

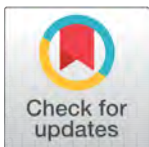
# Investigating the Computational Thinking Ability of Young School Students Across Grade Levels in Two Different Types of Romanian Educational Institutions

Zoltan Katai , Erika Osztian  and Beata Lorincz 

Department of Mathematics and Informatics, Sapientia Hungarian University of Transylvania, Romania

## ABSTRACT

Over the last decade, continuous efforts have been made to bring computational thinking (CT) closer to K-12 education. These focused endeavors implicitly suggest that the current curricula do not sufficiently contribute to the development of learners' CT. On the other hand, since CT is a combined skill with cross-disciplinary implications, one might conclude that even without an explicit focus on CS education, students' CT might develop latently as they advance with the current curriculum. We have proposed to test whether differences exist in how 3rd-, 5th-, 7th- and 9th-grade learners from two Romanian educational institutions (girls vs. boys from Art vs. Theoretical school; 214 subjects with no prior experience in CT) relate to learning tasks that require a certain level of CT. The testing tool was inspired by the AlgoRythmics dance choreography illustration of the linear search algorithm and has the potential to reveal different levels of abstracting. Findings emphasize the need for a purposeful and coordinated CS infusion into K-9 education in order to accelerate students' CT development.



**Received** 2020-08-13  
**Revised** 2020-09-24  
**Accepted** 2021-02-16  
**Published** 2021-07-15

**Corresponding Author**  
Zoltan Katai,  
[katai\\_zoltan@ms.sapientia.ro](mailto:katai_zoltan@ms.sapientia.ro)

Sapientia University, Romania,  
Târgu-Mureş - Corunca, 1C,  
540485, Op. 9., Cp. 4.

**DOI** <https://doi.org/10.7821/naer.2021.7.640>

**Pages:** 214-233

Distributed under  
Creative Commons CC BY 4.0

**Copyright:** © The Author(s)

**Keywords** COMPUTER-ASSISTED INSTRUCTION, ALGORITHMS, CURRICULUM, EDUCATIONAL TESTING, GENDER ROLES

## 1 INTRODUCTION

During the last decade, sustained efforts have been made to bring computational thinking (CT) closer to K-12 education and even to K-9 education. It is worth mentioning in this regard two complementary implementational approaches to the “CT for all” initiative: (1) introducing new computing courses; and (2) infusing CT into the existing courses (Brown, Sentance, Crick, & Humphreys, 2014). For example, Computer Science (CS) has been mandatory for pupils in the UK since 2014 from the age of five (Brown et al., 2014; Román-González, Pérez-González, Moreno-León, & Robles, 2018) and a growing recognition of the importance of CS education (CSE) becomes visible in other countries too (European Schoolnet, 2015; Grover & Pea, 2018). In addition, Mannila et al. (2014) report on a survey distributed to K-9 teachers which aims at revealing the extent to which the different aspects

## OPEN ACCESS

of CT already form part of the current curricula both in various European countries and in the U.S.

This focused attempt implicitly suggest that current curricula do not adequately contribute to the development of learners' CT. Several studies emphasize that computing students lack a variety of skills that programming would require (Ahadi, Lister, Lal, Leinonen, & Hellas, 2017; Evans & Simkin, 1989; Simon, Chen, Lewandowski, McCartney, & Sanders, 2006). On the other hand, since CT is a combined skill with cross-disciplinary implications (Feaster, Ali, Zhai, & Hallstrom, 2014), it seems plausible that even without an explicit focus on CS education learners' CT might develop latently as they advance with the current curriculum. For example, Lewandowski and his colleagues report on a series of computing projects that sought to identify the commonsense knowledge beginners bring to CS study (Lewandowski et al., 2010). While most previous research focused on assessing the CT level of certain age groups, this study tried to answer the following key question: Does any detectable incidental progress take place in students' CT during their K-9 education? Instead of using one of the general CT tests proposed by previous works in this field (Román-González et al., 2018; Shute, Sun, & Asbell-Clarke, 2017), we chose to follow the same approach that the authors of the aforementioned computing project had applied. More precisely, we focused on checking possible differences concerning the way in which 3rd-, 5th-, 7th- and 9th-grade learners (without any explicit prior experience in CT) relate to a specific learning task that implies a certain level of CT.

The rest of the paper is organized as follows. The literature review included in Section 2 presents the context of the study and identifies the knowledge gap that we addressed by investigating (and comparing) the developmental trajectory of K-9 students' CT skills in a STEM (Theoretical school) and a non-STEM (Art school) institution. Section 3 defines CT from the perspective of current research and introduces our *ad hoc* testing tool adapted to it. Sections 4 and 5 describe the research questions along with the experiment performed. In Sections 6 and 7, we show our findings and interpret them from the perspectives of grade level, educational program and gender. Finally, Section 8 provides a brief summary of the paper and highlights some implications of our findings.

## 2 REVIEW OF RELATED LITERATURE

In recent years, there has been an increasing interest in CT education at each school grade level with several approaches being analyzed from different perspectives. By way of example, a recent review by Zhang and Nouri (2019) systematically examined the CT skills that can be obtained through Scratch programming at different age groups (5-9, 9-12, 12-15). Other studies emphasize the effectiveness of unplugged methods to introduce CT in the early years of primary education. Del Olmo-Muñoz, Cózar-Gutiérrez and González-Calero (2020) report on the convenience of combining unplugged and plugged-in activities for 2nd-grade students. Some research works investigated the benefits of using metaphors (figures of speech that describe objects or actions which, despite not being literally true, help explain abstract ideas) when teaching CT. In turn, Pérez-Marín, Hijón-Neira, Bacelo, and

Pizarro (2020) concluded that mixing the use of metaphors with a block-based programming environment could be an effective method to promote CT knowledge acquisition in primary education ([4th-, 5th- and 6th-grade] 9-to-12-year-old students).

Because of the gender stereotypes associated with CT and the gender gap existing in the computing field (Grover & Pea, 2013), research results have been frequently analyzed from a gender perspective too. For example, del Olmo-Muñoz et al. (2020) reached the conclusion that, despite the possible initial differences, all children can develop CT-related skills regardless of their gender if a carefully designed instruction is put into play. These authors also observed how girls showed lower motivation than boys when plugged-in methods were applied; female learners' interest slightly increased with unplugged approaches, though.

