A Co-Instructional Model to Develop High School CS Teachers in Historically Underrepresented Communities

Building Capacity with a Purpose

Andrew Lapetina
Electrical and Computer Engineering and Computer Science
University of Detroit-Mercy
Detroit, MI, United States
lapetiaj@udmercy.edu

Richard Hill
College of Engineering and Science
University of Detroit-Mercy
Detroit, MI, United States
hillrc@udmercy.edu

Aman Yadav
Educational Psychology and Educational Technology
Michigan State University
East Lansing, MI, United States
ayadav@msu.edu

Michael Lachney
Educational Psychology and Educational Technology
Michigan State University
East Lansing, MI, United States
lachneym@msu.edu

Madison Allen Kuyenga
Educational Psychology and Educational Technology
Michigan State University
East Lansing, MI, United States
allenm72@msu.edu

ABSTRACT

Many secondary school teachers in historically lower performing districts find themselves asked to teach CS without adequate training. They find themselves struggling to build a CS program without the environment necessary to foster student engagement and success. To improve educator preparedness, and transform school cultures, Innovating Detroit's Robotics Agile Workforce (iDRAW) offers one year of co-instruction by a university instructor with a high school teacher to better prepare the teacher to provide rigorous CS instruction independently in future years. Reflections from this university co-instructor are contained, with insights into the preparation of CS teachers and initiation of CS courses in high schools in the future.

CCS CONCEPTS

• Computer Science Education

KEYWORDS

Computer science education, teacher preparedness, co-instruction

ACM Reference format:

Andrew Lapetina, Richard Hill, Aman Yadav, Michael Lachney and Madison Kuyenga. 2024. A Co-Instructional Model to Develop High School CS Teachers in Historically Underrepresented Communities: Building Capacity with a Purpose. In *Proceedings of Conference for Research on Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT '24). May 16-17 Atlanta, GA. ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3653666.3656067

RESPECT 2024, May 16–17, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-0626-4/24/05. https://doi.org/10.1145/3653666.3656067

1 INTRODUCTION

Imagine a high school that had no sports or physical education whatsoever and began a volleyball team. The school assigned 20 random students to the team, and made a teacher who liked to run on weekends the coach. The teacher went to a one-week volleyball camp the summer before, and was offered 4 virtual meetings with other first-year coaches about volleyball coaching over their first season. No one would be surprised if the athletes did not like the sport, the teacher felt unqualified, and the team had a losing record. Further, the quality of the coach would have little influence on the performance of the team.

Consider a school which instead had several sports and a robust physical education program. The school is considering adding another sports team. After asking for input from students and parents, and finding a strong interest, the school decides to start a field hockey team, and recruits intrigued athletes from complementary sports such as soccer and basketball. In looking for a coach, they ask teachers who have played field hockey or are familiar with the sport. This teacher is sent to a few training sessions in the year before the season starts, and the school hires an assistant coach from a nearby school district with experience in the sport.

CS education policy struggles with a tension between access and rigor. Considering the analogies above, the former school is offering access to a sport, while the latter is providing rigorous access to physical education, and augmenting this access with another option. The movement for every student to learn to code believes that programming is as essential as reading and writing for youth, and this can manifest in a state or district mandate to offer CS, without adequately preparing teachers for this task. But, merely enhanced access to CS educational resources via web-based learning has not closed the achievement gap between white, wealthy students and students of color [1].

These two schools present different approaches to starting a program, and speak to differing approaches schools take to providing CS education. The obvious contrasts inform very local policies on developing CS teachers, and introducing CS to schools. iDRAW, a program at the University of Detroit-Mercy which offers dual-enrolled engineering and computer science courses at high schools in the metro-Detroit region, has attempted to help schools avoid the former model, and adopt the latter model, and with proper support, could help many schools build comprehensive CS and STEM programs, ultimately enhancing CS education for all.

At the heart of navigating this tension between access and rigor should be the question of "Why" a school is adding CS to their course offerings, and what they are hoping for their students as a result. This has a massive impact on the perspectives and perception the teacher has for themselves. Considering our analogy of sports programs, it seems that the former school added volleyball to provide access. The latter school, already well resourced, looks to add access and rigor. The experience of iDRAW, which has trained 8 high school CS teachers, at very different schools, has put this tension into relief as some schools have attempted to merely add access, but without the school culture which may be essential to develop rigorous access. The students at all high schools deserve rigorous access to computer science, but at question is how present policies and programs can create that culture of rigorous access.

