories in CER [36] identified a list of common learning theories and related influential contributions. The use of theories in CER articles in the central venues of TOCE, CSE, SIGCSE Bulletin, SIGCSE Technical Symposium (TS), ICER, ITiCSE, Koli Calling, and ACE were inspected. Top referenced learning theories were identified, including flow-theory, self-efficacy, and the largely debunked learning styles theory [36]. The findings also show that a large majority of influential papers were found to have only a limited reach within CER [36].

Use of research design and research methodology has been the target of many analyses. For example, a recent review of 427 papers in 2014 and 2015, published in SIGCSE TS, ICER, ITiCSE, TOCE, and CSE, found that 80% of papers included some form of empirical evaluation, and that many papers lacked rigor in reporting and did not consistently follow reporting norms [37]. While empirical research is on the increase, it still lacks in reporting quality. In recent years, venues such as SIGCSE TS, and TOCE have started to pay increased attention into reporting of empirical work in their review criteria. It has been estimated that this will eventually lead to better support for replication and meta-analyses in the future [37]. Other reviews to investigate methodology use in CER include those of [28], [33], [38], and findings show, e.g., extensive use of statistical methods beyond descriptives [28], and reveal what statistical methods are commonly used in CER [38]. A review [39] reported that 54% of empirical papers in SIGCSE TS (2014-2015) used surveys for data collection, and a small percentage reported use of qualitative methods, while another review [40] showed that 32% of papers in TOCE, CSE, and ICER (2013-2017) used both qualitative and quantitative data. Other reviews have looked, e.g., at the replicability of CER studies [41].

Scientometric analyses of CER publications are also starting to emerge. One study analyzed the collaboration networks of SIGCSE, ITiCSE, and ICER, revealing the dominance of high-income countries, especially the United States, and that, while collaboration between authors from different institutions is growing, collaboration still takes mostly place within countries rather than between countries [42]. The geographical diversity in the SIGCSE Technical Symposium has been investigated, with results showing strong US dominance [43], [44]. One scientometric study investigated the publications in the ASEE/IEEE Frontiers in Education (FIE) conference from the viewpoint of most cited papers, influential authors, and internalization [45], while another analysis of FIE looked at sources of funding [46]. Another investigation analyzed the keywords in ITiCSE and ICER papers [47], showing, for instance, a steady trend of research on introductory programming courses. The history of computational thinking (CT), a subarea of CER research, was scientometrically mapped by a recent study [48].

The combined body of literature reviews, meta-research, and scientometric analyses of CER paints a picture of CER evolving over the decades from experience reports to empirical research; increased attention paid to learning theories, research methods, and reporting rigor; sustained focus on programming education; and dominance of high-income countries, especially the US. Up-to-date, scientometric studies have targeted one or two publication venues, missing a holistic and more in-depth analysis of the CER field. No previous scientometric or other analyses have covered all central CER publication venues, over many decades, and offered a holistic analysis of citations, keyword trends, foundational work, dissemination, and core institutions that create CER. While a number of reviews and meta-reviews exist, their scope is limited to a restricted set of articles within a specified topic. All these lacks in the previous analyses together constitute the research gap for this paper.

C. SCIENTOMETRICS

The increased quality in structured databases such as Scopus and Web of Science, combined with new methods of data science and network analysis have resulted in rising popularity of scientometric studies [48], [49]. With new tools and methods, one can go far beyond simple quantities and descriptives. The new tools provide researchers means to conduct transparent and reproducible studies of published research. In this research, we apply state-of-the-art methods to offer quantitative in-depth views to CER. The tools used include the Bibliometrix package [49] for analysis, OpenRefine[1] for data screening, and Gephi[2] for visualizations.

D. RESEARCH QUESTIONS

As CER as a research field has evolved and matured over the years, its history offers a unique opportunity to study the evolution of central trends and themes in research topics (what CER studies), foundational work (what the studies are based on), knowledge creators (who does CER), and finally the dissemination (how CER is published). The research questions for this paper read as follows:

1) How has the publication profile of CER evolved in terms of keyword trends and topics of research? (RQ_1)
2) What do citation and co-citation metrics reveal about the foundational work in the CER discipline? (RQ_2)
3) How have the knowledge creators (authors and institutions) and their collaboration shaped the discipline of CER and its communities? (RQ_3)
4) How can the central venues of dissemination of CER be characterized with regards to their publication profiles and citation practices? (RQ_4)

The paper is structured as follows: data and methods will be presented first in section II, followed by evolution of keyword trends in section III, foundational work (section IV), creators (section V), and finally dissemination in section VI.

[1]https://openrefine.org
[2]https://gephi.org

TABLE 1. Descriptive statistics of the scientometric dataset.

Data set statistics	
Documents	16453
Distinct Venues	1840
Author Keywords (DE)	16290
Period	1954–2020
Mean citations per article	7.785
Document types	
Journal article	3345
Conference article or book chapter	13108
Authorship data	
Authors	21269
Author appearances	45479
Authors of single-authored documents	2917
Authors of multi-authored documents	18352
Single-authored documents	4421
Mean documents per author	0.774
Mean authors per document	1.29

The results will be discussed in section VII. Finally, the article is concluded in section VIII.

II. METHODS AND DATA

A. DATA SOURCE

The data were extracted from the Scopus database on September 1st, 2021. In addition to having most of the Web of Science titles, the Scopus database has a wider selection of technical conferences that are relevant to research questions of this article [50]. Compared to the other databases, Scopus is well maintained, and it uses rigorous selection criteria for inclusion of journals and conference proceedings [51], [52].

B. DEDICATED VENUES SEARCH

All publications from well-known CER venues were included in the data extraction. These included the two major journals dedicated to CER, which are *Computer Science Education (CSE)*, established in 1988 [14, p. 1], and *ACM Transactions on Computing Education (TOCE)*, whose inaugural issue was published in March 2009 [53]. ACM TOCE was formerly known as *Journal on Educational Resources in Computing (JERIC)*, which was launched in 2001, and had a special focus on educational resources such as educational technology in computing education [53]. All articles from CSE, ACM TOCE, and ACM JERIC were included. Then, key conferences were added. These included ACM's Special Interest Group in Computer Science Education (SIGCSE)'s annual symposium, which was started in 1970 [7], [54], ITiCSE (Innovation and Technology in Computer Science Education) which was founded in 1996, ACE (Australasian Computing Education Conference) (1996), Koli Calling (2001), and ICER (International Computing Education Research conference) (2005) [7]. ACM's new Global Computing Education Conference (CompED) was established to serve scholars outside North America and Europe, and was arranged for the first time in 2019. SIGCSE, ITiCSE, Koli Calling, ICER, ACE, and CompED are the main conferences dedicated to computing education research (e.g., [7], [14], [54]). A new conference Computer Science Education Research Conference (CSERC) was established in 2011, an in the K-12 context, there are ISSEP (The International Conference on Informatics in Schools) launched in 2005 and WIPSCE (Workshop in Primary and Secondary Computing Education), with roots in the German computing education community.

All papers published in the aforementioned conferences were included in the sample. For those conference proceedings that were missing some of their early years in the Scopus database (e.g., Koli Calling and the ACM SIGCSE Symposium), the missing data were manually retrieved from the ACM Digital Library and conference archives, where available.

C. KEYWORD SEARCH

A line needed to be drawn with regards to the amount of noise (non-CER articles) brought by less CER-focused publishing venues. For that reason some important venues were excluded. Most importantly, especially in the early days of computing education, CER results have been published in venues such as engineering education journals, generic computing journals, educational technology journals, and educational journals [7]. For example, the *IEEE Symposium on Visual Languages and Human-Centric Computing*, started in 1984 often included computing education research results [54]. Other venues known to occasionally publish important CER contributions—among other topics—include *IEEE Transactions on Education, Communications of the ACM, IEEE/ASEE Frontiers in Education Conference, Informatics in Education*, *LATiCE (Learning And Teaching in Computing and Engineering)*, as well as the Psychology of Programming Interest Group (PPIG), formed in 1987, and the Empirical Studies of Programmers (ESP), with its first workshop in 1986 [54]. Due to the large numbers of non-CER articles, those venues were excluded from full inclusion, but in order to still capture a representative sample of CER from outside the realm of the dedicated venues, a search query was performed in Scopus to find additional relevant research in computing education that has been published in venues that are not mainly devoted to this field. To identify the best suited search terms, several iterations of search queries were conducted, followed by examining the resulting papers for relevance to the study. The following search terms were used.

> "COMPUTING EDUCATION" OR "COMPUTER SCIENCE EDUCATION" OR "COMPUTER SCIENCE LEARN*" OR "LEARNING COMPUTER SCIENCE" OR "TEACHING COMPUTER SCIENCE" OR "COMPUTER SCIENCE TEACH*" OR "COMPUTER SCIENCE STUDENT*" OR "COMPUTER SCIENCE COURSE*" OR "COMPUTER SCIENCE CURRICUL*" OR "COMPUTING LEARN*" OR "TEACHING COMPUTING" OR "COMPUTING TEACH*" OR "COMPUTING STUDENT*" OR "COMPUTING COURSE*" OR "COMPUTING CURRICUL*"

D. INCLUSION AND EXCLUSION CRITERIA

Results were limited to original articles published in conferences or journals, written in the English language.

Rapid responses, letters to the editors, view points, trade journals, errata, systematic reviews and conference reviews were excluded as they represent secondary syntheses of already-published articles, and therefore, including such articles would amplify articles included in these reviews. CER has evolved from tools-research and experience reports to empirical research with increased attention in research methods and theories. As experience reports and tools-research have always been important parts of CER, and still are, we kept all such articles in the dataset. Articles published during 2021 were excluded to allow comparison across complete years.

The keyword "informatics education" is commonly used in German-speaking communities. However, a separate keyword search in Scopus resulted in several thousand contributions, many in the health informatics domain. By excluding health-related publication venues, and the articles already present in our dataset, the result set was limited to some 187 articles. A closer inspection revealed that those articles represent a diverse group of research areas, such as education technology and information technology (IT). A decision was made to exclude these articles, since many of them did not belong to CER.

After removing duplicates and uneligible articles, the number of resulting articles from combining both queries (full venues and keyword search) was 16,453 (see Fig. 1).

E. DATA SCREENING

Data screening means inspecting the data for errors and inconsistencies and fixing these errors in order to maximise the signal and minimise noise to prepare the data for analysis. For data screening, four steps were performed:

1) Author names were cleaned based on Scopus ID, so authors with different name spellings were combined.
2) Conference proceedings titles were checked and different editions of the same conference were combined (e.g., "SIGCSE 2018 Proceedings" and "Proceedings of the 2019 ACM SIGCSE Symposium"). Variations of the title of the proceedings for the same conference were also combined (e.g., "SIGCSE" and "SIGCSE Symposium"). Conferences with name changes were also combined. Similarly, variations in journal names were cleaned (e.g., "Computers and Education" and "Computers & Education").
3) Institutions were manually cleaned and variations were combined (e.g., "University of Helsinki", "Helsinki University" and "The University of Helsinki").
4) Keywords with similar meaning were combined including singular and plural forms (e.g., network and networks); abbreviations with their full spellings (e.g., "artificial intelligence" and "AI"), and variations of the same keyword, e.g. ("introductory courses", "introduction to CS", "CS1/CS2"), ("MOOC", "MOOCs", "massive open online course" and "massive open online courses"). The OpenRefine tool by Google was used to facilitate the process.

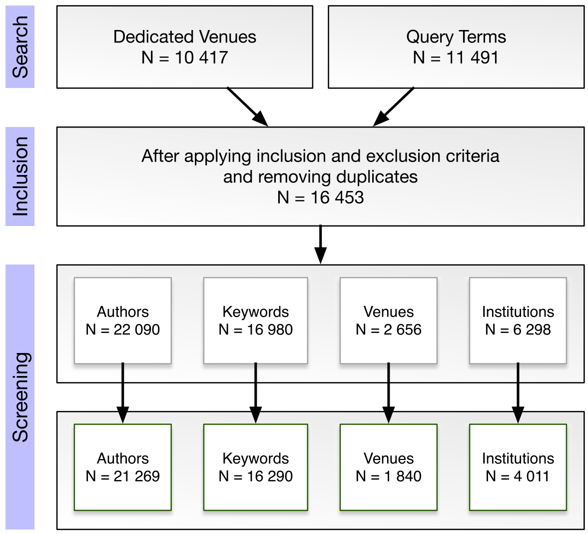


FIGURE 1. Search and data screening.

The screening process and results are outlined in Fig. 1.

F. ANALYSIS: METHODS AND TOOLS

The screened data set was analyzed using the R statistical language with the Bibliometrix package [49]. The frequencies of keywords, articles, citations, and plots were calculated from the Scopus metadata. A network of institutions was constructed from the affiliations of the co-authors of each article. Two institutions were considered connected if authors from both institutions collaborated on the same article. The network was plotted using the open-source software application Gephi [55] with the Fruchterman-Reingold layout algorithm. To study the presence of the rich-club phenomenon, the rich club coefficient (Ø) was computed as the ratio of the existing number of connections between the highly connected core (rich club) and the maximum connections possible. The boundary of the rich club core was computed using the method of decreasing rank order [56]. To exclude the possibility that the high degree of connectivity among the connected core nodes was a result of random chance, the normalized rich-club coefficient (Ønorm) was also computed. The Ønorm is the ratio of relative value of Ø to the mean value of Ø computed in 10 000 random networks that were matched for size and degree distribution.

A co-citation network was constructed by considering two papers connected when they were cited by the same paper. The network size was limited to a maximum of 100 most referenced papers. To map the subgroup of references that are commonly cited together—representing themes of theoretical or foundational articles—community detection was done using Louvain modularity, a value of 1 was set for the modularity, and for the community detection, edge weights were prioritised [57]. The network was plotted using the

Fruchterman Reingold layout algorithm and communities were color-coded. The node size is proportional to the degree centrality, i.e., the number of references connected to each node. A keyword co-occurrence network was constructed by considering each pair of two author keywords that appear in the same manuscript as connected. Groups or communities of keywords that are frequently listed together or strongly connected represent research themes. Louvain modularity was applied for community detection, with the same parameters as in the detection of foundational (co-cited) articles. The keyword network was plotted in the same way as the co-citation network.

III. EVOLUTION OF TRENDS AND THEMES

In order to gain a picture of the evolution of trends and themes of research, we briefly discuss the evolution of CER through the decades (III-A), followed by an analysis of keyword trends in the past two decades, the time period in which keywords have been available (III-B), and concluding with an analysis of the most influential topics of research, as revealed by keyword clusters (III-C).

A. SIX DECADES OF CER

Table 2 presents influential (highly cited) articles in the data over the decades of CER. In our data, prior to 1970, academic publications on CER were scarce. Since the 1950s, universities have considered how to arrange computing education. In 1951, some early textbooks on programming were launched; in the 1960s, the software industry grew, and universities started to offer training in computing and form computer science departments. In 1960, ACM started their education committee, and, in 1968, the ACM Computing Curriculum (CC) '68 [58] was published. The CC'68 became an authoritative guideline for establishing computing education in the US. Reviews of curricula, such as that of Stanford, and recommendations for establishing computer science departments were published [59]. A survey covering 25 US CS departments with students graduating in 1966-1967 revealed diversity in programs and, e.g., the wide teaching of FORTRAN, ALGOL and MAD, and the presence of AI in many curricula [60]. Another survey of computing curricula found that most computer professionals back then had reached their positions through apprenticeship or self-training [62]. The influential articles prior to 1970 are curriculum guidelines [61], recommendations, surveys, or reviews, which contributed to building the basis for teaching the then-new discipline of computer science. Publication venues for CER at this time were rare, and many contributions were documents such as e.g. curriculum guidelines rather than research articles.

The 1970s are marked by the development of communities of practice in CER, along with the formation of first conferences, journals, and magazines. The new computing curriculum CC'78 [63] introduced a shift from a mathematically oriented view of computer science (CS) to a more diverse discipline, including hands-on work, programming, and applications [63]. Influential work includes Marvin Minsky's Turing Award lecture [64] and recommendations for colleges, such as guidance for CS in small colleges [67]. Work on tools and educational technologies started, such as a report on an educational technology for learning programming: a diagnostic compiler that allows preparing, debugging, and executing simple programs [65]. A survey of CER literature post ACM's 1968 curriculum recommendations shows that CER to that date included research on tools and pedagogical aids, activity reports, and course descriptions, but only a limited number of empirical research papers [66]. Central to this decade were debates on programming languages and how to teach programming, theory versus practice, and the role of mathematics versus demands for practical skills from the software industry. New discussion threads included those on programming teamwork, human-computer interaction, and professional accreditation [7].

In the 1980s, the debates on programming language and contributions on how to teach programming continued [68]. Influential contributions included new curriculum recommendations [70], research on tools, and programming environments, such as the "TRY system" for program testing [71]. The decade was also marked with a growing number of empirical research, such as those of predicting success of freshmen [69], and predicting performance on introductory programming courses [72]. Although empirical research was starting to appear, it still took more than 20 years for CER to mature as a primarily empirical research field [1]. In 1989, Denning's influential article [89] was published, coining the term "computing", and made a remarkable impact on the 1991 Computing Curriculum. The 1980s also marked the maturing of Human Computer Interaction (HCI) and usability into research fields, Papert's radical ideas, and demands for more experimental computer science [7].

