Concept Maps in Computer Science Education

Donovan McMurray
Georgia Institute of Technology
Brooklyn, USA
donovan@gatech.edu

ABSTRACT

Concept Maps have been studied in education on a wide range of subjects. There is a great deal of evidence for their effectiveness in helping students form deep connections of underlying concepts over other forms of knowledge representation. However, not much research has been done on the effectiveness of Concept Maps in Computer Science education. In this paper, I document a set of experiments exploring in which ways the benefits of Concept Maps transfer to the domain of Computer Science education, and also what limitations arise in testing Concept Maps in an abstract domain like Computer Science

AUTHOR KEYWORDS

Concept Maps; Introductory Computer Science Education

ACM CLASSIFICATION KEYWORDS

K.3.2 Computer and Information Science Education

INTRODUCTION

A Concept Map (CM) is a graph representation of a subject's underlying ideas, where nodes are concepts at different hierarchical levels and edges are relationships between concepts. According to Novak and Cañas, Concept Mapping is "powerful for the facilitation of meaningful learning [because] it serves as a kind of template or scaffold to help to organize knowledge and to structure it, even though the structure must be built up piece by piece with small units of interacting concept and propositional frameworks."[6] Figure 1 is an example of a concept map, where the subject of the map is concept mapping itself. This meta concept map provides both an example of the method and its underlying structure.

Many other studies have gone on to analyze CM effectiveness. A meta-analysis of 19 studies on concept mapping by Horton, et al., found that concept mapping "had generally medium positive effects on students' achievement, and large positive effects on students'