Most studies on CT include assessing methods and tools as well. In a recent systematic review, Tang, Yin, Lin, Hadad, and Zhai (2020) examined several research works on CT assessment implemented across all educational levels. Their analysis revealed that a large number of studies had focused on cultivating CT in elementary and middle school grade levels. Some research gaps were also identified. Amongst other things, these authors observed that: (1) CT assessment is lacking in informal contexts; and also that: (2) compared to assessments infusing CT with STEM subjects, the integration of CT assessment into non-STEM contexts has not been sufficiently developed either.

Another particularity of prior research on CT is that it mostly concentrated on certain age groups. However, some researchers suggest that findings should be interpreted as inter-related pieces of a whole. For instance, Grover and Pea (2013) refer to the path of developing CT elements in children; and Tang et al. (2020) underline the importance of finding resources to ensure a complete developmental trajectory of students' CT skills.

In addition, a vast majority of the studies focused on methods aimed to explicitly improve students' CT skills. The extent to which some CT skills of students latently develop as they advance with their current CT-free curriculum—it includes no explicit attempts to improve students' CT—is a poorly researched area.

In the light of the research gap outlined above, we suggest testing and comparing the CT skills of 3rd-, 5th-, 7th- and 9th-grade boys and girls (with no explicit prior experience in CT) from two educational programs (STEM-oriented vs. non-STEM-oriented). The following sections will serve to present the testing tool that we designed for this purpose.

### 3 CREATING A COMPUTATIONAL MODEL TO INVESTIGATE CT ABILITY

Teaching curricula are nowadays confronted with the concept of CT and teachers struggle with questions such as “What is exactly meant by CT?” and “How is CT measured or taught?” Tang et al. (2020). CT has become more popular since J. Wing (2006) promoted its introduction in K-12 education. According to her definition, CT is a “universally applicable attitude and skill set that everyone, and not just computer scientists, would be eager to learn and use” (p. 33). Aho (2012) in turn redefines the concept emphasizing that “models of computation are the heart of computation and computational thinking” (p. 834). Finally,

in an appraisal of Aho's revised definition, [Denning \(2017\)](#) highlights the importance of computational models and algorithms, describing the steps of the latter as the design of the "way to control any machine that implements the model" (p. 35).

Regarding the nature of CT, it can be described as a skill rather than a knowledge set of facts and information ([Denning, 2017](#)). Skills are considered tacit knowledge, abilities that can be enhanced by regular practice. Accordingly, [Zhang and Nouri \(2019\)](#) positioned CT at the intersection of computing and problem-solving. When measuring CT instead of evaluating students' knowledge, attention should be directed to their competences. Recent research has focused on detecting CT through competence-based assessment ([Feaster et al., 2014](#); [Shute et al., 2017](#)). Some of these studies involved students who had not begun to study programming yet. [Román-González et al. \(2018\)](#) stressed that CT ability can be demonstrated not only by means of computer programming but also within unplugged contexts ([Brackmann et al., 2017](#); [Rodríguez, Kennicutt, Rader, & Camp, 2017](#)). Along these lines, an efficient method for the investigation of CT would be to engage students with learning environments where CT skills can prevail.

Since algorithms are central to CT ([Zhang & Nouri, 2019](#)), the learning environment created by us confronts students with "masked computer algorithms." We chose dance as a metaphor. Learners are asked to identify with a flamenco dancer looking for his match in a dancer sequence. We took into account Denning's ([2017](#)) reminder that algorithms must control some computational model and that step sequences requiring human judgment should not be considered algorithms in the context of CT. In addition, [Katai \(2014\)](#) emphasizes that algorithm visualization systems should support the human user in assimilating the algorithm-processing role of blind computers. He argues that visualizations showing information with extra meanings for learners may obstruct them in following strict computer algorithms and suggests that wisely applied hiding might result in more effective algorithm visualization due to its higher epistemic fidelity. Accordingly, the dancer sequence members wear on their back the numbers they represent and the boy has to ask different girls for a dance to realize who his match is (the girl dancing the same choreography and wearing the same number).

The environment was inspired by the AlgoRhythmic videos, and more precisely, by the dance choreography illustration of the linear search algorithm. A peculiarity of this video lies in the fact that it also implements the above-suggested selective hiding (girls are wearing numbers on their back). Once again, since the boy is forced to perform the comparing operations explicitly (by inviting girls to dance), the scheme can be perceived as a computational model claimed by [Aho \(2012\)](#) and [Denning \(2009, 2017\)](#).

An attempt was made to investigate CT ability by measuring (1) to what extent students intuit the linear search algorithm before viewing the AlgoRhythmic video; and (2) how well they could assimilate the algorithm they were introduced to. This algorithm includes most of the algorithmic elements suggested in the CSTA standard with respect to the 3rd-to-5th grade interval [CSTA \(2017\)](#). Furthermore, in order to be able to identify the differences in students' CT and to develop a suitable tool for each grade, the tasks included in the learning environment involved different levels of abstraction [Román-González et al. \(2018\)](#).

While designing specific tasks, we also drew on the definition that Shute et al. (2017) gave to CT: “the conceptual foundation required to solve problems effectively and efficiently (i.e. algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts” (p. 151). These authors’ CT model comprised the following facets: decomposition; abstraction; algorithms; debugging; iteration; and generalization. The assignments incorporated in our testing instrument aim to reveal such components. We accordingly assumed that students’ correct answers to the different tasks reflect their CT skills, since these required them to decompose the algorithmic searching process into a succession of dances (representing comparisons) and to iterate through the girl sequence until the boy meets his match. As for the effectiveness and efficiency factor, some tasks are built around the concept of algorithm complexity. These assignments required quite advanced abstraction insofar as students needed to imagine and generalize the best- and worst-case behaviors of an algorithm that was presented to them, as a dance choreography, for a single specific input.

## 4 RESEARCH QUESTIONS

The main research question was: do any detectable differences exist in how 3rd-, 5th-, 7th- and 9th-grade learners (with no explicit prior experience in CT) relate to learning tasks that require a certain degree of CT? More specifically:

- What is the potential CT growth pace? (RQ 1)
- Do CT growth rate and pace depend on the characteristics of the current curriculum (Art vs. Theoretical school)? (RQ 2)
- To what extent can students of different grades assimilate a basic computer algorithm (linear search) (RQ 3)
- Does evidence of advanced CT exist at different grade levels? (RQ 4)
- Do CT growth rate and pace depend on learners’ gender? (RQ 5)

## 5 METHODS

### 5.1 Participants

The experiment took place in Romania, where K-12 education is divided into three stages: primary school (grades 0-4); gymnasium (grades 5-8); and high school (grades 9-12). In addition, the school curriculum revolves around three main lines: theoretical; vocational (e.g. arts); and technological. Regarding CS teaching, this has been a mandatory subject for many years in high school classes (9-12 grades) in the case of natural sciences, mathematics and computer science and intensive computer science programs. Starting from the 2017-18 academic year, the ICT syllabus corresponding to the gymnasium level was enlarged with CS concepts (algorithms, computing, etc.) too.