Central characteristics of the 1990s are the "info-computational turn", which started already in the 1980s; scientists from other fields started looking at their fields as information processes; problem solving becoming a central concept in CER; and diversification of academic contexts, such as those recommended by Computing Curricula 1991 [7]. Also, discussions on pedagogical aspects increased, as is demonstrated by a reflection on constructivism in CER [73], [74], and a paper on active learning [77]. Programming continued to be a central topic of research [75], [78]. One influential paper of the decade is a survey about the Entity-Relationship (ER) models [76], an important topic in software engineering.

In 2000s, the new ACM/IEEE computing curricula split computing into five disciplines of computer engineering, computer science, software engineering, information technology, and information systems, making it no longer a one-size fits all [7]. In addition, CER became established as a research discipline with new venues of dissemination established, and publication of the influential book of Fincher in 2004 [14]. Influential papers of the 2000s include a study on game-based learning in K-12 [79], and several studies on programming: difficulties of novice programmers [80], a multi-national

TABLE 2. Influential CER contributions in Scopus by decade.

	Paper	Title	Publishing Venue	Year	Cites
Pre 1970	[58]	Curriculum '68: Recommendations for Academic Programs in Computer Science	Communications of the ACM	1968	231
	[59]	A University's Educational Program in Computer Science	Communications of the ACM	1967	20
	[60]	Master's Level Computer Science Curricula	Communications of the ACM	1968	7
	[61]	Information Science in a PhD Computer Science Program	Communications of the ACM	1969	6
	[62]	Status of Computer Sciences Curricula in Colleges and Universities	Communications of the ACM	1964	5
1970–1979	[63]	Curriculum '78: Recommendations for the Undergraduate Program in Computer Science	Communications of the ACM	1979	206
	[64]	Form and Content in Computer Science (1970 ACM Turing Lecture)	Journal of the ACM	1970	73
	[65]	The Design and Implementation of a Table Driven, Interactive Diagnostic Programming System	Communications of the ACM	1976	29
	[66]	A Survey of the Literature in Computer Science Education Since Curriculum '68	Communications of the ACM	1977	28
	[67]	A Computer Science Course Program for Small Colleges	Communications of the ACM	1974	27
1980–1989	[68]	Learning to Program = Learning to Construct Mechanisms and Explanations	Communications of the ACM	1986	393
	[69]	Predicting the Success of Freshmen in a Computer Science Major	Communications of the ACM	1984	89
	[70]	A Model Curriculum for a Liberal Arts Degree in Computer Science	Communications of the ACM	1986	82
	[71]	The Try System -OR- How To Avoid Testing Student Programs	SIGCSE Bulletin	1989	77
	[72]	Predicting Performance in an Introductory Computer Science Course	Communications of the ACM	1985	75
1990–1999	[73], [74]	Constructivism in Computer Science Education	SIGCSE Bulletin	1998	231
	[75]	The Case for Case Studies of Programming Problems	Communications of the ACM	1992	153
	[76]	Temporal Entity-Relationship Models-A Survey	IEEE Tr. Know. Data. Eng.	1999	129
	[77]	Active Learning and its use in Computer Science	SIGCSE Bulletin	1996	103
	[78]	The Effects of Comments and Identifier Names on Program Comprehensibility	J. of Progr. Languages	1996	101
2000–2009	[79]	Digital Game-Based Learning In High School Computer Science Education	Computers & Education	2009	953
	[80]	A Study of the Difficulties of Novice Programmers	SIGCSE Bulletin	2005	519
	[81]	A Multi-National, Multi-Institutional Study of Assessment of Programming Skills	ITiCSE WGR	2001	431
	[82]	A Meta-Study of Algorithm Visualization Effectiveness	J. Visual. Lang & Computing	2002	399
	[83]	The LilyPad Arduino: Using Computational Textiles in CSE	SIGCHI Conference	2008	341
2010-2020	[84]	The Scratch Programming Language and Environment	ACM TOCE	2010	592
	[85]	Failure Rates in Introductory Programming Revisited	ITiCSE Proceedings	2014	271
	[86]	Online Python Tutor: Embeddable Web-Based Program Visualisation	ACM SIGCSE Proceedings	2013	189
	[87]	To Block or Not to Block: Students' Perceptions of Blocks-Based Programming	ICIDC Proceedings	2015	176
	[88]	The Fairy Performance Assessment: Measuring Computational Thinking in Middle School	ACM SIGCSE Proceedings	2012	173

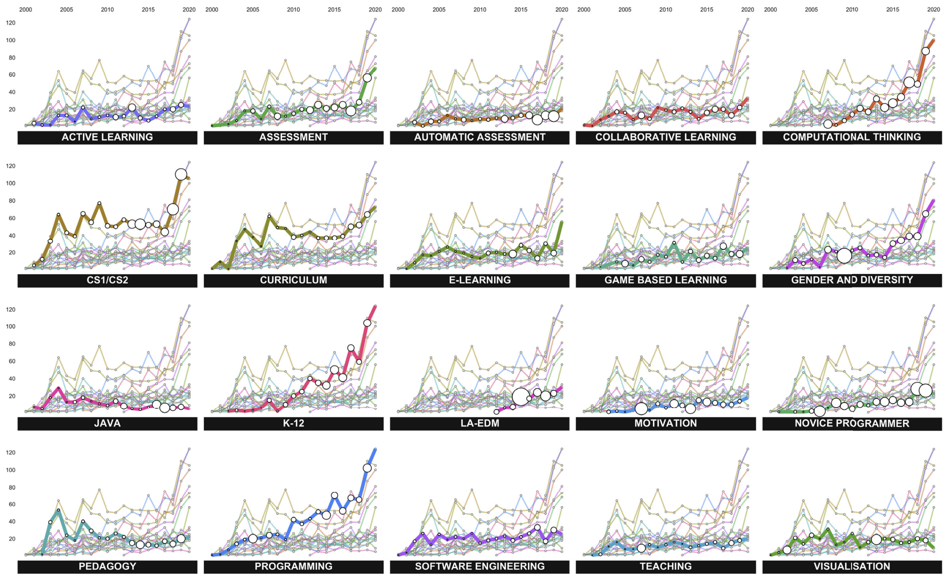
study of assessment of programming skills [81], and an article about the role of Arduino and e-textile design in computing education [83]. Algorithm visualization [82] attracted a lot of interest. A large amount of research was on computer programming. An increase in empirical research papers was seen, as CER continued to build its identity as a respectable academic field [1].

From the 2010s onwards, CER became more established as a research discipline with formation and maturing of publication venues, establishment of professorships in CER, and a steep increase in the number of publications [1]. The highly cited papers include a paper on the Scratch programming language [84], a study on block-programming in K-12 [87], a paper about failure rates in programming courses [85], a visualization tool for Python [86], and a paper about measuring computational thinking (CT) skills [88].

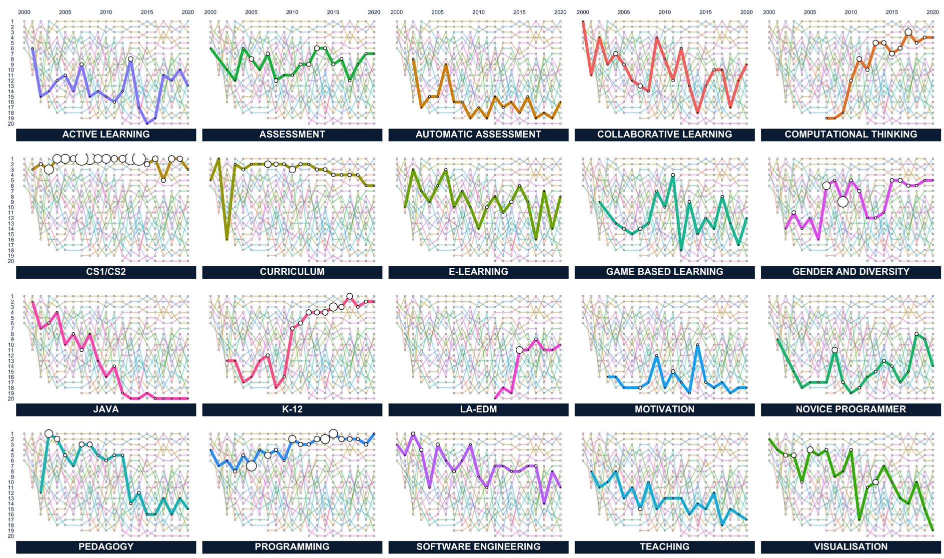
B. ANALYSIS OF KEYWORDS

A unique view to influential topics, themes, and trends in CER during various time periods is revealed by the analysis of keywords. The use of keywords prior to 2000 was rare and inconsistent, and while some publications prior to 2000 listed "themes" of research, those themes are not suitable for analysis. Therefore, reliable analysis of keywords is possible only post 2000. We have plotted the frequencies of top keywords between 2000-2020 (Fig. 2a), ranks of top keywords between 2000-2020 (Fig. 2b), proportions of top keywords between 2000-2020 (Fig. 3a) and proportions of top keywords between 2011-2020 (Fig. 3b).

Analysis of the top 20 keywords reveal several things. First, as observed in Sec. III-A, teaching and learning of programming has always been a dominant theme in CER. Our keyword analysis confirms, loud and clear, that teaching and learning programming continues to be a dominating area of research in the 2000s and 2010s, too. The frequencies of keywords **CS1/CS2** (referring to introductory courses in programming), **novice programmer** and **programming** have all increased during the past two decades (Fig. 2a), and the ranks (Fig. 2b) show that the keyword CS1/CS2, which refers to research on introductory programming courses, has the highest overall rankings among all top 20 keywords, while programming, referring more generally to teaching and learning programming, also has a high ranking. The number of citations of papers with programming-related keywords is also remarkable. As revealed in Sec. III-A, and confirmed in the analysis of keywords, the roots of programming education as a central topic in CER are deep, and this trend continues in the 2000s and 2010s. The strong emphasis on programming is clearly visible in previous analyses of publications in many central venues of CER, too [1], [22], [45]. During all decades of CER, the choice of programming language has been a central concern and has sparked many heated discussions and debates [7]. Our analysis shows that the keyword **Java** has experienced a downward trend, and no longer is among the top 20 keywords post 2010 (Fig. 3b).



(a) Frequencies of Top 20 Keywords

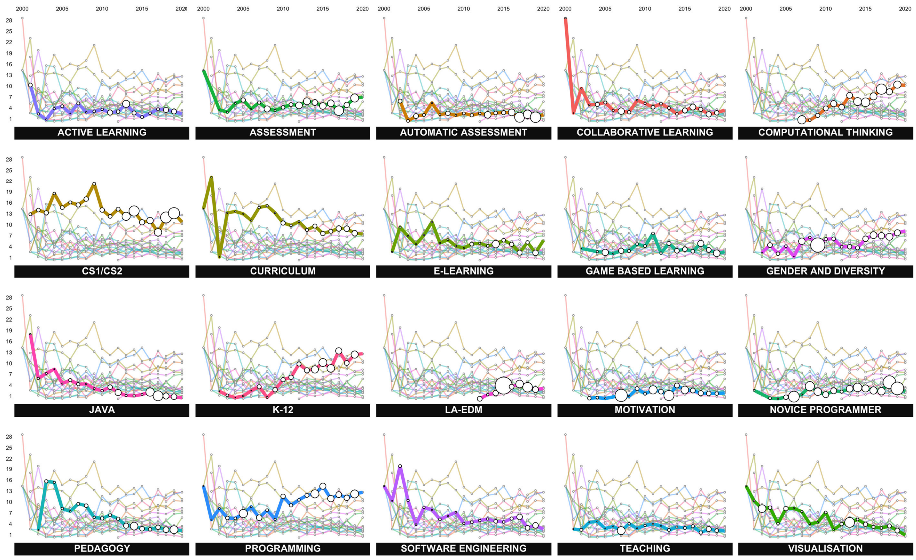


(b) Ranks of Top 20 Keywords

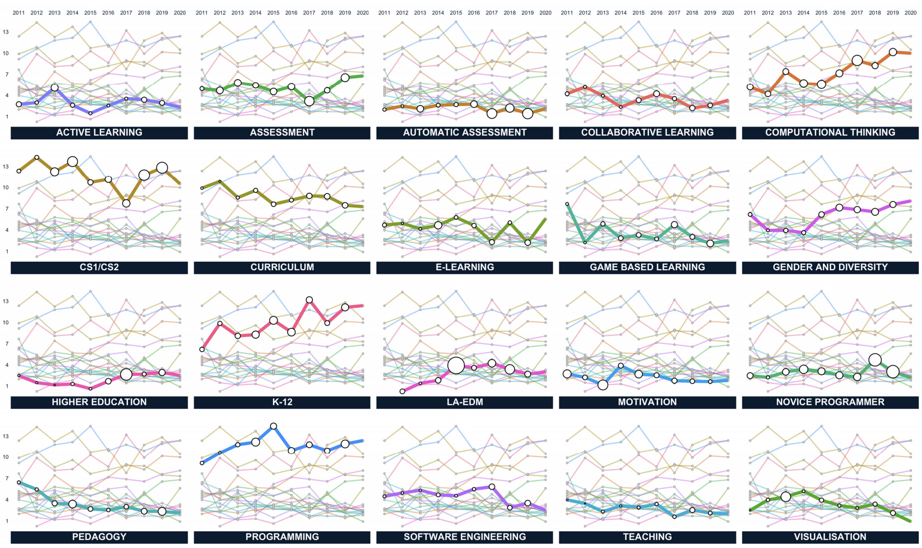
FIGURE 2. **Frequencies and ranks of top 20 keywords (2000-2020).**

Second, two top keywords with a remarkable and steep increasing trend over the past two decades are **K-12** and **computational thinking** (CT). Starting from 2006, when also Jeanette Wing's seminal discussion on computational thinking [90] was published, the rankings of computational thinking and K-12 skyrocketed (Fig. 2b), which is also reflected in the remarkable increase in proportions of these keywords both post 2000, (Fig. 3a), and in the 2010s (Fig. 3b), when these keywords reached the highest proportions within all top keywords, together with CS1/CS2. Similar findings are seen in related meta-analyses [48]. Third, the keyword **LA-EDM** (learning analytics and educational data mining) experienced a steep increase in popularity from 2012 onwards, reaching the top 20 keywords post 2000, despite being rarely seen as a keyword prior to 2010. The explanation is that 2010 was the time when a steep increase in overall research on learning analytics and educational data mining begun [91], and CER followed along with the trend. The beginning of 2010 sparked a steep increase in articles, published books, and new events on learning analytics and educational data mining, such as the annual conference on Learning Analytics and Knowledge (LAK), which was launched in 2011 [91].

Fourth, an increasing trend in research on **gender and diversity** is seen. The keywords **assessment** and **automatic assessment** are also among the top 20, assessment experiencing a turbulent trajectory over the top rankings (Fig. 2b).



(a) Proportions of Top 20 Keywords (2000-2020)



(b) Proportions of Top 20 Keywords (2011-2020)

FIGURE 3. Proportions of top 20 keywords (2000-2020) and (2011-2020). The circles indicate relative share of citations.

In addition to exceptionally popular keywords such as CS1/CS2 and computational thinking, the top 20 keywords include pedagogical keywords, such as **active learning**, and **collaborative learning**, **motivation**, **teaching**, and **pedagogy**, which show that pedagogical considerations are important, or at least that they are often inserted as keywords in articles. Since the emergence and growth of the software industry in the 1960s, **software engineering** has always been a crucial part of CER [7], [63], and much continues to be so in the 2000s and 2010s, while educational technologies, **e-learning**, **game-based learning**, and **visualization** are also well visible in the top 20 topics of research in CER.

C. KEYWORD CLUSTERS

In order to gain an alternative and complementing view about the central thematic areas of CER, we conducted an analysis of frequently co-occurring keywords. The network of keyword co-occurrence was constructed on the basis of keyword communities identified by the Louvain modularity algorithm. Fig. 4 visualises the clusters of interconnected keywords, which represent the major themes of research within CER, post 2000. The analysis identified a total of nine (9) clusters. The four larger and dominating clusters are the orange, red, pink, and green clusters. Smaller clusters consist of blue, light-blue, light-magenta, purple, and gold clusters. While the nodes in the clusters are interconnected, so are the clusters.