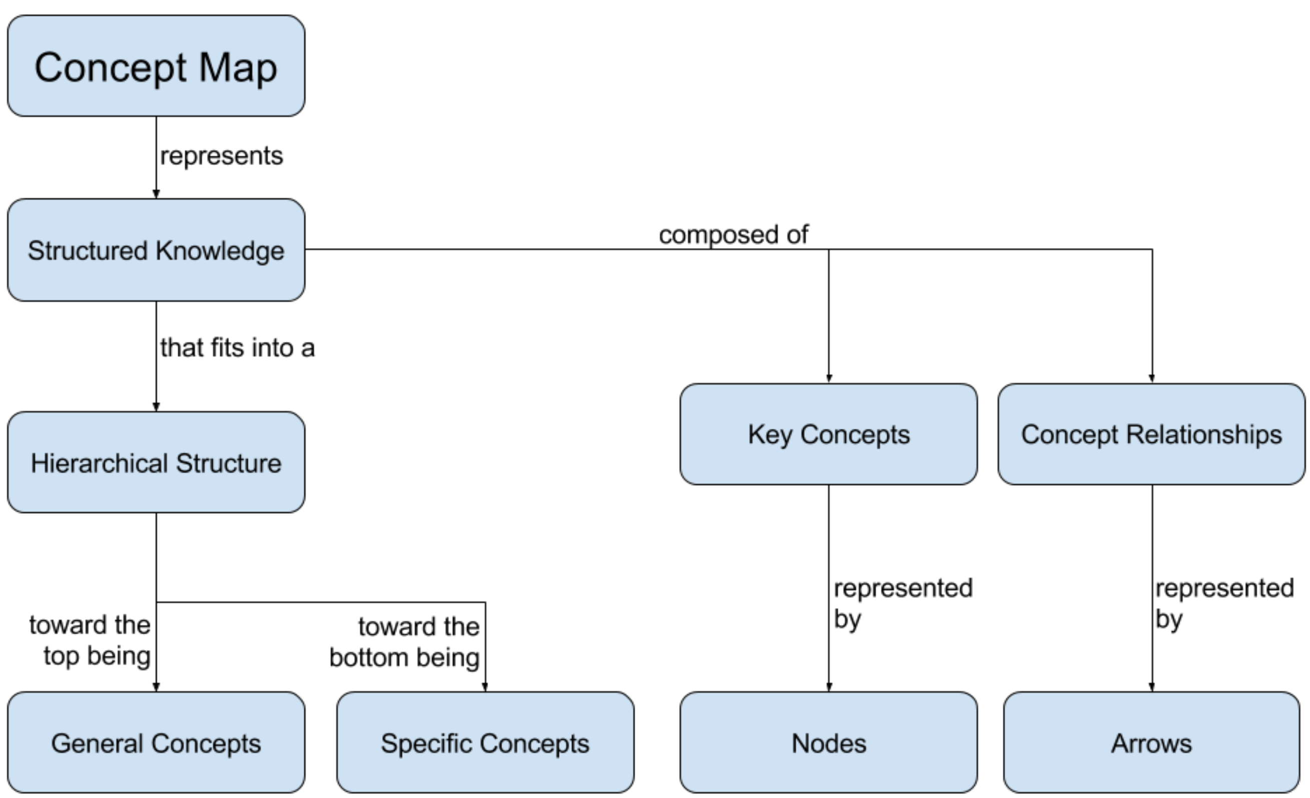

Figure 1: Concept Map Demonstration

attitudes."[3] Furthermore, many studies examining specific subjects have found the use of concept mapping to be beneficial, as well. Salleh and Ismail found that concept mapping "is one of the promising methods that can be effectively used to teach history in secondary students"[7]; Aein and Aliakbari found concept mapping to be effective in boosting the critical thinking of nursing students[1]; Alsomaidan found that English as a Second Language learners comprehended an English conversation better when accompanied with a concept map[2]; and Martínez, et al., found concept maps to improve engineering students' physics test scores by 21.77%[5].

Despite the copious amounts of research on CM effectiveness in all of the above subjects, I did not find any studies examining concept mapping on the field of Computer Science (CS). In 2008, Keppens and Hay offer some suggestions on applying concept mapping to programming education, but admits that the quantitative data for computer science specifically is missing.[4] Still in 2017, Wei and Yue make the following observation on concept maps: "Although widely researched and utilized in many other disciplines, the uses of CMs in Computer Science (CS) education have been relatively scarce."[8]

The research described in this paper aims to fill in this blind spot of research on the effectiveness of Concept Mapping in CS education. This is a worthwhile endeavor because most of the education domains in which concept mapping has been applied are far less abstract than CS, so making assumptions about CMs in CS from research in these more concrete domains could be problematic.

EXPERIMENT DESIGN

I conducted two experiments to explore CMs in CS, a depth-of-data in-person experiment, and a breadth-of-data online experiment.

Introduction to CS Experiment Design (EXP1)

The first experiment (EXP1) was part of an after school tutoring program at Global Technology Preparatory School in Harlem, NY. The students chose my after school tutoring program out of 6 total options through an organization called Citizen Schools. A number of different after school tutoring programs were offered along with my Introduction to Programming course, including a neuroscience course, an entrepreneurship course, and a photography course. While the students self-selected the tutoring course itself, I administered the tests to each student with their consent, and no personal or identifying information was retained. The programming environment used was Pencil Code.10

This experiment repeatedly tested the same group of 9 middle schoolers on introductory CS principles after receiving a lesson on the subject and a review sheet. The review sheet alternated from a CM to a non-hierarchical bulleted list of information. Each week a post-test was administered following the review materials. For an example of each review material type, see Figure 2.

The lesson structure for the five weeks of the experiment went as follows:

Week 1: Loops (Concept Map)
Week 2: User Input (No Concept Map)
Week 3: Functions (Concept Map)
Week 4: Debugging (No Concept Map)
Week 5: Recursion (Concept Map)

The post-test followed a similar format each week. Here are the questions from the first week's post-test:

What are the names of the three loops that were used in today's lesson?
Answer: for loop, while loop, forever loop

Which loop makes the turtle keep going until the program stops?
Answer: forever

List two shapes that are easier to draw using a loop.
Answer: triangle, square (other answers accepted)

What shape would the following program make?
```
pen red
for x in [1..8]
  fd 100
  rt 90
```
Answer: a square

To test the difference between test scores after a CM review with test scores after a non-CM review, I planned to use standard hypothesis testing using a t-test and the following null hypothesis:

H_{0_EXP1}: *Providing students with a Concept Map review for introductory CS lessons will have no difference over non-hierarchical, non-graphical review materials on students' test scores.*

The significance level to test H_{0_EXP1} will be $\alpha = 0.05$.

Traveling Salesman Problem Experiment Design (EXP2)

The second experiment (EXP2) was aimed at getting more data than was possible in EXP1. In order to achieve this larger-breadth of data collection, some sacrifices needed to be made in terms of how much depth could be collected.

1. To ensure participants would complete the experiment, it must be designed to take less than 10 minutes.
2. To allow the maximum number of participants, an easy to grasp problem must be used.

