sciendo

Research on Education and Media
Vol. 16, N. 2, Year 2024
ISSN: 2037-0830 – DOI: 10.2478/rem-2024-0018

Effective Instructional Strategies for the Development of Computational Thinking in Primary Education: A Systematic Literature Review

Francesco C. Ugolini[a], Panagiotis Kakavas[b]

[a]Università Guglielmo Marconi, Roma, f.ugolini@unimarconi.it, https://orcid.org/0000-0002-7887-7983
[b]43rd Primary School of Patras, panoskakavas86@gmail.com, https://orcid.org/0009-0000-6229-0363

Abstract

This study consists of a 17-year (2006-2021) systematic literature review on the effective instructional strategies for developing Computational Thinking (CT) in primary school students (K-5). The aim of this paper is to identify instructional strategies that have been implemented and evaluated by means of a pre- and post-test, with the aim of developing CT as a thinking process (i.e. going beyond the mere acquisition of CS concepts). The analysis considers the overall pedagogical approach, with the intention of overcoming both the archetypal approaches to coding and the assessment tools used to measure CT acquisition.

The results show that there is a strong difference between K-3 and 4-5 classes, the former being more the context for educational robotics or unplugged activities; they also show that a task-based approach is prevalent, whose effectiveness is limited to the first access to CS activities, namely improving self-efficacy and reducing learning anxiety. Nevertheless, this study also identifies some interesting models of problem solving (such as IGGIA or CPS) that go beyond the puzzle-based approach, as well as two significant teaching strategies (IDC and DBL) that try to promote more meaningful learning (typically including elements of constructionism), while enabling curricular learning objectives.

Keywords: *Computational Thinking, instructional strategies, primary education, systematic literature review, content analysis.*

Introduction

1.1. Computational Thinking as a thought process

In recent years, activities aimed at developing Computational Thinking (CT) in primary and kindergarten (K-5) education have become increasingly widespread and have received much attention at both policy and research levels. Despite this, the term 'computational thinking' itself has remained ambiguous, appearing mostly as an 'umbrella term' (Roman-Gonzalez et al., 2019, p. 80). In our previous systematic review on CT in primary education (Kakavas & Ugolini, 2019), we were able to identify how papers claiming to document an improvement in CT in K-6 children actually referred to a variety of its operational definitions and, consequently, to different assessment tools, some of which focused exclusively on the acquisition of programming concepts.

While the number of primary empirical studies is increasing, we believe that the time is ripe for a more specific systematic review that takes a strictly pedagogical perspective, assuming that CT is a thinking process, as was the original intention of Jeannette Wing (2006, 2008). This review will therefore focus on the instructional strategies used in K-5 schools that are considered to be effective according to this specific perspective.

As a theoretical definition of CT, we can now consider the so-called Cuny-Snyder-Wing definition to be widely accepted: CT is "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively executed by an information-processing agent" (Wing, 2010; Brennan & Resnick, 2012, p. 2). However, we can report another one that emphasises more the relationship with the digital world:

CT "is the mental skills and practices for designing computations that make computers do work for us, and for explaining and interpreting the world as a complex of information processes" (Denning & Tedre, 2019, p. 4). In any case, in both cases, CT constructs go beyond the mere acquisition of computer science (CS) concepts, and therefore both are appropriate for this paper.

As we said before, the landscape of operational definitions seems to be more heterogeneous. In our previous review, we were able to highlight three of them: the three-dimensional model of Brennan and Resnick (2012), which includes 7 CT concepts, 4 CT practices and 3 CT perspectives; the one established by CSTA and ISTE (2011), which lists a minimal set of 6 characteristics strongly related to CS; and a third one, which describes four steps of a problem-solving process (Anderson, 2016) and which can be easily found on the web under the name of four pillars. All three are relevant according to certain aspects, but they reveal different approaches to coding activities. We can identify two such approaches, which we can define in some way as archetypal; we will describe them in the next paragraph.

1.2. Two archetypal approaches to coding activities

For the purposes of this review, we will refer to two archetypal approaches to coding activities aimed at developing CT in K-5.

The first is related to Seymour Papert's constructionism (Papert, 1980, 1993), a learning paradigm that underlies the well-known programming environment Scratch (Resnick et al., 2009); this approach was eventually theorized by Mitchel Resnick under the name of creative learning (Resnick, 2017). Thus, its roots predate the revision of CT by Jeannette Wing (2006, 2008), with which it merged thanks to Brennan and Resnick (2012). In line with the principles of constructionism, it is based on the creation and sharing of computational objects and essentially sees programming as a creative act and an opportunity for self-expression. Scratch, both a programming language and an online project-sharing environment, was designed to meet these needs. In addition to the typical constructs of computer programming (the CT Concepts), Brennan and Resnick's operational definition also identifies 4 CT Practices that the "young designers" engaged in while creating their projects and, most importantly, 3 CT Perspectives that represent the development of the children's "understanding of themselves, their relationships with others, and the technological world around them" (Brennan & Resnick, 2012, p. 10) as an effect of their programming activity: Express, Connect, and Question. Activities related to this approach, in addition to the adoption of Scratch as a project-sharing environment, are characterized by the so-called "4 Ps" (Project, Passion, Peers and Play) and the creative learning spiral (Resnick, 2017). It typically finds concrete realization in out-of-school contexts, such as the Computer Clubhouses founded by Resnick himself, or the CoderDojos, although in particular Karen Brennan (2013) has tried to propose some strategies to include this type of activity in the school curriculum.

At the other ideal extreme we find the "puzzle-based" approach; CS learning is structured in sequences of puzzles of increasing difficulty, based on computer programming concepts. The standard problem presents an agent inside a grid, whose moves must be programmed to make it reach a specific goal. The best-known site offering this type of puzzle is code.org, which is sponsored by major computer companies to promote computer programming in schools. Their goals are broad: their main initiative is actually "The Hour of Code," a one-hour introduction to CS designed to be accessible to the widest number of people. This approach responds well to curricular needs, but from the point of view of pedagogical significance it seems rather limited, following a behaviorist approach. Nevertheless, it is based on a problem-solving process - closer to the Four Pillars definition - although the puzzles seem less complex and meaningful. Since such puzzles can be easily created with paper and pencil, many fully unplugged activities are based on this approach.

As mentioned above, neither of the two archetypal approaches seems to be fully adequate from a pedagogical point of view to effectively pursue curricular goals in K-5 education. Therefore, the precise purpose of this work is to identify in research papers approaches to coding and CT development that attempt to provide points of synthesis, either by identifying points of contact between a creative approach and curricular objectives, or by integrating, within a problem-based approach, elements aimed at deeper learning.

1.3. The instructional strategies

The purpose of this paper is to identify the main instructional strategies that are most effective from a pedagogical point of view. The term "instructional strategy" refers to instructional design, but despite its widespread use - or perhaps because of it - it is often imprecisely defined. Since we need to compare the strategies employed by a variety of papers, a too narrow operational definition of it seems inappropriate for the purposes of this review. Nevertheless, we can identify some elements (Bonaiuti, 2014; Bonaiuti, Calvani & Ranieri, 2016); a teaching strategy:

1. must be recognizable either by its name, by its appearance or even by its historical background;
2. should be adaptable to different contexts, as it could be based on specific procedures;

3. should have practical utility.

An instructional strategy is also typically self-contained in terms of duration. Strictly speaking, it can be implemented in 1-hour activities. For the purposes of this review, we will also consider some longer duration methods, provided they are well defined and result in well-structured shorter learning activities. However, we will distinguish them from the general approaches; namely, we will not consider educational robotics, unplugged activities, or on-screen programming activities as instructional strategies; similarly, the two archetypal approaches mentioned above may be a source of inspiration for instructional strategies, but they are not instructional strategies in themselves.

Finally, the construct of instructional strategy is one that is used in literature reviews (see, for example, the well-known meta-analysis by Marzano, Pickering, and Pollock (2002)); as they aim to provide guidance to practitioners, instructional strategies should not only be adaptable and easy to implement, but their effectiveness should also be proven to some extent, namely through experimental research (Bonaiuti, 2014). This is precisely the purpose of this review.

Identifying instructional strategies from research papers in order to compare them is not always straightforward. In order to do so, we first structure our analysis by means of a categorization based on the two archetypal approaches mentioned above in the "Results" section. In the discussion section, we re-structure the findings by age group in order to better advise scholars and practitioners, and finally, we specify the instructional strategies identified and discuss their level of effectiveness.

1.4. Related work

The systematic literature review studies reveal the current state of research and provide support for identifying research gaps in a specific area (Lo, 2020). Considering the importance of "effective instructional strategies" for the development of CT skills in elementary school students, to our knowledge, there is no review study that attempts to systematically address this issue. However, some review studies with certain similarities have been reported.

In particular, some of the reported studies were related to educational CT practices, CT strategies, and CT teaching approaches. The study by Kakavas and Ugolini (2019) has shown how CT has grown among primary school students by providing an overview of the context/setting in which CT has been implemented, identifying the "learning context" in which CT has been used, and highlighting how CT has been evaluated by presenting the learning outcomes for children who have participated in the different CT activities. Ogegbo and Rammarain (2021) analyzed and discussed the results of 23 studies by highlighting the "methodology", "different strategies", and assessment practices used to promote the integration of CT in science classrooms. They also provided an overview of how CT is taught in science classrooms and described the tools available for teaching CT in science classrooms. De Jong and Jeuring (2020) identified the instructional interventions/approaches used to teach CT in higher education, followed by a discussion of their effectiveness and how they are evaluated. Kakavas (2019) highlighted the "teaching practices" and "learning frameworks" used to support the development of both computational and creative thinking skills in elementary school students. In an attempt to answer the question of whether collaboration influences CT education, Lai and Wong (2021) examined the general effects of collaborative versus individual problem solving in CT via programming on "cognitive," "social," and "affective" skills. Palts and Pedaste (2020) proposed a three-stage model for CT skill development (problem definition, problem solving, and solution analysis) consisting of ten CT skills (problem formulation, abstraction, problem reformulation, decomposition, data collection and analysis, algorithm design, parallelization and iteration, automation, generalization, and evaluation) after conducting a systematic literature review of scientific studies to determine the defined CT dimensions. Taslibeyaz, Kursun, and Karaman (2020) examined the process of developing CT skills in learning environments, while also identifying the "conceptual understanding" and "measurement approaches" in the papers examined. Relevant studies have been conducted in the field of programming in K-12 education. More specifically, the systematic literature review by Fagerlund, Häkkinen, Vesisenaho, and Viiri (2020) focused on studies using Scratch in K-9 to examine what kind of CT has been evaluated in Scratch at the primary level of education. In this regard, Zhang and Nouri (2019) also provided evidence of CT cultivation through programming in Scratch by examining the CT skills that emerge through Scratch in K-9 based on empirical evidence. Additionally, Tikva and Tambouris (2021) developed a "conceptual model" (consisting of 6 CT domains: Knowledge Base, Learning Strategies, Assessment, Tools, Factors, and Capacity Building) based on their review study mapping CT through programming in K-12 education. In addition, Popat and Starkey (2019) reviewed the research to analyze the educational outcomes for students who learn to code in school, while Da Cruz Alves, Gresse Von Wangenheim and Hauck (2019) conducted a systematic review study focused on analyzing the "code created" by the students to infer CT skills related to algorithms and programming.

In addition, there are some review studies that present relative similarities regarding the systematic mapping of CT assessment in general education. The study by Haseski and Ilic (2019) aimed to investigate the characteristics of "paper and pencil" data collection instruments developed to measure CT based on several variables. Vinu Varghese and Renumol (2021) identified both the interventions for the development of CT competencies and the methods used to assess these competencies. Similarly, the review by Tang, Yin, Lin, Hadad, and Zhai (2020) focused on the ways in which CT has been assessed in the literature, while the study by Lin, Yin, Tang, Hadad, and Zhai (2020) synthesized empirical evidence on the assessment of "maker activities" in papers that used precise assessment tools in "maker activities" to measure learning outcomes obtained in different learning settings across educational levels. Finally, Cutumisu, Adams, and Lu (2019) worked on the grouping of different CT assessments used after reviewing 39 empirical studies coded by the specific competencies outlined in existing CT frameworks to identify and classify the main characteristics of existing CT assessments.

