Expanding computer science education in schools: understanding teacher experiences and challenges

Aman Yadav, Sarah Gretter, Susanne Hambrusch & Phil Sands

To cite this article: Aman Yadav, Sarah Gretter, Susanne Hambrusch & Phil Sands (2016): Expanding computer science education in schools: understanding teacher experiences and challenges, Computer Science Education, DOI: 10.1080/08993408.2016.1257418

To link to this article: http://dx.doi.org/10.1080/08993408.2016.1257418

Published online: 05 Dec 2016.
Expanding computer science education in schools: understanding teacher experiences and challenges

Aman Yadav\textsuperscript{a}, Sarah Gretter\textsuperscript{a}, Susanne Hambrusch\textsuperscript{b} and Phil Sands\textsuperscript{b,‡}

\textsuperscript{a}Department of Counseling, Educational Psychology and Special Education, Michigan State University, East Lansing, MI, USA; \textsuperscript{b}Department of Computer Science, Purdue University, West Lafayette, IN, USA

\textbf{ABSTRACT}

The increased push for teaching computer science (CS) in schools in the United States requires training a large number of new K-12 teachers. The current efforts to increase the number of CS teachers have predominantly focused on training teachers from other content areas. In order to support these beginning CS teachers, we need to better understand their experiences and challenges encountered in the classroom. This study investigated U.S. CS teachers' perspectives on the demands of teaching computer science and support needed to ensure quality teaching. Results suggested that teachers face a number of challenges, including isolation, lack of adequate computer science background, and limited professional development resources.

\textbf{1. Introduction: computer science education}

Recently, there has been an increased push for computer science education across the globe, including the United States of America (CS for All), the United Kingdom (Computing at School), Australia (Digital Technologies), and Mexico (Escherle, Ramirez-ramirez, Basawapatna, Maiello, & Nolazco-florez, 2016). These efforts to introduce computer science ideas in the curriculum across the primary and secondary education highlight the need to prepare today's students to a world heavily influenced by computing (CSTA & ISTE, 2011; Selby, 2015; Yadav, Hong, & Stephenson, 2016) and to move students from being consumers of technology to creators and producers (Buckingham, 2015). The Royal Society (2012) report on the state of computing curriculum in the U.K. highlighted a dwindling enthusiasm for computing given the shortage of teachers with sufficient subject matter knowledge, lessons taught by non-specialist teachers as well as negative perceptions of computing. The report recommended a restart of computing education in the U.K., including overhauling ICT in schools, and increasing the number of computing teachers and their professional development. Based on those recommendations, the national curriculum in England led the effort to guide computing education in primary and secondary schools. Similarly, in the United States a number of organizations, such as Code.org and College Board have been leading efforts to expand...
computer science education in schools. In order to expand computer science education, the White House announced CS for All “to empower a generation of American students with the computer science skills they need to thrive in a digital economy” (Smith, 2016).

While the field of computer science is expanding in primary and secondary education, there is a critical need to train a significant number of computing teachers (Gal-Ezer & Stephenson, 2010; Lye & Koh, 2014; Ni & Guzdial, 2012). While computer science teacher training efforts have taken place in some countries, such as Israel over the past decades (Gal-Ezer, Beer, Harel, & Yehudai, 1995; Lapidot & Hazzan, 2003; Ragonis, Hazzan, & Gal-Ezer, 2010; Tucker et al., 2004), other countries such as the U.K. and U.S.A still lag in producing significant number of teachers prepared to teaching computing due to issues of lack of teacher certification and initial training opportunities (CSTA, 2013; Gal-Ezer & Stephenson, 2010; Sentance & Csizmadia, 2016). For example, within the U.K., training new teachers with background in computer science remains a challenge (Sentance, Humphreys, & Dorling, 2014). A recent report by Computing at School Scotland found that in spite of the efforts to expand computer science in schools, there has been a 25% drop in CS teachers over the last decade, which is disproportionately down when compared to teachers of English (4% drop), Mathematics (6% drop), and Physics (10% drop) (CAS, 2016). The report also highlighted that schools struggle to fill vacancies for computer science teachers and schools with only one CS teacher are vulnerable to losing that teacher. A first step towards addressing questions of teacher preparation is to understand the experiences of current computer science teachers and the challenges they encounter in the classroom. This study aimed to understand computer science teachers’ perspectives and the demands of teaching computer science in order to inform the training of future CS teachers. For that purpose, we interviewed current high school teachers in the United States to understand the challenges they face in computer science classrooms. The results from this study help provide a “big picture” of what is needed to move forward to train thousands of future CS teachers and has implications for teacher training not only in the U.S., but also other countries. The following sections provide an overview of the current state of computer science teacher preparation.

1.1. K-12 computer science education

The demand for professionals with computing experience, the interest of undergraduates in computer science (CS) as a major, and the overall drive to expand computer science course offerings in primary and secondary schools has increased considerably in the last few years (Strickland, 2014; Taylor & Miller, 2015). The number of schools interested in offering computer science courses has exceeded expectations and many schools have shown interest in the Exploring Computer Science course, Code.org activities, and the new PLTW CS curriculum (Yadav et al., 2016). Despite all these activities and interest, many schools are unable to offer computer science courses due to the difficulty in hiring computer science teachers (CAS, 2016; Google, 2015a). To meet the growing demand and increase the role of computing in K-12, a major investment is needed to train teachers to teach computing and computing-related topics (Cuny, 2012); this includes developing teachers’ knowledge of computing concepts and how to deliver those ideas to students.