Given these constraints, a proper introductory CS lesson was out of the question. I therefore decided to use a common CS problem as my test, the well-known Traveling

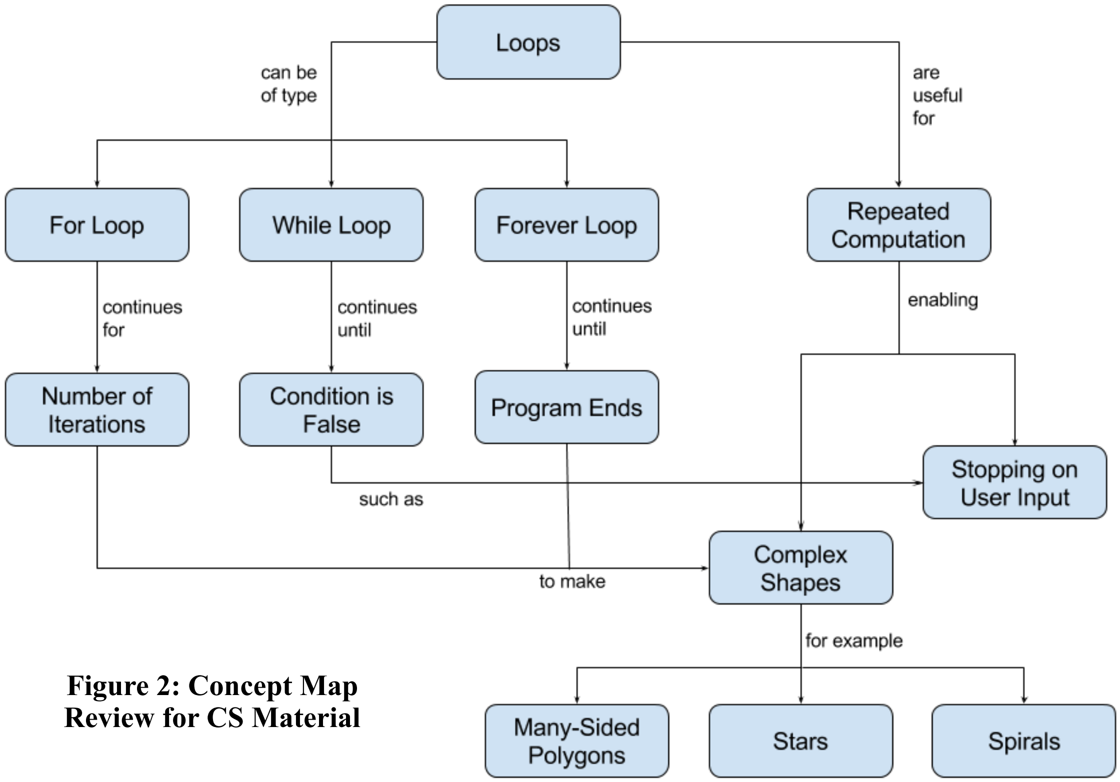

Figure 2: Concept Map Review for CS Material

- You can make interactive programs in PencilCode that let the user control the turtle. These interactions can be used in games.
- To control the turtle with the keyboard, you can use `keydown` and `pressed`
- `keydown` is used when you want the user to press a key once
- `pressed` can be used in a `forever` loop when you want the user to be able to hold down a key to repeat an effect
- To control the turtle with the mouse, you can use `lastmouse`, `click`, and `button`
- `lastmouse` can allow the user to hover the mouse to move the turtle
- `click` can allow the user to tell the turtle to move somewhere by clicking there
- `button` puts a button on the screen, which can have an effect when clicked
- All of these interactive components can be combined into games or other interactive programs that engage the user

Figure 3: Non-Hierarchical Bulleted List Review for CS Material

Salesman Problem (TSP).

The participants volunteered their participation in the experiment, and I only tracked completed experiments. I advertised the experiment on Facebook and Twitter. I also posted the experiment to the internal forum for the Online Master's of Computer Science course CS6460: Education Technology. In order to incentivize participants to complete the experiment, I offered a $25 Amazon gift card to those who completed the experiment and provided their email address. Additionally for CS6460 students, I offered participation credits as laid out by the course structure.
EXP2 consisted of the following questions:

1. *Please enter your level of computer experience.*
   - I am a complete novice.
   - I am good with computers, but I have no programming experience.
   - I have some programming experience.
   - I am a professional programmer or have a Computer Science degree or equivalent.