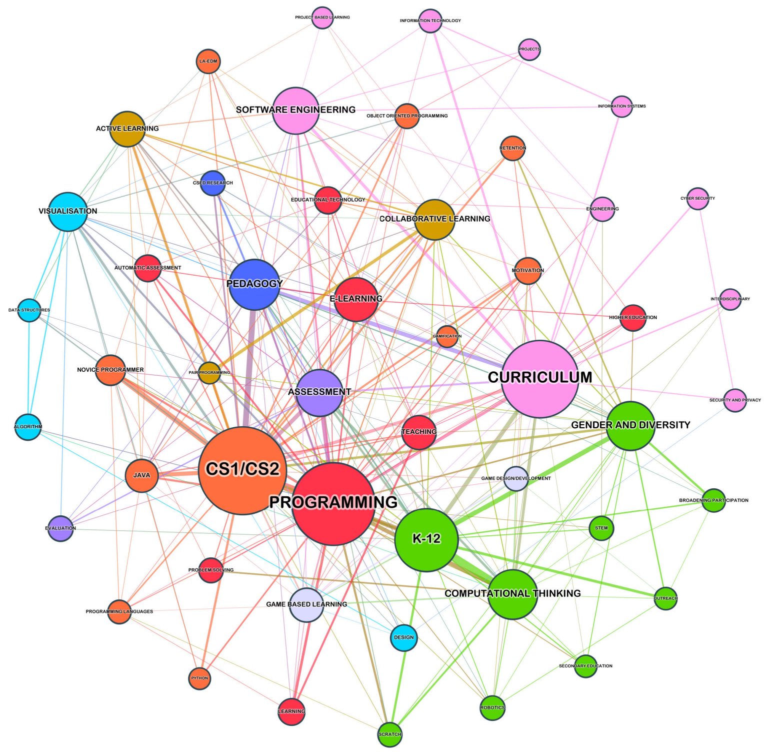


FIGURE 4. **Associations between keywords, with keywords "computer science education" and "computing education" removed. The circle size represents count of keywords, edge thickness denotes frequency of co-occurrence, and colors indicate clusters of keywords.**

The presence of two large and interconnected **red** and **orange** clusters that center around programming-related topics confirms the dominance of research on teaching and learning to program in CER. While the **orange cluster** centers around first courses (CS1/CS2), object-orientation (OOP), Java, and gamification, the **red cluster** focuses more on teaching aids and tools, such as: e-learning, educational technology, and automatic assessment. The large **green cluster** confirms the rising popularity of **K-12 and computational thinking**, and complements the findings with keywords that are typically found together: gender and diversity is highly visible in this cluster, together with broadening participation, and common tools in the K-12 domain: Scratch and robotics. Finally, the large **pink cluster** includes topics centered around **curriculum**, focused on software engineering, information systems, projects, and project-based learning, representing topics typically associated with software engineering. The smaller clusters center around game based learning (light-magenta), collaborative learning, active learning and pair programming (gold), pedagogy (blue), visualization, algorithm design and data structures (light-blue), and evaluation and assessment (purple).

IV. THE BUILDING BLOCKS OF CER

In the co-citation analysis, we investigate the lists of references of the CER articles in our dataset. More specifically, we investigate what groups of articles are typically co-cited by CER researchers. Fig. 5 shows a network of the most co-cited papers, and we interpret these frequently co-cited constellations of papers to be foundational articles: the building blocks of CER. These papers are not necessarily papers in our dataset, but contain papers outside of CER, e.g., learning theoretical contributions, influential reviews, and methodology works. They are the most co-cited papers by the papers in our dataset. The papers are listed in Appendix A. The building blocks are organized in clusters based on the themes of research. Two separate clusters form the building blocks of programming research, with the green cluster having a basis more on general educational theories, while the orange cluster bases its research more on CER-originated theories and pedagogies. Another large cluster consists of the foundational papers on research about K-12 and computational thinking. Smaller clusters make the foundations on educational issues (gray), automated feedback and assessment (light green), meta-analyses (blue), and emotional aspects in programming (brown). It must be noted that we are interested in what CER research co-cites, regardless of the domain of the cited articles. By doing this we are able to investigate to what extent contributions of CER cite works from its own domain (CER), and on the other hand, from other domains, such as the learning sciences, mathematics education, or methodological literature. We follow approaches from previous co-citation analyses [92], [93].

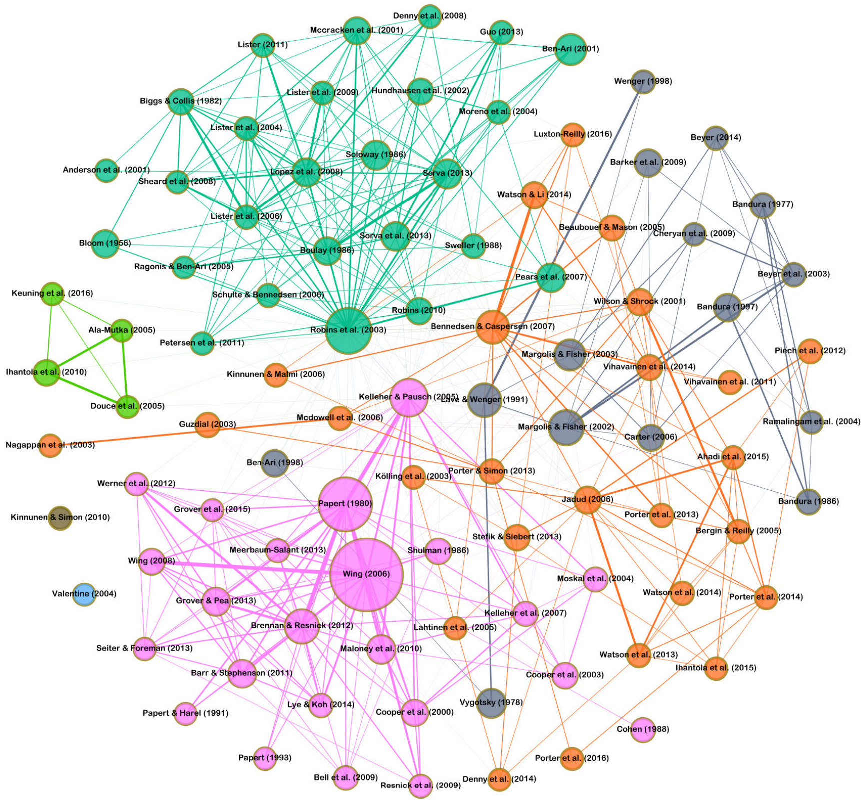


FIGURE 5. Foundational papers (co-citation network).

A. GREEN CLUSTER: THE TRADITIONAL TRACK OF PROGRAMMING RESEARCH

The green cluster includes 26 papers, which are likely to be used as a frame in a lot of the research on learning and teaching computer programming. The top three papers in this cluster are a review of learning and teaching programming [10], an influential reflection on how introductory programming should be taught through algorithm development [68], and an influential paper about the role of constructivism in computer science education [94], where Ben-Ari discusses the importance of distinguishing constructivism in computing education from constructivism in natural sciences. The other papers include influential reviews on introductory programming [8], reviews on program visualizations, papers on exam questions, learning objectives, and articles about the relationship between reading, tracing, and writing in programming. Besides actual studies on computer programming, the foundational papers include five major theoretical foundations of learning, including the SOLO taxonomy [95], cognitive load in problem solving [96], Bloom's taxonomy [97], revised Bloom's taxonomy, and constructivism in computing education [94], with two additional papers discussing the role of SOLO taxonomy in programming. Previous work has identified Bloom's taxonomy, cognitive load theory, SOLO taxonomy, and self-efficacy theories as popular theories in CER [33], [36]. Out of the 26 foundational articles in this cluster, with the exception of the learning theoretical contributions, many papers are from the CER domain: influential reviews, meta-reviews or large-scale questionnaires on teaching and learning programming. A few papers are influential papers on visualization tools. There are no papers on research methods or research design in this cluster.

B. ORANGE CLUSTER: A MODERN TRACK OF PROGRAMMING RESEARCH

The orange cluster includes 26 papers, which are also likely to be used as a foundation in papers that deal with research on teaching and learning to program. Three influential papers are a survey among institutions around the world about failure rates in CS1 courses [98], a study of *compilation behaviour*: approaches that students take when they engage in repeated editing and compiling of their practice programs during learning [99], and a study of success factors in an introductory computer science course [100]. A large share of the other papers are studies on failure rates, success factors, predicting performance, or understanding reasons why students drop out of programming courses. A number of new pedagogical approaches appear, including the ''extreme apprenticeship'' method [101], and several papers on pair programming. As opposed to the green cluster, the orange cluster contains no learning theoretical constructs outside CER but includes pedagogies designed and contextualized specifically for learning

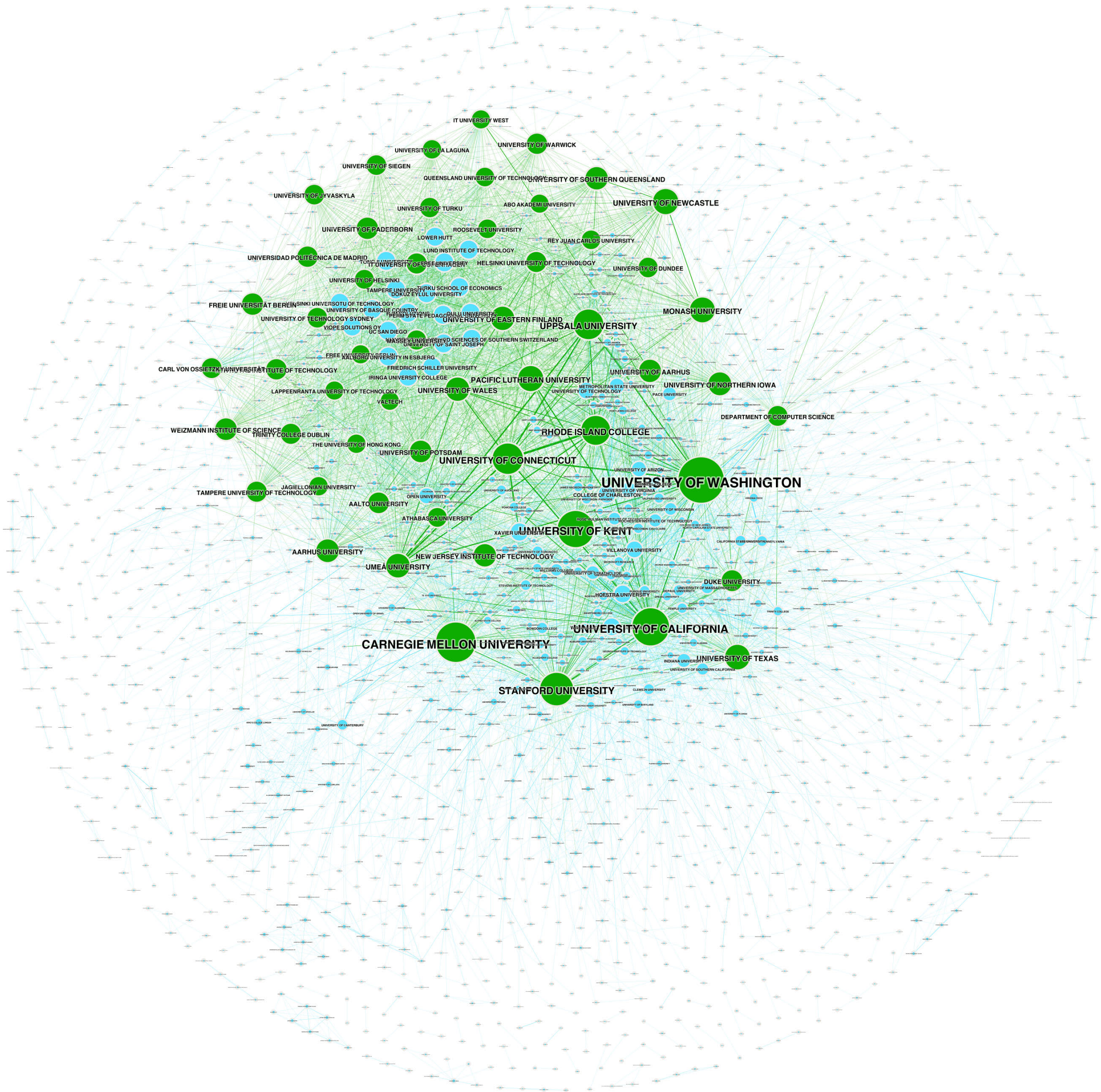


FIGURE 6. Rich club of CER including 52 institutions.

and teaching programming by CER researchers. While the green cluster leans towards common learning theories, the orange cluster leans on pedagogies developed inside CER and therefore presents a new wave of research on developing discipline-specific theories, constructs, and pedagogies. The orange cluster does not include any papers on research methods or research design.

C. PINK CLUSTER: CT AND K-12

The pink cluster consists of 23 papers, which form the foundation in a major share of papers that deal with research on the K-12 and computational thinking track. The most influential paper in this cluster is Jeanette Wing's discussion paper from 2006 [90], which marked the time when the term ''computational thinking'' entered the common computing education vocabulary. The cluster includes Papert's seminal book on Mindstorms [102], and a taxonomy of programming environments [103]. In addition, a number of reviews on CT and frameworks for assessment of CT skills are included. The cluster includes a number of foundational papers on educational technologies and pedagogies; LEGO Mindstorms, Scratch, Alice, and Computer Science Unplugged. This cluster contains one methodological contribution: Cohen's book on statistical power analysis [104], including the Cohen's d for estimating sample sizes and evaluating strengths of statistical claims. This suggests that research papers in the CT and K-12 track have used Cohen's methods, e.g. the effect size coefficient in their research on testing pedagogies, educational technologies or CT interventions and their relationship to, e.g., learning outcomes or motivation. These articles are co-cited by a major share of CER in the CT and K-12 domain. There are many recognised contributions to CT (e.g. [105]–[108]), but not all of them were found in the top co-cited articles in our analysis.

D. THE GRAY CLUSTER: FOUNDATIONS ON SOCIAL ASPECTS AND DIVERSITY

The gray cluster covers the foundations of research on social aspects and diversity, including a book on the gender-gap in computing [109], an influential book on situated learning, the social aspects of learning, stressing the importance of communities of practice, and knowledge creation in communities [110]. The cluster involves grand works on education including that of Vygotsky [111], Bandura's self-efficacy [112], and constructivism. Self-efficacy, constructivism, and communities of practice were identified as top referenced learning theories in a recent review of CER, dispite having a relatively limited overall reach within CER [36]. The cluster also includes articles about gender differences, and mental models in programming. The cluster represents the building blocks of work in CER research that deals with social aspects, diversity, communities of practice, and knowledge building in communities.

E. SMALL CLUSTERS: AUTOMATED FEEDBACK, META-ANALYSES, EMOTIONS

The light green cluster includes four influential papers on automated feedback and assessment in programming, forming the foundations for research on automatic assessment. The blue cluster is based on a seminal meta-analysis of SIGCSE proceedings [15], acting as a building block for reviews and meta-analyses of CER. The brown cluster includes a lone paper on the emotional toll of programming assignments in introductory programming, acting as a building block for research on affective aspects in programming.

V. THE KNOWLEDGE CREATORS OF CER

In this section we look into the knowledge creators of CER by zooming in on influential institutions and authors, and their collaboration networks. First, we conduct a ''rich-club'' analysis of CER institutions (V-A), followed by a peek into the most influential authors (V-B) and geographical distribution of CER (V-C).

A. A DOMINATING RICH CLUB OF 52 INSTITUTIONS

Our data contains some 4011 unique affiliations (institutions) that have produced CER. The functioning of collaborative groups is the backbone of efficient co-creation of knowledge [113]. In a network of collaborators, the concept of *rich club* refers to a situation in the network, where highly-connected nodes or hubs interact primarily among themselves, indicating a dominance, or oligarchy, of teamwork in a way, which may make the network less collaborative as a whole. In other words, a rich club is a small but dominating subset in a network [114]–[116]. In learning contexts, students who are left outside of a ''rich club'', may become isolated, lose motivation, or become underachievers [117]. It is not uncommon for high achieving students to build a rich club, excluding lower achievers from information exchange [117]. All forms of computing are done within networks of people, and the history of computing is also a history of discrimination, biases, well-functioning collaborative groups, and various forms of rich clubs. In this case, we make an investigation into institutional-level rich clubs within the discipline of CER.

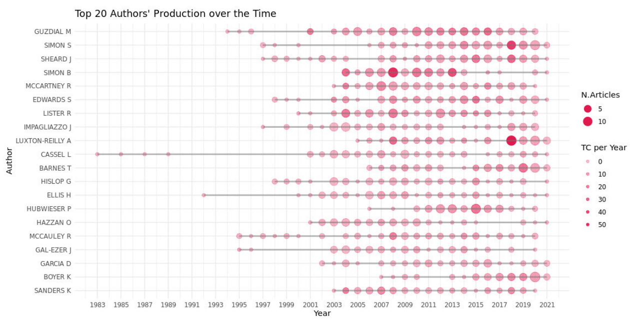


FIGURE 7. **20 influential authors' production.**

Our analysis shows that out of all the 4011 unique affiliations (institutions) in the data, some 1.3% ($n = 52$) belong to a rich club (see Fig. 6). The 52 institutions in the rich club (shown with green color in Fig 6, and listed in Appendix B) were involved in some 9% of all articles in the dataset. Those 9% of all articles attracted a share of 18% of citations out of all citations of articles in our data. The articles of rich-club institutions were cited on average some 15.1 times as compared to the average of 7.7 citations of all articles. The density of the rich club, which indicates how likely each of the members is to have interacted with all other members, was .88. This indicates that most members of the rich club have collaborated with most other members in the rich club. Some $n = 12$ (23%) of the institutions in the rich club are based in the US, $n = 9$ (17%) in Finland, $n = 5$ (10%) in Germany, and $n = 5$ (10%) in Australia, followed by $n = 4$ (8%) in Denmark, and $n = 4$ (8%) in the UK. Other countries include Spain ($n = 3$), Sweden ($n = 3$), and Canada, China, Ireland, Israel, New Zealand and Poland, all with one institution in the rich club. One institution in the rich club is multinational.