A total of 214 students (56% of them girls) from two public schools with a good reputation in their region participated in this study. Since only theoretical (T) and art (A)

specificities are available at the elementary level, the schools involved had these orientations (School-T and School-A). Eight classes of grades 3, 5, 7 and 9 were invited to take part in the experiment, with one class from both schools at each level: classes 3T, 3A (music); 5T, 5A (fine arts); 7T, 7A (fine arts); and 9T, 9A (fine arts) (see Table 1 for further details). We chose to start with 3rd-grade classes for the same reasons as [Settle, Goldberg, and Barr \(2013\)](#). This grade is considered the transition from “learning to read to reading to learn,” which led these authors to suggest a corollary between reading comprehension and CT skills. It is also our assumption that this two-year shift will reveal easily perceived advances regarding students’ thinking and skills. This partitioning likewise harmonizes with the classification provided in the CSTA standards ([CSTA, 2017](#)).

**Table 1** Participants’ demographic profile

	<b>Grade 3 9-to-10-year-olds</b>	<b>Grade 5 11-to-12-year-olds</b>	<b>Grade 7 13-to-14-year-olds</b>	<b>Grade 9 15-to-16-year-olds</b>	<b>All</b>
School T	32 (14 girls)	29 (13 girls)	29 (18 girls)	28 (13 girls)	118 (58 girls)
School A	25 (12 girls)	20 (13 girls)	25 (21 girls)	26 (17 girls)	96 (63 girls)

## 5.2 Procedure

The experiment developed during the first semester of the academic year, and the testing sessions took place in the computer laboratories of the participant schools with the exception of 3rd-grade classes. Since their curriculum does not include ICT classes and it is quite hard to keep learners busy, we suggested meeting at university labs in the hope that such an environment would help them stay more focused.

After a brief introduction, participants were requested to login to an online testing application (Socrative) that included the items listed in Table 2.

**Table 2** Questions and tasks included in the investigation procedure

<b>Metacode</b>	<b>Item</b>
Question 1	What is your gender? (male/female)
Question 2	How much do you like computer science? (1-10)
Question 3	How much do you love music and dance? (1-10)
Task 1	Students were shown the picture in Fig 1 and asked to specify the number of dances required for the boy to identify his match. It was emphasized that he does not see the numbers the girls are wearing and has to invite each girl for a dance to find out if she is wearing his number.
Video	Subjects were requested to watch the flamenco dance choreography illustrating the linear search algorithm (see Fig. 3). Since the algorithm unfolds quite slowly (because of the choreography), the video was played at double speed. They were asked to focus on the strategy applied by the boy in the video.
Question 4	To what extent did you understand the boy’s method (in your opinion)? (1-10)
Task 2.1	Students were shown a picture similar to that in Fig 1 with the difference that this one presented a girl sequence of length 10 (the girl wearing the boy’s number was the 7th or the 4th one depending on the position of the first one on the list). They were once again asked to specify the number of dances required for the boy to identify his match (it was emphasized that they had to implement the same method that the boy had applied).

*Continued on next page*

Table 2 continued

Task 2.2	Students were shown the picture in Fig 2 and asked to give two sequences of length 5 that required the minimum number (subtask 2.2.1) and the maximum number (subtask 2.2.2) of dances needed for the boy to find his match.
Task 2.3	Students were requested to generalize the algorithm for a girl sequence of length $x$ by giving the required number of dances for the best-case (subtask 2.3.1) and worst-case (subtask 2.3.2) scenarios.

The testing sessions for all groups lasted ca. 30-35 minutes. In the remaining 10-15 minutes, students had to interpret their answers (verbal probing).

### 5.3 Materials

The testing process consisted of six consecutive tasks (1, 2.1, 2.2.1, 2.2.2, 2.3.1, 2.3.2), some of which referred to pictures presenting the girl sequence from behind and the boy with his number on his chest (see Fig. 1). Using this method allowed us to make students see not only the number that the boy represents and the sequence personified by the girls but also that the girls' numbers are invisible to the boy. Other tasks required students to imagine the best- and worst-case behavior of the algorithm by providing number sequences that entail a minimum and maximum number of dances for the boy to find his match (see Fig. 2).



**Figure 1** Students were asked to indicate the number of dances required for the boy to identify his match.

Task 1 aimed to reveal the extent to which students intuit the linear search algorithm before viewing the AlgoRhythmics video. The purpose of Task 2 consisted in checking how deeply students had assimilated the algorithm they were introduced to (by watching the video; see Fig. 3). An important detail about Task 2 was that students had no possibility to indicate which end of the list their searching started at.

As mentioned above, abstraction arises as a critically important element of CT (J. M. Wing, 2008), two of its key aspects being: (1) the act of leaving unimpor-



**Figure 2** Students were asked to provide best- and worst-case sequences.



**Figure 3** The AlgoRythmics linear search video (<https://www.youtube.com/user/AlgoRythmics>)



tant aspects out of consideration; (2) the process of formulating general concepts by extracting common features from specific examples (Kramer, 2007). In addition to the need for students to extract an algorithm from a dance choreography, the subtasks included in Task 2 required different levels of abstraction.

- On the picture shown in Fig. 1 (included in Task 1) both the boy's number and the girl-sequence's length were 7. The video presents a similar scenario: the boy is looking for number 7 in a girl sequence with a length of 7. Instead, 10 corresponds to the boy's number and to the girl-sequence's length too on the picture attached to Task 2.1. During these assignments, students had to ignore such coincidences as an unimportant detail. (Verbal probing revealed that some learners were disturbed by this circumstance)
- We decided to establish neither that the sequence represented by the girls obligatorily contains the boy's number nor that the girls' numbers are different. In our view, the ability of students to abstract from the given samples (both the video and the pictures associated with Tasks 1 and 2.1 displayed different numbers including the boy's number) and, for example, to identify a sequence without the target value as the worst-case scenario constituted signs of advanced CT.
- The students' inability to indicate which end of the list their searching started at turned out to be particularly important in the case of Task 2.2. Once more, we considered it a sign of advanced CT if they addressed this "left-to-right or right-to-left dilemma" (as algorithms must operate correctly on any input data).