2. *Are you familiar with the Traveling Salesman Problem?*
   - Never heard of it.
   - I have heard of it but don't remember much.
   - Yes, I am familiar with it.
   - I consider myself an expert on it.

3. *What age are you?*
   - 15 or younger
   - 16 - 22
   - 23 - 29
   - 30 - 39
   - 40 - 49
   - 50 or older
   - I prefer not to say

4. *What is your gender?*
   - Female
   - Male
   - Non-binary
   - Other
   - I prefer not to say

5. *This experiment consists of 2 rounds. In each round, you must connect all of the points, visiting each point exactly once and returning to the starting point, while trying to minimize travel distance of the overall tour.*

   Round 1
   Solve the Traveling Salesman Problem by clicking here. It will open a new window, so don't worry about your progress being lost here.

6. *Strategy Review*
   Now, please review some characteristics of and strategies for the Traveling Salesman Problem by clicking here.

7. *Round 2*
   Solve the Traveling Salesman Problem again by clicking here.

8. *Do you think your ability to solve the TSP improved from Round 1 to Round 2?*
   - No, I think I got worse.
   - My skill was the same in both rounds.
   - I improved slightly in Round 2.
   - I was much better in Round 2.

9. *How big of an impact did the strategy review between rounds affect the change in your performance from Round 1 to Round 2?*
   - No impact
   - Some impact
   - Great impact
   - I don't know

Round 1 and Round 2 consist of an interactive TSP instance. Each respective round has the same configuration of points that all other participants see. Figure 4 shows the TSP interface that was used.