B. INFLUENTIAL AUTHORS

Fig. 7 shows the top 20 influential authors (in number of contributions in our data) and their production over time. The list is not intended as a ranking list, but rather as a sample of some of the giants of CER: authors who push the boundaries of CER. Many other influential authors exist. There is an equal balance of 10 females and 10 males within these 20 influential authors.

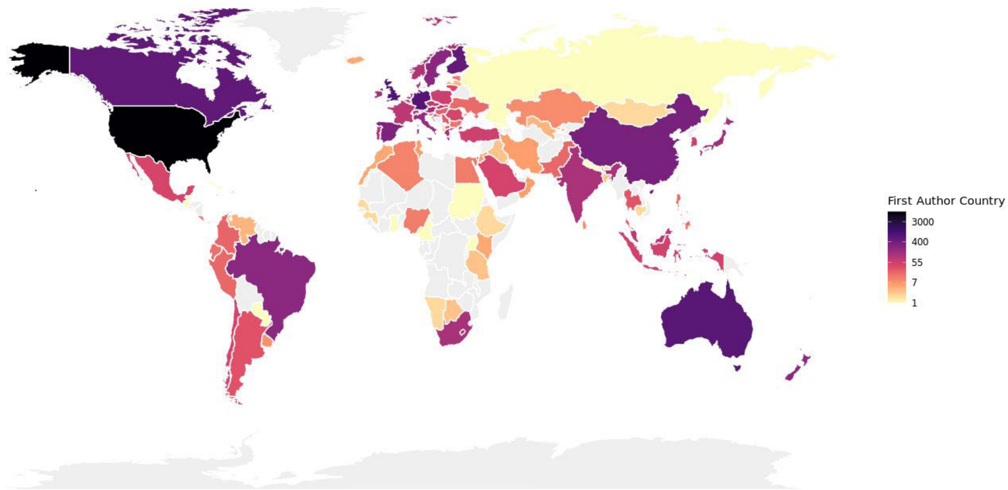


FIGURE 8. Global distribution of CER.

The sample of 20 influential authors include **Mark Guzdial**, a multiple-award-winning author and inventor of the media computation approach to learning introductory computing; **Simon**, whose research focuses on programming education, and who is also known for inventing the Simon's system for classifying CER; **Judithe Sheard**, a long-time and awarded CER researcher, focusing particularly in the areas of educational technology, student learning behavior, and assessment; **Beth Simon**, with her research focusing on online and technology-enhanced learning and curriculum development for K-12; **Robert McCartney**, an awarded CER researcher and former editor of the ACM Transactions on Computing Education; **Stephen H. Edwards**, focusing on various CER projects in automatic assessment, metrics, gamification, and innovative teaching methods; **Raymond Lister**, a computer scientist and CER researcher specialising in understanding the learning and teaching of computer programming; **John Impagliazzo**, author of numerous books, articles, contributor in computing education, computing history, and computing accreditation and outcome assessment, and receiver of many IEEE and ACM/SIGCSE awards; **Andrew Luxton-Reilly**, a significant contributor to the international CER community with several senior roles, including membership on the ACM SIGCSE Executive Board; **Lillian N. Cassel**, working currently on the ACM data science curriculum task force with numerous other interests and contributions within the computing and CER domains; **Tiffany Barnes**, her research focusing on AI for education, educational data mining, serious games for education, health, and energy, CER, and broadening participation; **Gregory Hislop**, with his current research focusing on the educational value of student participation in humanitarian free and open-source software (HFOSS) projects, a contributor to numerous initiatives and efforts in computing and CER; **Heidi Ellis**, a founding member of the HFOSS project, whose research interests include software engineering education, open-source software, and tools for biological data analysis; **Peter Hubwieser**, whose research focuses on empirical investigation of learning processes in computer science (definition, measurement, and evaluation of competencies, skills, and knowledge structures); **Orit Hazzan**, with her research focusing in computer science, software engineering and data science education; **Renée A. McCauley**, her scholarly activities revolving around computer science education, and including: teaching, curriculum development, and collaborative education-based research; **Judith Gal-Ezer**, her research in CER focusing, among other topics, on algorithmic thinking, the teaching of subjects such as recursion, complexity and efficiency, misconceptions in computer science, reduction and non-determinism; **Dan Garcia**, a leader of the "CSforALL" movement, winner of numerous awards, having served on numerous boards such as the ACM Education Board, the College Board Computer Science Principles Development Committee, and the most frequent SIGCSE author in its 50-year history with over 61 submissions; **Kristy Boyer**, with her research focusing on CER, AI in education, intelligent tutoring systems and dialogue systems; **Kate Sanders**, with her research interest in empirical CER, a participant in the bootstrapping CER workshop in the 2002-2003, and a prominent author in SIGCSE, ITICSE, and ICER.

C. COUNTRIES

The global distribution of CER, shown in Fig. 8 shows strong USA dominance in CER. The top ten contributing countries in CER are USA, Finland, Germany, Denmark, Australia, UK, Spain, Sweden, Canada and New Zealand.

VI. THE DISSEMINATION OF CER

Dissemination (Lat. disseminare, "scattering seeds") means broadcasting a message from a sender to a number of receivers. In the early days of CER, outlets such as SIGCSE Technical Symposium mostly published experience reports, while new venues in the 2000s, such as ICER started to demand papers with a clear theoretical base, which draw on previous research, and share a strong empirical basis. The venues of dissemination, and their habits and cultures play a crucial part in the evolution of a scientific discipline. From an author's perspective, the selection of a suitable publication venue is guided by a set of attributes including: intended audience, reputation of the venue, turnaround time, acceptance rates, page limits, citation metrics, and institutional rankings.

A. TOP 20 PUBLICATION OUTLETS OF CER

Our dataset included CER publications from 1840 distinct sources. Table 3 shows the top 20 publication outlets in the data with regards to numbers of articles published. Some 68% of publications in our dataset were published in the top 20 outlets, and those articles have attracted some 73% of all citations in the dataset. The top 20 publication outlets (Table 3) include the well-known venues that exclusively publish CER, introduced in Sec. II.

Top 20 venues (Table 3) include the two major journals of CER, *Computer Science Education* (CSE; row 7) and *ACM Transactions on Computing Education* (TOCE; row 12), formerly known as *Journal on Educational Resources in Computing* (JERIC, row 14), which was a pioneering venue in advancing tools-research in CER. ACM TOCE (row 12) has the second highest mean citation rate among the top 20 venues, and only 1% of its articles have been never cited. ACM JERIC (row 14) also has a high mean citation rate of 20.5. The mean citation rate of CSE is 11.9 (Table 3). With regards to conference proceedings, the ACM's Special Interest Group in Computer Science Education's (SIGCSE) Technical Symposium (row 1) has by far published the largest share of CER (some 23%) in our dataset. Other well-known CER outlets include ITiCSE (row 2), ACE (row 11), Koli Calling (row 6), and ICER (row 5). SIGCSE, ITiCSE, Koli Calling, ICER, and ACE form the core of conferences, which are dedicated to exclusively publishing CER [7], [14], [54]. The new CSERC conference is found on row 19, and the two well-known conferences dedicated to publishing K-12 research, ISSEP and WIPSCE are found on rows 8 and 13.

With regards to other venues, where CER is known to be occasionally published [7], our data shows publications in *IEEE Transactions on Education* (row 16), *Communications of the ACM* (row 18), *IEEE/ASEE Frontiers in Education Conference* (row 4), *IEEE Symposium on Visual Languages and Human-Centric Computing* (not in top 20), *Informatics in Education* (not in top 20), *LATiCE* (not in top 20), the Psychology of Programming Interest Group (PPIG), (not in top 20), and the Empirical Studies of Programmers (ESP), (not in top 20) [54].

In addition to these well-known venues, the top 20 venues (Table 3) include: the *ASEE Conference & Exposition* and *IEEE Educon*, both focusing on engineering education, the *ACM Annual Computer Science Conference*, publishing

TABLE 3. Top 20 venues for dissemination of CER.

	Publication Venue	Start	End	Cit	$\bar{x}$	M	0	5
1	SIGCSE Technical Symposium	1972	2020	.28	9.5	4	.23	.57
2	ITiCSE Conference	1997	2020	.16	7.3	2	.27	.67
3	SIGCSE Bulletin	1969	2009	.04	6.4	3	.22	.69
4	Frontiers in Education (FIE)	1982	2020	.02	3.2	1	.32	.8
5	ICER Conference	2005	2020	.06	15.1	6	.16	.42
6	Koli Calling Conference	2006	2020	.02	6.3	2	.2	.66
7	Computer Science Education	1988	2020	.04	11.9	5	.17	.49
8	ISSEP Conference	2005	2020	.01	4.5	2	.17	.69
9	ASEE Conference & Expo	1995	2020	.00	1.9	1	.39	.91
10	ACM Ann. CS Conference	1983	1990	.00	0.2	0	.94	.99
11	ACE Conference (Australasian)	1996	2020	.00	5.0	2	.18	.69
12	ACM TOCE	2010	2020	.04	21.8	10	.01	.28
13	WIPSCE Conference	2012	2020	.00	4.2	1	.25	.75
14	ACM JERIC Journal	2001	2009	.02	20.5	9	.09	.45
15	IEEE Educon Conference	2010	2020	.00	4.6	3	.1	.7
16	IEEE Transactions on Education	1973	2020	.01	15.5	9.5	.06	.29
17	SIGITE Conference	2004	2020	.00	3.5	2	.2	.77
18	Communications of the ACM	1964	2020	.02	28.2	7.5	.14	.42
19	CSERC Conference	2012	2020	.00	2.8	1	.31	.9
20	FDG Conf. (Digital Games)	2011	2019	.01	9.9	5	.06	.53
				.73	8.3		.23	.58

Start = First publication in dataset, End = Last publication in dataset, Cit = Share of citations to venue of all citations in dataset, $\bar{x}$ = Mean citations in venue, M = Median citations in venue, ***0*** *= Percentage of articles with zero citations in Scopus after a minimum of 2 years of publication,* ***5*** *= Percentage of articles with 5 or fewer citations in Scopus after 2 years of publication.*

broadly on computer science and related topics, the *Special Interest Group in Information Technology Education (SIGITE) Conference*, and the *International Conference on Foundations on Digital Games (FDG)*, occasionally publishing a paper or two e.g. on gamification aspects of CER. Some well-known venues outside the top-20 that publish CER, together with other topics, include *Informatics in Education* (top 35), *Conference on Software Engineering Education and Training* (CSEE&T, top 25), *Conference on Learning and Teaching in Computing and Engineering* (LATiCE, top 29), *ACM Global Computing Education Conference* (CompED, top 30), *Computer Applications in Engineering Education* (Top 38), and *Computers & Education* (top 26).

B. PUBLICATION AND CITATION PROFILES

The articles in our dataset were spread over a variety of venues. Eleven of the top 20 venues are dedicated to exclusively publishing CER. Many top venues in our data are in neighboring or closely connected disciplines, such as information technology education, general computer science, engineering education, or general education. Many of the central publication outlets of CER, including 14 out of the top 20 venues, are conferences, and indeed a major share of CER is published in conferences, with only two active journals dedicated exclusively to CER (JERIC was terminated in 2009). The conference-oriented publication tendency of CER is inherited from the CS discipline, where many conferences are also highly regarded, in contrast to many other disciplines of science, where journals are more highly regarded. Our analysis shows that journal articles of CER

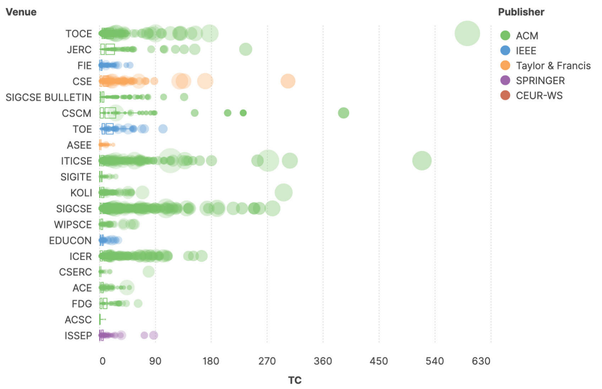


FIGURE 9. Citation profiles of CER outlets. Each dot represents an individual article. The size of the dot represents number of citations to the article. Color is the publisher. *TC* = Total Citations.

receive significantly more citations than conference articles, with an average of 11.92 citations (journal article) versus 6.72 citations (conference article), ($t(3844.79) = 9.15$, $p < .001$, Welch's t-test, two-tailed). So, while people tend to publish CER in conferences, journal articles are cited more.

Fig. 9 shows the citation profiles of the top 20 CER venues. Table 3 shows that in SIGCSE and ITiCSE, many articles have received a relatively small number of citations ($M = 4$, SIGCSE; $M = 2$, ITiCSE), with a few highly cited star papers, while e.g. ACM TOCE, and IEEE ToE appear to have healthier profiles of citations ($M = 10$, TOCE; $M = 9.5$ in IEEE ToE). Some 23% of all articles in the top 20 venues have never been cited according to our data, while some 58% of all articles in the top 20 venues have been cited for five or fewer times, including those with zero citations. For some of the venues, such as ACE and ISSEP, some 69% of articles have received five or fewer citations. For SIGCSE, which is by far the most popular venue in CER, some 57% of articles have received five or fewer citations and some 23% have never been cited. The modest citation rates in CER research have been observed also in previous research [45].

For comparison, in Computers & Education, which was the top 26 venue of CER publications in our data, the mean and median for citations were 42.8 and 14, with some 25% of articles being cited five or fewer times and 11.1% of articles never being cited. Publisher-wise, dissemination of CER is dominated by ACM (see Fig. 9), with 14 of the top 20 venues published through ACM.

VII. DISCUSSION

A. KEYWORD TRENDS (RQ_1)

Our first research question asked: *"How has the publication profile of CER evolved in terms of keyword trends and topics of research? (RQ_1)"* Our analysis of the highly cited papers in the active six decades of CER, as well as the analysis of absolute, relative, and ranked keyword trends post 2000, reveals several things.

First, the topics of the highly cited papers in the past decades connect well with the narrative of the evolution of CER [7]. The times before the 1970s were about laying the groundwork for CER. Publications focused on new curriculum guidelines and recommendations for the then-new discipline of computing. The 1970s were about seeking for identity with increased focus on hands-on work and applications, tools, pedagogical aids and course descriptions. The 1980s saw the emergence of empirical research, and the entry of new topics in software engineering. The 1990s demonstrated an increase in pedagogical considerations in CER, and the 2000s were marked by continuing dominance of teaching computer programming, new topics such as program visualization, and increase in empirical research. The 2010s are characterised by CER maturing as an independent field of study, and the increase of research in K-12 and computational thinking. Cluster analysis revealed two keyword-clusters formed around programming-related topics.

Second, analysis of the most cited papers, and analysis of keyword trends show that teaching and learning programming has by far been the all-time most popular topic of CER. The roots of programming as a central topic in CER are deep. Starting from the first programming textbooks in 1951, through the emergence and growth of the software industry in the 1960s, to the influential ACM curriculum recommendations in 1978, programming has had a strong foothold in CER [63]. A central debate in both the 1970s and 1980s was how to teach programming [7], [68]. In the 1980s also empirical research on programming started to appear [72], and in the 1990s, programming continued to be a central topic [75], [78]. The 2000s saw an ever increasing quantity of research and seminal papers on teaching and learning programming [8], [80], [81]. The trend continued into the 2010s, and the strong emphasis on programming is also clearly visible in previous analyses of publications in many central venues of CER [1], [15], [21], [22], [45].

The dominance of programming warrants some reflections. Programming is, in many ways, at the core of computing, making it an important topic of research. On the other hand, the typical CS curriculum is filled with many topics of equal importance to programming or e.g. data structures, such as theory of computation, databases, machine learning, and concurrency. This may make one wonder: why is CER so concentrated in first-year courses on introductory programming? A number of voices are stressing the importance of CER on topics such as machine learning [118]–[120], design research [108], [121], [122], and STEAM integration [123]–[125]. While entire generations of people are growing up in the middle of machine learning (ML) systems, this development seems to have been given only minor attention in CER. A small but growing body of research shows concrete examples of teaching ML to beginners [118], [119], [126], [127]. New social and ethical dilemmas created by AI also call for reshaping of related training in AI ethics [128], [129].