Of particular interest are the review studies that focus on "unplugged pedagogy" for teaching CT. In particular, Huang and Looi (2020) considered how research findings on unplugged pedagogy could advance CS/CT educational priorities by examining how unplugged pedagogy supports (a) students' CT development and (b) CS-for-all (CS accessible to a wide range of students). Battal, Adanir, and Gülbahar (2021) analyzed research studies that investigated the implementation of CS unplugged activities and found that most studies refer to "paper-pencil activities," "problem solving," "storytelling," "games," "tangible programming," and even "robotics" when using the term "CS unplugged." After reviewing 27 "picture books" and "graphic novels" for elementary students for CT elements, Ballard and Haroldson (2021) concluded that literature provides teachers with a non-programming "unplugged resource" for exposing students to CT and enhancing CT and CT-integrated instruction, while also personalizing learning based on CT readiness and interest levels. Chen, Yang, and others (2023) examined the influence of unplugged activities on improving students' CT skills in K-12 education. Their review showed that studies on CT skills were mainly (81.64%) conducted in CS and STEM education, with board and card games being the most common unplugged activities for promoting CT skills in K-12 education. The results also highlighted the opportunities of unplugged pedagogy for promoting CT skills in K-12 education. Finally, the review by Bati (2022) aimed to summarize the experimental evidence on CT and programming conducted in early childhood education in terms of the variables "plugged" versus "unplugged", age, and gender. The results showed that (a) age is an important factor in learning CT, (b) girls and boys perform similarly in CT and programming, and (c) programming and CT unplugged applications are one step ahead, considering the impact of concrete experiences.

Also of great interest are the systematic literature review studies that highlight the "pedagogical merit" focusing on educational games, robotics, and toys and kits for students' CT development. Regarding educational games, Sun, Guo, and Hu (2021) reviewed 22 empirical studies to determine the effectiveness of using educational games to improve students' CT skills and the influence of various factors in instructional design on the acquisition of CT skills, while Clark and Sengupta (2019) provided a critical review by analyzing the usefulness of incorporating modeling in "discipline-integrated games" from the perspective of CT and science as practice. Triantafyllou, Sapounidis, and Farhaoui (2024) synthesized knowledge on CT and gamification to improve education for the benefit of students by also specifying gamified applications of STEM. In addition, Giannakoulas and Xinogalos (2024) investigated the effects of serious games/educational games aimed at teaching CT skills to primary school students. Their results indicated that educational programming games can help primary school students develop CT skills and basic programming concepts. Regarding the educational use of robotics, Zhang, Luo, Zhu, and Yin (2021) evaluated current studies on improving K-12 students' CT and STEM attitudes and concluded with specific recommendations for teachers to strengthen the influence of educational robots on STEM attitudes, improve the persistence of their learning effects, and further explore their application models. Another review study by Bakala, Gerosa, Hourcade and Tejera (2021) aimed to explore "robot-mediated" activities to promote CT development in preschool children by investigating the robots used, the activities proposed, and the evaluation process, and provided useful recommendations for future research. Regarding the pedagogical use of various computational toys and kits, Hamilton, Clarke-Midura, Shumway, and Lee (2019) presented both a framework for categorizing "computational toys" and another framework for assessing "toy effectiveness" for teaching CT competencies, while Yu and Roque (2019) provided an overview of "computational toys and kits" that allow young children to explore computational ideas and practices. They examined the kits from four different perspectives on how they: (a) are designed, (b) support students in exploring computational concepts and practices, (c) support children in engaging in a variety of activities and projects, and how they (d) support students in exploring other domains of knowledge.

Furthermore, it is worth mentioning the related review studies that consider the school curriculum for the development of CT in students. The study by Curasma and Curasma (2020) aimed to provide a systematic mapping of the development of CT in the regular "basic education" of South American countries, by examining educational studies and policies, methodologies designed for teaching CT, languages and tools used, and techniques or methods for assessing CT. Moreover, Acevedo-Borrega, Valverde-Berrocoso, and Garrido-Arroyo (2022) focused on the emerging conceptualization of CT by simultaneously examining the roles of the school curriculum and teachers in the face of CT for updating the current situation with research from recent years and with the new data emerging to contribute to the "collective intelligence" on methodological strategies and educational resources. Moreover, Isharyadi and Juandi (2023) highlighted the benefits and challenges of CT in mathematics learning by noting the need for resource readiness, which includes teacher competence, teaching materials, assessment instruments, and the need for teachers to feel ready to prepare and design didactic sequences, while Fauzi, Kusumah, Nurlaelah and Juandi (2024) added that CT in mathematics education could also improve students' affective side, such as increasing creativity, confidence, and involvement in the learning process.

Complementarily, there are also review studies of great interest related to CT and early childhood/kindergarten education. Su and Yang (2023) focused their review study on how to effectively teach and learn CT in early childhood education by analyzing the content knowledge, tools, pedagogical design, assessment methods, and learning outcomes that emerged from the studies. Their findings indicated that with age-appropriate instructional design, students could develop early concepts and CT skills, as well as other related skills such as collaboration, communication, and problem solving. Zeng, Yang, and Bautista (2023) focused on the teaching and assessment of CT in early childhood education (ECE). They included studies with the three-dimensional CT framework proposed by Brennan and Resnick (2012) and showed how this framework could be modified to fit the context of ECE. Based on this systematic review, they sorted out the CT components that have been shown to be appropriate for young children and proposed a CT curriculum framework for ECE that includes CT concepts, CT practices, and CT perspectives. Finally, Franco Silva, Jose Dembogurski, and Silva Semaan (2022) presented a review study on teaching initiatives and tools on programming, robotics, and other playful tools for CT development in the early years, explicitly filling the gap of CT at the kindergarten level.

Finally, recent review studies point to the need to focus on pedagogical, didactic/instructional, and teacher training issues in order to successfully integrate CT into daily classroom practice. Specifically, the study by Espina, Vieira, and Magana (2024) showed that until recently, there were no pre-service or in-service programs for teachers to integrate CT into K-12 classrooms, and therefore there is a challenge in how educators and researchers prepare the next generation of students. Ortuño Meseguer and Luis Serrano (2024) addressed the question of what constitutes effective teacher training and implementation strategies in the classroom, which remains unclear. They emphasized the influence of school practices on curriculum design. The results showed that many teachers/educators try to integrate CT without adequate training. Robot programming prevails as the primary strategy and there is a high demand for training in this area, but the use of "unplugged" activities is limited. They noted that more research is needed before advocating for CT in primary education, especially in the early grades. Therefore, successful integration of CT into primary education requires more robust pedagogical approaches supported by appropriate teacher training.

Despite the above, a review study focusing on the effective instructional strategies used for the development of CT as a thinking process in primary school students (K-5) has not yet been reported. Thus, this systematic literature review would help us to outline the effective instructional strategies that have been used through Educational Robotics, plugged-in and/or unplugged activities as suggested by various researchers after conducting empirical research. Also, the review would help other researchers to better design their empirical research for students' CT development, as well as CT educators, ICT teachers, and primary school teachers to better design more effective lessons, projects, and worksheets to enable students to have a more complete experience in cultivating their CT skills.

The rest of this paper is organized as follows: in Section 2, the research methodology of the study is described. In Section 3, we present the two main foci of our analysis: the categorization by pedagogical approach and the assessment tools. The results of the study are presented in Section 4, followed by a discussion of the review in Section 5.

2. Methodology

2.1. Aim of this paper

The central aim of this paper is to review the literature from 2006 to 2021 to identify the most effective instructional strategies implemented and studied by international research aimed at developing CT as a thinking process in primary school students (K-5), in order to provide advice for further research and educational practice.

2.2. Theoretical reference

Our review was based on and aligned with the PRISMA 2020 statement (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) (Page et al., 2021a). The aim of the PRISMA statement is to help researchers improve the reporting of their systematic reviews and meta-analyses by providing a 27-item checklist and a four-phase flowchart for original and updated reviews. In particular, it provides updated reporting guidance for systematic reviews that reflects advances in methods for identifying, selecting, appraising, and synthesizing studies, so that authors can provide a transparent, complete, and accurate account of why the review was done, what it did, and what it found (Page et al., 2021a; 2021b).

2.3. Data collection

2.3.1 Databases examined

The current study reviews papers published in scholarly journals and proceedings of international conferences for the period from January 2006 to December 2021 by searching 10 well-known international electronic databases. All of these digital libraries contain a variety of papers related to the humanities, social sciences, education and technology. In particular, this paper makes use of the following electronic databases: Association for Computing Machinery (ACM) SpringerLink, Education Resources Information Center(ERIC), Bio-Medical Library, IEEE Xplore Digital Library, Wiley, Taylor & Francis Online, Ingenta Connect, Learning & Technology Library(LearnTechLib) and Science Direct.

2.3.2 Search Term

The search was performed using the search term "computational thinking" and was strictly limited to the period from January 2006 to December 2021. A total of 10,010 papers were identified.

2.3.3. Selection of papers to be included in the review

The selection of the reviewed papers was made according to the specific criteria established at the beginning of the process. As a result, a total of 31 papers were selected as relevant to the purpose of the current study. Specifically, the criteria for inclusion of each academic paper in the review are as follows:

(a) explicit reference to the term “computational thinking” in the “*title”* and/or “*abstract”* and/or “*keywords”*,
(b) completely written in the “*English language”*,
(c) having a “typical structure” of a “*scientific/full paper”* (no roundtables, posters, short papers, demonstrations, work-in-progress papers, plenary sessions, roundtables, panel discussions, PhD dissertation proposals/consortiums, editorials, brief talk descriptions, books, invited keynotes, keynote addresses or presentations, position papers, abstracts, extended abstracts, encyclopedias, etc.),
(d) already published in scientific journals or conference proceedings after having passed through the peer-review process,
(e) teaching CT without referring to or discussing general programming in education,
(f) a focus on *CT cultivation as a thought process* (namely as a problem-solving ability focusing also on CT-practices and/or perspectives rather than solely on CT or CS Concepts) and *primary school students* (K-5: Kindergarten to grade 5, less than 11 years old students) according to the Italian educational system,
(g) providing empirical research with sufficient information about the “*research methodology”* followed, the “*participants”* as well as the *“teaching material”* and the “*research procedure”* used (no pilot and preliminary studies, preliminary investigations, pilot experiences, early findings, initial studies, exploratory studies, first observations, initial implementations and evaluations),
(h) use of at least one CT-assessment tool that evaluates CT as a thought process, and not only acquisition of CT concepts,
(i) adoption of at least one CT-assessment tool which is valid and/or reliable to some extent, and is also easily re-usable and transferable in other studies and experimental conditions for students’ CT measurement,
(j) measuring the effect of the CT intervention by using both *pre-* and *post-assessment(s)* for the measurement of the students’ CT development after the proposed didactic intervention, and
(k) *not referring* to the subsequent issues:
  - teachers’/professional education,
  - teachers’ discourse practice,
  - teachers’ perspectives, preconceptions, conceptions, perceptions or practices on CT and/or their experiences on CT,
  - mathematical algorithmic and/or algebraic thinking,
  - students with special educational needs (disabilities, disorders, autism etc.),
  - talented or gifted students, as this educational group of students is specific and the results of the studies cannot be generalized to the common population of students, and
  - CT literature review studies.

Please note that papers that are reported in two or more databases (such as Ingenta Connect and ERIC) or that refer to the same study were mentioned once. Table 1 depicts the number of items/papers identified from each electronic library/database as well as the number of the academic papers included which finally constituted the “body” of this review paper.

Table 1. The number of papers included was obtained from each of the ten electronic databases according to the inclusion criteria set.