In Task 1, we regarded as correct answers the values which resulted from linear search, 2 corresponding to the right-to-left approach and 6 to the left-to-right one. The correct answers for Task 2.1 were 7 and 4.

Task 2.2.1 requested students to give a best-case input of length 5.

- (Variant 1a) A majority of students placed the value the boy is looking for (the key element) at one of the ends of the list. They simply assumed that the search began at the end they had chosen (probably at the left end of the list either from the boy's perspective or from theirs).
- Nevertheless, other students addressed the left-to-right or right-to-left dilemma quite ingeniously:
  - (Variant 2) situating the key element at both ends of the list or giving a constant sequence that contained only the searched value;
  - (Variant 3a) inserting the key element in the middle of the list.

Participants faced Task 2.2.2 (to give a length-5 worst-case input) only after having finished Task 2.2.1.

- (Variant 1b) Most students once again chose to put the key element at the end of the list, usually the opposite end relative to their best-case sequence.

- (Variant 3b) Some students gave a variant where the key element was in the middle of the list as the worst-case answer.
- (Variant 4) A few students provided a list which did not contain the key element.

When asked (verbal probing) about their Variant 3 answers, participants provided argumentations that resembled the key idea behind binary search optimization. The essence of their answers was that the larger part of a segment (luck is against the boy, who starts dancing at the “wrong end of the girl sequence”) becomes the smallest when the segment is divided into two equal parts.

All variants (1a, 1b, 2, 3a, 3b, 4) were considered correct answers in relation to Task 2.2. The correct answers for Tasks 2.3.1 and 2.3.2 were 1 (best case) and  $x/(x-1)$  (worst case), respectively.

## 6 RESULTS

Statistical analysis was performed using R statistical software. Since students' responses to Task-1 were binary data (0: incorrect; 1: correct) we used Fisher's exact test to compare the performance of each corresponding group. As for Task 2, which included five subtasks—2.1, 2.2.1, 2.2.2, 2.3.1, and 2.3.2—a decision was made to code students' responses from 0 to 5 (0/1 for each incorrect/correct response). In this case, the statistical tests applied were t-test and one-way or two-way ANOVA.

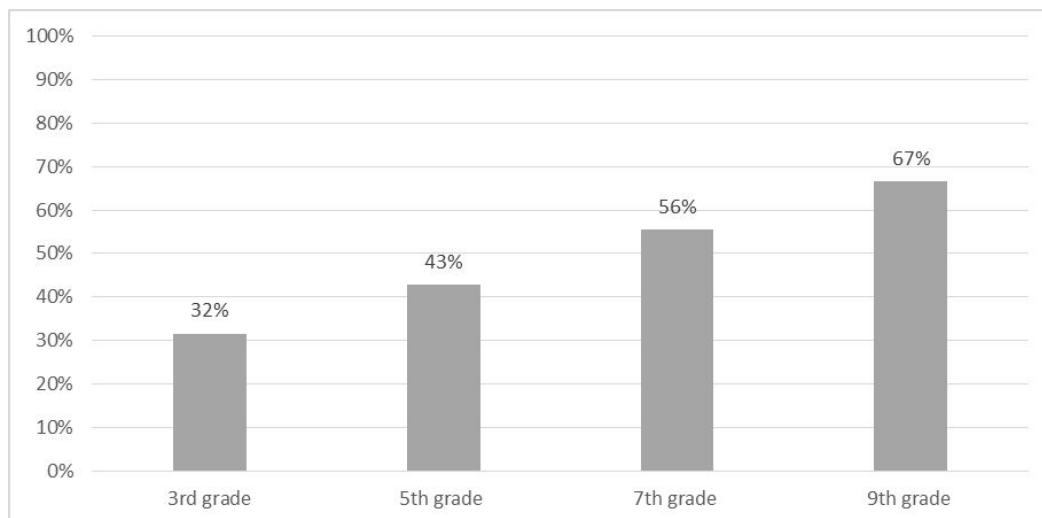
### 6.1 Results Grouped by Students' Grade

#### 6.1.1 Intuiting the Linear Search Algorithm (RQ 1)

A gradual, albeit moderate growth becomes visible (see Fig. 4) in view of the scores obtained for Task 1 (the extent to which students intuit the linear search algorithm) by all learners enrolled in grades 3, 5, 7 and 9 at both schools (see Table 3). While differences associated with the two-year shifts (3-5, 5-7, 7-9) were negligible, four-year shifts resulted in significant increases (Fisher's exact test: shift 3-7,  $p=0.01<0.05$ ; shift 5-9,  $p=0.01<0.05$ ).

**Table 3** All students' Task-1 and Task-2 performance percentages (means and standard deviations) grouped by grade (level)

	Task 1 M	Task 2 (2.1, 2.2.1, 2.2.2, 2.3.1, 2.3.2) M(SD)	Tasks 2.1, 2.2 M(SD)
Grade 3	0.32	0.65 (0.18)	0.83 (0.22)
Grade 5	0.43	0.80 (0.17)	0.84 (0.18)
Grade 7	0.56	0.80 (0.24)	0.85 (0.23)
Grade 9	0.67	0.79 (0.20)	0.89 (0.19)