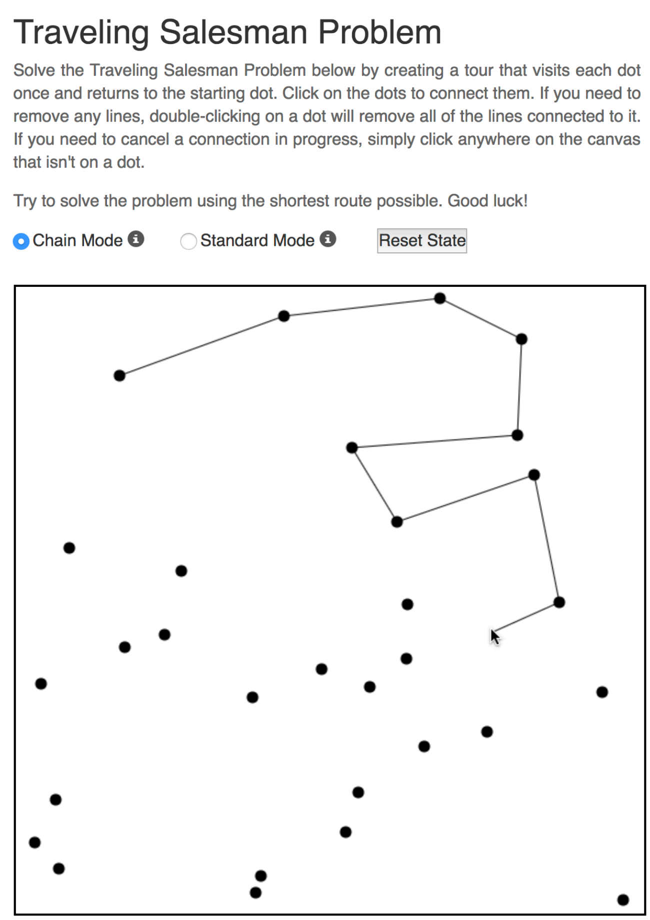

Figure 4: TSP Interface

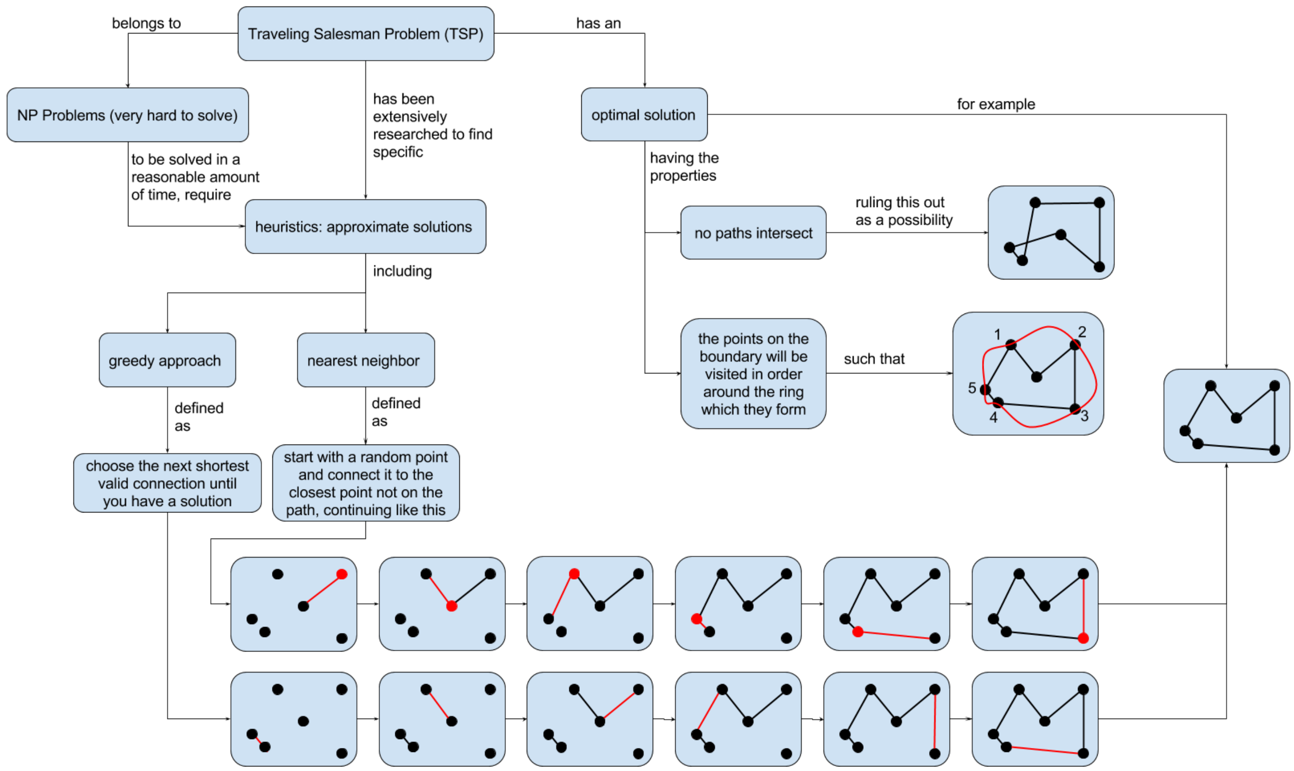

Figure 5: Concept Map Review for TSP

- The Traveling Salesman Problem (TSP) belongs to the class of problems called NP Problems.
- NP Problems require heuristics to be solved in a reasonable amount of time.
- TSP has been extensively researched to find specific heuristics for approximate solutions.
- One heuristic for finding a TSP solution is called the greedy approach.
- In the greedy approach, choose the next shortest valid connection until you have a solution.
- Another heuristic for finding a TSP solution is called nearest neighbor.
- In nearest neighbor, start with a random point and connect it to the closest point not on the path, continuing like this.
- TSP does have an optimal solution.
- One property of the optimal solution is that no paths will intersect.
- Another property of the optimal solution is that the points on the boundary will be visited in order around the ring which they form.

Figure 6: Non-Hierarchical Bulleted List Review for TSP

The difference in the control and the experimental groups was in the "Strategy Review" step, where the control group was shown a non-hierarchical bulleted list of TSP properties and strategies, and the experimental group was shown the same information but in a CM. Each review can be seen in Figure 3.

For EXP1 our null hypothesis definition relied on measuring student improvement on different weeks. This was a straightforward statistical analysis of the students' post-test scores. However, before defining the null hypothesis for the TSP, we need to define an improvement metric, which should measure how much better participants did in Round 2 than they did in Round 1.

$$TSP_{improvement} = \frac{TSP\_tour\_length_1 / TSP\_optimal\_tour\_length_1}{TSP\_tour\_length_2 / TSP\_optimal\_tour\_length_2}$$

Then the null hypothesis can be as follows:

H_{0_EXP2}: *Providing participants with a Concept Map overview of TSP strategies will have no difference over a non-hierarchical, non-graphical overview on participants' ability to approximate an optimal solution to an instance of the TSP manually.*

I planned to use a similar technique as I did in EXP1 to verify EXP2's results. The main difference is the way the null hypothesis is worded. The significance test and the significance level to test H_{0_EXP2} will be the same as it was in testing H_{0_EXP1}.