Other voices call for CER on rigorous pedagogies to add understanding of communities, habits, and cultures into the technology development process [108], [121], [122], [129]. Indeed, it has been argued that future crucial breakthroughs will not only be programming breakthroughs, but increasingly design breakthroughs [108], [121], [122], and that technology innovation requires practitioners well-versed in techno-solutionist methodologies, but increasingly also in social realities to a much larger extent than can be done with typical methods of, e.g., user requirement definition [122], [129]. While ML, design, and ethics have been niche areas in CER, their relevance is increasing rapidly. In addition, while the need for CER on ML is increasing, this also increases the relevance of CER on topics such as basic probability and statistics, currently almost non-existent [118]. While nearly the entirety of publications on CER focus on classical programming, research on how people learn ML and design is needed, not to mention the other underresearched topics [108], [118], [129]. We hope to see increasing trends of CER on these topics in the near future.

Third, keyword trends in the past two decades indicate that, in addition to programming, computational thinking (CT) and K-12 are top trends. Indeed, the keyword analysis shows that the keywords "K-12" and "computational thinking" (CT) are among the most popular keywords in the CER publications, with a significant increase in popularity over the past decade. While the skyrocketing trends of CT and K-12 were launched after the publication of the seminal paper of Wing in 2006 [90], the roots of CT are in the 1950s in the work of, e.g., Donald Knuth, Edsger Dijkstra, and many others [105], [130], and, e.g., in Seymour Papert's groundbreaking Mindstorms [102]. Other common labels for CT are, e.g., algorithmic thinking, computational making, and computational participation [131], [132]. After CT was made popular by Wing in 2006 [90], the number of CT-related publications began to grow, with the annual output of CT publications accelerating rapidly, reaching some 430 articles in 2019, with an annual percentage growth rate of 61.2% [48]. A wide repertoire of approaches for computing in K-12 exist, many of them focused on programming, with some 27.2% of CT articles including programming- and coding-related terms [48]. A common approach is block-based programming with Scratch [133], and educational robotics [134]. Also in the context of CT, a large amount of CER seems to concentrate on rule-driven programming or, e.g., logic puzzles. Future recommendations [135] include changing a public misconception of "computer science = programming", changing a common stereotype that only social misfits can do programming, increasing basic training on machine learning in CT [119], and abandoning logic puzzles in favour of well-established and brilliant pedagogical toolkits, such as the CS-Unplugged [136], [137].

Fourth, in addition to programming and computational thinking, the top topics of research include gender and diversity, assessment, and learning analytics & educational data mining. Other keywords in the top 20 include pedagogical keywords, such as collaboration in learning, and research on software engineering education, which has been a fundamental part of CER throughout the years. Also, research on educational technologies, game-based learning, and visualization are strongly visible in the top keywords. Research on educational technologies, games, and visualization tools all belong to the category of tools-research within CER.

From early on, an important category of CER has been that of tools and pedagogical aids, and this is reflected in the cluster analysis of keywords, especially in the red cluster with the focus on educational technologies, automatic assessment and e-learning. A survey of CER articles published prior to 1977 shows that reports on tools, pedagogical aids and course descriptions were common, while empirical research was rare [66]. Empirical research started to appear in the 1980s, e.g., in the Psychology of Programming Interest Group (PPIG) publications, and through the influential work of Soloway [68]. While empirical research is nowadays a fundamental part of CER, research within the tools category has remained equally important. System papers and experience reports feed research by presenting innovations and allowing the community to familiarize itself with novel ideas even before any more rigorous evaluations have been carried out. A crucial trajectory of CER is that of tools: from early diagnostic compilers [65] to the automatic assessment and visualization tools of today, as is demonstrated by the top keywords and keyword clusters.

B. BUILDING BLOCKS OF CER (RQ_2)

Our second research question asked: *"What do citation and co-citation metrics reveal about the foundational work in the CER discipline? (RQ_2)"* Our cluster analysis of co-cited papers shows several things with regard to how CER researchers build on previous work.

First, clusters of foundational work—articles that researchers of CER are likely to cite in their research—were formed around the most researched areas of CER. Two clusters were formed around teaching and learning programming, with their own separate orientations. While one cluster on programming (green, traditional track) includes a combination of influential reviews and works, it also includes a set of classical works on education and learning, namely the cognitive load theory, Bloom's taxonomy, SOLO taxonomy, and constructivism. Bloom's taxonomy, cognitive load theory, SOLO taxonomy, and self-efficacy theories have been identified as much used learning theories in CER [33], [36]. Another cluster on programming (orange, the modern track), contains no theoretical contributions on learning outside of CER. Instead, it includes pedagogical approaches designed for the purposes of learning and teaching programming, by CER researchers, such as methods called "extreme apprenticeship" and pair-programming.

Second, while the green cluster on programming leaned more heavily on classical work in education, the orange cluster was more oriented towards developing its own CER-specific constructs and pedagogies. Previous meta-work has investigated how CER builds on previous theories and models and how CER develops its own theoretical constructs (TC) [34]–[36], [138]. Findings from a recent review, looking at CER from the viewpoint of influential learning theories, identified Csikszentmihalyi's flow theory, learning styles theory, self-efficacy, problem-based learning, and communities of practice as the most used, despite having an overall limited reach within CER [36]. Other findings show that CER-specific TC's are being developed in the areas of emotions, attitudes, self-efficacy, and other areas [138], and cohorts of articles are emerging that use these constructs as their base [34], [35]. Also, much previous work seem to draw on education and psychology [34]. Overall, a modest use of TCs has been found, and in many cases TCs were merely mentioned in citing articles with no clear connection or deeper application of the TCs [35]. The findings, on one hand, show modest use of theories and background with no prevailing theoretical works that are broadly applied and, on the other hand, the maturation of CER, building its own theoretical constructs and claiming its independence [34], [35], [138].

A somewhat restricted set of learning-theory use has been observed in previous reviews [36]. This finding is supported by the findings of this paper. Previous research has also identified the use of outdated and largely debunked learning theories, such as the learning styles theory, within CER [36]. While development of CER-specific theories of learning is important, it is also important that researchers build their work upon previous research. It seems fair to recommend that CER researchers become better informed of the rich repertoire of available learning theories (eg. [36]), aim to avoid debunked and outdated theories, and do not build their research only upon the most commonly used theories. New and innovative openings may be found by learning to appreciate the richness of available theoretical contributions.

Third, one major (pink) cluster was formed around computational thinking and K-12. This cluster includes a heavy presence of foundational tools and pedagogies in CER, including LEGO Mindstorms, Alice, Scratch, and CS-Unplugged. In addition to the most co-cited works in the K-12 and CT domains, revealed by our analysis, there exists other influential contributions in this domain, too. Indeed, it is important to ask, why does a majority of researchers co-cite these specific works and not some others. Other influential contributions include those authored by e.g. Denning and Tedre [105], Dagiene and Stupuriene [106], Mannila *et al.* [107], and Pears *et al.* [108], and from the times before Wing's discussion paper in 2006 [90], influential works include those of e.g. Knuth, Dijkstra, and many others [105], [130]. While Wing's paper was not a research paper but a discussion paper, it marked the starting point for the new trend of CT and K-12 research, and has gained enormous popularity, and high numbers of cites and co-cites. But it is indeed relevant to ask, to which extent the majority of CER researchers on this domain are aware of other influential works beyond the most co-cited ones, and what could be done to increase the community's awareness? A recent analysis about the publication trends in CT [48] delves in this theme with more depth. One possible explanation for amplified citations to certain works is the Matthew effect [139], which can be summarized as "the rich get richer and poor get poorer", and may explain the heightened visibility of contributions by researchers of acknowledged standing and reduced visibility of contributions by other authors [139]. Moreover, the amplifying of already cited research by some search engines may generate echo chambers, which lead to researchers overlooking other influential works, while only citing the recommended and most cited of previous works.

Fourth, the gray cluster represents foundations in research that deals with diversity, gender, and social aspects. This cluster includes a set of educational work including that of Vygotsky, Bandura's self-efficacy, and constructivism. Self-efficacy and constructivism have been identified as common learning theories in CER [33], [36]. Smaller clusters represent work on social aspects and diversity, automatic assessment, meta-research and emotions. In related research, self-efficacy and constructivism have been identified to be among the commonly cited learning theories in CER [36].

Fifth, only a single contribution on research methodologies was found in the foundational papers, in the computational thinking and K-12 cluster. Neither of the clusters on computer programming had any methodological contributions. Previous meta-research has looked into methodology use in CER papers [28], [33], [38], and found, e.g., that many papers make use of statistical techniques beyond descriptive statistics [28], and that commonly used statistical tests include the t-test, χ^2 test and Mann Whitney Wilcoxon test [38]. Our analysis found only one methodological contribution, Cohen's statistical power analysis [104], which introduces, e.g., Cohen's d for calculating effect sizes. This is in contrast to educational technology research, where methodological contributions are much more widely referred [92]. One explanation for the low number of methodological contributions is that CER researchers may be uneducated on the literature on research designs and research methods. Another explanation is that while statistical methods, and research designs may be commonly used in CER, citations for them are not widely presented. A recent review of CER published in 2014-2015 [37] supports the latter assumption, when it comes to CER published in the past decade. While quantitative evaluation methods are used by CER researchers, and papers frequently report results on pedagogies, curriculums, and tools, many papers lack properly reported research objectives, goals, research questions, or hypotheses, description of participants, study design, data collection and threats to validity [37]. It seems that while CER researchers are, at least in recent times, actively using research methods to conduct empirical studies, not all norms of reporting are fully met, including those of research design and research methods, as well as proper citations of methodology work.

The ad-hoc nature and lack of research rigor in CER begun to attract a lot of attention in the beginning of 2000s [140], [141]. It became well known that a large share of CER concentrated on describing course contexts and teaching practices in an anecdotal manner, and the need to increase methodological rigor, build connections to learning theories, and need to expand qualitative research became evident [140]–[143]. A series of workshops to educate academics with a computing background on how to design and conduct CER, conduct qualitative research such as phenomenography, and proposals for CER-specific research frameworks started to emerge [13], [140]–[143]. In 2008, an increasing number of papers with increased methodological rigor was observed in CER conferences [143]. While the value of empirical research has been on the increase for a while, it is relevant to highlight the important focus of CER in addressing and solving pragmatic challenges related to teaching and learning [144]. The series of research on developing research frameworks of CER, and on how to arrange workshops to train methodology to CER researchers forms an important part of CER. Future scientometric studies and reviews could have a special focus on this series of articles, which focuses on workshops and training on CER methodology.

C. CREATORS OF CER (RQ_3)

Our third research question asked: *"How have the knowledge creators (authors and institutions) and their collaboration shaped the discipline of CER and its communities? (RQ_3)"* Our analysis reveals the following.

First, our findings from analyzing the institution-level collaboration networks show the presence of an elite group of 52 institutions, which collaborate extensively with each other, and who conduct and publish a remarkable share of CER. The share of articles published by the rich core attracts significantly more citations than other articles in the dataset. Some in the research community might find such inequality alarming. On the other hand, it is likely that the success of the rich-club institutions has resulted from their ability to develop broad expertise and partnerships, and excel in both variety and quality of research. An important question is: to what extent do the core of elite universities have a wider impact on driving science forward, and to what extent are the less successful institutions able to benefit from their associations with the institutions of the rich core?

Research as an enterprise is fundamentally driven by collaborative relations and their dynamics. Previous research has shown cases in which elite circles of academic institutions overattract research funding, while at the same time they collaborate with members of the very same elite circles [116]. Membership in a "rich club" of academic institutions may offer easier access to other elite members, give strategic advantages as compared to non-elite members, and provide the elites with the potency to boost their power by controlling access to opportunities [116], [145]. In short, research funding may mainly go to rich clubs of science, who have control over topics and scholarships and who decide which research is valued, and which is not, potentially causing inequalities and discrimination. In this sense, future research could zoom in on the research produced by the rich core, identify leading trends and themes of research, the degree of interdisciplinary research, and the quality of collaborations with institutions that remain outside of the rich core.

Second, our findings show a diverse set of influential authors with an equal balance of female and male researchers, working on a diverse set of topics, and pushing the boundaries of CER. Third, geographical distribution of CER shows strong USA dominance, followed by a selection of high-income countries. The dominance of high-income countries in CER, especially USA, has been observed in a number of previous research studies [42]–[45], too. Also, previous research has shown that while collaboration in CER is growing, it still mostly happens within the same country [42]. It seems that only a few papers originate from, or address challenges of computing education in the Global South, a situation observed already back in 2010 [146]. For example, research from Africa, a home to 1.2 billion people, is rarely seen. The CER community needs to act in order to better serve the needs of computing education for all, in all countries, not just in a predominantly Western few.

D. DISSEMINATION CER (RQ_4)

Our fourth research question asked: *"How can the central venues of dissemination of CER be characterised with regards to their publication profiles and citation practices? (RQ_4)"* Our analysis shows the following. First, the top 20 publication outlets include all the well-known publication outlets of CER, and together they publish some 68% of all CER, attracting some 73% of all citations. Other common venues that publish CER are from closely connected disciplines, such as IT education, general computer science, engineering education, or general education. Second, a major share of CER is published in conferences. By far, the most influential venue with regards to numbers of publications is the SIGCSE Technical Symposium. The tendency to publish in conferences is inherited from the CS discipline. A major share of the CER conferences are published by ACM. Third, while CER is often published in conferences, journal articles in CER are significantly better cited. Fourth, many venues have a small group of highly cited articles and a large group of minimally cited articles. A healthier citation profile is found, e.g., in the ACM TOCE. Fifth, CER suffers from modest overall citation rates. A large share of CER is never cited, and over half of CER is cited five or fewer times. Similar observations have been found in previous research, too [45]. One exception is CER published in *Computers & Education*, attracting a significantly higher number of citations than in any of the top 20 CER venues. Of course, citations are not the only measure of impact, as many articles may be read by interested educational practitioners. In any case, it is good to ask if the current situation is desirable, and if not, what the CER community can do to address the issue.

E. FUTURE RESEARCH

A number of avenues for future research exist. While this research offers a macro-level view of CER on a wide range of topics, areas, authors, and geographical areas, more focused scientometric reviews could be conducted to gain deeper insights into specific research niches, work of specific communities of authors, citation practices, and CER conducted in specific geographical areas. Future research could focus on differences in author networks and keyword trends in different communities of CER around the globe. More in-depth analysis of co-citations could zoom in on how learning theories, methodologies and ideas outside the field of CER are used and applied in CER. While the field of CER has been heavily dominated by USA, and topics such as teaching classical programming, it is important to focus future studies on evolving themes, and on underrepresented geographical areas. Future research could also investigate research produced by institutions in the rich core, identify leading trends and themes, degrees of interdisciplinary research, and patterns of collaboration and networking. Ideas for future research also include publication and citation practices. One area for future research is to investigate how CER is received and talked about in social media, blogs, news and media.

F. LIMITATIONS

The following limitations apply to this research. First, several caveats apply to the data. While Elsevier's Scopus database is well maintained and, in most cases, more accurate than WoS (Web of Science) [147], it is not perfect. The earlier years especially have some issues: missing fields, inconsistent and unstructured keywords, references not perfectly recorded, and mistakes in publication venue names. As one example, an influential article from 2003 [10], published in the journal *Computer Science Education*, is misclassified in Scopus as being published in *International Journal of Phytoremediation*. Another example is reviews that fail to be classified as reviews, and therefore get included in the dataset. Even though we have manually checked author and venue names, used manual and algorithmic methods for combining keywords, and given our best efforts to fix inconsistencies in the databases, detecting and manually correcting all such mistakes is not possible. Data prior to 2000 were found more vulnerable to mistakes and mis-classifications. For example, keywords use was very inconsistent before 2000, and therefore we have conducted keyword analysis only post 2000. Data prior to 1970 were unreliable and needed to be subjected to many manual checkups. In all, by including data from the well-known dedicated venues, carefully controlling the articles retrieved through keywords searches, and with extensive cleaning of the data, we have reached a representative, if not comprehensive, sample of CER.

Second, scientometric methods provide a quantitative view and remain shallow without a qualitative perspective. Scientometrics as a method is severely limited as compared to systematic reviews and meta-reviews, which can capture a number of important perspectives that are out of reach for scientometrics. With scientometrics, it is not possible to analyze methodological rigor, how learning theories are exactly applied, the research designs, the empirical or experimental nature of the work (if not available in article metadata), or the replicability of studies. On the other hand, reviews are typically restricted to a few venues with a limited timeline, and cannot provide a holistic analysis of trends, themes, communities, venues, and their evolution, which can be done with scientometrics. Indeed, both of these two approaches, scientometrics and systematic reviews, are needed, and they complement each other in building a holistic picture of the evolving discipline of CER. In order to turn the numbers into meaning, we have included a combination of perspectives, and involved experts with decades of experience in the field to create an understanding of things.