Training thousands of new computer science teachers to meet this goal is a daunting task. One of the recommendations is to develop and deploy strategies for existing teachers from other content areas to become effective CS teachers through professional development
The efforts for teacher training in the United Kingdom provide potential pathways to get new teachers into computing. For example, the Department of Education in the U.K. provides support, such as tax-free scholarships and fully funded subject knowledge enhancement courses to train teachers to teach computer science. Computing at School has also developed a certificate in computer science teaching, which teachers can complete on either a guided or an independent route to demonstrate their competence in teaching the new Computing curriculum. Hazzan, Gal-Ezer, and Blum (2008) presented a four-component model to increase computer science education in Israel, which could inform other countries as they engage in growing CS in elementary and secondary classrooms. Their proposed model includes four components necessary for a successful CS education program: “A well defined high school CS curriculum (including textbooks and teaching guides), a mandatory CS teaching license, teacher preparation programs (and in-service training), and research in CS education”. However, training teachers to teach computer science raises many questions for educators and educational researchers regarding the best practices for preparing teachers. How can we deliver content knowledge and promote effective instructional practices for inservice teachers? How do teachers acquire knowledge, including the knowledge of subject matter, pedagogy, and pedagogical content knowledge (PCK) in CS? What is the minimal set of computing skills that a teacher needs to be an effective computer science educator?

Within the United States, the challenge of recruiting, training, and retaining computer science teachers is impacted by the fact that computer science teacher certification is deeply flawed (CSTA, 2013). A number of recent reports have found that there are few pathways to become a CS teacher in teacher education programs and only a couple of states in the U.S. actually have certification requirements to teach computer science (Century et al., 2013; CSTA, 2013). Similar challenges exist in the U.K. to train CS teachers through initial teacher education institutions, where programs are struggling to recruit enough high-quality preservice teacher applicants (CAS, 2016).

There is a limited number of teacher preparation programs in the United States that offer computer science teacher certification programs as the primary licensure areas (CSTA, 2015). The teacher certification programs that do offer computer science endorsements have “no tangible relationship to what is needed to teach in a computer science classroom” (Gal-Ezer & Stephenson, 2010, p. 63). Those responsible for creating, implementing, and enforcing teacher education policies do not have a clear understanding of what constitutes computer science (Gal-Ezer & Stephenson, 2010; Khoury, 2007). Computer science is typically confounded with other subject areas, such as Technology Education/Educational Technology, Industrial or Instructional Technology, Management Information Systems, or even the use of computers to support learning in other subject areas (Google, 2015b; Khoury, 2007). Given the inconsistent teacher certification demands and difficulty recruiting preservice teachers to focus on computer science teaching, it is amazing that there are any computer science teachers at all.

The lack of adequate teacher certification pathways for computer science often leads to teachers with little or no computer science training being assigned to teach computer science courses (Gal-Ezer & Stephenson, 2010). A recent CAS report also highlighted that CS teachers are also asked to teach as many as four non-computing classes, which could contribute to teacher attrition (CAS, 2016). Recent survey results conducted by the Computer Science Teachers Association offer some insight into the current composition of computer
science teachers at the K-12 level (CSTA, 2015). Specifically, results have suggested that the majority of the teachers who teach computer science (57%) also teach other content areas. Furthermore, the majority of the teachers (62.3%) also reported that computer science was taught through the business, technology, or mathematics department in their schools. Hence, teachers who wished to teach computer science had to first meet teacher certification requirements in a second discipline and demonstrate expertise in that discipline even if they did not plan on teaching it in the future (Gal-Ezer & Stephenson, 2010).

Prior research has shown that as a result of having varied backgrounds, beginning computer science teachers continually struggle to meet pedagogical demands in the classroom as they lack adequate computer science content and pedagogical knowledge to effectively teach computer science (Gal-Ezer & Stephenson, 2010; Yadav et al., 2016). This is not surprising given that good teaching requires not only substantial understanding of the subject matter, but also a solid background of pedagogy and knowledge of the learners (Hill, Ball, & Shilling, 2008; Shulman, 1986). These two areas of knowledge, referred to as “content knowledge” and “pedagogical knowledge”, are highly dependent on each other (Shulman, 1986). Shulman extended this notion by introducing PCK, “which goes beyond the knowledge of subject matter per se to the dimension of subject matter knowledge for teaching … [and includes] the ways of representing and formulating the subject that make it comprehensible to others” (Shulman, 1986, p. 9). An effective method for increasing computer science teachers’ PCK would be to provide them with opportunities tied to their curricular needs, develop their knowledge of content and of students’ needs (Hill, Ball, & Shilling, 2008), engage them in continuous reflection and dialogue about teaching (Powell, Diamond, Burchinal, & Koehler, 2010), and support their identity development (Ni & Guzdial, 2012). However, studies examining computer science PCK are sparse; we know very little about how computer science teachers come to develop these knowledge systems and what challenges they face given the constraints of teacher licensure requirements.

Saeli et al. (2011) aimed to uncover PCK within the context of programming using four core questions: what are the reasons to teach programming; what are the concepts we need to teach programming; what are the most common difficulties/misconceptions students encounter while learning to program; and how to teach this topic. The authors found that responses to the four questions were not connected to each other and recommended additional work in order to portray programming PCK both from a general perspective as well as most frequently taught topics. In another study, Liberman, Kolikant, and Beeri (2012) explored relationship between three CS teachers’ content knowledge and PCK as they shifted from a procedural programming to teaching a new paradigm in Object-Oriented Programming. The authors found that even experienced CS teachers are in a new state (called regressed expert) in which they incorporate elements typical of novice teachers as well as expert teachers.