EXPERIMENT RESULTS

EXP1 Results

Here are the raw, week-by-week scores for EXP1. Due to absences and some students not turning in their tests, there were an average of 6 responses per week. These scores are all out of 100. They were not shared with the students. This allowed me to keep a strict rubric to get a good distribution of scores without being discouraging to the young students.

Week 1: 28, 14, 86, 57, 57, 100, 71, 86
Week 2: 0, 0, 50, 70, 40, 20, 0
Week 3: 20, 60, 80, 100, 70, 10
Week 4: 0, 100, 67, 0, 100
Week 5: 20, 80, 100, 100, 80

The mean of the control group tests (weeks 2 and 4) is $\mu = 37.22$ with a sample standard deviation of $s = 39.56$. The mean of the experimental group tests (weeks 1, 3, and 5) is $\mu = 64.70$ with a sample standard deviation of $s = 30.42$.

Using these values to test H_{0_EXP1} yields a t-statistic of 2.18, which has a p-value of 0.037. This passes the significance threshold for H_{0_EXP1}, meaning that the null hypothesis can be rejected, and CMs do in fact have positive impact on students' understanding of CS material. This result is in line with the previous findings on CMs being an effective teaching tool to improve students' understanding. As I mentioned in the Introduction, the finding of this paper extends this conclusion into the abstract subject of Computer Science, where CMs had not been tested yet.

EXP2 Results

EXP2 had two groups, as explained in the Experiment Design section, but it also ran in two separate trials. The first trial gave participants an easy TSP instance in Round 1 (consisting of 10 nodes to connect), followed by a more challenging TSP instance (consisting of 30 nodes). The second trial used a TSP instance with 30 nodes in both Round 1 and Round 2 in order to see if keeping the difficulty constant had an impact on participants' perception of their overall performance improvement.

	Control	Experiment
Trial 1		
Mean	0.959	0.948
Standard Dev	0.083	0.069
Trial 2		
Mean	1.258	1.189
Standard Dev	0.223	0.119

Table 1: TSP Participant Improvement Scores

It's clear looking at the descriptive statistics in Table 1 that the data does not support rejecting H_{0_EXP2}, but for completeness, the p-value for Trial 1 is 0.62 and the p-value for Trial 2 is 0.93 (since the improvement actually went down for the experimental group).

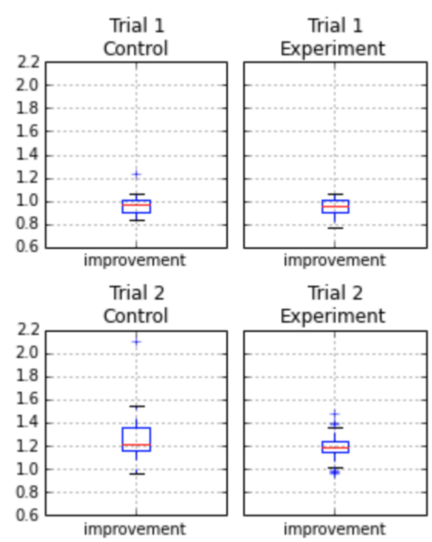

Figure 7: TSP Participant Improvement Scores Boxplot

Not being able to reject the null hypothesis can result from a number of issues:

1. There is no underlying effect.
2. There is not enough data.
3. There is a flaw in the experiment.

Issue 1 is unlikely, given the strong results around CMs in other subjects along with the results of EXP1 for CS specifically. Issue 2 is a possibility, but over both trials there were 124 participants in the TSP experiment, so it's unlikely that getting more participants for this experiment would change the results. This leaves issue 3, which is the most likely answer. Intuitively (albeit, only in hindsight), TSP is not a good test for CMs because people are already pretty good at the problem without any training, so participants can play around with the TSP interface from Figure 4.

Despite not being able to reject H_{0_EXP2}, there are still some interesting features of the data from EXP2 that are a good starting point for further research. First, my data was skewed to those with a strong CS background. There is some evidence that an improvement could be measured on a more novice participant-base. Second, the participants in EXP2 had the perception that the CM review was more helpful, which could still be a valuable finding in favor of CMs.