A data point to consider is that only 35% of school principals in Michigan believe that their guidance counselors consider it important to offer CS opportunities [2]. If the school administration does not believe that students will be shepherded into CS with gusto and support, there is reason to fear schools will adopt the former model of our analogy, even as access to CS curriculum increases.

A novel program presented here takes a uniquely qualified university co-instructor from the University of Detroit-Mercy, a small university with a teaching focus, and pairs them with a high school teacher. Over an entire school year, the two co-teach a dual-enrolled, university course in computer science (APCSP), with a culturally relevant and sustaining pedagogy.

The university co-instructor works with 2-3 high school teachers in a given year, in an effort to comprehensively improve CS education in the metro-Detroit region. Because this is the primary employment objective of this instructor, it refines the practices of the university co-instructor, and potentially creates a community of CS teachers for mutual support.

2 GOALS AND METRICS

The goals of iDRAW, which guide the reflection of this paper are:

1. to provide a rigorous CS learning experience to a high school class
2. develop the teacher's content knowledge so they can teach independently, and
3. support the high school in adopting a culture supportive of CS and engineering instruction.

To measure progress at these goals, the following metrics are considered here:

1. Performance on the APCSP Exam
2. Evaluation of teacher content knowledge.
3. Continuation of the instruction of APCSP and other engineering courses

3 DESCRIPTION OF PRACTICE

This program seeks to offer sustained, comprehensive pedagogical and content knowledge training to high school computer science teachers through a full school year of co-instruction with a university instructor. Every day, the university co-instructor and high school teacher are present in the classroom. Each day, one of them leads the lesson, with the other providing support via leading a segment of class, leading a small group of students, or parallel management of project-based work time. Weekly, the two collaboratively plan for an hour, to ensure proper focus on appropriate topics, coordination with school-based schedules, and check for the teacher's understanding of core content.

This co-instructional relationship yields many benefits for teachers. Teachers feel less burdened to master the nuances of delivering a completely new computer science lesson every day, giving them space and time to ruminate on content and identify their misconceptions. Weekly meetings amount to 40 hours of individual professional development for the high school teachers with an experienced computer science instructor. And regularly observing the university instructor assist students in debugging, vocabulary practice, and creative programming helps high school teachers learn how to effectively perform these complex teaching tasks. Last, because the university instructor handles most grading and planning for the first year, the time commitment for a high school teacher taking on this responsibility is reduced.

In some of the schools where the university co-instructor has worked, other dual-enrolled STEM courses are provided and co-taught, such as mechatronics and CAD. While the focus of this paper will be on the co-instruction of CS, these other engineering courses bring students who did not believe they were interested in CS into the CS classroom, and very importantly mixes students who have taken CS with other students in the school in rigorous engineering courses. In the interest of identifying future policy steps and outline successful strategies for enhancing CS teaching in secondary schools, this aspect of iDRAW is essential, and will be discussed towards the end of the paper.

This program is part of an NSF-funded Research Practitioner Partnership (NSF grant 2122349) between the University of Detroit-Mercy, Michigan State University (MSU), and Detroit Public Schools Community District (DPSCD). These three partners seek to support one another in advancing CS education in the metro-Detroit region. The University of Detroit-Mercy houses the program and provides the university co-instructor, along with program leadership. DPSCD suggests schools and teachers for co-instruction, and communicates the value of the program to school communities. MSU analyzes data arising from the program, offering critical examination of program practices to improve outcomes for teachers and students.

4 CONTEXT

Many high schools in Michigan aspire to offer CS coursework to students, but struggle to find qualified and eager teachers, with only 50% of urban high schools in the state offering CS [2]. Complicating matters are staffing shortages of certified STEM educators, combined with rigid graduation requirements in math and science, leading to challenges in scheduling teachers. While the growth of accessible but rigorous curriculums such as code.org, CodeHS, and Khan Academy have provided useful tools for teaching, they rely on a model of a single week of summer training followed by a few hours sprinkled throughout the year. Additionally, models where professionals from industry join a classroom to assist a teacher are widespread, but these professionals lack pedagogical training or experiences germane to teaching rigorous content at scale, over the course of a year.

Here, a uniquely qualified university co-instructor pairs with a high school teacher to deliver content with strong pedagogical skill and experience. In addition to a Ph.D in engineering, the university co-instructor has 5 years experience as a traditional high school teacher in DPSCD. This combination of training permits the university co-instructor to build relationships with high school communities, demonstrate delivery of rigorous content, and appropriately pace the course to meet APCSP expectations. Further, the university co-instructor has been trained in culturally relevant and sustaining pedagogy in computer science, which allows them to demonstrate its inclusion into the CS curriculum.