VIII. CONCLUSION

Over the recent decades, computing education research (CER) has matured into a respectable field of research. New conferences and venues have been formed, methodological quality and rigor has improved, own discipline-specific constructs and pedagogies have been developed, and experience reports and tool-papers have been complemented with an increasing amount of empirical research. While CER has become more international, it is still dominated by few high-income countries, especially the USA. As teaching and learning computer programming still strongly dominates CER as a research topic, it is necessary to reflect on the need for diversifying the common themes of research. Separate threads of CER build on traditional learning theories as well as developing their own discipline-specific pedagogies and constructs of learning. The lack of citations of research methods raises the issue of how to increase the methodological knowledge and skills of CER researchers. While the discipline of CER is led by an elite group of 52 institutions, it is relevant to reflect to what extent the less successful institutions are able to benefit from their association with the rich core. Finally, the low citation rates in many central venues of dissemination call for a response from the CER community.

APPENDIX A
CO-CITED PAPERS

The Pink Cluster (Computational Thinking and K-12)

1) J. M. Wing, "Computational thinking," Communications of the ACM, vol. 49, no. 3, pp. 33–35, 2006
2) S. Papert, MINDSTORMS: Children, Computers, and Powerful Ideas. Basic Books, 1980
3) C. Kelleher and R. Pausch, "Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers," ACM Comput. Surv., vol. 37, pp. 83–137, June 2005
4) K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in AERA, 2012

5) J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch programming language and environment," ACM Trans. Comput. Educ., vol. 10, Nov. 2010
6) S. Grover and R. Pea, "Computational thinking in k–12: A review of the state of the field," Educational Researcher, vol. 42, no. 1, pp. 38–43, 2013
7) V. Barr and C. Stephenson, "Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?," ACM Inroads, vol. 2, pp. 48–54, Feb. 2011
8) S. Cooper, W. Dann, and R. Pausch, "Alice: A 3-d tool for introductory programming concepts," J. Comput. Sci. Coll., vol. 15, pp. 107–116, apr 2000
9) J. M. Wing, "Computational Thinking and Thinking About Computing," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 366, no. 1881, pp. 3717–3725, 2008
10) L. Shulman, "Those who understand: Knowledge growth in teaching," Educational Researcher, vol. 15, no. 2, pp. 4–14, 1986
11) S. Cooper, W. Dann, and R. Pausch, "Teaching objects-first in introductory computer science," in Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03, (New York, NY, USA), pp. 191–195, Association for Computing Machinery, 2003
12) C. Kelleher, R. Pausch, and S. Kiesler, Storytelling Alice Motivates Middle School Girls to Learn Computer Programming, pp. 1455–1464. New York, NY, USA: Association for Computing Machinery, 2007
13) B. Moskal, D. Lurie, and S. Cooper, "Evaluating the effectiveness of a new instructional approach," SIGCSE Bull., vol. 36, pp. 75–79, mar 2004
14) J. Cohen, Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates, second edition ed., 1988
15) Bell, T, Alexander, J, Freeman, I & Grimley, M 2009, 'Computer science unplugged: school students doing real computing without computers', New Zealand Journal of Applied Computing and Information Technology, vol. 13, no. 1, pp. 20-29, 2009.
16) S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for k-12?," Computers in Human Behavior, vol. 41, pp. 51–61, 2014
17) M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, *et al.*, "Scratch: programming for all," Communications of the ACM, vol. 52, no. 11, pp. 60–67, 2009
18) O. Meerbaum-Salant, M. Armoni, and M. M. Ben-Ari, "Learning computer science concepts with scratch," Computer Science Education, vol. 23, no. 3, pp. 239–264, 2013
19) L. Werner, J. Denner, S. Campe, and D. C. Kawamoto, "The fairy performance assessment: Measuring computational thinking in middle school," in Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, SIGCSE '12, (New York, NY, USA), pp. 215–220, Association for Computing Machinery, 2012
20) S. Papert, The Children's Machine: Rethinking School in the Age of the Computer. New York, NY, USA: Basic Books, 1993
21) S. Papert and I. Harel, "Situating constructionism," in Constructionism (S. Papert and I. Harel, eds.), vol. 36, pp. 1–11, Ablex Publishing Corporation, 1991
22) L. Seiter and B. Foreman, "Modeling the learning progressions of computational thinking of primary grade students," in Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research, ICER '13, (New York, NY, USA), pp. 59–66, Association for Computing Machinery, 2013
23) S. Grover, R. Pea, and S. Cooper, "Designing for deeper learning in a blended computer science course for middle school students," Computer Science Education, vol. 25, no. 2, pp. 199–237, 2015

The Orange Cluster (A Modern Track of Programming Research)

1) J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," SIGCSE Bull., vol. 39, pp. 32– 36, June 2007
2) M. C. Jadud, "Methods and tools for exploring novice compilation behaviour," in Proceedings of the Second International Workshop on Computing Education Research, ICER '06, (New York, NY, USA), pp. 73–84, Association for Computing Machinery, 2006
3) B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: A study of twelve factors," in Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education, SIGCSE '01, (New York, NY, USA), pp. 184–188, Association for Computing Machinery, 2001
4) C. Watson and F. W. Li, "Failure rates in introductory programming revisited," in Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, ITiCSE '14, (New York, NY, USA), pp. 39–44, Association for Computing Machinery, 2014
5) A. Stefik and S. Siebert, "An empirical investigation into programming language syntax," ACM Trans. Comput. Educ., vol. 13, nov 2013
6) T. Beaubouef and J. Mason, "Why the high attrition rate for computer science students: Some thoughts and observations," SIGCSE Bull., vol. 37, pp. 103–106, jun 2005

7) A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," in Proceedings of the Tenth Annual Conference on International Computing Education Research, ICER '14, (New York, NY, USA), pp. 19–26, Association for Computing Machinery, 2014
8) M. Guzdial, "A media computation course for non-majors," in Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '03, (New York, NY, USA), pp. 104–108, Association for Computing Machinery, 2003
9) L. Porter and B. Simon, "Retaining nearly one-third more majors with a trio of instructional best practices in cs1," in Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13, (New York, NY, USA), pp. 165–170, Association for Computing Machinery, 2013
10) C. Piech, M. Sahami, D. Koller, S. Cooper, and P. Blikstein, "Modeling how students learn to program," in Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, SIGCSE '12, (New York, NY, USA), pp. 153–160, Association for Computing Machinery, 2012
11) C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "Pair programming improves student retention, confidence, and program quality," Commun. ACM, vol. 49, pp. 90–95, aug 2006
12) C. Watson, F. W. Li, and J. L. Godwin, "Predicting performance in an introductory programming course by logging and analyzing student programming behavior," in 2013 IEEE 13th International Conference on Advanced Learning Technologies, pp. 319–323, 2013
13) S. Bergin and R. Reilly, "Programming: Factors that influence success," in Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '05, (New York, NY, USA), pp. 411–415, Association for Computing Machinery, 2005
14) L. Porter, M. Guzdial, C. McDowell, and B. Simon, "Success in introductory programming: What works?," Commun. ACM, vol. 56, pp. 34–36, aug 2013
15) A. Vihavainen, M. Paksula, and M. Luukkainen, "Extreme Apprenticeship Method in Teaching Programming for Beginners," in Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education, SIGCSE '11, (New York, NY, USA), pp. 93–98, ACM, 2011
16) P. Kinnunen and L. Malmi, "Why students drop out cs1 course?," in Proceedings of the Second International Workshop on Computing Education Research, ICER '06, (New York, NY, USA), pp. 97–108, Association for Computing Machinery, 2006
17) C. Watson, F. W. Li, and J. L. Godwin, "No tests required: Comparing traditional and dynamic predictors of programming success," in Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE '14, (New York, NY, USA), pp. 469–474, Association for Computing Machinery, 2014
18) A. Luxton-Reilly, "Learning to program is easy," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '16, (New York, NY, USA), pp. 284–289, Association for Computing Machinery, 2016
19) M. Kölling, B. Quig, A. Patterson, and J. Rosenberg, "The bluej system and its pedagogy," Computer Science Education, vol. 13, no. 4, pp. 249–268, 2003
20) A. Ahadi, R. Lister, H. Haapala, and A. Vihavainen, "Exploring machine learning methods to automatically identify students in need of assistance," in Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ICER '15, (New York, NY, USA), pp. 121–130, Association for Computing Machinery, 2015
21) L. Porter, D. Zingaro, and R. Lister, "Predicting student success using fine grain clicker data," in Proceedings of the Tenth Annual Conference on International Computing Education Research, ICER '14, (New York, NY, USA), pp. 51–58, Association for Computing Machinery, 2014
22) E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," SIGCSE Bull., vol. 37, pp. 14–18, June 2005
23) P. Denny, A. Luxton-Reilly, and D. Carpenter, "Enhancing syntax error messages appears ineffectual," in Proceedings of the 2014 Conference on Innovation Technology in Computer Science Education, ITiCSE '14, (New York, NY, USA), pp. 273–278, Association for Computing Machinery, 2014
24) L. Porter, D. Bouvier, Q. Cutts, S. Grissom, C. Lee, R. McCartney, D. Zingaro, and B. Simon, "A multi-institutional study of peer instruction in introductory computing," in Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16, (New York, NY, USA), pp. 358–363, Association for Computing Machinery, 2016
25) P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, M. A. Rubio, J. Sheard, B. Skupas, J. Spacco, C. Szabo, and D. Toll, "Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies," in Proceedings of the 2015 ITiCSE on Working Group Reports, ITICSE-WGR '15, (New York, NY, USA), pp. 41–63, ACM, 2015
26) N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik, "Improving the cs1 experience with pair programming," in Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03,

(New York, NY, USA), pp. 359–362, Association for Computing Machinery, 2003

The Green Cluster (The Traditional Track of Programming Research)

1) A. Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," Computer Science Education, vol. 13, no. 2, pp. 137–172, 2003
2) M. Ben-Ari, "Constructivism in computer science education," Journal of Computers in Mathematics and Science Teaching, vol. 20, no. 1, pp. 45–73, 2001
3) E. Soloway, "Learning to Program = Learning to Construct Mechanisms and Explanations," Communications of the ACM, vol. 29, pp. 850–858, Sept. 1986
4) A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson, "A Survey of Literature on the Teaching of Introductory Programming," SIGCSE Bulletin, vol. 39, pp. 204–223, December 2007
5) J. Sorva, "Notional machines and introductory programming education," ACM Trans. Comput. Educ., vol. 13, July 2013
6) J. Sorva, V. Karavirta, and L. Malmi, "A review of generic program visualization systems for introductory programming education," ACM Trans. Comput. Educ., vol. 13, Nov. 2013
7) M. Lopez, J. Whalley, P. Robbins, and R. Lister, "Relationships between reading, tracing and writing skills in introductory programming," in Proceedings of the Fourth International Workshop on Computing Education Research, ICER '08, (New York, NY, USA), pp. 101–112, Association for Computing Machinery, 2008
8) B. D. Boulay, "Some difficulties of learning to program," Journal of Educational Computing Research, vol. 2, no. 1, pp. 57–73, 1986
9) B. S. Bloom, M. D. Englehart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain. New York, NY, USA: Longmans, 1956
10) M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz, "A Multi-national, Multi-institutional Study of Assessment of Programming Skills of First- year CS Students," in Working group reports from ITiCSE on Innovation and technology in computer science education, ITiCSE-WGR '01, (New York, NY, USA), pp. 125–180, ACM, 2001
11) A. Robins, "Learning edge momentum: a new account of outcomes in cs1," Computer Science Education, vol. 20, no. 1, pp. 37–71, 2010
12) C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," Journal of Visual Languages & Computing, vol. 13, no. 3, pp. 259–290, 2002
13) J. B. Biggs and K. F. Collins, Evaluating the Quality of Learning: The Solo Taxonomy: Structure of the Observed Learning Outcome. 1982
14) J. Sweller, "Cognitive load during problem solving: Effects on learning," Cognitive Science, vol. 12, no. 2, pp. 257–285, 1988
15) R. Lister, B. Simon, E. Thompson, J. L. Whalley, and C. Prasad, "Not seeing the forest for the trees: Novice programmers and the solo taxonomy," SIGCSE Bull., vol. 38, pp. 118–122, jun 2006
16) P. J. Guo, "Online python tutor: Embeddable web-based program visualization for cs education," in Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13, (New York, NY, USA), pp. 579–584, Association for Computing Machinery, 2013
17) R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon, and L. Thomas, "A multi-national study of reading and tracing skills in novice programmers," in Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education, ITiCSE-WGR '04, (New York, NY, USA), pp. 119–150, Association for Computing Machinery, 2004
18) A. Moreno, N. Myller, E. Sutinen, and M. Ben-Ari, "Visualizing programs with jeliot 3," in Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '04, (New York, NY, USA), pp. 373–376, Association for Computing Machinery, 2004
19) C. Schulte and J. Bennedsen, "What do teachers teach in introductory programming?," in Proceedings of the Second International Workshop on Computing Education Research, ICER '06, (New York, NY, USA), pp. 17– 28, Association for Computing Machinery, 2006
20) R. Lister, C. Fidge, and D. Teague, "Further evidence of a relationship between explaining, tracing and writing skills in introductory programming," SIGCSE Bull., vol. 41, pp. 161–165, jul 2009
21) A. Petersen, M. Craig, and D. Zingaro, "Reviewing cs1 exam question content," in Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, SIGCSE '11, (New York, NY, USA), pp. 631– 636, Association for Computing Machinery, 2011
22) L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York, NY, USA: Addison Wesley Longman, Inc., 2001
23) P. Denny, A. Luxton-Reilly, and B. Simon, "Evaluating a new exam question: Parsons problems," in Proceedings of the Fourth International Workshop on Computing Education Research, ICER '08, (New York,

NY, USA), pp. 113–124, Association for Computing Machinery, 2008

24) R. Lister, "Concrete and other neo-piagetian forms of reasoning in the novice programmer," in Thirteenth Australasian Computing Education Conference (ACE), 2011
25) N. Ragonis and M. Ben-Ari, "A long-term investigation of the comprehension of oop concepts by novices," Computer Science Education, vol. 15, no. 3, pp. 203–221, 2005
26) J. Sheard, A. Carbone, R. Lister, B. Simon, E. Thompson, and J. L. Whalley, "Going solo to assess novice programmers," in Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '08, (New York, NY, USA), pp. 209–213, Association for Computing Machinery, 2008

Gray Cluster (Foundations on Social Aspects and Diversity)

1) J. Margolis and A. Fisher, Unlocking the Clubhouse: Women in Computing. MIT Press, 2002
2) J. Lave and E. Wenger, Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, 1991
3) A. Fisher and J. Margolis, "Geek mythology," Bulletin of Science, Technology and Society, vol. 23, no. 1, pp. 17–20, 2003
4) L. S. Vygotsky, Mind in Society: The Development of Higher Psychological Processes. Cambridge, Mass., USA: Harvard University Press, 1978
5) A. Bandura, "Self-efficacy: the exercise of control," Self-efficacy: the exercise of control, 1-35. WH Freeman and Company. 1997
6) L. J. Barker, C. McDowell, and K. Kalahar, "Exploring factors that influence computer science introductory course students to persist in the major," in Proceedings of the 40th ACM Technical Symposium on Computer Science Education, SIGCSE '09, (New York, NY, USA), pp. 153–157, Association for Computing Machinery, 2009
7) M. Ben-Ari, "Constructivism in computer science education," in SIGCSE '98: Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, (New York, NY, USA), pp. 257–261, ACM, 1998
8) L. Carter, "Why students with an apparent aptitude for computer science don't choose to major in computer science," in Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '06, (New York, NY, USA), pp. 27–31, Association for Computing Machinery, 2006
9) A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," Psychological Review, vol. 84, no. 2, pp. 191–215, 1977
10) A. Bandura, "The explanatory and predictive scope of self-efficacy theory," Journal of Social and Clinical Psychology, vol. 4, no. 3, pp. 359–373, 1986
11) S. Beyer, K. Rynes, J. Perrault, K. Hay, and S. Haller, "Gender differences in computer science students," in Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, SIGCSE '03, (New York, NY, USA), pp. 49–53, Association for Computing Machinery, 2003
12) S. Beyer, "Why are women underrepresented in computer science? gender differences in stereotypes, self-efficacy, values, and interests and predictors of future cs course-taking and grades," Computer Science Education, vol. 24, no. 2-3, pp. 153–192, 2014
13) E. Wenger, "Communities of practice: Learning as a social system," Systems Thinker, 1998
14) S. Cheryan, V. Plaut, P. Davies, and C. Steele, "Ambient belonging: How stereotypical cues impact gender participation in computer science," Journal of Personality and Social Psychology, vol. 97, no. 6, pp. 1045–1060, 2009
15) V. Ramalingam, D. LaBelle, and S. Wiedenbeck, "Self-efficacy and mental models in learning to program," in Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '04, (New York, NY, USA), pp. 171–175, Association for Computing Machinery, 2004

Light Green Cluster (Automated Feedback and Assessment in Programming)

1) P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä, "Review of Recent Systems for Automatic Assessment of Programming Assignments," in Proceedings of the 10th Koli Calling International Conference on Computing Education Research, Koli Calling '10, (New York, NY, USA), pp. 86–93, Association for Computing Machinery, 2010
2) H. Keuning, J. Jeuring, and B. Heeren, "Towards a systematic review of automated feedback generation for programming exercises," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '16, (New York, NY, USA), pp. 41–46, Association for Computing Machinery, 2016
3) K. M. Ala-Mutka, "A survey of automated assessment approaches for programming assignments," Computer Science Education, vol. 15, no. 2, pp. 83–102, 2005
4) C. Douce, D. Livingstone, and J. Orwell, "Automatic test-based assessment of programming: A review," J. Educ. Resour. Comput., vol. 5, pp. 4–es, sep 2005