Data-Bases searched	Total papers/items identified in search	Total papers meeting the inclusion criteria
ACM	2,788	1
Springer	2,685	6
ERIC	561	8
Bio-Medical Library	63	-
IEEE Xplore Digital Library	726	3
Wiley	421	3
Taylor and Francis Online	579	1
Ingenta Connect	975	-
LearnTechLib	756	-
Science Direct	456	9
Total	10,010	31

Finally, after applying the "above criteria", from the 10,010 papers, a total of 31 papers met the specific criteria set for inclusion in the review. Twenty-eight of them were published in "scientific journals" and four in "international conferences". For their inclusion and citation in the main body of this paper, a specific code ([1], [2],...[31]) has been given to each paper (see Appendix A for the full references).

The twenty-eight "journal papers" are published in scientific journals related to computing, education, and technology. All journals are indexed/abstracted in various databases such as Google Scholar, Scopus, EBSCO, ProQuest, INSPEC, Copernicus, Gale, PsycINFO, ERA, Information Science Abstracts, Academic Journal Guide, Cambridge Scientific Abstracts, SCImago, Computer Science Index, MathEduc, Education Research Index, and ERIH PLUS.

The distribution of reviewed papers by year of publication is shown in Figure 1.

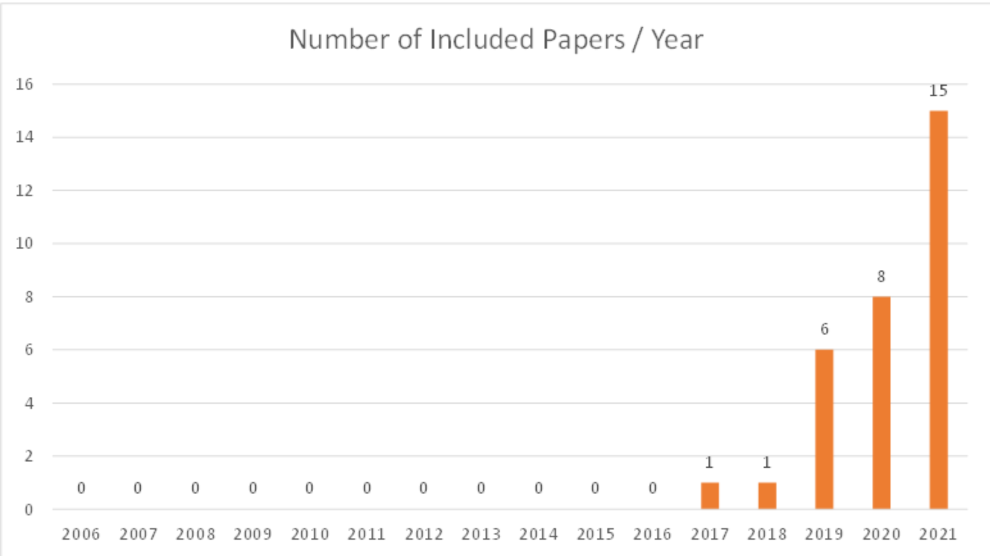

Fig. 1 Distribution of the reviewed papers by the year of publication.

It is evident that over the past 5 years-and especially in 2019-2021-there has been a noticeable increase in studies that seek to systematically cultivate CT skills in elementary students. This is an important step that shows how CT research is maturing through the use of (a) more comprehensive and effective teaching practices, and (b) more valid CT assessment tools that can reliably measure student improvement.

3. Data analysis

3.1. Categorization of paper

In line with the aims of this paper and the two archetypal approaches described above, the 31 papers that met the above inclusion criteria were categorized into the following four groups according to pedagogical approach for further analysis:

(a) *Creative*
(b) *Problem-solving*
(c) *Mature*
(d) *Interdisciplinary*

(a) and (b) are to some extent related to the two archetypal approaches described in section 1.2, while we have included in group (c) those papers that offer a teaching strategy that is able to overcome the limitations of the two basic approaches (we therefore call them "mature").

Of course, the boundaries are blurred, and none of the 31 papers refer exactly to one of the two models. Therefore, in the categorization process, the two authors of this review independently placed each study into a category, while the final choice was the result of mutual agreement. We also decided to avoid a generic fifth category and tried to assign each paper to a category as far as possible; some positions may therefore appear somewhat forced. Further details of the criteria used to identify the categories are given in the next section.

We then placed in a separate category (d) those studies that start from an interdisciplinary approach in which coding activities are intertwined with another subject area. This is a very specific approach, based on the awareness that CT "influences research in almost all disciplines, both in the sciences and in the humanities" (Bundy, 2007, p. 67). Such work is influenced by teaching strategies that are specific to their discipline and therefore deserve their own category.

Table 2 shows the 31 papers grouped by a pedagogy approach.

Table 2. Included papers by pedagogical approach.

(a) Creative	(b) Problem-Solving	(c) Mature	(d) Interdisciplinary
[9] Jiang & Li, 2021	[1] Angeli & Valanides, 2020	[4] Chou, 2020	[2] Angeli et al., 2019
[11] Jin et al., 2021	[3] Brackmann et al., 2017	[10] Jiang & Wong, 2021	[8] Hsu & Liang, 2021
[15] Mouza et al., 2020	[5] Del Olmo-Munoz et al., 2020	[12] Lin et al., 2020	[18] Ntourou et al., 2021
[19] Özcan et al., 2021	[6] Fanchamps et al., 2021	[14] Matere et al., 2021	
[20] Özmutlu et al., 2021	[7] Hooshyar et al., 2021	[17] Noh & Lee, 2019	
[21] Perez-Marín et al., 2020	[13] Ma et al., 2021	[24] Taylor & Baeck, 2019	
[23] Relkin et al., 2021	[16] Muñoz-Repiso & Caballero-González, 2019	[26] Tran, 2019	
[29] Yildiz-Durak, 2018	[22] Qu & Fok, 2021	[27] Wang et al., 2020	
[30] Yılmaz Ince & Koc, 2020	[25] Tonbuloğlu & Tonbuloğlou, 2019	[28] Wei et al., 2021	
		[31] Zapata-Caceres et al., 2021	
9 papers	9 papers	10 papers	3 papers

3.2. Assessment Tool

In order to properly review the results of the selected papers, we cannot ignore the issue of the CT assessment tools used. As pointed out by Román-Gonzalez, Moreno-León, & Robles (2019), different tools aimed at measuring CT actually assess different aspects of this construct, to the point that they promote the adoption of an "assessment system" that combines several tools, rather than using a single tool.

The authors have made a classification of the tools commonly used in research to date. Among all the 7 categories, we focus here especially on the "diagnostic tools" and the "perception-attitude scales", as they fit better to a pre- and post-assessment design, which was one of our inclusion criteria. Nevertheless, some papers adopt a larger "assessment system" (Grover, 2015), which also includes other types of tools, such as "formative-iterative" or "data mining". Therefore, in our review, for each paper, we will distinguish the tools used for the pre- and post-assessment from all other possible ones.

The diagnostic tools "aim to measure the CT aptitude level of the subject [and] that they can be administered in a pure pretest condition" (Román-Gonzalez, Moreno-León, & Robles, 2019, p. 81). The best known of these tools is the Computational Thinking Test or CTt (Román-González, 2015; Román-González, Pérez-González, & Jiménez-Fernández, 2017), which has also been adapted for the first grades of primary school under the name Beginner CTt or BCTt (Zapata-Cáceres et al., 2020). The tool is composed of 28 items, each of which is a puzzle presented either in a maze or in a canvas interface.

The perception-attitude scales "aim to assess subjects' perceptions (e.g., self-efficacy perceptions) and attitudes not only toward CT" (Román-Gonzalez, Moreno-León, & Robles, 2019, p. 82). The best known of these scales is the Computational Thinking Levels Scale (CTLS) (Korkmaz et al., 2015), which measures five factors related to five higher-order cognitive skills: creativity, algorithmic thinking, cooperation, critical thinking, and problem solving.

In our previous review (Kakavas & Ugolini, 2019), we found that CTt and CTLS are the most commonly used tools. Therefore, we have given them a privileged position in our analysis, specifying all other tools used.

However, it is important to note that although both claim to assess CT, they do not measure the same facets of this construct. Román-Gonzalez et al. (2019) related the types of tools to the three dimensions of Brennan and Resnick's (2012) operational definition, highlighting that diagnostic tools, such as the CTt, have an excellent adequacy in measuring CT concepts and a moderate adequacy in measuring CT practices, while perception-attitude scales, such as the CTLS, have an excellent adequacy in measuring CT perspectives and a moderate adequacy in measuring CT practices. Therefore, a combination of both would be a better way to fully assess CT.

Therefore, as far as this review is concerned, while presenting and discussing the results of the studies examined and, consequently, the effectiveness of the instructional strategies, we will have to take into account the type of assessment tool adopted and, each time we detect an improvement in CT, we will specify it.

4. Results

4.1. "Creative" approach

Although none of the papers selected directly refer to Resnick's "creative learning" model, we decided to group together under this label the studies that somehow adopt one or more features of this approach, namely:

- an orientation to self-expression of coding;
- a project-based approach (rather than a puzzle-based one);
- a focus on peer collaboration;
- the predominantly out-of-school or after-school context, free from specific curricular objectives;
- awareness of the use of Scratch, as an environment and not only as an easy-to-use block-based language;
- more or less explicit references to constructionism and its authors, like Papert or Resnick.

Therefore, we have not included here papers that aim to integrate creative learning in curricular contexts, which are better placed within the "mature" approach; papers [11], [15], [19], [20], [21], [29], and [30] are placed in this category because their context is out-of-school or after-school. Paper [23], although its context is in-school, is the study that most refers to a constructionist approach using Marina Bers' (2019) theoretical framework. Paper [9] is placed here because it focuses on Scratch and, although it is not widely mentioned, it addresses collaboration within pairs, emphasizing the storytelling part of programming.

As can be seen in Table 3, the typical study belonging to this category analyzes long-term programs, addresses 5th grade students in an after-school or out-of-school context, and adopts a single-group experimental design; it also focuses mostly on self-perception, as the majority of papers assess CT through a CT perceptions-attitudes scale.

The paper of [23] deserves a separate analysis: it is the only research of this group that addresses 1st and 2nd grade students, it adopts a well-structured approach that refers more strictly to a constructionist paradigm (Marina Bers' "Computer as Another Language" - CAL - approach (Bers, 2019)), its methodology is very rigorous both in terms of sample size (848 children involved) and experimental design (with experimental and control groups); it uses a diagnostic assessment tool, the TechCheck (Relkin et al., 2020), which is consistent with the overall approach. The research shows a significant improvement in CT, which is greater for 1st grade students than for 2nd grade students; this is explained by a higher baseline level of CT, but we note that it may also be due to 3 hours of additional instructional time based on teacher feedback in the 1st grade program. The specificity of the CAL curriculum is that it links coding to literacy, provides children with opportunities for self-expression, and fosters creativity. Not surprisingly, among all the dimensions captured by the TechCheck instrument, algorithms, modularity, and representation showed the highest percentage of improvers; these three dimensions are actually the most involved in the CAL curriculum, as it challenges children to rework story elements from literature into programming.

Table 3. Creative approach papers.