Initially, iDRAW was supported by the State of Michigan as a workforce development program. APCSP was one of three STEM courses developed through this grant, the others being mechatronics and CAD. Now iDRAW is funded by the NSF CS for All program, which primarily supports instruction of APCSP. As an innovative solution to broadening participation in CS instruction inclusively, this program has refined its techniques and model over 5 years. Four of the high schools involved over the five year lifespan of iDRAW offer multiple dual-enrolled engineering courses, and four have only offered APCSP as a dual-enrolled course.

5 AUDIENCE

The co-instructional model described here exists primarily to support teachers asked to teach CS without prior adequate content knowledge. None of the high school teachers involved completed majors or minors in CS in college, or had certificates to teach CS from the state. Two of the eight had previously taught CS in some capacity. In most cases, the school or district had an interest, desire, or mandate to increase CS instruction, and identified a high school teacher to teach CS which could meet other school objectives, namely scheduling for state-required courses. This selected high school teacher, in search of support, was connected with the program, and the relationship developed positively from there.

The schools involved are diverse in demographics and size. Research papers under review will document this in great quantitiative detail, but three of the high schools are nearly entirely African American students, two are nearly entirely Latino students, two have very high numbers of English language learning students, and two are very diverse on many metrics such as race, ethnicity, and native language. Of the 8 high school co-instructors, three are African American women, three are white women, one is a white male and one is a Hispanic male.

A district or state which was interested in replicating this model could do so in partnership with a university committed to dual-enrolled instruction. The most difficult element to reproduce is the university co-instructor, because the training required amounts to a minimum of a master's degree in science or engineering with significant CS experience, a secondary teaching certification, and several years of classroom teaching experience. The pool of candidates is very small [3], but the regional impact of the position is extremely high. Conceivably, a regional educational association or business conglomerate could support such a position, tying it to education and economic development.

One of the major influences on the success of the program is the selection of school and teacher to co-instruct with. Based on the observations of the university co-instructor, the ideal teacher would have a strong interest in teaching CS, a track record of offering rigorous coursework to students in other fields, and a passion for learning new skills rapidly. The ideal school would have a full class of students prepared to take APCSP level coursework, but not already have a full complement of CS teachers. Most importantly, the school would have leadership interested in developing a rigorous STEM program, and commit to engaging with the university program. Reasons for this are discussed in the outcomes section.

6 OUTCOMES

6.1 Goal 1- Student Outcomes

In all instances, students were offered the opportunity for rigorous instruction, and at every school involved, at least one student passed the AP exam during the year of co-instruction. One of these schools had never had a student pass an AP exam. The first year of the program (the COVID shortened 2019-2020 school year), over 30% of Latinx test takers in Michigan who passed the APCSP exam were iDRAW students, and average scores exceeded a 3.

Subsequent years of implementation have seen drastic declines in average student scores in co-instructed classrooms, and classes taught by high school teachers after the co-instructor moved to other classes. Both cases require separate investigation, and while future papers will analyze them more deeply, initial reflections are provided here.

The primary reason for lower scores in classes where the co-instructor is present is the process by which students are enrolled in the class by guidance offices and administration. Because this is often the first time APCSP is offered within these high schools, guidance counselors are challenged to identify and place appropriate students in the course, and struggle keeping students enrolled in the course through the year. In short, these classes have often been highly populated by students with a lack of interest who do not wish to learn computer science. Consequently, a few students with passion are able to pass the exam, but the average scores are not near the national average. Students with a desire to be placed in computer science typically thrive, enjoy the class, and perform acceptably well on the exam.

The reason high school teachers have not had tremendous success with student performance after their year of co-instruction is related to the second goal. But, the earliest iterations of the co-instructional model did not include all of the features described in the description of practice. High school teachers were not required to teach at least a section of every lesson, which led to them often passively taking in content. Therefore, the inclination of researchers is that their content knowledge was not sufficient to teach the course with great effectiveness.

6.2 Goal 2- Teacher Content Knowledge

To test high school teacher content knowledge and growth, the teachers are given a 20 question, multiple choice test of APCSP-aligned questions in the beginning and end of the school year, as a pretest and post-test. This practice was implemented after the first year of co-teaching because of a perceived lack of growth in teacher content knowledge. A cohort of 3 teachers were the first questioned in this fashion, and showed no growth in content knowledge over the course of 1 year. The mean number of correct answers was 10 for the pretest, and 11.3 for the post-test, with no outliers. Currently, another cohort of 3 teachers has taken the pre-test, and will take the post test in May of 2024.