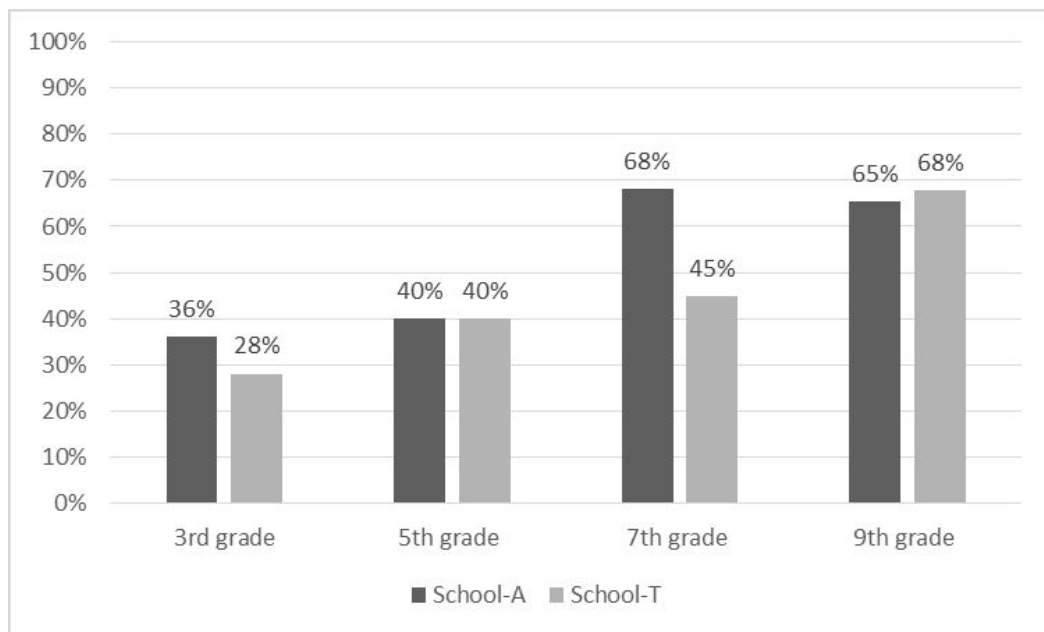
**Figure 4** All students' Task-1 performance percentages grouped by grade level.

### 6.1.2 Art vs. Theoretical School (RQ 2)

Investigating the two participant schools separately allowed us to check that, overall, students enrolled in art programs performed slightly better than their counterparts from theoretical school (School A: 53%, School T: 46%; Fisher's exact test:  $p=0.33>0.05$ ). Moreover, growth stages differed significantly (see Fig. 5). The following facts were taken into account while interpreting the results from this perspective:

- The elementary level curriculum at School A includes only the music program. The fine arts program starts on grade 5. Consequently, the students from class 5A (fine arts) participated in an elementary level educational program (at different "theoretical schools") similar to that of their counterparts from class 5T. In accordance with this, grade 5 classes performed equally well.
- A nationwide test is taken by all students at the end of the 8th grade. In order to register for a high school, a student must choose a list of secondary education centers they would like to study at. Afterwards, a national computer system carries out the distribution of students according to the order of their preferences and their admission grades. Therefore, a possible reason for the minimal decrease from class 7 to class 9 in School A could be that most of the students interested in CS move to classes that offer a higher level of CS teaching. Instead, the substantial increase from class 7T to class 9T can be explained by the fact that only well-performing students (based on their gymnasium final exam results) are admitted to School T's 9th grade classes. Once again, it becomes evident that classes 9A and 9T performed similarly.

In the light of the above-mentioned aspects, the most relevant comparisons from the perspective of the "Art versus Theoretical program analysis" are those between classes 3A and 3T, 7A and 7T and the growth from grade 3A to grade 7A vs. the one identified from



**Figure 5** All students' Task-1 performance percentages grouped by grade and by the schools they belong to.

3T to 7T. Focusing on these pairs, we noticed that, while a significant growth took place from grade 3 to 7 (32%) in School A (Fisher's exact test:  $p=0.04 < 0.05$ ) —mainly due to the increase from grade 5 to 7— (28%), growth was relatively small (17%) in School T. It additionally becomes evident that Class 3A clearly outperformed Class 3T (36% vs. 28%) and also that Class 7A performed substantially better (68% vs. 45%; Fisher's exact test:  $p=0.1$ ) than Class 7T.

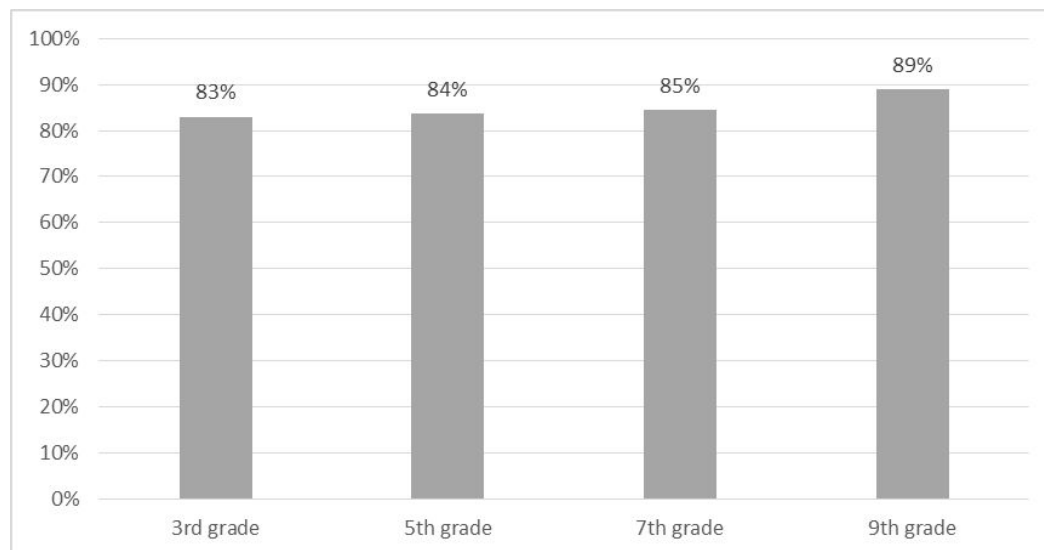
### 6.1.3 Assimilating the Linear Search Algorithm (RQ 3)

Students' performances at Task 2 can also be seen as CT measures, since the goal of these assignments was to reveal the extent to which learners were able to extract the linear search algorithm from the dance choreography presented to them.

Task-2 results (see Table 3) were analyzed using a one-way Analysis of Variance (ANOVA) test where grade (3rd, 5th, 7th, 9th) acted as the independent variable and Task-2 scores were the dependent variable (Levene's test showed that equal variances could be assumed:  $p=0.07 > 0.05$ ). Results suggest significant differences:  $F(210,3)=7.79$ ,  $p=0.000 < 0.05$ , partial  $\eta^2 = 0.1$ . As can be inferred from Table 3, 3rd-grade students (65%) were lagging behind their 5th-, 7th- and 9th-grade colleagues, who performed equally well (80%, 80%, 79%). An additional planned contrasts test (3rd grade vs. 5th, 7th and 9th grades) confirmed this observation ( $t(210)=4.83$ ,  $p=0.000$ ). Analyzing the responses of 3rd-grade students allowed us to notice that their performance was significantly lower than that of their higher-grade counterparts in Task 2.3.2 (only 12% of them answered x). This seems plausible, since denoting an arbitrary value by a letter (for example x) is not an

accessible concept yet for this age group (Kramer, 2007).