Study	Research Design	Context	Grade	# Participants (Exp. Gp. + Ctrl Gp. if any)	Intervention	Duration	Pre-/post-Assessment Tool(s)	Other assessment tools	Instructional strategy or approach	Notes
[9]	1 Group	In school	5th	336	*Scratch*	5 weeks	CTLS			
[11]	Exp. Gp. + Ctrl Gp.	After School	3rd	11 + 10	*Scratch* Extensive Scaffolding	14 weeks	CTt (Chinese version) CTLS (adapted)	*Dr. Scratch* Interviews	Extensive Scaffolding	
[15]	1 Group + Mixed Methods	After School	4th-5th	138 (including 6th grade students)	Program	1 Semester (may be repeated)	CT Concepts	Interviews Observations Prj analysis	Many small strategies Focus on Peer collaboration	Main assessment on concepts
[19]	Exp. Gp + Math Gp + Ctrl Gp	After School (lab., rand.)	4th	58+52+64	20 hrs program AlgoDijital	20 h per 10 weeks	Tran's CT Test Matrix reasoning task Spatial Orientation Test		Long program ending with *Scratch* project	
[20]	1 Group pre-, post- and delayed	After School	5th	24	Promoting Self-Efficacy	2 days (8 hours)	CTS Self-efficacy Scale (≠ CTLS):			
[21]	1 Group	Out of School	4th-5th	51 (only 4th and 5th grade students who actually did both pre- and post- test)	MECOPROG	6 hours (1/2 h per week)	CTt + Content Test		MECOPROG (metaphor based)	
[23]	Exp. Gp + Ctrl Gp.	In school	1st and 2nd	667 + 181	CAL Kibo Curriculum	12-15 hours (1-2 h per week)	TechCheck		CAL Kibo	
[29]	2 Groups, comparative	After School	5th	110	*Scratch* vs. *Alice*	8 weeks	Engagement Scale Reflective Skills for Problem-solving Scale CTLS			Comparative
[30]	1 Group	Out of School	5th	3	Programme	10 days in 2 weeks	CTLS	Satisfacion Form Prj Diaries		Also 29 6th grade participants

The study [15] also has some specificities: it analyzes a very long program (1 semester, over 2.5 years, and some participants attended more than once), it overtly adopts a mixed methods approach, and the quantitative results address only the acquisition of CT concepts, leaving the most interesting information to the qualitative data collection. While the former shows a significant improvement, it is worth noting that the peer collaboration aspects are often mentioned by the participants as the most interesting from a pedagogical point of view. For example, participation in the Scratch online community was particularly fostered by the educators, who routinely devoted the first 10 minutes of each lesson to having the children review other projects in order to be "inspired" by them, a typical feature of Resnick's (2017) "creative learning" approach.

Resnick [21] discusses the effectiveness of a particular metaphor-based strategy (MECOPROG), comparing increases in CT (measured by CTt) and knowledge of programming skills. Although we have placed it in this category - because of its theoretical framework and the use of Scratch, considered as an "authoring environment" - it also focuses on the relationship between CT as a thought process (measured by CTt) and the acquisition of curricular outcomes (measured by a specific tool), so we can see here a curricular perspective of the intervention.

[9], [11], [20], [29] and [30] share the adoption of a perception-attitude scale. [9], [11], [29] and [30] use the CTLS, while [20] uses another scale that focuses more on self-efficacy - the CT self-efficacy scale (Gülbahar, Kert & Kalelioğlu, 2019) - which is consistent with the intervention that specifically aims to promote self-efficacy. Although [11] is a small study in terms of the number of participants, it is interesting for several reasons: first, it addresses 3rd grade students, all girls; second, it assesses CT through multiple tools (namely CTt + CTLS), differentiating CT skills and self-efficacy; and finally, it argues for the importance of structured scaffolding in a constructivist approach; the result is that extensive scaffolding in a Scratch-based creative approach improves both CT skills and self-efficacy.

In particular, [29] compares two coding environments, Alice and Scratch, according to engagement, reflective skills and CTLS; while there is a greater improvement in engagement and reflective skills with Scratch, CTLS has a better result with Alice.

[9] also focuses on Scratch, with an improvement in CT (measured with CTLS). [30] studies a longer program, but it is less significant for us because there are only 3 5th grade participants. [19] studies a long-term program; although it includes some problem-solving oriented activities at the beginning, it leads the students to create their own Scratch project, showing a significant efficacy on CT (by Tran's test), and thus a kind of near-transfer from coding activity to CT development.

[9], [29] and [30] give us some interesting information about the 5 factors measured by CTLS, related to the specific intervention. In [9], the greater improvement is in Cooperativity, with a less significant improvement in Critical Thinking and Creativity, while Algorithmic Thinking and Problem Solving are stable. Although the declared independent variable is the mere use of Scratch, the pedagogical approach is strongly cooperative, as the children are grouped in pairs with specific roles, followed by a discussion between the pairs. In [30], the dimension of algorithmic thinking shows the highest improvement. The intervention is actually a 2-week full immersion, but the children are significantly older, as only 3 of them are in 5th grade. However, there is one factor that is significantly stable across three papers ([9], [29], and [30]), and

that is problem solving. We can assume that the kind of problem solving measured by the CTLS self-perception scale is more suited to a different approach to coding than the "creative" one, in which the problems are posed within projects and are therefore less structured than, for example, puzzles.

4.2. Problem solving approach

We have grouped in this category the studies that are based on a set of tasks or challenges that children are asked to solve. Such tasks do not necessarily look like the code.org puzzles: some of the interventions focus on some interesting problems, as we will see below.

From a methodological point of view, we find that the studies in this category use a very rigorous design. 7 out of 9 studies use both an experimental and a control group, and the other two use a mixed methods design. 5 out of 9 studies randomly assign students to each group. In addition, the intervention in 4 of them was implemented in a specific context, even if participation was somehow linked to one or more schools (see Table 4).

For a more detailed analysis, we divide the studies into two subcategories according to school level, since the approaches can vary considerably both in terms of research design and intervention; thus, we have 5 papers focusing on 5th grade students and 4 others focusing on K-3 students. and 4 other papers focusing on K-3 children.

Table 4. Problem solving approach papers.

Study	Research Design	Context	Grade	# Participants (Exp. Gp. + Ctrl Gp. if any)	Intervention	Duration	Pre-/post- Assessment Tool(s)	Other assessment tools	Instructional strategy or approach	Notes
[1]	Two Groups Comparative For a specific RQ	In School (lab. rand.)	K (5-6 y.o.)	50	Bee-Bot 3 PS scenarios 2 scaffolding types	Two 40 minutes sessions, separated by 2 days	Color test (s.m.) Spatial relations test (s.m.) CT Assessment rubric (s. m.) (pre- and post- 2nd scenario)		3 PS Scenarios 2 scaffolding types to overcome the limitation of the robot's interface	
[3]	Exp. Gp + Ctrl. Gp.	In School (not rand.)	5th	20+23	Unplugged activities	10 hours in 5 weeks	CTt	Task results	Unplugged, linked to 4 pillars	Also 16+14 6th grade
[5]	Exp. Gp + Ctrl. Gp. + unified follow-up	After School (lab., not rand.)	2nd	42+42	Unplugged activities	5 x 45 min in 8 weeks	Bebras (pre-, mid- e post-) Motivation Test (mid- e post-)		Combination unplugged (first) and plugged-in (then)	
[6]	Exp. Gp + Ctrl. Gp	After School (lab., rand.)	5th	15+15	15 SRA-programming tasks in Bomberbot	5 x 1h, 15 programming tasks	CTt		SRA + Pairs	
[7]	Exp. Gp + Ctrl. Gp	In School (rand.)	5th	36+43	Autothinking	60-75 min session	CTt (Adapt.) Learning Attitude Scale (Adapt.)		Adaptive environment	
[13]	Exp. Gp + Ctrl. Gp	In School (not rand.)	5th	31+32	IGGIA	15 weeks 90 min/w	CTt CTLS (ad. China)	*Dr. Scratch* Focus Gp.	IGGIA Problem Solving Model	
[16]	Exp. Gp + Ctrl. Gp	In School (not rand.)	K (3-6 y.o.)	67+64	BeeBot + TangibleK + group act.	7 sessions, 4h per days	SSS Rubric (Adapt.)		TangibleK Storytelling Group act.	
[22]	1 Group + Mixed Methods	Out-of-School (rand.)	1st-3rd	32	Ed. Robotics 12 lessons, 11 problem solving tasks	4 weeks 3 lessons/w 90min/lesson	Rubric Scoring Week 1-2 vs. Week 3-4	Semi-str. Interviews Open obser.	Robot-Student Interaction	3 levels model for R-S Interaction
[25]	1 Group + Mixed Methods	In School (not rand.)	5th	114	10 lessons unplugged problems	10 weeks 2 h/w	CTLS	Daybook + Observation forms	Unplugged	

4.2.1 Papers addressing 5th grade students

We found 5 papers with 5th grade students. 4 of them ([3], [6], [7] and [13]) adopt a research design with both an experimental and a control group (two of them with randomized allocation), with the number of participants in the experimental group varying from 15 to 36; all of these studies assess CT using the CTt (or its adapted versions); one of them ([13]) adopts a much more complex assessment system, including the CTLS and other assessment tools, in addition to pre- and post-tests, such as Dr. Scratch and a focus group, thus better covering the construct of CT (see section 3.2). The fifth paper ([25]) adopts a mixed methods approach, considering a single group of 114 participants and using CTLS as a pre- and post-test, combined with other qualitative data collection tools. Such a finding confirms that CTt and CTLS are widely used in research with 5th grade students.

Three papers ([3], [6] and [7]) effectively focus on the tool through which the learning activities were carried out: [3] deals with unplugged activities, [6] focuses on the Sense-Reason-Act feature of a virtual robot programming environment, and [7] evaluates the effectiveness of a specific adaptive software in terms of both CT and learning attitude. [3] finds that the improvement in CT with unplugged activities is somehow comparable to a similar improvement previously found with the use of code.org puzzles; the authors seem to be aware of the fact that unplugged activities are ultimately not sufficient on their own, and therefore they question the point at which an actual computing device would be needed to further improve CT. Within a virtual robot programming environment, [6] compares the usual linear programming approach with the Sense-Reason-Act feature (which corresponds to the typical behavior of robots) and finds that the latter promotes the development of a higher level of CT, thanks to which students can solve complex CT tasks. The study [7] claims the effectiveness of an adaptive environment in cultivating CT, but its main problem is that the study compares its use with the traditional transmissive approach, whereas a comparison with a traditional puzzle-based approach would have been more interesting. Nevertheless, the study finds a significant improvement in the "debugging" factor of CTt, which is related to the extreme focus on feedback of the adaptive environment.

A slightly different approach is taken in [13], which studies a specific problem-solving model called IGGIA (Identify, Gather Data, Generate, Implement, Assess), which strongly characterizes CT as a problem-solving process and associates one or more CT components to each phase. The intervention is halfway between a project-based one (it uses Scratch) and a task-based one, since during a 15-week program it includes several activities, each with a strong disciplinary stamp, materialized in a project. Nevertheless, we have included this paper in this group because it actually focuses on a specific problem-solving model, although it differs from the other task-based studies. In fact, the results show that a typical problem-solving instruction provokes an improvement in both CT (measured by CTt) and self-perception (measured by CTLS), and therefore the authors can conclude that they confirm the conceptualization of CT as a problem-solving ability.

4.2.2 Papers addressing K-3 students

We found 4 papers addressing K-3 students: 2 of them ([1] and [16]) focus on kindergarten, [22] covers grades 1to3, while [5] addresses 2nd grade students; except for the latter, which focuses on unplugged activities, the other three papers actually refer to educational robotics. We can also consider papers [1], [16], and [22] as a whole from a methodological point of view, since all three use a CT assessment rubric, although [1] uses a self-made one, [16] adapted the SSS rubric from the Tangible K project (Bers, 2010), and [22] uses a CT rubric (Leonard et al., 2016). It should be noted that the latter, while using a single-group mixed methods design, also includes semi-structured interviews and open observations ( ). Paper [5], in addition to a motivation test, collects data through adapted Bebras tasks, which are functional to assess the transfer of CT skills (Román-Gonzalez, Moreno-León & Robles, 2019).

In terms of this review, paper [1] seems to be the most interesting of this category; in the context of a scenario-based robotics activity, it focuses on two types of scaffolding: individualistic, kinesthetic, spatially oriented and manipulative, using direction cards, versus a collaborative writing task activity with deeper teacher involvement. In this case, the only difference between the two groups is the type of scaffolding. Although all activities lead to an improvement in both CT and spatial relations, the research confirms the hypothesis derived from neuroscience that scaffolding of the first type leads to a better improvement in boys, while scaffolding of the second type leads to a better improvement in girls.

The study [16] is less interesting from a pedagogical point of view, since it compares an experimental group that participated in a robotics intervention with a control group that didn't; not surprisingly, the children in the experimental group improved their CT skills significantly more than the children in the control group, who improved their CT anyway due to the maturation effect and the testing effect. Although the authors refer to Marina Bers' TangibleK framework, they actually adopt BeeBot, with several carpets or mats, and therefore the intervention doesn't seem to implement Marina Bers' constructionist principles.