For the first potential avenue for further research, first let's look at Figure 8 to see the trend for improvement scores broken up by how participants answered question 1, level of CS experience. This figure shows the improvement scores across all experience levels: None, Some, Much, and Professional. As you can see, the improvement score medians decrease as CS experience goes up, indicating that participants with less CS experience might have more to learn regarding the TSP. As was mentioned earlier, the population I asked to participate in my experiment was skewed to have more CS experience (notice that in Trial 2 there were actually no participants with a CS experience level of None). This fact can be seen clearly in Figure 9, which is a histogram of all participants' CS experience levels. I computed a Chi-Square test on these four difference groups of participants, but there was no statistically significant difference between them. However, that could be due to the low number of novice participants, so I think an experiment targeted at CS novices would be worthwhile to see if the CM effect could be measured.

For the second potential avenue for further research, the median interpretation of the review step's effectiveness by the participants with little prior exposure to the TSP jumped up a level for both trials, as can be seen in Figure 10. In terms of means, in Trial 1 the TSP-unfamiliar participants interpreted the review effectiveness with a mean score of $\mu = 1.32$ in the control group and $\mu = 1.60$ in the experimental group, and there was an even greater jump in Trial 2, with the mean score being $\mu = 1.00$ for the control group and $\mu = 1.60$ for the experimental group.

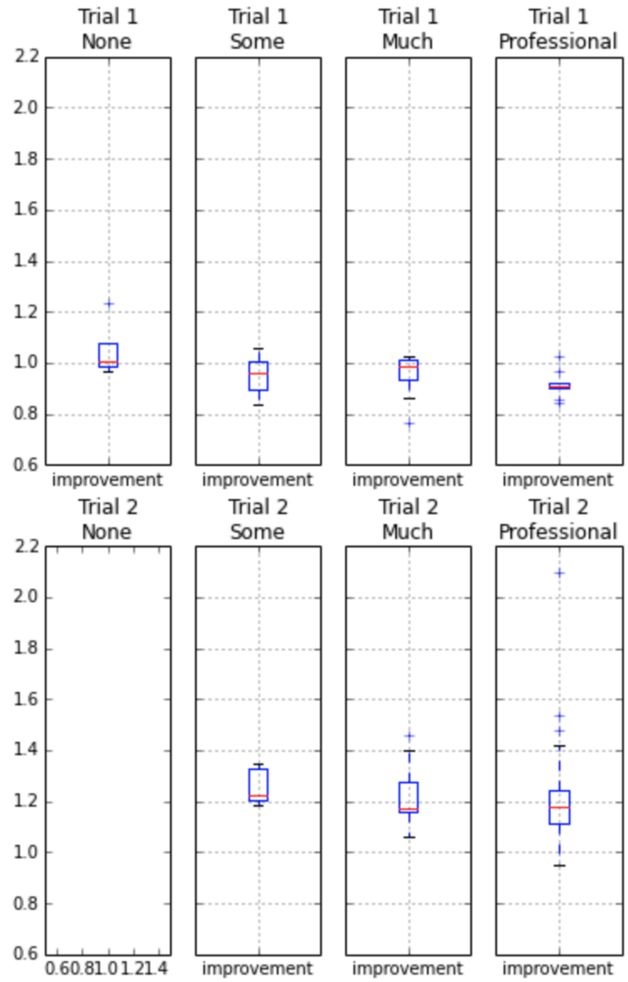

Figure 8: TSP Improvement Scores by CS Experience

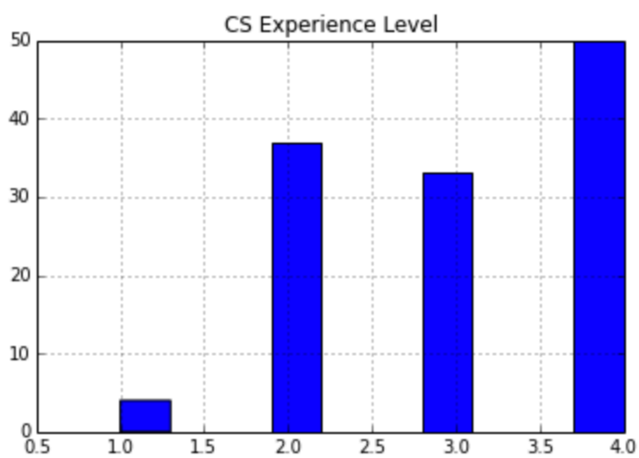

Figure 9: TSP Experiment CS Experience Levels Histogram

For a future experiment, targeting participants with little familiarity of the TSP problem and measuring how effective they find a CM review could also prove to be a fruitful research path.