Research on experiences of novice teachers in other disciplines offers some insight into how computer science teachers without adequate background come to navigate the classroom, along with challenges they face in the classroom. Prior work has suggested that beginning teachers face a number of challenges, including: teaching outside one’s area, feeling isolated, teaching multiple subject areas, classroom management, and insufficient planning time (Conway, 2001; Feiman-Nemser, Schwille, Carver, & Yusko, 1999; Lortie, 1975; Veenman, 1984). Veenman (1984) conducted a meta-analysis of 83 studies from across the globe that focused on problems that teachers encounter in the classroom. After an extensive review
of the studies, Veenman identified the 24 most perceived problems including: assessing student work, heavy teaching load leading to reduced planning time, lesson planning, knowledge of subject matter, and inadequate guidance and support. Other researchers have also suggested that beginning teachers report similar challenges and concerns that include: loneliness and isolation (Corley, 1998), inadequate preparation time, heavy teaching load (Dollase, 1992; Ganser, 1999), difficulty in lesson planning (Dollase, 1992; Hertzog, 2002), and struggles in assessing student learning (Hebert & Worthy, 2001). Given the problems that novice teachers have identified, researchers have argued that these teachers need support during their first years of teaching to increase their content and pedagogical knowledge, self-efficacy, and beliefs about what it means to be a successful teacher (Huling, Resta, & Yeargain, 2012; Martin, 2008). These strategies – known as induction programs – help support teachers through mentoring and reduces teacher attrition rates during the first few years (Fantilli & McDougall, 2009; Huling et al., 2012; Ingersoll & Smith, 2004; Martin, 2008). Liberman et al. (2012) recommended that allowing CS teachers to emulate a lab session might also “improve their understanding of (at least certain aspects of) the CK [content knowledge] and encourage them to develop an initial PCK, by emulating the pressure of students in this contexts, [and] their possible questions” (p. 282).

1.2. The present study

As discussed previously, computer science teachers come with varied backgrounds, which means that professional development and induction programs tailored to provide mentoring and support would be beneficial for them. However, we know very little about computer science teachers’ experiences, challenges of teaching computer science, and support strategies they believe would be valuable to them. The current demand to train more teachers in computing (Yadav et al., 2016; Cuny, 2012), along with the high risk of attrition of computing teachers (CAS, 2016) and the lack of clarity surrounding teaching licensure programs in computer science (Gal-Ezer & Stephenson, 2010) makes it imperative for us to understand problems that they face when teaching computer science in the classroom. Hence, the present study examined teachers’ perspectives about teaching computer science, challenges they face in the classroom, and areas they believe are needed for them to be successful in the classroom. Specifically, this study addressed the following research question: What challenges do CS teachers face in the classroom and what support systems they perceive would be helpful?

2. Method

2.1. Participants

Twenty-four high school computer science teachers from across the United States of America participated in the study. Participants included 13 males and 10 females. The majority of participants were Caucasian (N = 22) with one teacher identifying as Hispanic and another one identifying as Asian. In addition to computer science, the majority of the teachers also taught either business education (N = 8) or mathematics (N = 6). Participants cited in this paper are included in Appendix 1 along with their pseudonyms, gender, prior general and computer science teaching experience, and teaching licensure.
2.2. Measures

A semi-structured interview protocol was used to elicit teachers’ views on teaching computer science and to discuss their experiences in a computer science classroom. Teachers were also asked to share how teaching computer science compared to teaching any other subject area. The interview questions provided them with opportunities to share the challenges that they faced in terms of the content as well as the pedagogical needs of a CS classroom. The interview was pilot-tested with two computer science teachers to examine the clarity of questions and was modified based on their feedback. The data from the two pilot teachers were not included in the data analysis.

2.3. Procedure

We invited computer science teachers to participate in the study via mailing lists of computer science teachers. Teachers first completed an online background questionnaire, which took about 5 minutes and was used to collect demographic information. Participants were then contacted to schedule an interview aiming to better understand the challenges that computer science teachers faced in their classroom. On average, the interviews lasted about 39 min and 35 s (SD = 13 min 19 s) and were conducted either via face-to-face or by phone depending upon the proximity of the researchers and the teachers. Participants were compensated for their time. We used data saturation as the factor in determining an adequate sample size for the qualitative study (Mason, 2010). More specifically, we decided to conduct interviews as long as they offered new insights into computer science teachers’ practices and stopped collecting data once the interviews stopped shedding any additional information teachers’ perspectives about teaching computer science (Charmaz, 2006). Based on the data saturation concept, we interviewed 24 teachers to address the research question.

2.4. Data analysis

All interviews were transcribed and imported into a qualitative analysis software called Dedoose. The interview transcripts were coded with Dedoose using a detailed line-by-line analysis to generate initial codes from the raw data. During this phase of coding, the researchers stuck closely to the data to generate initial codes and help researchers in “defining what [was] going on in the data and grapple with the meanings underlying it” (Charmaz, 2006, p. 49). Once the initial coding was done, we used an iterative process to continually merge and collapse related codes until they could not be collapsed any further. The final themes were, then, collapsed into conceptually based categories to provide researchers with a broader conceptual structure of teachers’ perspectives of teaching computer science. In order to establish stability of themes, we conducted interrater reliability using the themes that emerged from the data. Two raters coded one-third of the data. The original coder created a codebook in an Excel file with a list of themes and their corresponding definitions in two columns. Randomly selected raw data were included in a separate sheet, which was coded by the second rater using themes and descriptions from the first sheet. Creswell (2013) recommended that for this approach, arguing that it is “important to have agreement on the text segments we [are] assigning to codes than to have the same, exact passage coded” (p. 201). Cohen’s kappa was run to determine the agreement between the two raters and based on guidelines from Landis and Koch (1977) suggested a substantial agreement, $k = .80, p < .05$. 
3. Results

The interview data revealed a number of codes related to teachers’ experiences in computer science classrooms. The qualitative analysis generated an initial list of 58 codes, which were collapsed into eight conceptual themes. The themes were then categorized into three overarching categories: (i) challenges of teaching in a computer science classroom, (ii) compounding factors that influence teaching in a computer science classroom, and (iii) professional needs of computer science teachers. Below, we discuss each of the overarching categories and corresponding themes.