In the context of the first three years of primary school, and thanks to a mixed methods design, [22] investigates student-robot interaction, focusing specifically on the features of robotics that characterize it as different from on-screen programming: in addition to the programming-computing dimension, the authors also consider the dimensions of observational investigation (students observe the robots' "responses") and participatory investigation (students physically interact with the robots' sensors to explore their "responses"), within a broader conception of S-R interaction. The results, among the 6 sub-dimensions of the CT rubric show a significant improvement in abstraction, logical thinking, and the ability to analyze and implement solutions, allowing the authors to conclude that interacting with a robot helps to abstract real-world situations to the programming process.

While papers [1], [16] and [22] focus on educational robotics from different perspectives, paper [5] deals with unplugged activities. Better, it supports a mixed approach that combines both plugged and unplugged activities and finds that it increases both CT (measured by Bebras tasks) and motivation. From a strictly pedagogical point of view, we have to say that the research adopts only code.org puzzles (also due to a methodological issue related to the comparability between their plugged and unplugged versions), but nevertheless we think it is important to study a mixed approach, especially since the fully unplugged one is considered more as a B-plan when material conditions do not allow a plugged solution, rather than an informed choice.

As an overall conclusion for the problem-solving category, we can observe that only 3 papers ([6], [7] and [13]) out of 9 are of the on-screen type, while the others focus either on robotics ([1], [16] and [22]) or on unplugged activities ([3],

[5] and [25]). Furthermore, although the paper [6] deals with on-screen activities, it simulates a typical robot behavior, Sense-Reason-Act. We can conclude that a task-based approach is better suited for this kind of off-screen activity, even if some of these studies go further into some specific aspects, such as teacher scaffolding, student-robot interaction, unplugged + plugged combination. Only one of the two remaining papers studies the benefits of an on-screen activity (using an adaptive software for programming), while the last one ([13]) takes a significantly different approach, dealing with a specific problem-solving model, as we explained above.

4.3. Mature pedagogy approach

In this section, we present the main category of this review according to its objectives. To be included in this category, a study must both address curricular outcomes (thus overcoming the limitations of the "creative" approach) and explicitly test (with both a pre- and post-test) an instructional strategy different from, or in addition to, problem solving (thus overcoming its limitations). For this reason, we decided to call this category "mature".

Ten studies belong to this category (see Table 5). From a methodological point of view, we can see that almost all of the studies - with only one exception [17] - use a pre- and post-assessment, but also collect data through other tools, namely observation, project analysis, interviews, data-driven learning analytics, or even product scoring. We can conclude that studies that are more pedagogically oriented will not only rely on a quantitative pre-/post-assessment but will need a more complex picture of the results.

Table 5. Mature pedagogy approach papers.

Study	Research Design	Context	Grade	# Participants (Exp. Gp. + Ctrl Gp. if any)	Intervention	Duration	Pre-/post-Assessment Tool(s)	Other assessment tools	Instructional strategy or approach
[4]	1 Group + follow-up	In School (rand.)	3rd	11	*ScratchJr* on tablet Program	8 weeks	CT Competence Test	Learning Observation Doc.	Review-Copy-Create
[10]	2 Groups	In School (rand. per class)	4th-5th	129	Comb. Unplugged + plugged-in	5 weeks	s.m. validated	Semi-str. Int. Observation	Problem (unplugged) to Project Pattern recognition
[12]	1 Group	In School	K	7	Robotics (mBot)	2 weeks	Bebras	Observation	GBL+TUI Five E Framework
[14]	2 Groups	(rand.)	4th-5th	25+24	DBL approach	6 days, 3 hrs/d	Pre- comp. progr.skills Post- modified CTt	Product Scoring by Experts	DBL
[17]	1 Group	In School	5th	68	Educational Robotics	11 weeks	Bebras Torrance Creativity Test		Creative Problem Solving (structured ID)
[24]	3 Groups comparative	In School	4th-5th	191	Robotics	14 weeks, 1h/w	CTt Motivation Test	2 Rubrics	Grouping (by gender, by role)
[26]	1 Group + Mixed Methods	In School	3rd	183	Unplugged + code.org	10 weeks, 1h/w	s.m.	Interview	Pair Programming Kolb Cycle 3 environment
[27]	1 Group	In School	5th	48	Interest-Driven Creator	1 semester 1h/w	CTt (selection) Content Test CTLS (PS + Creat.)	Project analysis	Interest-Driven Creator (IDC)
[28]	2 Groups Mixed Methods	In School (rand. per class)	4th	84+87	CT Curriculum *Scratch*-based	16 weeks 40 min/w	*DrScratch* Kong's Prog. SE Scale	Semistructured Interviews	Partial Pair Programming
[31]	3 Groups comparative + CG + follow-up	In School (rand.)	1st-5th	76+72+28	Playing with Blue Ant Code	5 weeks 10 min play/w	BCTt (pre-, post- e follow-up).	Blue-Ant-Code (in-game data driven learning analytics) Questionnaires	Problem solving + Game-Based Learning + Collaborative Learning

Four studies ([24], [26], [28] and [31]) deal with collaborative strategies, albeit from different perspectives: two of them ([26] and [28]) focus on pair programming (Campe & Denner, 2020); another one [31] deals mainly with game-based learning (GBL), specifically comparing a collaborative way of playing with a non-collaborative one; finally, [24] focuses on the effectiveness of different modes of grouping in coding activities, by role and gender. The research shows the extreme versatility of collaborative strategies. For example, both [26] and [28] study pair programming, but from two opposite perspectives: [26] starts from a puzzle-based approach and the quantitative results show an improvement in typical CS knowledge (algorithm, loops and debugging), while [28], although lowering the strategy to a partial pair programming, clearly refers to a constructivist approach, accompanying the students in appropriating their projects, with a significant improvement in both CT and programming self-efficacy. [31] is very detailed, especially from a theoretical point of view: it combines GBL and collaborative learning strategies to cover all three dimensions of Brennan and Resnick's definition; the authors also go through all primary school grades to find out which CT skills can be learned in each grade of primary school, finding that collaboration is particularly effective in the early grades, namely for low percentile students and students with special needs. [24] compares the effects of different compositions of 3-4 person groups by role and gender on their performance in a robotics activity, their CT level (by CTt) and their motivation to learn. The study shows that the gender composition doesn't show a significant difference on any of the dependent variables, while the assigned roles, whether fixed or rotating, do. It should be noted, however, that [28], although dealing with pairs, finds that gender composition has an impact on both CT (of Dr. *Scratch*) and programming self-efficacy: girls in girl-girl pairs do not improve their CT, while those in boy-girl pairs do; girls do not improve their programming self-efficacy when paired with a boy or with another girl.

The other six papers present a structure in their instructional design aimed at integrating learning objectives related to the school curriculum and elements that can improve creative thinking. Ideally, we can order them according to their degree of affinity with a problem-solving approach or with a constructivist creative approach. According to this order, the description of these strategies, together with the main results of the studies, is summarized in Table 6. The closer the strategies are to the "problem-solving" polarity, the more they are based on a problem-solving approach (e.g., [12] is based on a specific way of designing robotics problems according to a 5-E framework; [10] moves from unplugged puzzle-based activities to a constructionist-based plugged-in approach; [17] explicitly refers to a "creative problem-solving" approach). Similarly, the closer the strategies are to the "creative constructionist" polarity, the more they adopt some specific elements of a constructionist approach; for example, [27], although its strategy includes a problem-solving phase, leads to the creation of projects to be shared; [14] focuses on reflection in activities that promote both problem-solving and creativity; [4] focuses on constructionist-based CT practices, namely "reusing and remixing", and students are encouraged to create their own original projects. All these study results show an improvement in CT, but most of them bring more refined data, for example, on retention, acquisition of higher-level skills, creativity, or interest in learning, while not ignoring the acquisition of curricular content.

Table 6. Mature instructional strategies or approaches, ordered by their affinity to problem-solving or constructionism approach

	Study	Grade and duration	Instructional Strategy / Approach	Results
Problem-Solving	[12]	K 2 weeks	Educational Robotics problems, with direction cards, carefully designed according to Game-Based Learning approach, the theory of the Tangible User Interface and a five E framework (Engage, Explore, Explain, Elaborate Evaluate).	Small study (7 children); improvement of both CT (via Bebras) and learning interest.
	[10]	4^{th}-5^{th} 5 weeks	Program organization moving from problem-based unplugged activities (focused on pattern-recognition) to constructionist-based plugged-in programming activities.	Advantages on conditionals, operators and pattern recognition (via specific validated test), more significant in 4^{th} grade children.
	[17]	5^{th} 11 weeks	Creative Problem Solving (CPS). Well-structured Instructional Design consisting of five stages. Based on a Problem-Solving process, it implements several strategies aiming at promoting creativity and divergent thinking (like brainstorming and SCAMPER)	Improvement both of CT (via Bebras) and creativity, namely regarding fluency. Children starting with a lower level improve the best.
	[28]	5^{th} 1 semester	Interest-Driven Creator (IDC). 45 minutes-long sessions, leading to the creation of projects to share. Within those sessions, we have an “Interest arousing phase”, a “Problem-Solving phase” and “Idea expressing phase”. Therefore a specific Problem-Solving phase is embedded within a constructivist-based structure.	Improvement of CT (via a selection of CTt items) and content skills. Improvement of Problem solving self-perception (via CTLS items). Tendency to an improvement of creativity (via non structured observation).
	[14]	4^{th}-5^{th} 6 days	Design-Based Learning. Students are engaged in solving real-life design problems, while reflecting on their learning process. They use design activities to meet curriculum objectives. Students design a final project, by the means of specific design worksheets which assist them to foster problem-solving abilities and creativity.	With the same acquisition of curricular content, the analysis of final projects (with a scoring scheme) denotes, in the experimental group, the acquisition of higher level skills based on the three dimensions considered (value/quality; novelty; program concept)
Creative constructionist	[4]	3^{rd} 10 weeks	Review-Copy-Create cycle 10 two hour long sessions. For each session: - 20 minutes Review phase (discussion of students take-home assignments); - 40 minutes Copy stage (presentation of programming skills by the teacher);	Significant improvement and good retention, for all CT Concepts (except for Loops). Improvement of “Reusing and Remixing” CT Practice.

			- 40 minutes Create stage (students modify the examples adding their own ideas).	

4.4. Interdisciplinary approach

As mentioned above, we have placed in this group those studies that start from an interdisciplinary approach, where coding activities are intertwined with another subject area. Their instructional strategies are therefore much more focused on a disciplinary approach, and so are their results (see Table 7).

In [2], educational robotics is used for 1st-2nd grade students to make them acquire knowledge about the water cycle, so it is problem-solving oriented with some storytelling elements related to this specific content. Although it addresses a small group of students (20) without a control group, the research shows a learning gain both for CT (through a specific assessment rubric) and for disciplinary content.

[8] addresses 3rd grade students using a GBL approach in English as a foreign language classrooms, comparing two versions: an unplugged (though game-based) one and a plugged one using educational robots. Both approaches improve both CT (through a self-made test) and English skills; at the same time, the research shows that the plugged-in approach reduces learning anxiety and improves critical thinking, while the unplugged approach increases the cooperative tendency.

[18] targets 5th grade students and uses a physical computing approach (using *Scratch for Arduino*) to get children to perform computational experiments to enhance physics learning. Although there is also a learning gain in CT (by Bebras), the main focus of the study is the acquisition of higher-level disciplinary skills; children in the experimental group performed better than those in the control group, who only do physical experiments, especially in terms of understanding the correct model.

All these studies show how programming activities are able to promote learning in specific disciplinary areas of the curriculum; the improvement of CT, although present to some extent, remains in the background.