After eliminating Task-2.3 scores from our analysis we repeated the ANOVA test for students' Task-2.1 and Task-2.2 performances (Levene's test showed that homogeneity of variance was achieved again:  $p=0.47>0.05$ ). It surprised us that no significant differences (see Fig. 6) were found between the four grade levels under study ( $F(210,3)=0.83, p=0.47>0.5$ ).



**Figure 6** All students' Task-2 (subtasks 2.1, 2.2.1 and 2.2.2) performance percentages grouped by grade (level).

Our analysis of students' Task-2 scores (dependent variable) grouped by school (independent variable: Art or Theoretical), revealed no significant differences either (independent samples t-test; School A: 75%,  $SD=0.22$ ; School T: 77%,  $SD=0.20$ ;  $t(212)=0.74, p=0.45>0.05$ ). A two-way ANOVA was conducted in order to investigate whether A- and T-students performed equally well at each grade. The two independent variables were grade (3rd, 5th, 7th, 9th) and school (Art, Theoretical), the dependent variable being students' Task-2 score (Levene's test showed equal variances:  $p=0.7>0.05$ ). No interaction was detected ( $p=0.12>0.05$ ).

#### **6.1.4 Detecting Advanced CT Ability (RQ 4)**

Those variants which reflected that students addressed the “left-to-right or right-to-left dilemma” (or gave a sequence without the searched value as worst-case input), were considered signals of advanced CT in Task 2.2. Based on Task-2.2 results, 23% of 3rd-grade students, 17% of students enrolled in 5th grade, 36% of 7th-grade learners and 42% of their 9th-grade counterparts can be included in this promising category (students with advanced CT skills).

## 6.2 Results Grouped by Gender (RQ 5)

As the next step, we compared students' performance according to their gender. We found that male and female participants performed equally well in every task. For instance, the Task-1 score of both male and female students was 49%, the corresponding results for Task-2 being 74% and 77%, respectively. The examination of male and female students' scores for the two schools separately revealed no significant differences either. Interestingly, as for Task 1, boys outperformed girls in School T (boys' score: 52%; girls' score: 40%) while girls were more successful in School A (boys' score: 45%; girls' score: 57%); however, differences were not significant (Fisher's exact tests: for School T  $p=0.2>0.05$ ; for School A  $p=0.3>0.05$ ). Concerning Task 2, female students performed slightly better in both schools (School T: boys' score: 76%; girls' score: 78%; School A: boys' score: 73%; girls' score: 77%). A two-way ANOVA served to investigate whether gender groups performed equally well at each grade. The two independent variables were grade (3rd, 5th, 7th, 9th) and gender (male, female), the dependent variable being students' Task-2 score (Levene's test showed equal variances:  $p=0.55>0.05$ ). No interaction was detected ( $p=0.3>0.05$ ).

We also observed that, despite the similar performances, boys rated significantly higher in questions 2 (*How much do you like computer science?*) and 4 (*How much did you understand the linear search method?*) than girls. Two independent sample t-tests were conducted with gender as the independent variable in both. The dependent variables were students' Question-2 and Question-4 answers, respectively. Both tests showed significant differences (Question 2: male score ( $M=8.77$ ,  $SD=1.92$ ), female score ( $M=7.73$ ,  $SD=2.14$ ),  $t(212)=3.68$ ,  $p=0.0002$ ; Question 4: male score ( $M=8.24$ ,  $SD=2.22$ ), female score ( $M=7.40$ ,  $SD=2.51$ ),  $t(212)=2.52$ ,  $p=0.01$ ). As could be expected, girls rated significantly higher in Question 3 (*How much do you love music and dance?*).

## 6.3 The Importance of Computational Model-Based Algorithms

We finally examined students' wrong answers and noticed that some students (grade 3: 21%; grade 5: 22%; grade 7: 7%; grade 9: 11%) gave value 5 (or 1) as their Task-1 response. Similarly, some students provided for Task 2.1 value 6 (or 3). In both cases these students gave one less value than the correct answer, although they could have seen in the video that the boy's last dance was with his match. These participants subsequently justified (during verbal probing stage) their Task-1 (or Task-2.1) answers as follows: "After 5 (or 1) dances, the girl that the boy was looking for was the next one." This explanation suggests that such students lost the boy's point of view when they saw that the next girl was the searched one. According to Katai (2014), this phenomenon probably suggests that these students might also find it harder to identify with the computer's perspective than those who answered correctly (6 or 2 for Task 1 and 7 or 4 for Task 2.1).

## 7 DISCUSSION

These are the main contributions made with this paper: (1) A new testing tool (built around the linear search dance choreography from the AlgoRhythmic collection) is proposed to pro-

vide some measure of young school students' CT skills; (2) An examination has been made of the pace at which some young students' CT skills (girls vs. boys from Art vs. Theoretical schools) latently develop as they advance with their current CS free curriculum; (3) The extent to which students of different ages can assimilate a basic computer algorithm (linear search) has been investigated.

With regard to RQ-1 results, they revealed an implicit contribution of the current K-9 curriculum to students' CT, although that contribution is relatively low, since significant increases could only be detected with respect to four-year shifts. This finding matches those of previous studies which also examined students' basic understanding of CS concepts before they receive any formal instruction in this field [Simon et al. \(2006\)](#). By way of example, [Gibson and Kelly \(2005\)](#) found that pre-college students showed algorithmic understanding of how to solve a variety of search problems (being able to consider and reflect on the problem-solving process). In addition to the contributions made by the aforementioned studies, our research reveals the pace at which young school students grow in CT.

The conclusions reached in another related study are even more consistent with our results. [Brackmann et al. \(2017\)](#) investigated the effectiveness of unplugged activities in the development of CT skills in 5th-grade (10-11-year-old) and 6th-grade (11-12-year-old) students, none of whom had prior formal programming experience. Both classes were divided into control and experimental groups. Whereas lessons involving CT unplugged activities were assigned to the experimental group (for five weeks), the control group did not receive any intervention. The Computational Thinking Test proposed by [Román-González, Pérez-González, and Jiménez-Fernández \(2017\)](#) was administered as pre- and post-test to all participants with the following post-test results: (1) the experimental group performed significantly better than the control one; (2) 6th-grade students (in both groups) moderately outperformed their 5th-grade counterparts (though differences were not statistically significant).