Small Clusters: Meta-analyses, Emotions

1) P. Kinnunen and B. Simon, "Experiencing programming assignments in cs1: The emotional toll," in Proceedings of the Sixth International Workshop on

Computing Education Research, ICER '10, (New York, NY, USA), pp. 77–86, Association for Computing Machinery, 2010
2) D. W. Valentine, "Cs educational research: A meta-analysis of sigcse technical symposium proceedings," SIGCSE Bull., vol. 36, pp. 255–259, Mar. 2004

APPENDIX B
RICH-CLUB INSTITUTIONS

1) UNIVERSITY OF TECHNOLOGY SYDNEY, Australia
2) UNIVERSITY OF SOUTHERN QUEENSLAND, Australia
3) UNIVERSITY OF NEWCASTLE, Australia
4) QUEENSLAND UNIVERSITY OF TECHNOLOGY, Australia
5) MONASH UNIVERSITY, Australia
6) ATHABASCA UNIVERSITY, Canada
7) THE UNIVERSITY OF HONG KONG, China
8) UNIVERSITY OF AARHUS, Denmark
9) IT UNIVERSITY OF COPENHAGEN, Denmark
10) AARHUS UNIVERSITY, Denmark
11) AALBORG UNIVERSITY, Denmark
12) UNIVERSITY OF TURKU, Finland
13) UNIVERSITY OF JYVÄSKYLÄ, Finland
14) UNIVERSITY OF HELSINKI, Finland
15) UNIVERSITY OF EASTERN FINLAND, Finland
16) TAMPERE UNIVERSITY OF TECHNOLOGY, Finland
17) LAPPEENRANTA UNIVERSITY OF TECHNOLOGY, Finland
18) HELSINKI UNIVERSITY OF TECHNOLOGY, Finland
19) AALTO UNIVERSITY, Finland
20) UNIVERSITY OF SIEGEN, Germany
21) UNIVERSITY OF POTSDAM, Germany
22) UNIVERSITY OF PADERBORN, Germany
23) FREE UNIVERSITY BERLIN, Germany
24) CARL VON OSSIETZKY UNIVERSITÄT, Germany
25) TRINITY COLLEGE DUBLIN, Ireland
26) WEIZMANN INSTITUTE OF SCIENCE, Israel
27) MASSEY UNIVERSITY, New Zealand
28) JAGIELLONIAN UNIVERSITY, Poland
29) UNIVERSITY OF LA LAGUNA, Spain
30) UNIVERSIDAD POLITÉCNICA DE MADRID, Spain
31) REY JUAN CARLOS UNIVERSITY, Spain
32) UPPSALA UNIVERSITY, Sweden
33) UMEÅ UNIVERSITY, Sweden
34) KTH ROYAL INSTITUTE OF TECHNOLOGY, Sweden
35) ABO AKADEMI UNIVERSITY, Finland
36) UNIVERSITY OF WARWICK, UK
37) UNIVERSITY OF WALES, UK
38) UNIVERSITY OF KENT, UK
39) UNIVERSITY OF DUNDEE, UK
40) UNIVERSITY OF WASHINGTON, USA
41) UNIVERSITY OF NORTHERN IOWA, USA
42) UNIVERSITY OF CONNECTICUT, USA
43) ROOSEVELT UNIVERSITY, USA
44) RHODE ISLAND COLLEGE, USA
45) PACIFIC LUTHERAN UNIVERSITY, USA
46) NEW JERSEY INSTITUTE OF TECHNOLOGY, USA
47) CARNEGIE MELLON UNIVERSITY, USA
48) STANFORD UNIVERSITY, USA
49) DUKE UNIVERSITY, USA
50) UNIVERSITY OF TEXAS, USA
51) UNIVERSITY OF CALIFORNIA, USA
52) VALTECH, Multinational

REFERENCES

[1] S. Simon, "Emergence of computing education as a research discipline," Ph.D. dissertation, Dept. School Sci., Aalto Univ., Espoo, Finland, 2015.
[2] S. A. Fincher and A. V. Robins, "An important and timely field," in *The Cambridge Handbook Computing Education Research*, S. A. Fincher and A. V. Robins, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2019, pp. 1–8.
[3] M. Tedre, *The Science of Computing: Shaping a Discipline*. New York, NY, USA: CRC Press, 2014.
[4] W. Aspray, "Was early entry a competitive advantage? U.S. universities that entered computing in the 1940s," *IEEE Ann. Hist. Comput.*, vol. 22, no. 3, pp. 42–87, Jul. 2000.
[5] I. B. Cohen and G. W. Welch, *Makin' Numbers: Howard Aiken and the Computer*. Cambridge, MA, USA: MIT Press, 1999.
[6] A. W. Jacobson, *Proceedings of the First Conference on Training Personnel for the Computing Machine Field*. Detroit, MI, USA: Wayne Univ. Press, 1955.
[7] M. Tedre, A. Simon, and L. Malmi, "Changing aims of computing education: A historical survey," *Comput. Sci. Educ.*, vol. 28, no. 2, pp. 158–186, Apr. 2018.
[8] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. and Paterson, "A survey of literature on the teaching of introductory programming," *SIGCSE Bull.*, vol. 39, pp. 204–223, Dec. 2007.
[9] M. Guzdial and B. du Boulay, "The history of computing education research," in *The Cambridge Handbook Computing Education Research*, S. A. Fincher and A. V. Robins, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2019, pp. 11–39.
[10] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, Mar. 2003.
[11] M. Goldweber, M. Clark, and S. Fincher, "The relationship between CS education research and the SIGCSE community," in *Proc. 35th SIGCSE Tech. Symp. Comput. Sci. Educ.*, Leeds, U.K., 2004, pp. 228–229.
[12] J. J. Randolph, "Computer science education research at the crossroads: A methodological review of the computer science education research: 2000–2005," Ph.D. dissertation, Dept. Educ., Utah State Univ., Logan, UT, USA, 2007.
[13] S. Fincher and J. Tenenberg, "Using theory to inform capacity-building: Bootstrapping communities of practice in computer science education research," *J. Eng. Educ.*, vol. 95, no. 4, pp. 265–277, Oct. 2006.
[14] S. Fincher and M. Petre, *Computer Science Education Research*. Oxfordshire, U.K.: Taylor & Francis, 2004.
[15] D. W. Valentine, "CS educational research: A meta-analysis of SIGCSE technical symposium proceedings," in *Proc. 35th SIGCSE Tech. Symp. Comput. Sci. Educ.*, 2004, pp. 255–259.
[16] A. Pears, S. Seidman, C. Eney, P. Kinnunen, and L. Malmi, "Constructing a core literature for computing education research," *ACM SIGCSE Bull.*, vol. 37, pp. 152–161, Dec. 2005.
[17] Simon, "A classification of recent australasian computing education publications," *Comput. Sci. Educ.*, vol. 17, no. 3, pp. 155–169, Sep. 2007.
[18] M. Joy, J. Sinclair, S. Sun, J. Sitthiworachart, and J. López-González, "Categorising computer science education research," *Educ. Inf. Technol.*, vol. 14, pp. 105–126, Jun. 2009.

[19] S. Simon, A. Carbone, M. de Raadt, R. Lister, M. Hamilton, and J. Sheard, "Classifying computing education papers: Process and results," in *Proc. 4th Int. Workshop Comput. Educ. Res.*, New York, NY, USA, 2008, pp. 161–172.

[20] Simon, "A picture of the growing ICER community," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, New York, NY, USA, Aug. 2016, pp. 153–159.

[21] Simon, "Twenty-two years of ACE," in *Proc. 22nd Australas. Comput. Educ. Conf.*, New York, NY, USA, Feb. 2020, pp. 203–210.

[22] Simon and J. Sheard, "Twenty-four years of ITiCSE papers," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, New York, NY, USA, Jun. 2020, pp. 5–11.

[23] Simon, "Twenty-four years of ITiCSE authors," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, New York, NY, USA, Jun. 2020, pp. 205–211.

[24] Simon, "The Koli Calling community," in *Proc. 16th Koli Calling Int. Conf. Comput. Educ. Res.*, New York, NY, USA, Nov. 2016, pp. 101–109.

[25] J. Randolph, R. Bednarik, P. Silander, J. Gonzalez, N. Myller, and E. Sutinen, "A critical analysis of the research methodologies reported in the full papers of the proceedings of ICALT 2004," in *Proc. 5th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, 2005, pp. 10–14.

[26] S. Simon, "Informatics in education and Koli Calling: A comparative analysis," *Informat. Educ.*, vol. 8, no. 1, pp. 101–114, Apr. 2009.

[27] S. Simon, J. Sheard, A. Carbone, M. de Raadt, M. Hamilton, R. F. Lister, and E. Thompson, "Eight years of computing education papers at NACCQ," in *Proc. 21st Annu. Conf. Nat. Advisory Committee Comput. Qualifications (NACCQ)*. Auckland, New Zealand: National Advisory Committee on Computing Qualifications. 2008, pp. 101–107.

[28] J. Sheard, S. Simon, M. Hamilton, and J. Lönnberg, "Analysis of research into the teaching and learning of programming," in *Proc. 5th Int. Workshop Comput. Educ. Res.*, New York, NY, USA, 2009, pp. 93–104.

[29] R. P. Medeiros, G. L. Ramalho, and T. P. Falcão, "A systematic literature review on teaching and learning introductory programming in higher education," *IEEE Trans. Educ.*, vol. 62, no. 2, pp. 77–90, May 2019.

[30] A. Vihavainen, J. Airaksinen, and C. Watson, "A systematic review of approaches for teaching introductory programming and their influence on success," in *Proc. 10th Annu. Conf. Int. Comput. Educ. Res.*, New York, NY, USA, 2014, pp. 19–26.

[31] V. Garneli, M. N. Giannakos, and K. Chorianopoulos, "Computing education in K-12 schools: A review of the literature," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Mar. 2015, pp. 543–551.

[32] C. Szabo, J. Sheard, A. Luxton-Reilly, Simon, B. A. Becker, and L. Ott, "Fifteen years of introductory programming in schools: A global overview of K-12 initiatives," in *Proc. 19th Koli Calling Int. Conf. Comput. Educ. Res.*, New York, NY, USA, Nov. 2019, pp. 1–9.

[33] L. Malmi, J. Sheard, Simon, R. Bednarik, J. Helminen, A. Korhonen, N. Myller, J. Sorva, and A. Taherkhani, "Characterizing research in computing education: A preliminary analysis of the literature," in *Proc. 6th Int. Workshop Comput. Educ. Res.*, New York, NY, USA, 2010, pp. 3–12.

[34] L. Malmi, A. Taherkhani, J. Sheard, Simon, R. Bednarik, J. Helminen, P. Kinnunen, A. Korhonen, N. Myller, and J. Sorva, "Theoretical underpinnings of computing education research: What is the evidence?" in *Proc. 10th Annu. Conf. Int. Comput. Educ. Res.*, New York, NY, USA, 2014, pp. 27–34.

[35] L. Malmi, J. Sheard, P. Kinnunen, Simon, and J. Sinclair, "Computing education theories: What are they and how are they used?" in *Proc. ACM Conf. Int. Comput. Educ. Res.*, New York, NY, USA, Jul. 2019, pp. 187–197.

[36] C. Szabo, N. Falkner, A. Petersen, H. Bort, K. Cunningham, P. Donaldson, A. Hellas, J. Robinson, and J. Sheard, "Review and use of learning theories within computer science education research: Primer for researchers and practitioners," in *Proc. Working Group Reports Innov. Technol. iComput. Sci. Educ.*, New York, NY, USA, 2019, pp. 89–109.

[37] S. Heckman, J. C. Carver, M. Sherriff, and A. Al-zubidy, "A systematic literature review of empiricism and norms of reporting in computing education research literature," *ACM Trans. Comput. Educ.*, vol. 22, pp. 1–44, Oct. 2021.

[38] K. Sanders, J. Sheard, B. A. Becker, A. Eckerdal, S. Hamouda, and Simon, "Inferential statistics in computing education research: A methodological review," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, New York, NY, USA, Jul. 2019, pp. 177–185.

[39] A. Al-Zubidy, J. C. Carver, S. Heckman, and M. Sherriff, "A (updated) review of empiricism at the SIGCSE technical symposium," in *Proc. 47th ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Feb. 2016, pp. 120–125.

[40] L. Margulieux, T. A. Ketenci, and A. Decker, "Review of measurements used in computing education research and suggestions for increasing standardization," *Comput. Sci. Educ.*, vol. 29, no. 1, pp. 49–78, Jan. 2019.

[41] Q. Hao, D. H. Smith IV, N. Iriumi, M. Tsikerdekis, and A. J. Ko, "A systematic investigation of replications in computing education research," *ACM Trans. Comput. Educ.*, vol. 19, pp. 1–18, Aug. 2019.

[42] J. Zhang, A. Luxton-Reilly, P. Denny, and J. Whalley, "Scientific collaboration network analysis for computing education conferences," in *Proc. 26th ACM Conf. Innov. Technol. Comput. Sci. Educ.*, New York, NY, USA, Jun. 2021, pp. 582–588.

[43] A. Settle, B. A. Becker, R. Duran, V. Kumar, and A. Luxton-Reilly, "Improving global participation in the SIGCSE technical symposium: Panel," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Feb. 2020, pp. 483–484.

[44] B. A. Becker, A. Settle, A. Luxton-Reilly, B. B. Morrison, and C. Laxer, "Expanding opportunities: Assessing and addressing geographic diversity at the SIGCSE technical symposium," in *Proc. 52nd ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Mar. 2021, pp. 281–287.

[45] M. Apiola, M. Tedre, S. López-Pernas, M. Saqr, M. Daniels, and A. Pears, "A scientometric journey through the FIE Bookshelf: 1982-2020," in *Proc. ASEE/IEEE Frontiers Educ.*, Feb. 2021, pp. 1–9.

[46] R. Rosenblatt, "Investigating co-authorship, research themes, and funding streams for the frontiers in education community," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2021, pp. 1–9.

[47] Z. Papamitsiou, M. Giannakos, Simon, and A. Luxton-Reilly, "Computing education research landscape through an analysis of keywords," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, New York, NY, USA, 2020, pp. 102–112.

[48] M. Saqr, K. Ng, S. S. Oyelere, and M. Tedre, "People, ideas, milestones: A scientometric study of computational thinking," *ACM Trans. Comput. Educ.*, vol. 21, pp. 1–17, Mar. 2021.

[49] M. Aria and C. Cuccurullo, "BiblioMetrix: An R-tool for comprehensive science mapping analysis," *J. Informetrics*, vol. 11, no. 4, pp. 959–975, Nov. 2017.

[50] M. Norris and C. Oppenheim, "Comparing alternatives to the web of science for coverage of the social sciences' literature," *J. Informetrics*, vol. 1, no. 2, pp. 161–169, Apr. 2007.

[51] J. Baas, M. Schotten, A. Plume, G. Côté, and R. Karimi, "Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies," *Quant. Sci. Stud.*, vol. 1, pp. 377–386, Feb. 2020.

[52] V. K. Singh, P. Singh, M. Karmakar, J. Leta, and P. Mayr, "The journal coverage of web of science, scopus and dimensions: A comparative analysis," *Scientometrics*, vol. 126, no. 6, pp. 5113–5142, Jun. 2021.

[53] J. Tenenberg and R. McCartney, "Introducing the ACM transactions on computing education," *ACM Trans. Comput. Educ.*, vol. 9, pp. 1–13, Mar. 2009.

[54] M. Guzdial and B. du Boulay, *The History of Computing Education Research*. Cambridge, U.K.: Cambridge Univ. Press, 2019, pp. 11–39.

[55] M. Bastian, S. Heymann, and M. Jacomy, "GEPHI: An open source software for exploring and manipulating networks," in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2009, pp. 361–362.

[56] A. Ma and R. J. Mondragón, "Rich-cores in networks," *PLoS ONE*, vol. 10, Jul. 2015, Art. no. e0119678.

[57] P. D. Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized Louvain method for community detection in large networks," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Nov. 2011, pp. 88–93.

[58] W. F. Atchison, S. D. Conte, J. W. Hamblen, T. E. Hull, T. A. Keenan, W. B. Kehl, E. J. McCluskey, S. O. Navarro, W. C. Rheinboldt, E. J. Schweppe, W. Viavant, and J. M. David Young, "Curriculum 68: Recommendations for academic programs in computer science: A report of the ACM curriculum committee on computer science," *Commun. ACM*, vol. 11, no. 3, pp. 151–197, 1968.

[59] G. E. Forsythe, "A university's educational program in computer science," *Commun. ACM*, vol. 10, no. 1, pp. 3–11, 1967.

[60] R. W. Elliot, "Master's level computer science curricula," *Commun. ACM*, vol. 11, pp. 507–508, Jul. 1968.