3.1. Challenges of teaching in a computer science classroom

Overall, teachers faced a number of challenges while teaching computer science, including several sub-themes that related to the challenges of becoming proficient in the content and pedagogical aspects of computer science teaching. Participants discussed challenges related to meeting both content and pedagogical needs of students in the classroom, which are compounded for beginning teachers due to their lack of computer science teacher preparation.

Teachers also discussed the difficulty in assessing student learning in computer science due to the lack of organized assessment items. Themes related to the difficulty of teaching computer science are outlined below.

3.1.1. Content challenges

One of the main themes that emerged from the interviews was that teachers ($N = 13$) initially struggle with teaching computer science because of their limited content knowledge. Specifically, teachers expressed that a beginning computer science teacher without a computer science background has to learn concepts herself, while helping students understand them at the same time. This is highlighted by one teacher’s comment:

I think in general anything that’s new to me that I don’t have a lot of experience with is challenging to teach because now I’m not only trying to understand it myself but I’m trying to distill it in a way to help students understand it better. (Amanda)

Another critical issue for teachers was the relationship between computer science courses and programming experience. Without the necessary background and understanding of programming constructs, teachers expressed their struggles to meet the needs of students in the classroom:

I don’t know how you do it, but if someone with no programming experience, if they’re gonna be teaching this course, Python [a programming language] is gonna hit them in the face like “what is this?” (Gregorio)

Beginning computer science teachers that lack content knowledge are presented with the challenge of not being able to explore concepts in depth. This, in turn, limits their ability to meet the learning demands of students. As the same teacher stated, “Can I get real in depth on most things? Not really. At least not right now. So that’s kind of where I’m at.” (Gregorio)

3.1.2. Pedagogical challenges

Another major concern for the teachers in our study ($N = 15$) was the pedagogical challenge of teaching computer science. Because of the student-centered nature of CS work, keeping all students engaged and focused in the course content can be difficult. For example, one
teacher mentioned that the presence of computers in the classroom could be distracting as students often get off-task. He stated,

I would say with students today being able to capture all of their interests at one time and still have them stay on task is always a challenge. Especially when you're in a room with 20 computers and they all connect to the Internet and every student knows there's so much out there. (Delmar)

Similarly, another teacher highlighted that the teacher–student ratio in a computer science classroom is even more challenging because of needs of individual learners, stating:

The big challenge that I face when I teach CS is that it's very … because of its student-centered nature teaching the subject, there's a lot of one-on-one during class time and this year I have 32 students in one section, and there is one me and 32 of them. (Amanda)

Another pedagogical challenge that teachers brought up was the uniqueness of the problem-solving approaches that each student might use when working on a project, making it complicated to address all students' needs. This sentiment was highlighted in the following comment:

Having the knowledge and ability to approach a student and understand where they’re coming from, and then being able to take them to the next step … Because student A will approach it one way, student B will be coming at it from another way, and I could look, at the same … The students are creating code, so I can look at the codes on the monitor, but understanding things well and being able to meet them at their level and guide them – That has been a bit of a struggle. (Kenneth)

A recurring theme in our interviews was the difficulty in meeting student needs on an individual basis. Adding the challenge of evaluating student progress stresses the importance of pedagogical training for teachers of computer science courses.

3.1.3. Assessment challenges

Not only did teachers report difficulties related to how to deliver computer science content, they (N = 10) also discussed challenges related to evaluating student learning in their classroom. It was oft-mentioned that the lack of access to quality computer science assessment tools makes it difficult for them to accurately gauge what students are learning. For example, one teacher stated that his lack of computer science background made it difficult for him to come up with assessment items:

For me, not having a computer science background or even my major, I struggle coming up with good ways to evaluate the students and to know how much should I grade. Should I be grading every activity? Or should I be pausing after a certain amount and saying, “Ok we’re gonna have a quiz here.” Or “you need to perform a small program here.” So I'm not sure what to do there. (Kenneth)

Another teacher communicated that assessing students was difficult given the collaborative nature of some computer science assignments; these tasks in their eyes made it laborious to accurately evaluate what students know. This teacher stated that when students were working on a project, it was hard to know whether they understood the concepts themselves or whether they were able to complete the assignment based on help from peers. She said,

In a high school class there's a lot of, it's hard to just grade them by a few quizzes and tests. That work that's done in class, you want to try to give them credits for doing that, but a lot of times, even then, if they've done it in class, you don't know if they've gotten help from their neighbor. So, is it worth grading that if it's just kind of practice work? Is it a true assessment or true worth of what they know or is it just kind of ok they practiced this, how much is that worth? (Cynthia)
Teachers in our study also talked about using rubrics to grade student projects; however, rubrics run the risk of informing students as to how to do the minimal amount of work to obtain a passing grade on the project. They often times reduce creativity and expression rather than enhance or inform it. One of the teachers stated,

Rubrics are hard because you don’t want to give a kid a rubric that’s so defined that they’re just using it as a checklist and they’re not really stretching themselves. I like one of my rubrics – the top score is knock my socks off. You know, I wanna see something that you can be proud of. (Amanda)

Together, content, pedagogy, and assessment form a triad of challenges that CS teachers are confronted within the classroom. For novice CS teachers specifically, these problems can be particularly challenging if they do not have the support or prior experience needed to resolve them.

3.2. Compounding factors that influence teaching computer science

While the issues outlined above are related to meeting pedagogical needs in a computer science classroom, teachers also discussed a number of additional factors that compound those challenges. These factors relate to the lack of focus on computer science teacher preparation, solitary nature of CS teachers within K-12 schools, and limited information technology support received within individual schools. The following subsections outline these particular challenges.