These findings likewise suggest, on the one hand, that students' CT ability tends to increase (moderately) with grades. If we admit that CT is mainly a problem-solving ability, it seems quite natural for students' cognitive development and maturity to entail some improvement in their CT skills too ([Ackerman & Rolfhus, 1999](#)). On the other hand, the significantly better results obtained by the experimental group suggest that students' CT skill development can and should be accelerated by purposeful CT-oriented activities. While these authors investigated only two consecutive grade levels, our study extends the analysis to the entire K-9 educational process.

Regarding the "School T vs. School A comparison" (RQ 2) the differences detected revealed that young school students' progress in CT could be affected by factors such as the educational program, additionally suggesting that art education may have extra potential to enhance students' CT. This finding reminded us of several successful initiatives which combine arts and CS education ([Guzdial & Tew, 2006](#); [Laamanen, Jormanainen, & Sutinen, 2015](#); [Simon, Kinnunen, Porter, & Zazkis, 2010](#); [Tew, Mccracken, & Guzdial, 2005](#); [Wood, Muhl, & Hicks, 2016](#)).

With respect to the excellent results obtained for RQ 3 in Task 2 (including lower-grade learners' ability to catch up with their higher-grade counterparts), they hint at a real possibility to teach CS-based concepts successfully in primary and secondary education. This finding is in accordance with numerous existing experiences which provide evidence thereof [Gander et al. \(2013\)](#). For example, [Atmatzidou and Demetriadis \(2016\)](#) investigated the development of students' CT skills in the context of educational robotics learning activity with an experiment that involved students belonging to different age groups (15- and 18-year-olds) too. Participants carried out educational robotics learning activities for 11 weeks (2 hours per week) and their CT skills were evaluated at different phases throughout the learning process. Results show that students reached the same level of CT skills development regardless of their age. This result can be interpreted as meaning that 15-year-old students were able to assimilate the content these authors elaborated as effectively as their 18-year-old counterparts. Similarly, our results reveal that some basic computer algorithms can successfully be taught even to 3rd-grade students. Such findings will surely help teachers and policymakers identify the lowest grade where some CT-oriented concepts can be introduced.

Results regarding RQ 4 should be interpreted with caution. [Román-González et al. \(2018\)](#) recently addressed the subject of detecting CT talent and came to the conclusion that 'computationally talented' students can be detected in middle school. Their study revealed that these students have the ability to gain between 1 and 2 years in CSE standards compared to regular learners. The fact that our research only addressed the "current status" of students' CT prevents us from drawing direct conclusions about their possible CT talent. Our results should rather be seen as potential early signs (at primary and secondary levels) of a promising CT ability.

Findings related to the gender dimension (RQ 5) are consistent with those obtained in studies according to which no significant correlation exists between gender and programming course outcomes ([Ahadi et al., 2017](#); [Byrne & Lyons, 2001](#); [Werth, 1986](#)). For instance, [Lau and Yuen \(2009\)](#) did not find any significant gender differences in the programming performance of two hundred and seventeen secondary school students aged 14 to 19. More recently, the authors of an already mentioned study ([Atmatzidou & Demetriadis, 2016](#)) checked that male and female students reached the same level of CT skills development as well. Our findings, which extend such results to primary education too, match the conclusions reached in the study by [del Olmo-Muñoz et al. \(2020\)](#).

Results regarding Questions 2 and 4 are also coherent with previous research that identified female learners' self-conceptions in computing (e.g. self-efficacy) and their perceived lack of confidence amongst peers (despite their obvious abilities and successes) as important factors which discourage them from entering academic computing and staying therein [Ahadi et al. \(2017\)](#); [Stout and Tamer \(2016\)](#).

Concerning our results on the importance of computational model-based algorithms (Section 6.3) it is worth mentioning the study authored by [Onorato and Schvaneveld \(1986\)](#). Their subjects' assignment consisted in explaining how to find a name on a telephone directory either to a person or to a computer. They analyzed the natural language



descriptions of this task provided by naïve (with no programming experience), beginner (CS1) and expert (with a good deal of programming experience) students. Considering the explanations given to a computer, the authors found differences between all three groups. These results not only emphasize the importance of computational model-based algorithms in the context of CT (Denning, 2017) but also confirm the relevance of Katai's (2014) conclusion that algorithm visualization systems should support non-blind learners in assimilating the algorithm processing role of blind computers.

## 7.1 Limitations

The results of this study should be cautiously interpreted/generalized because the sample size was moderate and only two specific institutions were involved. Another limitation relates to its context: while 5th-, 7th- and 9th- grade participants performed the experiment within their usual environment, 3rd-grade students were invited to visit the labs of our university. In addition, the assessment tool covered a limited number of CT aspects.

## 8 CONCLUSIONS

A recent report by the Joint Informatics Europe & ACM Europe Working Group on Informatics Education emphasizes that: (1) computational thinking is an important ability that everyone should possess; (2) computer-science-based concepts, abilities and skills are teachable and must be included in the primary, and particularly in the secondary school curriculum (Gander et al., 2013). In this spirit, we designed and developed a novel testing method and tool to measure students' CT skills at different grade levels (3rd, 5th, 7th, 9th) across two different types of schools (art and theoretical) with both genders. Even though our study revealed supporting evidence that the current K-9 curriculum, even without an explicit focus on CS education, might contribute to students' CT skills, such contribution is in fact relatively low, since significant increases could only be detected with respect to four-year shifts (from grades 3 to 7 and from grades 5 to 9). The detected progress took place regardless of students' gender, though it could be affected by factors such as the educational program. Results suggest that art education may have extra potential to help develop students' CT skills. These findings emphasize the need for a purposeful and coordinated CS infusion into K-9 education so that students' CT development can be accelerated.

Another conclusion of this research is that choosing the right methods and tools make it possible for certain CT-related concepts to be taught as effectively to younger students as to older ones. This result supports the idea that CT promotion should start at the primary level. Besides, confirmation has once again been obtained that girls perform as well as boys in CT-oriented learning environments (although they continue to rate their interest in CS and self-efficacy beliefs lower than their male counterparts).

Not only curriculum developers and the computing education research community but also CS teachers can benefit from this research. The presented learning environment and the attached testing tool could prove suitable both to promote CT and to notice signs of advanced CT ability. Furthermore, the AlgoRhythmics environment (including ten different

complexity searching and sorting algorithms) provides an excellent opportunity to extend the core idea of this method to other algorithms as well as to calibrate the resulting tool for different learner categories.