[61] G. Salton, "Information science in a Ph.D. computer science program," *Commun. ACM*, vol. 12, pp. 111–117, Feb. 1969.

[62] W. F. Atchison and J. W. Hamblen, "Status of computer sciences curricula in colleges and universities," *Commun. ACM*, vol. 7, no. 4, pp. 225–227, Apr. 1964.

[63] R. H. Austing, B. H. Barnes, D. T. Bonnette, G. L. Engel, and G. Stokes, "Curriculum '78: Recommendations for the undergraduate program in computer science— A report of the ACM curriculum committee on computer science," *Commun. ACM*, vol. 22, no. 3, pp. 147–166, Mar. 1979.

[64] M. Minsky, "Form and content in computer science (1970 ACM Turing lecture)," *J. ACM*, vol. 17, no. 2, pp. 197–215, Apr. 1970.

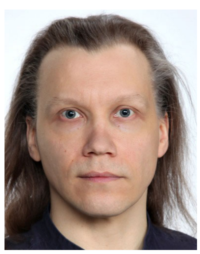
[65] T. R. Wilcox, A. M. Davis, and M. H. Tindall, "The design and implementation of a table driven, interactive diagnostic programming system," *Commun. ACM*, vol. 19, pp. 609–616, Nov. 1976.
[66] R. H. Austing, B. H. Barnes, and G. L. Engel, "A survey of the literature in computer science education since curriculum '68," *Commun. ACM*, vol. 20, no. 1, pp. 13–21, Jan. 1977.
[67] G. L. Engel, "A computer science course program for small colleges," *Commun. ACM*, vol. 16, pp. 139–147, Mar. 1973.
[68] E. Soloway, "Learning to program=learning to construct mechanisms and explanations," *Commun. ACM*, vol. 29, pp. 850–858, Sep. 1986.
[69] P. F. Campbell and G. P. McCabe, "Predicting the success of freshmen in a computer science major," *Commun. ACM*, vol. 27, pp. 1108–1113, Nov. 1984.
[70] N. E. Gibbs and A. B. Tucker, "A model curriculum for a liberal arts degree in computer science," *Commun. ACM*, vol. 29, pp. 202–210, Mar. 1986.
[71] K. A. Reek, "The try system-or-how to avoid testing student programs," *SIGCSE Bull.*, vol. 21, pp. 112–116, Feb. 1989.
[72] D. F. Butcher and W. A. Muth, "Predicting performance in an introductory computer science course," *Commun. ACM*, vol. 28, pp. 263–268, Mar. 1985.
[73] M. Ben-Ari, "Constructivism in computer science education," *SIGCSE Bull.*, vol. 30, pp. 257–261, Mar. 1998.
[74] J. Smith, "Constructivism in computer science education," in *Proc. 53rd ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Mar. 2022, p. 1171.
[75] M. C. Linn and M. J. Clancy, "The case for case studies of programming problems," *Commun. ACM*, vol. 35, pp. 121–132, Mar. 1992.
[76] H. Gregersen and C. S. Jensen, "Temporal entity-relationship models—A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 3, pp. 464–497, 1999.
[77] J. J. McConnell, "Active learning and its use in computer science," *SIGCUE Outlook*, vol. 24, pp. 52–54, Jan. 1996.
[78] A. Takang, P. Grugg, and R. Macredie, "The effects of comments and identifier names on program comprehensibility: An experimental investigation," *J. Prog. Lang.*, vol. 4, no. 3, pp. 143–167, 1996.
[79] M. Papastergiou, "Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation," *Comput. Educ.*, vol. 52, no. 1, pp. 1–12, 2009, doi: 10.1016/j.compEdu.2008.06.004.
[80] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bull.*, vol. 37, no. 3, pp. 14–18, Sep. 2005.
[81] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz, "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students," in *Proc. Work. Group Rep. ITiCSE Innov. Technol. Comput. Sci. Educ.*, New York, NY, USA, 2001, pp. 125–180.
[82] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," *J. Vis. Lang. Comput.*, vol. 13, no. 3, pp. 259–290, Jun. 2002.
[83] L. Buechley, M. Eisenberg, J. Catchen, and A. Crockett, "The LilyPad arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education," in *Proc. 26th Annu. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2008, pp. 423–432.
[84] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch programming language and environment," *ACM Trans. Comput. Educ.*, vol. 10, pp. 1–15, Nov. 2010.
[85] C. Watson and F. W. B. Li, "Failure rates in introductory programming revisited," in *Proc. Conf. Innov. Technol. Comput. Sci. Educ.*, New York, NY, USA, 2014, pp. 39–44.
[86] P. J. Guo, "Online Python tutor: Embeddable web-based program visualization for cs education," in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, 2013, pp. 579–584.
[87] D. Weintrop and U. Wilensky, "To block or not to block, that is the question: Students' perceptions of blocks-based programming," in *Proc. 14th Int. Conf. Interact. Design Children*, New York, NY, USA, Jun. 2015, pp. 199–208.
[88] L. Werner, J. Denner, S. Campe, and D. C. Kawamoto, "The fairy performance assessment: Measuring computational thinking in middle school," in *Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, 2012, pp. 215–220.
[89] P. J. Denning, D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner, and P. R. and Young, "Computing as a discipline," *Commun. ACM*, vol. 32, no. 1, pp. 9–23, 1989.
[90] J. Wing, "Computational thinking," *Commun. ACM*, vol. 49, pp. 33–35, Mar. 2006.
[91] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 10, no. 3, p. e1355, 2020.
[92] T. Valtonen, S. López-Pernas, M. Saqr, H. Vartiainen, E. T. Sointu, and M. Tedre, "The nature and building blocks of educational technology research," *Comput. Hum. Behav.*, vol. 128, May 2022, Art. no. 107123.
[93] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *J. Amer. Soc. Inf. Sci.*, vol. 24, no. 4, pp. 265–269, 1973.
[94] M. Ben-Ari, "Constructivism in computer science education," *J. Comput. Math. Sci. Teaching*, vol. 20, no. 1, pp. 45–73, 2001.
[95] J. B. Biggs and K. F. Collins, *Evaluating the Quality of Learning: The Solo Taxonomy: Structure of the Observed Learning Outcome*. New York, NY, USA: Academic, 1982.
[96] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988.
[97] B. S. Bloom, M. D. Englehart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. New York, NY, USA: Longmans, 1956.
[98] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," *ACM SIGCSE Bull.*, vol. 39, no. 2, pp. 32–36, Jun. 2007.
[99] M. C. Jadud, "Methods and tools for exploring novice compilation behaviour," in *Proc. Int. Workshop Comput. Educ. Res.*, New York, NY, USA, 2006, pp. 73–84.
[100] B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: A study of twelve factors," in *Proc. 22nd SIGCSE Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, 2001, pp. 184–188.
[101] A. Vihavainen, M. Paksula, and M. Luukkainen, "Extreme apprenticeship method in teaching programming for beginners," in *Proc. 42nd ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, 2011, pp. 93–98.
[102] S. Papert, *MINDSTORMS: Children, Computing, Powerful Ideas*. New York, NY, USA: Basic Books, 1980.
[103] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers," *ACM Comput. Surv.*, vol. 37, pp. 83–137, Jun. 2005.
[104] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Avenue Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1988.
[105] P. J. Denning and M. Tedre, *Computational Thinking* (Essential Knowledge Series). Cambridge, MA, USA: MIT Press, 2019.
[106] V. Dagiene and G. Stupuriene, "Bebras—A sustainable community building model for the concept based learning of informatics and computational thinking," *Inf. Educ.*, vol. 15, no. 1, pp. 25–44, Apr. 2016.
[107] L. Mannila, V. Dagiene, B. Demo, N. Grgurina, C. Mirolo, L. Rolandsson, and A. Settle, "Computational thinking in K-9 education," in *Proc. Technol. Comput. Sci. Educ. Conf.*, New York, NY, USA, 2014, pp. 1–29.
[108] A. Pears, M. Tedre, T. Valtonen, and H. Vartiainen, "What makes computational thinking so troublesome?" in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2021, pp. 1–7.
[109] J. Margolis and A. Fisher, *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA, USA: MIT Press, 2002.
[110] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1991.
[111] L. S. Vygotsky, *Mind Society: The Development Higher Psychology Processes*. Cambridge, MA, USA: Harvard University Press, 1978.
[112] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," *Psychol. Rev.*, vol. 84, no. 2, pp. 191–215, 1977.
[113] M. Saqr, J. Nouri, H. Vartiainen, and M. Tedre, "Robustness and rich clubs in collaborative learning groups: A learning analytics study using network science," *Sci. Rep.*, vol. 10, no. 1, Dec. 2020, Art. no. 14445.
[114] S. Zhou and R. J. Mondragon, "The rich-club phenomenon in the internet topology," *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 180–182, Mar. 2004.
[115] V. Colizza, A. Flammini, M. A. Serrano, and A. Vespignani, "Detecting rich-club ordering in complex networks," *Nature Phys.*, vol. 2, no. 2, pp. 110–115, Feb. 2006.
[116] A. Ma, R. J. Mondragón, and V. Latora, "Anatomy of funded research in science," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 48, pp. 14760–14765, Dec. 2015.
[117] L. M. Vaquero and M. Cebrian, "The rich club phenomenon in the classroom," *Sci. Rep.*, vol. 3, no. 1, Dec. 2013, Art. no. 1174.
[118] R. B. Shapiro, R. Fiebrink, and P. Norvig, "How machine learning impacts the undergraduate computing curriculum," *Commun. ACM*, vol. 61, no. 11, pp. 27–29, Oct. 2018.

[119] H. Vartiainen, T. Toivonen, I. Jormanainen, J. Kahila, M. Tedre, and T. Valtonen, "Machine learning for middle schoolers: Learning through data-driven design," *Int. J. Child-Computer Interact.*, vol. 29, Sep. 2021, Art. no. 100281.
[120] M. Tedre, P. Denning, and T. Toivonen, "Ct 2.0," in *Proc. 21st Koli Calling Int. Conf. Comput. Educ. Res.*, 2021, pp. 1–8.
[121] K. Kelly, *The Inevitable: Understanding the 12 Technological Forces That Will Shape Our Future*. London, U.K.: Penguin Books, 2017.
[122] M. Apiola and E. Sutinen, "Design science research for learning software engineering and computational thinking: Four cases," *Comput. Appl. Eng. Educ.*, vol. 29, no. 1, pp. 83–101, Oct. 2020.
[123] R. Root-Bernstein, M. Van Dyke, A. Peruski, and M. Root-Bernstein, "Correlation between tools for thinking: Arts, crafts, and design avocations and scientific achievement among STEMM professionals," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 6, pp. 1910–1917, Feb. 2019.
[124] D. Skorton and A. Bear, *The Integration of the Humanities and Arts With Sciences, Engineering, and Medicine in Higher Education: Branches From the Same Tree*. Washington, DC, USA: The National Academies Press, 2018.
[125] R. Root-Bernstein, "STEMM education should get 'HACD,'" *Science*, vol. 361, no. 6397, pp. 22–23, 2018.
[126] T. Toivonen, I. Jormanainen, J. Kahila, M. Tedre, T. Valtonen, and H. Vartiainen, "Co-designing machine learning apps in K–12 with primary school children," in *Proc. IEEE 20th Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2020, pp. 308–310.
[127] R. Mariescu-Istodor and I. Jormanainen, "Machine learning for high school students," in *Proc. 19th Koli Calling Int. Conf. Comput. Educ. Res.*, Nov. 2019, pp. 1–9.
[128] C. Fiesler, N. Garrett, and N. Beard, "What do we teach when we teach tech ethics?: A syllabi analysis," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, New York, NY, USA, Feb. 2020, pp. 289–295.
[129] I. D. Raji, M. K. Scheuerman, and R. Amironesei, "You Can't sit with U.S.: Exclusionary pedagogy in AI ethics education," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, New York, NY, USA, Mar. 2021, pp. 515–525.
[130] M. Tedre and P. J. Denning, "The long quest for computational thinking," in *Proc. 16th Koli Calling Int. Conf. Comput. Educ. Res.*, New York, NY, USA, Nov. 2016, pp. 120–129.
[131] Y. B. Kafai, "From computational thinking to computational participation in K-12 education," *Commun. ACM*, vol. 59, pp. 26–27, Jul. 2016.
[132] Y. B. Kafai and Q. Burke, "The social turn in K-12 programming: Moving from computational thinking to computational participation," in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, 2013, pp. 603–608.
[133] J. Fagerlund, P. Häkkinen, M. Vesisenaho, and J. Viiri, "Computational thinking in programming with scratch in primary schools: A systematic review," *Comput. Appl. Eng. Educ.*, vol. 29, no. 1, pp. 12–28, Jan. 2021.
[134] A. Shipepe, I. Jormanainen, and E. Sutinen, "Educational robotics initiatives in Namibia and worldwide," in *Proc. 8th Int. Conf. Technol. Ecosyst. Enhancing Multiculturality*, New York, NY, USA, Oct. 2020, pp. 48–53.
[135] P. J. Denning and M. Tedre, "Computational thinking for professionals," *Commun. ACM*, vol. 64, pp. 30–33, Nov. 2021.
[136] J. Bell and T. Bell, "Integrating computational thinking with a music education context," *Informat. Educ.*, vol. 17, no. 2, pp. 151–166, Oct. 2018.
[137] T. Bell, I. H. Witten, and M. Fellows. (1998). *Computer Science Unplugged Off-Line Activities and Games for All Ages*. [Online]. Available: https://csunplugged.org/
[138] L. Malmi, J. Sheard, P. Kinnunen, Simon, and J. Sinclair, "Theories and models of emotions, attitudes, and self-efficacy in the context of programming education," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, New York, NY, USA, Aug. 2020, pp. 36–47.
[139] R. K. Merton, "The Matthew effect in science," *Science*, vol. 159, no. 3810, pp. 56–63, Jan. 1968.
[140] A. Pears, M. Daniels, and A. Berglund, "Describing computer science education research: An academic process view," *Simul. Ser.*, vol. 34, no. 1, pp. 99–104, 2002.
[141] A. N. Pears and M. Daniels, "Structuring csed research studies: Connecting the pieces," *ACM SIGCSE Bull.*, vol. 35, no. 3, pp. 149–153, Sep. 2003.
[142] A. Berglund, M. Daniels, and A. Pears, "Qualitative research projects in computing education research: An overview," in *Proc. 8th Australas. Conf. Comput. Educ.*, vol. 52, 2006, pp. 25–33.
[143] A. Berglund, I. Box, A. Eckerdal, R. Lister, and A. Pears, "Learning educational research methods through collaborative research: The phicer initiative," in *Proc. 10th Conf. Australas. Comput. Educ.*, vol. 78, 2008, pp. 35–42.
[144] M. Daniels and A. Pears, "Models and methods for computing education research," in *Proc. 14th Australas. Comput. Educ. Conf.*, vol. 123, 2012, pp. 95–102.
[145] M. Szell and R. Sinatra, "Research funding goes to rich clubs," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 48, pp. 14749–14750, Dec. 2015.
[146] E. Sutinen, "Koli Calling: From the ten past years to the future: A developing country's perspective," in *Proc. 10th Koli Calling Int. Conf. Comput. Educ. Res.*, 2010, pp. 124–127.
[147] F. Franceschini, D. Maisano, and L. Mastrogiacomo, "Empirical analysis and classification of database errors in scopus and web of science," *J. Informetrics*, vol. 10, no. 4, pp. 933–953, Nov. 2016.

MIKKO APIOLA received the Ph.D. degree in computer science from the University of Helsinki, Finland. He has worked in IT industry and in several teaching and research positions in Finnish universities. He is currently an Adjunct Professor at the University of Turku's Department of Computing, Finland; and the University of Eastern Finland's School of Computing. His research interests include computing education research, learning analytics, and ICT for development (ICT4D).

MOHAMMED SAQR received the Ph.D. degree in learning analytics from Stockholm University. He currently works as a Senior Researcher at the University of Eastern Finland on artificial intelligence, big data in education, network science, and scientometrics. He is particularly interested in research methods, including network analysis, temporal networks, machine learning, process, and sequence mining as well as temporal processes in general. He is an active member of several scientific organizations and acts as an academic editor in leading academic publications.

SONSOLES LÓPEZ-PERNAS (Graduate Student Member, IEEE) received the bachelor's and master's degrees in telecommunications engineering and the Ph.D. degree in telematics engineering from the Universidad Politécnica de Madrid (UPM). Since 2015, she has been working as a Researcher with the Next Generation Internet Group (GING), UPM. She is currently an Assistant Professor with the Department of Computer Science UPM. Her research interests include learning analytics, technology enhanced learning, computer science education, and bibliometrics.



MATTI TEDRE is currently a Professor at the School of Computing, University of Eastern Finland. He is the author of *The Science of Computing: Shaping a Discipline* (CRC Press, 2014), *Computational Thinking* with Peter J. Denning (MIT Press, 2019), and about 100 other publications. His research interests include philosophy of computer science, educational technology, computer science education research, the disciplinary history of computing, and ICT4D.