Table 7. Interdisciplinary approach papers

Study	Research Design	Context	Grade	# Participants (Exp. Gp. + Ctrl Gp. if any)	Intervention	Duration	Pre-/post-Assessment Tool(s)	Other assessment tools	Instructional strategy or approach
[2]	1 Group	Out-of-school	1st-2nd	20	Educational Robotics (Blue-Bot) 2 scenarios	2 days (45 min/d)	Knowledge Test (discipl.) CT Assessment Rubric (s.m.)		Problem Solving + Storytelling (disciplinary based)
[8]	2 Groups Comparative	In School	3rd	24+24	Plugged-in or unplugged English activities	10 weeks 2x40min/w	s.m. Test (English + CT) Self Perception Quest. (learning anxiety + Crit. Th + Coop. Tendency)		Game-Based Learning Coop. Learning (pairs) Plugged-in (robotics) vs. Unplugged
[18]	2 Groups	In School	5th	15+18	*Scratch* 4 *Arduino*	4 weeks (1 activity/w)	Bebras + MSLQ (3 dim.) + Content quest. Programming quest (only XG)	Interviews	Physical Computing + Coop. Learning

5. Discussion: Effective Instructional Strategies

In keeping with the research goal of this review, we will now discuss the findings to outline the most effective instructional strategies for CT gain as they emerged from the research we reviewed. To better serve our purpose of providing suggestions for researchers and practitioners, we will organize the discussion by grade level (K, 1st-3rd, and 4th-5th), although we will refer to the aforementioned categories to better outline the instructional strategies.

5.1. Kindergarten

Only 3 papers out of 31 actually deal with kindergarten, and all three use educational robotics to promote CT: two of them use BeeBot and the third uses mBot. All three are mainly problem-based; we decided to place [12] in the "mature" category, since the proposed activity follows a specific learning cycle (the 5E learning cycle). Although [16] refers to the constructivist-based TangibleK approach, it actually proposes a challenge-based activity with some elements of storytelling; [1], although referring to a fully scenario-based approach, focuses on a specific issue, namely the scaffolding type with regard to genre.

The research shows that a scenario-based educational robotics significantly improves CT, which results from the paper [16], which has the longest duration (7 sessions compared to 2 in the other studies), the largest sample, and a control group composed of children who didn't participate in any robotics activity. [12] proposes an interesting theoretical model for the design and implementation of educational robotics activities based on GBL principles, it adopts a tangible user interface and follows a 5E learning cycle. However, although the empirical research shows some promising learning gains, the small number of children involved (7) requires further research. The most interesting finding of [1] is that the use of direction cards to follow instructions (individualistic, kinesthetic, spatially oriented, and manipulative scaffolding type) is more effective for boys, while working with the teacher for the same purpose (collaborative, writing scaffolding type) is more effective for girls, although both types imply an improvement in CT for both genres.

Finally, CT is mainly assessed through the use of specific rubrics; paper [12] uses Bebras questions of different levels of difficulty combined with observation.

5.2. Grades 1st to 3rd

We found 9 out of 31 papers dealing with grades^{1} to^{3}, including [31], which deals with all five grades of primary school, but as mentioned above, its main results cover grades^{1} to^{3} best.

Three of them ([2], [22], and [23]) include children in^{1st} grade. In a kind of continuity with the studies dealing with kindergarten, all three use educational robotics, but in this case only one of them ([22], $^{1st-3rd}$ grade) actually focuses on problem-based activities (using a Chinese robot - Kazi EV5 - similar to LEGO Mindstorms), dealing specifically with robot-student interaction. [2] ($^{1st-2nd}$ grade) adopts a disciplinary approach (water cycle), using a disciplinary-based storytelling in 2 scenarios. [23] adopts a creative approach, using KIBO in the context of the "Computer as Another Language" framework, which explicitly teaches coding as a symbolic representation system for expressive purposes.

We can consider the latter separately, both for the very high number of its sample (667 children belonging to the EG, 181 to the CG), and for its theoretical consistency, due to the structure of the CAL-KIBO framework and the appropriateness of the assessment tool used, TechCheck, developed and validated by the same research group. The study found a significant improvement in CT (using TechCheck), which was greater for students in the^{1st} grade. It should be noted that this specific approach naturally leads to an improvement in the representation, modularity and algorithmic dimensions of CT.

Conversely, both studies [2] and [22] deal with small samples with only one group, although [22] declares a mixed-methods approach that makes use of a broader assessment. [2] finds that adding discipline-based storytelling to educational robotics implies a learning gain in both CT (by assessment rubric) and content knowledge (related to the water cycle), but for the latter, a control group would give more significant results. The quantitative results of [22] show that a complex approach to robot-student interaction, including not only computer-programming interaction but also observational and participatory investigation, leads to a CT gain that is particularly significant in the abstraction dimension.

Overall, the studies confirm that educational robotics improves CT, but the most interesting fact is that an expression-based approach like CAL-KIBO improves its representation, modularity, and algorithm dimensions, while an investigation-based approach improves its abstraction, reasoning, and problem-solving analysis dimensions. Three other studies ([5], [26] and [8]) focus on unplugged activities with^{2nd} and^{3rd} grade students. We can consider [8] as a kind of bridge, since it compares, from an interdisciplinary point of view, a group of students using educational robotics with another group participating in unplugged activities, and shows that the former reduce their learning anxiety and improve their critical thinking skills, while the latter improve their cooperative tendencies, although both improve CT and English learning.

[5] and [26] both use code.org puzzles, clearly moving away from a problem-solving approach; [5] focuses on a mixed approach, comparing a fully plugged-in program with one that includes both unplugged and plugged-in activities, showing that^{2nd} graders who participate in the latter improve both their CT (through Bebras) and their motivation, making this mixed approach a good solution for a first approach to programming. The work of Bebras et al [26] offers an interesting approach as it combines unplugged activities to pair programming and the adoption of an experiential learning cycle; the results are mostly significant from a qualitative point of view, showing an improvement in CT related to everyday life and soft skills.

In conclusion, unplugged activities are often related to a puzzle-based approach, acquiring its advantages in terms of first access to programming (reducing learning anxiety and improving critical thinking skills); combining them either with other plugged activities (including robotics) or with other instructional strategies can lead to a more complete CT acquisition. Two papers ([4] and [11]), both with^{3rd} grade students, examine the effects of on-screen programming. Both take a constructionist perspective, using *ScratchJr* and *Scratch* respectively [4] specifically highlights the "reuse and remix" of CT practice, adopting a Review-Copy-Create strategy that also improves all CT concepts (except loops) and ensures good retention. [11] shows the effectiveness of extended scaffolding for young girls in promoting CT skills and self-efficacy.

They both present a long-term program (8 weeks and 14 weeks, respectively), demonstrating the efficacy of the constructionist approach in this type of intervention.

Finally, we include the paper [31], even though it addresses all primary school grades, because it has an advantage over the first ones in that it uses a GBL approach and the BCTt as an assessment tool. Not surprisingly, GBL combined with a collaborative approach improves CT in early grades, but also in low-percentile students and in students with special needs.

5.3. Grades 4th and 5 th

The vast majority of the papers (18 to 31) deal with grades 4th and 5 th Some of them, according to the specificity of the school system of each country, also address grade 6, although we have decided to ignore them, due to the aims of this review.

We can identify 4 main groups of strategies on which the research focused: collaborative strategies; problem-solving oriented strategies; creative strategies; environment/tool based strategies.

5.3.1 Collaborative strategies

According to the research, in order to improve CT, collaborative strategies aimed at 4th and 5th grade students should be well structured and, above all, students should assume fixed roles within their group. This is mainly a result of [24], which specifically addresses this issue as an independent variable, but it is not denied by other studies that also include collaborative strategies, although they do not specifically focus on them: [15] and [28] partially use pair programming within constructivist-based long programs; [18] uses a similar structure with groups of three students (two programmers and one evaluator). All these studies show a significant improvement in CT.

As a partial counter-evidence, we can refer to [9], which mainly focuses on a *Scratch-based* intervention: students were divided into groups of two children, one of whom dealt with character design and the other with script design. The study shows a small improvement in CT with CTLS, with only the cooperativity dimension of the scale showing a medium effect size. It must be said that the duration of the intervention was not too long (5 weeks), given its "creative" approach.

5.3.2 Problem solving oriented strategies

We can identify two groups of problem-solving oriented strategies: those that are primarily puzzle-based (or however they move from a puzzle-based approach), and those that consider a more structured approach to problems.

We can include unplugged activities in the first group. In fact, as stated by [3], within a puzzle-based approach, unplugged and plugged activities have the same effect on CT (through CTt, and thus on CT concepts and, partially, on CT practices). Such unplugged activities are mostly effective as a first access to CT, namely with 4th grade students, as specifically stated by [10]; [25] detects an improvement in CT in terms of self-efficacy (by CTLS), ironically in all dimensions of the scale, except the problem solving one, somehow confirming the fact that there are different approaches to problem solving that don't fit the puzzle-based one. Such results support the design of long-term programs that start with a puzzle-based, possibly unplugged, approach that facilitates access to CT, and then lead to a project-based approach that is more meaningful for students; [19] adopts this kind of structure and shows a significant improvement in CT (by Tran's CT test). [10] reiterates the need to identify the point at which the unplugged activities lose their effectiveness, in this case stating that until that point we can consider CT as a purely cognitive skill that can be disconnected from computer programming.

On the other hand, two papers, [13] and [17], identify two deeper approaches to problem solving. In [13], the IGGIA model, a strong conceptualization of CT as a problem-solving process, is studied, consisting of five stages, each of which is associated with a teaching activity and a learning activity, as well as one or more CT sub-processes (abstraction, decomposition, data analysis...). The results of the study, based on a combination of CTt and CTLS, show a significant improvement in CT (in this case, the problem-solving dimension of CTLS was the one that increased the most). A slightly different model is offered by [17]; it also consists of five stages related to CT subprocesses, but it also includes activities aimed at fostering creativity and divergent thinking within a structured four-component instructional design. Such a model is called "creative problem solving" and it somehow combines the two main approaches to CT. The research shows that such a model, applied to an 11-week educational robotics program, significantly improves both CT (from Bebras) and creativity.

Papers [13] and [17] show that appropriate instructional design can effectively improve CT as a problem-solving process, without resorting to a mere puzzle-based approach, which can be useful as an entry point to CT.

5.3.3 Creative strategies

We can identify 3 strategies that clearly refer to a creative approach.

The first one is the metaphor-based approach called MECOPROG [21], which combines the use of specific metaphors, useful to make some computational concepts easy to understand, with the use of *Scratch*. The results show a significant improvement in CT (through CTt), especially for 4th grade students, and it also improves specific CS content.

The second one is the Design-Based Learning (DBL), which structures the typical project orientation of the creative approach. Paper [14] shows its effectiveness in a short-term intervention more in the quality of the final project than in the CT, measured by CTt.

The third, Interest-Driven Creator, is perhaps the most interesting as it mixes a problem-solving approach with a project-based approach. It includes, in each 1-hour session within a long-term program (1 semester), an interest-arousing phase, a problem-solving phase, and an idea-expressing phase. [27] shows its effectiveness in developing CT using parts of CTt (directions and sequences, loops and conditional statements) and parts of CTLS (creativity and problem solving) and finds that they all improve significantly, except for creativity, which increases but not significantly. The improvement in the problem-solving dimension of the CTLS is perhaps the most interesting, as several problem-solving oriented approaches didn't significantly improve this dimension of the scale. [27] suggest that this may be due to the way problems are approached in this particular strategy. However, it must be said that the study did not have a control group, and with such a long duration, some of these improvements are to be expected. More research is needed.

5.3.4 Environment/tool based strategies

Some papers actually test specific tools or environments. Such research is somehow on the border of our research goals, since it doesn't focus on pedagogy *strictu senso*. However, we can identify three studies that implicitly follow a pedagogical approach and deserve to be mentioned.

For example, when testing the adaptive software Autothinking, [7] actually tests a higher quality of feedback to the students' activity. The research shows an outperformance of the students belonging to the experimental group, but at the same time we must say that the use of this software is compared to the traditional transmissive way of teaching. Probably more research is needed, aimed at comparing an adaptive-based feedback with the behavioral one typical of a puzzle-based approach.