## REFERENCES

- Ackerman, P. L., & Rolffhus, E. L. (1999). The locus of adult intelligence: Knowledge, abilities, and nonability traits. *Psychology and Aging, 14*(2), 314–330. <https://doi.org/10.1037/0882-7974.14.2.314>
- Ahadi, A., Lister, R., Lal, S., Leinonen, J., & Hellas, A. (2017, January). Performance and Consistency in Learning to Program. *Proceedings of the Nineteenth Australasian Computing Education Conference* (pp. 11–16). ACM. <https://doi.org/10.1145/3013499.3013503>
- Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal, 55*(7), 832–835. <https://doi.org/10.1093/comjnl/bxs074>
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems, 75*, 661–670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of Computational Thinking Skills through Unplugged Activities in Primary School. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 65–72). ACM.
- Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 9–9. <https://doi.org/10.1145/2602484>
- Byrne, P., & Lyons, G. (2001). The effect of student attributes on success in programming. *ACM SIGCSE Bulletin, 33*(3), 49–52. Retrieved from <https://dx.doi.org/10.1145/507758.377467>
- CSTA. (2017). *CSTA K-12 Computer Science Standards, Revised*. Retrieved from <https://www.csteachers.org/page/standards>
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education, 150*, 103832–103832. <https://doi.org/10.1016/j.compedu.2020.103832>
- Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM, 52*(6), 28–30. <https://doi.org/10.1145/1516046.1516054>
- Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39. <https://doi.org/10.1145/2998438>
- European Schoolnet. (2015). *Computing our future. Computer programming and coding: priorities, school and initiatives across Europe*. Retrieved from <http://www.eun.org/resources/detail?publicationID=661>
- Evans, G. E., & Simkin, M. G. (1989). What best predicts computer proficiency? *Communications of the ACM, 32*(11), 1322–1327. <https://doi.org/10.1145/68814.68817>
- Feaster, Y., Ali, F., Zhai, J., & Hallstrom, J. O. (2014, June). Serious toys: three years of teaching computer science concepts in K-12 classrooms. *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 69–74). ACM. <https://doi.org/10.1145/2591708.2591732>
- Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., Mcgettrick, A., & Meyer, B. (2013). *Informatics education: Europe cannot afford to miss the boat*.

- Gibson, J. P., & Kelly, J. (2005, October). Software engineering as a model of understanding for learning and problem solving. *Proceedings of the first international workshop on Computing education research* (pp. 87–97). ACM. <https://doi.org/10.1145/1089786.1089795>
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12. *Educational Researcher*, 42, 38–43. <https://doi.org/10.3102/0013189x12463051>
- Grover, S., & Pea, R. (2018). Computational thinking: a competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19–38). Bloomsbury Academic.
- Guzdial, M., & Tew, A. E. (2006, September). Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. *Proceedings of the second international workshop on Computing education research* (pp. 51–58). ACM. <https://doi.org/10.1145/1151588.1151597>
- Katai, Z. (2014, June). Selective hiding for improved algorithmic visualization. *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 33–38). ACM. <https://doi.org/10.1145/2591708.2591734>
- Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 36–42. <https://doi.org/10.1145/1232743.1232745>
- Laamanen, M., Jormanainen, I., & Sutinen, E. (2015, November). Theater robotics for human technology education. *Proceedings of the 15th Koli Calling Conference on Computing Education Research* (pp. 127–131). ACM. <https://doi.org/10.1145/2828959.2828975>
- Lau, W. W. F., & Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40(4), 696–712. <https://doi.org/10.1111/j.1467-8535.2008.00847.x>
- Lewandowski, G., Bouvier, D. J., Chen, T. Y., McCartney, R., Sanders, K., Simon, B., & Vandegrift, T. (2010). Commonsense understanding of concurrency: computing students and concert tickets. *Communications of the ACM*, 53(7), 60–70. <https://doi.org/10.1145/1785414.1785438>
- Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). *Computational thinking in K-9 education*. <https://doi.org/10.1145/2713609.2713610>
- 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, 105849–105849. <https://doi.org/10.1016/j.chb.2018.12.027>
- Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017, March). Assessing Computational Thinking in CS Unplugged Activities. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 501–506). ACM. <https://doi.org/10.1145/3017680.3017779>
- 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. <https://doi.org/10.1016/j.chb.2016.08.047>
- Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18, 47–58. <https://doi.org/10.1016/j.ijcci.2018.06.004>
- Settle, A., Goldberg, D. S., & Barr, V. (2013, July). Beyond computer science: computational thinking across disciplines. *Proceedings of the 18th ACM conference on Innovation and technology in computer science education* (pp. 311–312). ACM. <https://doi.org/10.1145/2462476.2462511>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Simon, B., Chen, T. Y., Lewandowski, G., McCartney, R., & Sanders, K. (2006, September). Com-

- monsense computing: what students know before we teach (episode 1: sorting). *Proceedings of the second international workshop on Computing education research* (pp. 29–40). ACM. <https://doi.org/10.1145/1151588.1151594>
- Simon, B., Kinnunen, P., Porter, L., & Zazkis, D. (2010, June). Experience report: CS1 for majors with media computation. *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 214–218). ACM. <https://doi.org/10.1145/1822090.1822151>
- Stout, J., & Tamer, B. (2016, February). Collaborative learning eliminates the negative impact of gender stereotypes on women's self-concept. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 496–496). ACM. <https://doi.org/10.1145/2839509.2850515>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798–103798. <https://doi.org/10.1016/j.compedu.2019.103798>
- Tew, A. E., Mcracken, W. M., & Guzdial, M. (2005, October). Impact of alternative introductory courses on programming concept understanding. *Proceedings of the first international workshop on Computing education research* (pp. 25–35). ACM. <https://doi.org/10.1145/1089786.1089789>
- Werth, L. H. (1986). Predicting student performance in a beginning computer science class. *ACM SIGCSE Bulletin*, 18(1), 138–143. <https://doi.org/10.1145/953055.5701>
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Wood, Z. J., Muhl, P., & Hicks, K. (2016, February). Computational Art: Introducing High School Students to Computing via Art. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 261–266). ACM. <https://doi.org/10.1145/2839509.2844614>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607–103607. <https://doi.org/10.1016/j.compedu.2019.103607>