To address this lack of growth, the co-teaching model has evolved in several ways. First, scheduling has been implemented to allow for weekly, hour-long content focused meetings between the high school teacher and the university co-instructor. These meetings help check for the understanding of the high school teacher, allow for co-instructional planning. Second, previous co-instruction has had high school teachers lead entire lessons while the university instructor would lead other lessons. Now, co-instruction generally takes one of three shapes: Either the university co-instructor leads 80% of a lesson, with the high school teacher leading one segment of class, the high school teacher leads a small group of students who have struggled or were absent in completing a larger task or assignment, or the two co-manage student work time on large projects. This forces both parties to be present in teaching during class time, and maximizes the content absorption of the high school teacher, every day.

Interestingly, two high school teachers completed a year of co-instruction, and took this 20 question multiple choice test after having taught independently for 3 years. They scored 15 and 19 correct answers, significantly outperforming teachers who had just finished their year of co-instruction. While growth data is not available for these teachers, it indicates that they have greater content knowledge than their less experienced peers. Future data will reveal the importance of the modifications to co-instruction described here.

Another observation is that while the high school teachers can improve on their understanding of specific ideas of computer science, they lack the ability to craft a complex program of their own, as is required for the AP Exam. The high school teachers are never required to do something like this, as students are when completing courses. So, there is never a pressure-filled situation to learn how to develop software which will force the teachers to begin thinking in a creative computational way.

In a professional development activity run by this program, with 15 CS educators present, only 3 teachers were able to create a program which contained a loop, a list, a conditional, a function, and parameters. One of those three completed the year of co-instruction, and afterwards taught CS independently for 3 years. Of the 12 who could not, 3 were teachers who had completed the year of co-instruction, but had not yet taught independently. This co-instructional model alone does not cultivate the ability to creatively write software to meet criteria, even if it may increase teacher content knowledge.

Outside evaluation of the RPP was conducted by American Institutes of Research, which included quantitative surveys and qualitative interviews about the program. Unanimously,the high school teachers found the experience helpful in their development as CS teachers. While this qualitative feedback is helpful, an increase in sample size and comparison to other interventions such as the engagement of technical professionals in the classroom would be worthwhile.

6.3 Goal 3- Sustained Program Development

So far, the university co-instructor has partnered with 8 high school teachers in CS, and 4 high school teachers in other engineering content. Of the 8, 4 are teaching rigorous CS independently, 1 is teaching computer science, 1 has left their position to work in administration, and 2 are currently in their year of co-instruction. The two in their year of co-instruction are prepared to teach independently. In courses other than CS, 3 of the 4 high school teachers are still teaching their dual-enrolled course at their school, and the fourth moved to a different state. The importance of this is evident when discussing implications and next steps, and in light of the sports program analogy from the introduction.

In terms of overall program building, all of the schools where the university co-instructor worked still offer APCSP, and two of the three where the program is ongoing are planning to continue offering APCSP next year. Only one high school teacher who co-instructed is still teaching, but not APCSP.

7 POSITIONALITY STATEMENT

The lead author of this paper is a white, American-born male possessing a Ph.D. in civil engineering. Subsequently, he completed an alternative certification to teach by teaching in majority African American, Title 1 high schools for five years before beginning the work described in this paper. The author is cognizant that his biases and prejudgments may be present here, but he made a conscious effort to confront those with evaluation from a diverse set of peers.

8 LIMITATIONS AND ASSUMPTIONS

The greatest limitation in evaluating this program is the very small sample size of teachers, schools, and university instructors who have completed the co-instructional model. By design, the program is small in scale and provides an exceptionally high level of support to the high school teachers as they become a CS educator. While some educational interventions are designed for rapid and widescale deployment and growth, this is not, to ensure high quality support to complicated pedagogical and content knowledge development.

Secondly, each school has particular idiosyncrasies which affect the impact of the program on their educational framework. Some high schools have multiple early college and dual enrollment programs that students are simultaneously enrolled in, while at others this is the only dual enrollment option.

Finally, there is legitimate concern that the use of this single university co-instructor is a product of a force of personality rather than a sustainable model for rigorous engineering education at scale. Expansion of the program either via replication in other cities or the use of other university co-instructors will reveal more information about methods described.