3.2.1. Lack of CS teacher preparation

The challenges beginning computer science teachers face with regard to meeting content and pedagogical needs of students could be due to the lack of teacher preparation programs in computer science. Teachers in our study discussed that the lack of computer science teacher preparation programs in their own background was a challenge and that they have learned to teach computer science principally on their own. While there are computing teachers who have formal background in teaching (N = 8), they do not necessarily have the CS content knowledge needed to teach computer science. On the other hand, there are teachers (N = 11) who have experience in programming (through undergraduate computer science courses), but do not have a teaching background. Teachers discussed how they have self-taught themselves to meet the demands of teaching CS. For example, one teacher stated

I didn't have any really formal training in that. And so it was kind of “here's a book.” And I kind of learned some things at home, did some stuff on the Internet, you know, kind of did it that way. (Gregorio)

Another teacher expressed a similar sentiment, stating, “When I decided I would teach programming, then I got books and I did self-teaching things” (Martha). Some teachers (N = 6) also shared that during their beginning years of teaching they would sometimes let students take control of their learning and think through the problems together in class. This was highlighted by one teacher, who said,

I remember one time I struggled for two hours trying to figure something out, so I introduced it as a problem and didn’t tell them I didn’t know the answer … and the kids figured it out and we moved on (Fermin)

Similarly, another teacher expressed that taking advantage of promising students as instructors can provide classroom support, stating, “The kids teach me a lot and I learn along with
them, so, I guess I'm more self-taught than anything” (Ismael). While being able to use the classroom as a setting for collaboration and learning with students, participants’ experiences reflected an underlying detachment between CS teaching and teaching in other areas of the profession.

3.2.2. Isolation

One of the biggest challenges of being a computer science teacher in a K-12 school is the solitude of working without peers in one’s content area. Teachers in the study (N = 13) talked about this isolation and how traditional content area teachers, such as Mathematics or English, have support groups of other teachers who can help them brainstorm ideas related to teaching the subject area in question. On the other hand, a computer science teacher is generally the only one at the school teaching CS (typically in addition to other subject areas). This is highlighted by the following teacher’s comment:

I mean textbooks, all that material. Now, I don’t have the textbook. I have to build all my own content. There's only one of me where we have a bunch of English teachers, a bunch of math, they work together whereas I have to do all my content alone, I don’t have anybody else. Well I have the business department, but I'm the only one that does programming. (Adam)

Another teacher expressed a similar sentiment, stating:

Grading is very hard in computer science. Coming up with materials and you are alone. If you have a question, if something doesn’t work you are on your own. (Anne).

While there are a number of online communities for teachers to engage in, teachers value spontaneous discussions that come up as a result of being in the same building. The same teacher added:

It is a challenge. A big challenge. I know there's a lot of communities, Internet communities and such. But somehow having someone face-to-face that you talk to or work with is just not there. (Anne).

3.2.3. IT challenges

While personal support was an area of concern for some teachers, teachers (N = 16) also discussed the need for reliable support from information technology teams within the school district to meet computer science teaching demands. Teachers discussed the unavailability of the latest technologies, as schools don’t always have resources to upgrade hardware. One teacher highlighted this fact by stating that:

Our computer lab, the equipment is actually six years old. So, trying to find programs that will work on the older equipment or the software. Internet connections can sometimes be sketchy. It's just working through some older equipment and the financial constraints of not being able to upgrade and stuff. (Chris)

Teachers also talked about the slow process of getting IT staff to update software as teachers don’t have administrator privileges on computers in their classrooms. The following teacher comment highlighted this:

I have to get them to come in. And that even includes updates. I’ll get a message that there's a new update available for this. I can't even do it. And so, it has to wait and every problem. I try to fix any computer problems in my class because every problem requires sending a work order in, and having them get to it and stuff. (Anne).

A similar view was expressed by another teacher who stated:
Some of the environments that we were trying to use, we struggled to get them installed on the students' computers, you know it was one of those things where I had to get a technical person to come in here because I don't have administrative rights to modify their computers. (Irene)

Both the technical and the personal struggles therefore revealed the obstacles that CS teachers may face on a daily basis.

3.3. **Supporting computer science teachers**

Identifying the demands and challenges of teaching computer science is a good first step, and there are a number of things that the computer science education community can do to support CS teachers. During the interviews, teachers in our study identified some key areas that would meet their teaching needs. Specifically, teachers discussed the need to access quality teaching material through centralized resources and to develop active learning communities for teachers, both at the local level as well as more globally online. This section presents the needs that teachers identified during the interviews.

3.3.1. **Organized repository**

Teachers in our study (\(N = 15\)) talked about online resources and expressed that while there is an abundance of teaching resources online, it can be challenging to assess their quality. For example, one teacher stated,

> You know the Internet is big and there are some good things out there but they’re hard to find. So I’m looking forward to resources that will make me more effective. (Hal)

The teachers mentioned the need for a better organization of online resources in one central repository organized by level and topic, and which would also provide teachers with the ability to rate such activities. One teacher commented,

> It would be really, really nice to have some kind of online repository where people can post worksheets and ideas. Have some kind of tagging system where you can search. Maybe even a ranking system where if I’ve used it, I can add a comment at the bottom saying, “hey, I used this when I taught Boolean logic with a group of intro students and it worked really well, I rank it 4 out of 5 stars,” kind of thing. (Jolina)

One of the reasons teachers wanted such a system was to support beginning CS teachers, who need to sift through online resources to find ones that match their pedagogical and content knowledge needs.