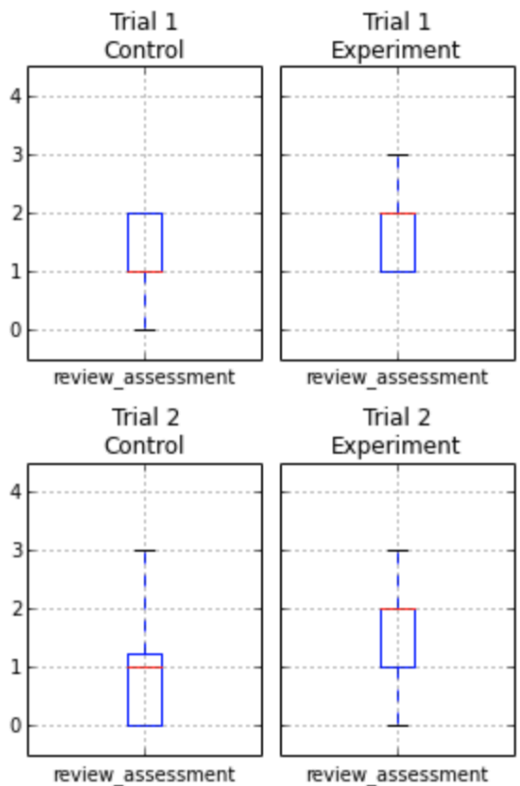

Figure 10: TSP-Unfamiliar Participants' Assessment of the Review Step

CONCLUSION

There has been much study into the effectiveness of CMs as an instructional tool in a wide range of subjects. This prior work implied a strong chance that this method would also be effective in the domain of CS education. However, it couldn't be taken for granted that this would be the case, since many of the earlier studies used subjects that were more concrete like physical science and engineering.

After running an experiment in a traditional school setting of the effectiveness of CMs in CS education, the data suggests that the CM method does transfer to the abstract subject of CS. The middle school students who had weekly tutoring on CS fundamentals showed a significantly significant improvement when the end-of-lesson review materials were a visual concept map over a non-hierarchical bulleted list.

This improved educational effect of CM reviews did not carry over to the specific problem of TSP, however. It is likely that when participants are faced with a spatial problem like TSP, they don't find a review of any kind necessary, and are instead able to fiddle around to find a solution as opposed to following a set of strategies and heuristics that need to be learned. In doing the TSP experiment, I did come across some interesting findings, such that CS novices did seem to improve more than CS veterans, and participants unfamiliar with the TSP altogether thought that the CM review was a more effective review material. These final "stretch" findings require further research to be scientifically sound, though.

ACKOWLEDGEMENTS

I would like to thank Sandy Davis for her unfaltering support and immensely helpful suggestions in the course of this research project.

REFERENCES

1. Aein, F., & Aliakbari, F. (2017). Effectiveness of concept mapping and traditional linear nursing care plans on critical thinking skills in clinical pediatric nursing course. Journal of Education and Health Promotion, 6, 13.

2. Alhomaidan, A. M. (2015). The Effectiveness of Concept Mapping on Learning: A Study in a Saudi College-Level Context. American Journal of Educational Research, 3(8), 1010-1014.

3. Horton, McConney, Gallow, Woods, Senn, and Hamelin, An investigation of the effectiveness of concept mapping as an instructional tool. Science Education 77(1): 95-1 11 (1993)

4. Keppens, Jeroen & Hay, David (2008) Concept map assessment for teaching computer programming Computer Science Education Vol. 18, Iss. 1, 2008

5. Martínez, et al (2013) The Effectiveness of Concept Maps in Teaching Physics Concepts Applied to Engineering Education: Experimental Comparison of the Amount of Learning Achieved With and Without Concept Maps J Sci Educ Technol (2013) 22:204–214 DOI 10.1007/s10956-012-9386-8

6. Novak, Joseph D. & Cañas, Alberto J., The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008

7. Salleh S.S., Ismail R. (2013) Effectiveness of Concept Map Approach in Teaching History Subject. In: Zaman H.B., Robinson P., Olivier P., Shih T.K., Velastin S. (eds) Advances in Visual Informatics. IVIC 2013. Lecture Notes in Computer Science, vol 8237. Springer, Cham

8. Wei Wei and Kwok-Bun Yue (2017) Concept Mapping In Computer Science Education. JCSC 32, 4 (April 2017)