9 IMPLICATIONS AND NEXT STEPS

Reflecting on the goals and outcomes described, a few conclusions seem evident:

9.1 Passive co-instruction will not lead to learning by osmosis.

A historical flaw of the program was the assumption that high school teachers could absorb content knowledge by being present for lessons taught by the university co-instructor. The failure of teachers to grow in content knowledge belies this. However, the content knowledge levels of teachers who completed the year of co-instruction, then taught independently for 3 years shows that this combination of experiences may lead to strong teacher preparation. While this model has refined the approach of the instruction of CS, the addition of data from this current school year will prove powerful in interpreting how adjustments to the co-teaching model are working. This school year, teachers are expected to teach a segment of a lesson on almost all days, and weekly meetings regarding content reinforce teacher understanding. Future articles emerging from this RPP will provide more information about this issue.

9.2 Schools need to build a culture of rigorous STEM coursework, rather than simply offer access

As described in the introduction, high schools often receive a grant or mandate, and then begin a CS course without much thought. A more reasonable approach would be for a teacher to initiate the process of building student capacity and interest in CS, and then, with a supportive guidance office and principal, acquire the resources necessary for a CS course. This has been the approach in 3 of the high schools, with the other two being some combination. Realistically, the most successful CS classes exist in schools where computational thinking is present in many courses, and students may see CS in the contexts of robotics, GIS, AI language models, and more.

A conclusion from this experience report is that the most important piece of CS education which can occur in a school is when a CS-fluent student teaches a CS-curious student in a context outside of a CS classroom. But, this is very rare in a school where APCSP is the only rigorous CS course. Consider Laura, a junior excelling in APCSP. Without a suitable follow-up course, she will likely never have the opportunity to instruct other students in her high school about computer science. But, if she could enroll in a dual-enrolled mechatronics course her senior year, she could teach basic CS content to Ramone, a sophomore eager to build robots but leery of coding. Further, this interaction makes the job of both the CS and mechatronics teacher vastly easier.

9.3 Scaling demands strong university co-instructors and interested universities

To scale this model up, more university co-instructors with comfort teaching in high school will be essential. Smaller colleges and universities have a unique role in providing nimble support to STEM education through dual-enrollment in secondary schools. One could envision a university with a strong GIS program offering computer science and Python-focused GIS courses in high schools, while another with a strong aerospace engineering program offering drone programming and computer science. Universities could capitalize on their strengths to offer rigorous, integrated coursework to high school students to both engage more young people and build capacity among secondary educators. Conceivably, this is a pathway for many smaller universities to boost enrollment by increasing their attractiveness to high school students in their regions.

Embedded within this concept is the need for more university co-instructors trained for and excited about the prospect of teaching in a secondary school environment. At a minimum, the university co-instructor needs a master's level education in an engineering field, and significant experience effectively teaching high school students in a classroom environment. While this is a challenging set of skills to identify, a surprising number of candidates exist [3], and a few in a metro region could have a dramatic effect over a long period of time, due to the nature of the co-instructional model.

Most obviously, students studying to become STEM teachers would benefit tremendously from taking a single computer science course while completing certification. Teachers with a strong knowledge base will be able to instruct students more effectively, and even if CS is not their primary area of instruction, it will provide high schools with a stronger pool of potential educators in future years. States considering adding a mandate of computer science coursework for high school students should provide secondary STEM educators a semester of introductory college computer science, offering them the experience to teach this content effectively.

Looking back at the analogy opening this paper, the objective of policymakers, and target of activism by researchers should be for more high schools to offer not just access, but rigorous, multidisciplinary access to computational thinking. The nature and specifics of this access should emerge from the unique strengths and experiences of the teachers within the high school staff, since this will be the driving force educating children. Completed with fidelity to provide just education, this could transform the opportunity of historically underrepresented groups in CS.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant 2122349 (Moving beyond access, increasing teacher knowledge to teach rigorous equity-focused high school computing). The authors are grateful for the generosity of this grant.

.

REFERENCES

[1] Code.org Advocacy Coalition, "2023 State of Computer Science Education," *Code.org Advocacy Coalition*, 2023. [online]. Available: [https://advocacy.code.org/state-handouts/Michigan.pdf] [Accessed January 15, 2024].

[2] Michigan Department of Education, "State of Computer Science Education in Michigan," *Michigan Department of Education*, 2020. [online]. Available: [https://www.michigan.gov/-/media/Project/Websites/mde/Year/2021/03/09/State_of_Computer_Science_Education_in_Michigan_Report.pdf] [Accessed January 15, 2024].

[3] National Center for Education Statistics, "Characteristics of Public School Teachers," Condition of Education, 2023. [online]. Available: https://nces.ed.gov/programs/coe/indicator/clr/public-school-teachers [Accessed January 15, 2024].