Paper [6] tests an approach to programming that follows a Sense-Reason-Act logic, which is compared to the traditional linear approach to programming and shows that it improves students' ability to solve complex CT tasks.

Finally, paper [29] compares *Alice* and *Scratch*, which is strongly associated with a creative learning approach. *Alice*, which is more storytelling oriented than *Scratch*, shows a significant improvement in CT (by CTLS), except again for the problem solving dimension.

The synthesis of the main results from our review of the instructional strategies used in primary education is presented in Table 3. Teachers and educators may find this useful, depending on the level of their school and their needs.

5.4. Recommendations for future research

After analyzing the main findings of the reviewed studies, we can identify several aspects that deserve to be better addressed in future research.

The first one is a theoretical issue: although all the papers refer to a theoretical framework, this aspect should be better explained in order to clarify what the study's approach to CT is. It is essential to make explicit whether CT is intended as a pure problem-solving ability that emerges in solving puzzles or tasks, or whether it is linked to creativity and self-expression in a project-based constructionist approach. The concept of problem solving itself can be very different, as shown by the heterogeneous results of the CTLS "problem solving" dimension across studies.

The second issue is somewhat related to the first and concerns the assessment tools used. A vigilant awareness of the approach underlying each tool is strongly recommended. Even if a more complex system of assessment is recommended for each individual study to better cover CT as a whole, for grades 4 and 5, the research shows a widespread use of CTt and CTLS for pre- and post-tests in experimental or quasi-experimental designs; using these tools, separately or together, can help to compare results and make more effective second-order syntheses (such as meta-analyses). For grades K to 3, the research on assessment tools appears to be less mature, although some tools are emerging and therefore merit further investigation.

The third issue concerns the need for better refinement of the usual conditions of the control groups. First of all, it has to be said that more than 1 out of 3 studies we analyzed did not use a control group, with a higher proportion in the "creative" and "mature" categories, which are the most interesting according to our objectives. But secondly, in several cases the ordinary conditions correspond to a traditional CS course; the more the intervention claims to be mature from a pedagogical point of view, the more it should be compared to an archetypal approach, namely the puzzle-based one.

Table 8. Synthesis of the effectiveness of the Instructional Strategies in developing CT in K-5 education, according to this review

Grade	Overall pedagogical approach	Specific Instructional Strategy	Results in terms of CT development
K	Educational Robotics	Scenario-based	An individualistic, kinesthetic, spatially-oriented, and manipulative-based scaffolding type is more effective for boys; a collaborative, writing scaffolding type is more effective for girls
1^{st}-3^{rd}	Educational Robotics	CAL-KIBO	Improves CT representation, modularity and algorithm dimensions
1^{st}-3^{rd}	Educational Robotics	Robot-Student interaction with investigation	Improves CT abstraction, logical thinking and analysis of problem solutions dimensions
2^{nd}-3^{rd}	Fully unplugged activities	Puzzle-based	Effective for first access to CS, reduction of learning anxiety
2^{nd}-3^{rd}	Combined unplugged + plugged in activities		Complete CT acquisition
3^{rd}	On screen programming	Review-Copy-Create	Improves CT Practice “Reusing and Remixing” and grants ~~a~~ good retention
1^{st}-4^{th}	On screen GBL	GBL + Collaborative Learning	Effective for early grades, for low percentile students and for students with special needs
4^{th}-5^{th}		Collaborative Learning	Effectiveness of fixed roles
4^{th}-5^{th}	Fully unplugged activities	Puzzle-based	Effective for first access to CS, improvement of self-efficacy
4^{th}-5^{th}	On screen programming / Educational Robotics	Advanced Problem-Solving strategies (IGGIA, Creative Problem-Solving)	Significant improvement of CT, as a Problem-Solving process
4^{th}-5^{t}	On screen programming	MECOPROG (metaphor based strategy)	Improvement of CT together with specific CS content
4^{th}-5^{th}	On screen programming + making (Arduino, 3D printing)	Design-Based Learning	Effectiveness in a short-duration intervention more in the quality of the final project rather than in the CT improvement
5^{th}	On screen programming	Interest-Driven Creator	Significant improvement of several CTt and CTLS dimensions; creativity does not improve significantly, while CTLS Problem-Solving dimension does.

5.5. Limitations

This study has several limitations. First, it was limited by the "specific search term" used. Second, it was limited to the period of publication dates of the reviewed papers (2006-2021). Third, it was also limited to the ten electronic databases searched. The number of papers included could have been higher if different and/or more digital libraries/databases had been searched.

We also point out that potential biases may arise from differences in original language and regional focus, as the papers reviewed describe activities that took place in different countries (e.g., USA, China, Turkey, Spain...). In addition, scholars from other countries may not be accustomed to publishing their research in indexed databases; therefore, research on CT in primary schools in these countries may be underrepresented. However, the findings from this review provide a useful synopsis of research on the most effective instructional strategies that have been used to develop CT as a thinking process, not just CT concepts.

Finally, this review adopts a categorization based on the theoretical assumption that a good instructional strategy in developing CT in primary schools should import into the curricula the benefits of creative constructionist approaches that are more often used in non-curricular out-of-school contexts and should also go beyond the mere puzzle-based behavioral approach. Of course, different theoretical premises can lead to different results.

6. Conclusions

The aim of this paper is to provide a snapshot of the current work and research around the way that CT has grown in elementary education students.

The first conclusion we can draw is that there is a strong difference between the activities aimed at developing CT, whether they address kindergarten and primary grades 1-3 (almost always related to Educational Robotics or an unplugged approach), and primary grades 4-5, which seem to adapt the pedagogical approach of the upper grades, both for on-screen programming and for Educational Robotics activities. This is also evidenced by the fact that assessment tools originally developed for high school and middle school (such as CTt and CTLS) have been adapted and often used in research on elementary grades 4-5, while tools specific to lower grades (such as TechCheck or BCTt) are still in development.

The second conclusion is that a task-based approach is still widespread across all grade levels. In the lower grades (K-2), this is mainly related to educational robotics scenarios or unplugged puzzles. Research shows that the efficacy of this approach is limited to the first access to CS activities, namely improving self-efficacy and reducing learning anxiety. Nevertheless, we can see that approaches to CT as a problem-solving process are diverse. It should be noted that the more mature they are (e.g., IGGIA or Creative Problem Solving models), the more the "problem solving" dimension of CTLS improves, while this is not the case with either puzzle-based or purely creative activities.

Another aspect of the research is peer collaboration, which seems to be effective for younger children. For grades 4-5, the design of fixed roles is recommended (e.g. in pair programming). In addition, in several programs of longer duration, there is a transition to a project-based approach and, in line with this framework, the learner's activity also moves from collaborative to individual, although the projects can be evaluated by peers according to the principles of constructionism.

In our analysis, we have chosen to pay close attention to the overcoming of both archetypal approaches, trying to identify those strategies that both try to foster much more meaningful learning (typically including some elements coming from constructionism) while achieving curricular learning objectives, labeling them as "mature". Among these, research has identified two that appear to be promising: Interest-Driven Creator (IDC) and Design-Based Learning (DBL). The former provides a structure that helps embed problem solving within a constructivist framework, with a significant improvement in many CTt and CTLS dimensions supporting the efficacy of this method (we again highlight the improvement in the CTLS problem solving dimension); the latter, adopted in a "making" intervention, also seems interesting for our purposes, as problem solving skills and creativity are fostered together in the design of a final project.

Nevertheless, both strategies need further research. On the one hand, IDC was tested in a long-term program without a control group: therefore, more refined studies are needed to show the real specificity of the effectiveness of this approach. On the other hand, DBL was tested in a very short-term intervention (6 days), and the most interesting findings come from the analysis of the quality of the final project rather than from the gap between a pre-test and a post-test. Further research is therefore recommended for both.

Authors contributions

The research was carried out by both authors, particularly in the process of categorization of the papers; Panagiotis Kakavas wrote paragraphs 1.3, 2 and 5.5; Francesco C. Ugolini wrote the remaining parts of the paper.

References

Acevedo-Borrega, J., Valverde-Berrocoso, J., & Garrido-Arroyo, M.d.C. (2022). Computational Thinking and Educational Technology: A Scoping Review of the Literature. *Education Sciences, 12*(39), 1-16.

Anderson, N.D. (2016). A call for Computational Thinking in undergraduate psychology. *Psychology Learning & Teaching*, 15, 226-234.

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction, 29*, 1-62.

Ballard, E. D., & Haroldson, R. (2021).Analysis of Computational Thinking in Children's Literature for K-6 Students: Literature as a Non-Programming Unplugged Resource. *Journal of Educational Computing Research*, 1-30.

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies, 27*, 2059-2082.

Battal, A., Adanir, G. A., & Gülbahar, Y. (2021). Computer Science Unplugged: A Systematic Literature Review. *Journal of Educational Technology Systems*, 1-24.

Bers, M.U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice, 12*(2).

Bers, M. U. (2019). Coding as Another Language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education, 6*(4), 499-528.

Bonaiuti, G. (2014). *Le strategie didattiche*. Roma: Carocci.

Bonaiuti, G., Calvani, A. & Ranieri, M. (2016). *Fondamenti di Didattica. Teoria e Prassi dei Dispositivi Formativi*. Roma: Carocci.

Brennan, K., & Resnick, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA'12) (pp. 1-25).

Brennan, K. (2013). *Best of both worlds: issues of structure and agency in computational creation, in and out of school*. Ph.D Dissertation. Cambridge, MA: MIT Media Lab.

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing Noted Reviews,* 1(2), 67-69.

Campe, S. & Denner, J. (2020). Peer Collaboration and Pair Programming. In S. Grover (Ed.). *Computer Science in K-12. An A to Z handbook on teaching programming* (pp. 171-179). Palo Alto, CA: Edfinity.

Chen, P., Yang, D., Metwally, A.H.S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education, 10*, 1-25.

Clark, D. B., & Sengupta, P. (2019). Reconceptualizing games for integrating computational thinking and science as practice: Collaborative agent-based disciplinarily-integrated games. *Interactive Learning Environments, 28*(3), 328-346.

CSTA-ISTE (2011). *Operational Definition of Computational Thinking for K-12 Education*. cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf.

Curasma, R. P., & Curasma, H. P. (2020). Computational thinking in school education in South America: Systematic review of the literature. In *Proceedings of the XXVII International Conference on Electronics, Electrical Engineering and Computing, INTERCON*, (pp. 1-4). Lima, Peru.

Cutumisu, M., Adams, C., & Lu, C. A. (2019). Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *Journal of Science Education and Technology, 28*, 651-676.

Da Cruz Alves, N., Gresse Von Wangenheim, Ch., & Hauck, J. C. R. (2019). Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education, 18*(1), 17-39.

De Jong, I., & Jeuring, J. (2020). Computational Thinking Interventions in Higher Education: A Scoping Literature Review of Interventions Used to Teach Computational Thinking. In *Proceedings of the 20th Koli Calling International Conference on Computing Education Research, Koli Calling '20*, (pp. 1-10). Koli, Finland.

Denning, P.J., & Tedre, M. (2019). *Computational Thinking*. Cambridge, MA: The MIT Press.

Espinal, Al., Vieira, C., & Magana, Al. J. (2024). Professional Development in Computational Thinking: A Systematic Literature Review. *ACM Transactions on Computing Education 24*(2), 1-24.

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2020). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 1-17.

Fauzi, A. L., Kusumah, Y. S., Nurlaelah, El., & Juandi, D. (2024). Computational Thinking in Mathematics Education: A Systematic Literature Review on its Implementation and Impact on Students' Learning. *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran, 10*(2), 640-653.

Franco Silva, Dembogurski, B. J., & Semaan, G. S. (2022). A literature review of computational thinking in early ages. *International Journal of Early Years Education, 31*(3), 753–772.

Giannakoulas, A., & Xinogalos, St. (2024). Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review. *Journal of Computers in Education, 11*, 1283–1325.