> I have built up a bank of 15 years of material that I’ve sort of developed just because I’ve done this for a long time, but especially when you look at this idea of CS10 K [10,000 CS teachers by 2016] with all these new teachers coming in, realistically most people, especially if they’re trained in another field, are not gonna be used to having to develop this many materials. (Jolina)

One beginning teacher highlighted a similar thought, stating

> Looking online was too advanced, I would love to see something at the first and second year level. I never did Python in college so it would be helpful. (Max)

Teachers also discussed the need for quality assessment materials that they could use in their classrooms:

> Well I would just like a really good bank of questions. Short questions that will test this one concept, you know, a for loop or a select statement, something like that. I just would like a good bank of those in a centralized place. (Max).
Material support was often cited by teachers as a need for improving their teaching practices. However, connecting with other teachers seemed to also be part of the elements that participants needed to complement their teaching toolboxes.

### 3.3.2. Community of practice

Teachers (N = 10) brought up the need to form a community of practicing CS teachers in order to address their isolation within schools. Participants stated that such a community would provide them with opportunities to share ideas about approaches to delivering CS content, share practical examples, and develop collaborative relationships. For example, one teacher expressed that:

> Just being able to talk with other teachers, computer science teachers and interact with them. To see how other people are using things in the classroom. Get ideas on stuff to do, ways to present the information in a new way. It was the kind of thing that I thought, “oh, if I could just get all the teachers up”, then they could see, oh wow, you can actually use Scratch programming for a lot of things and it’s not just something stuck down in the computer lab. (Chris)

Similarly, the following comments highlight the need for a community where CS teachers could share resources to allow them to deliver concepts in multiple ways,

> I need extra examples of ways to do different things would be nice. It’s nice to have one example of how something is done but to have multiple ways with different versions would be nice. (Ismael)

During the interviews, participants acknowledged that connecting with other professionals in their field was essential to developing their CS knowledge and pedagogy, along with their sense of belonging to the CS community. They also recognized their need for IT support, for teaching material, and for assessment strategies. Below, we further discuss these findings and draw implications from teachers’ perspectives on the challenges of teaching CS.

### 4. Discussion

Teachers in our study highlighted a number of challenges beginning computer science teachers face and discussed professional development needs they feel would have helped them adapt to the classroom. Teachers reported that during the first few years they themselves did not either have adequate content knowledge or pedagogical knowledge to teach computer science. Teachers with a formal background in teaching often didn’t have the CS content knowledge needed to teach CS. On the other hand, teachers with industry experience in programming did not have any teaching background to effectively deliver CS lessons. These teachers developed their repertoire on their own, sometimes while simultaneously teaching it to their students. These challenges are further exacerbated by the lack of computer science teacher preparation programs, insufficient professional development, and inadequate technology support at their schools.

Given that teachers in our study discussed how they developed knowledge to teach computer science on their own, teacher educators and professional development experts should train new computer science teachers by providing content and pedagogical support that is directly tied to teachers’ instructional goals and curricular framework (Century et al., 2013). The recent push to train more computer science teachers to meet the growing demand for computer science courses (Cuny, 2012) makes it imperative for us to consider what inservice teachers know and need in order to be successful in computer science classrooms.
Prior work on effective teaching has suggested a need for teachers to be fluent in both the knowledge of subject matter that is to be taught (content knowledge), the ways of teaching (pedagogical knowledge), and PCK. PCK includes knowing what makes certain topics easy or difficult, as well as ways of representing and formulating topics in order to make them comprehensible to others (Shulman, 1986). Professional development for inservice teachers is a crucial step in developing teachers’ PCK and propagating beneficial computer science teaching practices. However, one and done professional development approaches that rely heavily on workshops without structured ongoing engagement are known to have limited success (Dickinson & Caswell, 2007); hence, computer science education researchers need to develop ongoing and continuous PD opportunities to engage teachers in learning experiences designed to expand their understanding of computer science, pedagogical tools, and assessment.

Teachers in this study also commented on how teaching computer science can be a lonely enterprise in schools where there are few computer science teachers, or often only one per school. Computer science teachers typically work independently and often rely on virtual communities to stay in touch with colleagues, rather than through the continual collaborations that in-house teachers may have (Brown & Kölling, 2013). These findings support one of the challenges highlighted by the “Building an operating system for computer science education” (OS4CS) report that computer science teachers face isolation in the classroom (Century et al., 2013). The OS4CS report highlighted that “CS teachers don’t have colleagues close by to share ideas, information about the discipline, or provide instructional support and coaching”, which can affect motivation; hence, making it difficult to improve computer science instruction. The finding that computer science teachers feel isolated needs to be carefully addressed given that collaborating with other teachers within the same subject area has been found to be a significant predictor to recruiting and retaining teachers (Smith & Ingersoll, 2004).

Teachers in our study stressed the need of supporting computer science teachers through repositories and online communities of practice to allow them to collaborate and engage with peers. However, while repositories can provide computer science teachers with access to materials, teachers’ use of repositories is more nuanced and influenced by a number of factors. Fincher, Kölling, Utting, Brown, and Stevens (2010) identified key factors that impact computer science educators’ use of online repositories, which can inform the design of repositories. Fincher and colleagues examined how CS educators use Nifty repository, an online resource with a rich set of computing education-related items, mainly introductory programming assignments. Based on their findings, the authors provided recommendations for developing repositories for K-12 teachers, such as including social dimensions and Web 2.0 features, to allow users to engage with peers. Specifically, these features would allow users to rate and recommend materials, see how peers were adapting the materials to their own context, and feeding the adapted resources back into the repository.