Grover, S., (2015). 'Systems of assessments' for deeper learning of computational thinking in K-12. In *Proceedings of the 2015 Annual Meeting of the American Educational Research Association (AERA' 15)*, (pp. 1-9).

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: geçerlik ve güvenirlik çalıs¸ması. *Türk Bilgisayar ve Matematik Eğitimi Dergisi, 10*(1), 1-29.

Hamilton, M., Clarke-Midura, J., Shumway, J. F., & Lee V. R. (2019). An Emerging Technology Report on Computational Toys in Early Childhood. *Technology, Knowledge and Learning, 25*, 213-224.

Haseski, H., & İlic, İ. U. (2019). An Investigation of the Data Collection Instruments Developed to Measure Computational Thinking. *Informatics in Education, 18*(2), 297-319.

Huang, W., & Looi, C.-K. (2020). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education, 31*(1), 83-111.

Isharyadi, R., & Juandi, D. (2023). A Systematics Literature Review of Computational Thinking in Mathematics Education: Benefits and Challenges. *Formatif: Jurnal Ilmiah Pendidikan MIPA, 13*(1), 69-80.

Kakavas, P. (2019). Computational thinking and Creativity in K-6 Education. *Formamente, 14*(2), 93-108.

Kakavas, P., & Ugolini, F. C. (2019). Computational thinking in primary education: A systematic literature review. *Research on Education and Media, 11*(2), 64-94.

Korkmaz, Ö., Çakır, R., & Özden, M.Y. (2015). Computational Thinking Levels Scale (CTLS) adaptation for secondary school level. *Gazi Journal of Education Sciences*, *1*(2), 143-162.

Lai, X., & Wong, G.-K. (2021). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. *British Journal of Educational Technology*, 1-21.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology, 25*(6), 860-876.

Lin, Q., Yin, Y., Tang, X., Hadad, R., & Zhai, X. (2020). Assessing learning in technology-rich maker activities: A systematic review of empirical research. *Computers & Education, 157*, 1-62.

Lo, C. K. (2020). Systematic reviews on flipped learning in various education contexts. In Zawacki-Richter, O., Kerres, M., Bedenlier, S., Bond, M., & Buntins, K. (Eds.), *Systematic reviews in educational research: Methodology, perspectives and application*, (pp. 129-143). Springer VS, Wiesbaden.

Marzano, R.J., Pickering, D., & Pollock J.E. (2001). *Classroom Instruction that Works*. Alexandria, VA: ASCD.

Ogegbo, A. A., & Ramnarain, U. (2021). A systematic review of computational thinking in science classrooms. *Studies in Science Education, 58*(2), 203-230.

Ortuño Meseguer, G., & Luis Serrano, J. (2024). Implementation and training of primary education teachers in computational thinking: a systematic review. *Revista Iberoamericana de Educación a Distancia, 27*(1), 1-28.

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Gianville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P. & Moher, D. (2021a). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *PLoS Med 18(3)*.

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M. & Moher, D. (2021b). Updating guidance for reporting systematic reviews: Development of the PRISMA 2020 statement. *Journal of Clinical Epidemiology, 134*, 103-112.

Palts, T., & Pedaste, M. (2020). A Model for Developing Computational Thinking Skills. *Informatics in Education, 19*(1), 113-128.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1993). *The Children's Machine. Rethinking School in the Age of the Computer*. New York: Basic Books.

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review, *Computers & Education, 128*, 365-376.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.

Resnick M. (2017). *Lifelong Kindergarten. Cultivating Creativity through Projects, Passion, Peers, and Play*. Cambridge, CA: MIT Press.

Román-González, M. (2015). *Computational Thinking Test: Design Guidelines and Content Validation*. In *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)* (pp. 2436-2444). Barcelona: IATED.

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678-691.

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. In S.-C. Kong, & H. Abelson (Eds.), *Computational Thinking Education* (pp. 79-98). Singapore: Springer.

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open, 4*, 1-12.

Sun, L., Guo, Zh., & Hu, L. (2021). Educational games promote the development of students' computational thinking: A meta-analytic review. *Interactive Learning Environments*, 1-16.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 1-22.

Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to Develop Computational Thinking: A Systematic Review of Empirical Studies. *Informatics in Education, 19*(4), 701-719.

Tikva, C., & Tambouris, E. (2021). Mapping Computational Thinking through Programming in K-12 Education: A Conceptual Model based on a Systematic Literature Review. *Computers & Education, 162*, 1-38.

Triantafyllou, S. A., Sapounidis, T., Farhaoui, Y. (2024). Gamification and Computational Thinking in Education: A systematic literature review. *Salud, Ciencia y Tecnología - Serie de Conferencias, 3*, 1-25.

Vinu Varghese, V. V., & Renumol, V. G. (2021). Assessment Methods and Interventions to Develop Computational Thinking – A Literature Review. In *Proceedings of the 2nd International Conference on Innovative Trends in Information Technology, ICITIIT, (pp. 1-7)*. Kottayam, Kerala, India.

Wing, J. (2006). Computational Thinking. *Communication to the ACM*, 49(3), 33-35.

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society*, 366, 3717-3725.

Wing, J. (2010). *Computational Thinking: What and Why?* www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Yu, J., & Roque, R. (2019). A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction, 21*, 17-36.

M. Zapata-Caceres, E. Martın-Barroso, & M. Roman-Gonzalez (2020). Computational thinking test for beginners: Design and content validation. In *Proc. IEEE Glob. Eng. Educ. Conf., Porto, Portugal*, 2020 (pp. 1905-1914).

Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational Robots Improve K-12 Students' Computational Thinking and STEM Attitudes: Systematic Review. *Journal of Educational Computing Research*, 1-32.

Zeng, Y., Yang, W., & Bautista, Al. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review, 39*, 1-17.

Zhang, L., & Nouri, J. (2019). A systematic review of learning by computational thinking through Scratch in K-9, *Computers & Education, 141*, 1-36.

Annex A: Reviewed Papers

[1] Angeli, Ch. & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior, 105*, 105954, Science Direct.

[2] Angeli, Ch., Xerou, Ef., & Nicolau, M. (2019). Investigating K-2 Students' Computational Thinking Skills during a Problem-Solving Activity about the Water Cycle Using Educational Robotics. In *Proceedings of the 16th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA*, (pp. 1-7). Cagliari, Italy, ACM.

[3] Brackmann, Ch.P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. In *Proceedings of the 12th Workshop in Primary and Secondary Computing Education, WiPSCE '17*, (pp. 65-72). Nijmegen, Netherlands, ACM.

[4] Chou, P.-N. (2020). Using ScratchJr to Foster Young Children's Computational Thinking Competence: A Case Study in a Third-Grade Computer Class. *Journal of Educational Computing Research, 58*(3), 570–595, ERIC.

[5] Del Olmo-Muñoz, J., Cózar-Gutierrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education, 150*, 103832, Science Direct.

[6] Fanchamps, N., Slangen, L., Specht, M., & Hennissen, P. (2021). The Impact of SRA-Programming on Computational Thinking in a Visual Oriented Programming Environment. *Education and Information Technologies, 26*, 6479–6498, Springer.

[7] Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior, 114*, 106575, Science Direct.

[8] Hsu, T.-C., & Liang, Y.-S. (2021). Simultaneously Improving Computational Thinking and Foreign Language Learning: Interdisciplinary Media With Plugged and Unplugged Approaches. *Journal of Educational Computing Research, 59*(6), 1184–1207, ERIC.

[9] Jiang, B., & Li, Zh. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of Computers in Education, 8*, 505–525, Springer.

[10] Jiang, Sh., & Wong, G. (2021). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. *Journal of Computer Assisted Learning, 38*, 1-16, Wiley.

[11] Jin, Y., Sun, J., Ma, H., & Wang, X. (2021). The impact of different types of scaffolding in project-based learning on girls' computational thinking skills and self-efficacy. In *Proceedings of the 10th International Conference of Educational Innovation through Technology, EIT*T, (pp. 362-366). Chongqing, China, IEEE.

[12] Lin, S.-Y., Chien, S.-Y., Hsiao, C.-L., Hsia, C.-H., & Chao, K.-M. (2020). Enhancing Computational Thinking Capability of Preschool Children by Game-based Smart Toys. *Electronic Commerce Research and Applications, 44*, 101011, Science Direct.

[13] Ma, H., Zhao, M., Wang, H., Wan, X, Cavanaugh, T. W., & Liu, J. (2021). Promoting pupils' computational thinking skills and self-efficacy: A problem-solving instructional approach. *Educational Technology Research and Development, 69*, 1-18, Springer.

[14] Matere, I., Weng, C., Astatke, M., Hsia, C.-H., & Fan, C.-G. (2021). Effect of design-based learning on elementary students computational thinking skills in visual programming maker course. *Interactive Learning Environments*, 1-14, Taylor & Francis.

[15] Mouza, C., Pan, Y.-C., Yang, H., & Pollock, L. (2020). A Multiyear Investigation of Student Computational Thinking Concepts, Practices, and Perspectives in an After-School Computing Program. *Journal of Educational Computing Research, 58*(5), 1029–1056, ERIC.

[16] Muñoz-Repiso, A.G.-V., & Caballero-González, Y.-A. (2019). Robotics to Develop Computational Thinking in Early Childhood Education. *Media Education Research Journal, 27*(59), 63-72, ERIC.

[17] Noh, J., & Lee, J. (2019). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development, 68*, 1-22, Springer.

[18] Ntourou, V., Kalogiannakis, M., & Psycharis, S. (2021). A Study of the Impact of Arduino and Visual Programming In Self-Efficacy, Motivation, Computational Thinking and 5th Grade Students' Perceptions on Electricity. *Eurasia Journal of Mathematics, Science and Technology Education, 17*(5), 1-11, ERIC.

[19] Özcan, M.Ş., Çetinkaya, E., Goksun, T., & Kisbu-Sakarya, Y. (2021). Does learning to code influence cognitive skills of elementary school children? Findings from a randomized experiment. *British Journal of Educational Psychology, 91*(4), 1434-1455, Wiley.

[20] Özmutlu, M., Atay, D., & Erdoğan, B. (2021). Collaboration and engagement based coding training to enhance children's computational thinking self-efficacy. *Thinking Skills and Creativity, 40*, 100833, Science Direct.

[21] Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior, 105*, 105849, Science Direct.

[22] Qu, J.R., & Fok, P.K. (2021). Cultivating students' computational thinking through student–robot interactions in robotics education. *International Journal of Technology and Design Education, 32*, 1-20, Springer.

[23] Relkin, E., De Ruiter, L. E., & Bers, M. U. (2021). Learning to Code and the Acquisition of Computational Thinking by Young Children. *Computers & Education, 169*, 104222, Science Direct.

[24] Taylor, K., & Baek, Y. (2019). Grouping Matters in Computational Robotic Activities. *Computers in Human Behavior, 93*, 99-105, Science Direct.

[25] Tonbuloğlu1, T., & Tonbuloğlu, I. (2019). The Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students. *Informatics in Education, 18*(2), 403–426, ERIC.

[26] Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research, 57*(1), 3–31, ERIC.

[27] Wang, Y., Zhang, Y., Mao, A., Wang, J., & Li, N. (2020). The Research of Programming Teaching in Primary School on the Cultivation of Computational Thinking. In *Proceedings of the 9th International Conference of Educational Innovation through Technology, EITT*, (pp. 250-255). Porto, Portugal, IEEE.

[28] Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., & Kinshuk (2021). The Effectiveness of Partial Pair Programming on Elementary School Students' Computational Thinking Skills and Self-Efficacy. *Computers & Education, 160*, 104023, Science Direct.

[29] Yildiz-Durak, H. (2018). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning*, 1-17, Springer.

[30] Yılmaz Ince, E., & Koc, M. (2020). The consequences of robotics programming education on computational thinking skills: An intervention of the Young Engineer's Workshop (YEW). *Computer Applications in Engineering Education, 29*(1), 191-208, Wiley.

[31] Zapata-Caceres, M., Martín-Barroso, E. & Román-González, M. (2021). Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study. *IEEE Transactions on Learning Technologies, 14*(5), 576-589, IEEE.