Fincher and colleagues (2010) also discussed the importance of developing a community model of interaction rather than the traditional repository model, which would allow teachers to be a “community of people gathering in a place (such as a teacher common room), rather than that of a repository of things (such as a library or a database)” (p. 113). The authors provided a number of suggestions to help develop and maintain this community model of interaction to “support a community of teachers in developing, sharing and discussing resources and their teaching” (p. 112). The authors discussed how a sense of community was
built within the Greenroom, a community for teachers using Greenfoot software. They suggested that the Greenroom model includes having members use real names and personal picture as well as encouraging them to mark their geographical location. Furthermore, the controlled access to Greenroom by vetting teacher credentials also allows members to have confidence in the community. These features of Greenroom play a key role in allowing for a richer set of interactions between the teachers that go beyond simply sharing resources to having discussions around the material. In another study, Brown and Kölling (2013) discussed key features of three different virtual communities for computer science educators, including Greenroom, Blueroom, and CAS Online and examined how teachers used each of the three communities. The results provided an invaluable insight into how different factors influence computer science educators’ engagement within online communities. For example, the authors found that users’ teaching background and experience in computing may influence the types of discussions (e.g. asking programming-related vs. asking teaching-related questions) they would engage in. The results from this study can inform the development of other virtual communities of practice to support computer science teachers.

A community of practice is a group of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly (Wenger, McDermott, & Snyder, 2002). As such, a community of practice focused around teaching computer science can provide teachers with opportunities to reflect on their teaching practices with each other and gain multiple pedagogical perspectives, which significantly influences their thinking about CS teaching (Bryk, Camburn, & Louis, 1999). Within computer science, Ni, Guzdial, Tew, Morrison, and Galanos (2011) found that engaging teachers in a community of practice allowed them to feel a sense of belongingness within the discipline and see similarities in their teaching, which validated what they did in their own classrooms. The sustained nature of the community allowed teachers to be more confident in their own classroom practices and promoted reflection about their own teaching. Fincher and Tenenberg (2006) used the bootstrapping metaphor to argue that a community of practice for CS researchers could be supported “whereby a complex system emerges by starting simply and, bit by bit, developing more complex capabilities on top of the simpler ones”. This model could be used to: (1) bootstrap a novice CS teacher by providing entry points into content and pedagogical practices relevant to teaching computer science; (2) bootstrap a community of practice of CS teachers with similar skills and practices in shared teaching endeavors; and (3) bootstrap a wider community of CS teachers who engage in reflections around meeting CS pedagogical needs. The findings from this study have important implications for supporting computer science teachers through professional development opportunities and communities of practice as well as directions for future research. The following section provides implications and future directions as computer science education researchers create opportunities and strategies for supporting teachers.

4.1. Implications and future directions

One of the main implications from this study is the need for developing communities of practice for teachers to meet their curriculum needs (both content and pedagogical) and address the lonely enterprise of teaching computer science. Computer science education researchers and experienced and novice CS teachers could collaboratively develop regional
hubs, which provide induction programs (i.e. professional development, support, mentoring, etc.) focused on teaching computer science not as “a single, time-bound activity, but a continuing process of career development” (National Commission on Excellence in Teacher Education, 1985, p. 10). Within these teacher induction programs, experienced computer science teachers could serve as master teachers to provide mentoring support for beginning teachers. The Computing at Schools (CAS) Hubs in the United Kingdom is a promising approach to develop a network of computer science teachers and researchers, who come together for bite-sized professional development opportunities with a “specific aim of providing (at least) one idea that can be taken and tried in the classroom” (CAS, 2013, p. 5). Within the CAS community of practice, master teachers with excellent CS classroom practices and knowledge are recruited and trained to support and meet the needs of their local peers (Sentance, Humphreys, & Dorling, 2014). As we have argued elsewhere, this model of professional development opportunities would allow experienced computer science teachers and researchers to engage beginning computer science teachers in learning experiences designed to expand their understanding of computer science, pedagogical tools, and assessment (Yadav, Hambrusch, Korb, & Gretter, 2014).

We need to carefully consider how to engage computer science teachers in virtual communities of practice that go beyond merely accessing resources to engaging them in rich interactions around teaching and assessing student learning of computer science concepts in the classroom. As discussed previously, the findings and lessons learned from computer science educators’ use of repositories in general (Mitchell & Lutters, 2006), specific repositories like Nifty repository (Fincher et al., 2010), and virtual communities, such as the CAS Online (Brown & Kölling, 2013) can inform the development of new online communities for CS teachers. These communities of practice should not only be a place to share resources, but a place where CS teachers engage in rich discussions and reflective dialogue about teaching computer science with peers. Such a community would allow teachers, especially teachers with limited experience, to develop a clear and more elaborate understanding of teaching and learning related to computer science (Levin, 1995; Shoffner, 2009).

The fact that teachers in this study discussed both content and pedagogical challenges of teaching computer science during the beginning years highlights the need to prepare computer science teachers through traditional teacher education programs. Faculty in schools of education and computer science need to collaborate to develop teacher education curricula that prepare preservice teachers to teach computing principles at the primary and secondary levels. The teacher attrition rates in conjunction with the dearth of teacher education programs that prepare computer science teachers serve as a sounding alarm for CS education researchers and policy-makers. The current efforts to expand computing in K-12 schools will not be successful in the long term without a pipeline of new teachers who are well prepared to “combine pedagogical principles with computer science content to improve the learning experience for their students” (Yadav & Korb, 2012). Our prior work in preparing preservice teachers to teach computer science through a supplemental licensure program is a first step in this direction (See Yadav and Korb for a discussion of the program). However, this is a piecemeal approach, which would only yield tens of computer science teachers when there is a need for thousands. Hence, we need teacher educators and education policy-makers at the state level to work on changing policy and developing programs to make computer science teaching licensure a stand-alone program.
The current study used qualitative interviews to examine U.S. teachers’ perspectives of teaching high school computer science and challenges they face in the classroom. The results shed some light on how we can support computer science teachers in the classroom through professional development, teacher induction, and mentoring programs. Future research should examine how these strategies to support computer science teachers influence both teacher outcomes (such as, self-efficacy, and motivation) as well as student outcomes (such as, knowledge and attitudes of computing concepts). Future research could also examine how experienced computer science teachers could be used as mentors to help beginning CS teachers develop knowledge systems and practices to meet their pedagogical goals. For example, researchers could investigate how a video case library of exemplary pedagogical approaches from experienced computer science teachers might benefit beginning teachers.

This study had some limitations that should be acknowledged when considering its findings. One of the limitations was that while qualitative research can provide an in-depth understanding of participants’ perspectives on a specific issue, qualitative data from a small number of participants limits generalizability to the larger population of CS teachers. Another limitation of this study was that the teachers all came from the United States, so the challenges identified by them might not be relevant to other countries. Hence, researchers interested in examining CS teachers’ challenges in their own settings should take into account the particular context of the present study. Although limited to a single country-specific context, the results of this study address some important challenges in computer science education, which may or may not be present in other countries, but which warrant further inquiry. In summary, this study provided insights into computer science teachers’ experiences and the challenges they face in the classroom and could provide computer science researcher and teacher educators the opportunity to carefully design programs that would meet the needs of teachers.

Acknowledgments

We would like to thank all the teachers who participated in this study. This work was funded by National Science Foundation (NSF) under grant 1502462. The opinions expressed are those of the authors and do not necessarily reflect those of NSF.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was funded by National Science Foundation (NSF) under [grant number 1502462].

Notes on contributors

Aman Yadav is an associate professor in Educational Psychology and Educational Technology at Michigan State University. His research focuses on preparing teachers to teach computer science and embed computational thinking concepts.

Sarah Gretter is a doctoral candidate in Educational Psychology and Educational Technology at Michigan State University. Her research focuses on preparing teachers to integrate media information literacy in their pedagogy.
**Susanne Hambrusch** is a professor in Computer Science at Purdue University. Her research focuses on computer science education, parallel and distributed computation, and analysis of algorithms.

**Phil Sands** is a doctoral student in Educational Psychology and Educational Technology at Michigan State University. His research interests focus on computer science education and issues of broadening participation.

**ORCID**

Aman Yadav [http://orcid.org/0000-0003-4247-2033](http://orcid.org/0000-0003-4247-2033)

**References**


## Appendix 1. Teacher profiles

<table>
<thead>
<tr>
<th>Teacher</th>
<th>Gender</th>
<th>General teaching experience</th>
<th>CS teaching experience</th>
<th>Teaching licensure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>Male</td>
<td>12</td>
<td>12</td>
<td>Business Education</td>
</tr>
<tr>
<td>Benny</td>
<td>Male</td>
<td>10</td>
<td>10</td>
<td>None</td>
</tr>
<tr>
<td>Martha</td>
<td>Female</td>
<td>15</td>
<td>15</td>
<td>Business Education, CTE*</td>
</tr>
<tr>
<td>Cynthia</td>
<td>Female</td>
<td>14</td>
<td>13</td>
<td>Mathematics</td>
</tr>
<tr>
<td>Chris</td>
<td>Male</td>
<td>6</td>
<td>6</td>
<td>Bible</td>
</tr>
<tr>
<td>Anne</td>
<td>Female</td>
<td>35</td>
<td>30</td>
<td>Mathematics, Computer Education</td>
</tr>
<tr>
<td>Jolina</td>
<td>Female</td>
<td>16</td>
<td>15</td>
<td>Mathematics, Computer Science</td>
</tr>
<tr>
<td>Malika</td>
<td>Female</td>
<td>10</td>
<td>10</td>
<td>Business, Computer Science</td>
</tr>
<tr>
<td>Delmar</td>
<td>Male</td>
<td>8</td>
<td>8</td>
<td>Business Education</td>
</tr>
<tr>
<td>Tama</td>
<td>Female</td>
<td>11</td>
<td>11</td>
<td>Business Education</td>
</tr>
<tr>
<td>Tiera</td>
<td>Female</td>
<td>15</td>
<td>10</td>
<td>Business Education, Computer Endorsement</td>
</tr>
<tr>
<td>Wanda</td>
<td>Female</td>
<td>9</td>
<td>9</td>
<td>Business Education</td>
</tr>
<tr>
<td>Eugene</td>
<td>Male</td>
<td>17</td>
<td>9</td>
<td>Business Education</td>
</tr>
<tr>
<td>Fermin</td>
<td>Male</td>
<td>6</td>
<td>3</td>
<td>Technology Education</td>
</tr>
<tr>
<td>Gregorio</td>
<td>Male</td>
<td>17</td>
<td>7</td>
<td>Mathematics, Physical Science</td>
</tr>
<tr>
<td>Hal</td>
<td>Male</td>
<td>6</td>
<td>1</td>
<td>Mathematics, CTE</td>
</tr>
<tr>
<td>Patricia</td>
<td>Female</td>
<td>10</td>
<td>10</td>
<td>Business</td>
</tr>
<tr>
<td>Ismael</td>
<td>Male</td>
<td>6</td>
<td>1</td>
<td>Industrial Arts</td>
</tr>
<tr>
<td>Jeremiah</td>
<td>Male</td>
<td>8</td>
<td>1</td>
<td>Engineering and Technology Education</td>
</tr>
<tr>
<td>Amanda</td>
<td>Female</td>
<td>16</td>
<td>14</td>
<td>Mathematics and Computer Science</td>
</tr>
<tr>
<td>Kenneth</td>
<td>Male</td>
<td>13</td>
<td>1</td>
<td>Technology Education</td>
</tr>
<tr>
<td>Lucien</td>
<td>Male</td>
<td>5</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Max</td>
<td>Male</td>
<td>10</td>
<td>1</td>
<td>Physics, Science, Earth &amp; Space Science</td>
</tr>
<tr>
<td>Irene</td>
<td>Female</td>
<td>2</td>
<td>2</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

*CTE = Career & Technical Education.