Hills, T. (2010) Investigating mathematical search behavior and problem solving using network analyses. In Lesh, R., Galbraith, P., Blum, W., and Hurford, A. (Eds.) Modeling Students Mathematical Modeling Competencies. USA: Springer.

Investigating Mathematical Search Behavior and Problem Solving Using Network Analysis

Thomas Hills

University of Warwick

Contact information:
Thomas T. Hills
University of Warwick
CV4 7AL Coventry, UK
phone: +44 (0) 2476 575 527
fax: +44 (0) 2476 524 225
email: t.t.hills@warwick.ac.uk

Introduction

Some of the central tenets of constructivist learning follow from the idea that learning is a process of moving outwards from a central knowledge core. In *Adaptation and Intelligence* (1980), Piaget described the process of learning as very much similar to a process of biological adaptation, where new conceptual models come from progressive differentiations of pre-existing models. The closer the pre-existing model is to the objective, the more likely it is that the objective will be reached. In this way, learning, like evolution, acts via the modulation of existing material. Moreover, "the smaller the gap between the new and familiar becomes" the more likely it is that "novelty, instead of constituting an annoyance avoided by the subject, becomes a problem and invites searching" (Piaget, 1954). The basic pattern is one of a local-to-global search and growth process and variations and extensions of this theory of learning speak to ideas such as the zone of proximal development (Vygotsky, 1978), scaffolding (Wood et al., 1976; Pea, 2004), metacognition (Flavell, 1979; Schoenfeld, 1992), situated learning and the principles of transfer (Anderson et al., 1996; Bransford & Schwartz, 1999), and means-end analyses in general problem-solving (Newell and Simon, 1972). In the present work, the focus is on characterizing the "search" in a specific domain—mathematics—where the goal here is to introduce a methodology for characterizing trajectories of mathematical search in a known solution space. This includes both a novel task for exploring mathematical thinking (the math search task) and a cognitive hypothesis about the similarity between mathematical expressions (equation edit distance).

One way that learning trajectories have been characterized is to ask individuals at various stages in their understanding how they can explain a particular phenomenon (e.g., Smith et al., 1993). With this methodology, diSessa (1993) characterized what he called phenomenological primitives for physics understanding and points along the way towards deeper understandings. For example, the cognitive trajectory for the process of sucking water from a straw begins with thinking of sucking as pulling water, then sucking as creating a vacuum that pulls the water, followed by sucking as altering the balance of air pressure forces. Like the problem of sucking, for most problems we can present, the complete space of possible solutions is not known. However, if we have a complete solution space, we can move further towards characterizing these waypoints on the path to understanding. This would be equivalent to knowing all possible trajectories for understanding a given process, and further, using this space, we could begin to explore how different learning environments, involving different social, methodological, and theoretical approaches facilitate different paths to understanding and how these understandings are then more or less robust and transferable in different circumstances. Furthermore, we can construct hypotheses about how solutions are related to one another in cognition and how cognitive processes borrow from the old to construct the new. Different hypotheses will generate different cognitive landscapes, with different solutions being nearer one another. The landscapes can then be compared against observed data to determine which cognitive hypotheses are better representations of the cognitive landscape associated with a particular problem environment.

To investigate the cognitive landscape associated with basic mathematics, I created the math search task, involving search for solutions in a high-dimensional mathematical space (inspired by conversations with Walter Stroup). For example, provided with the problem "*=6*" and a list of five numbers "*1 2 3 4 5*", using each number no more than once, how many solutions can you find by combining two or more of the five numbers in basic a mathematical expression. "*2X3*", "*5+1*", "*4/2+5-1*", and "*3X2*" are all valid and unique solutions. In this problem, there are a total of 678 solutions. By allowing subjects to leave one number set—never to return—for another number set, we turn the task into a *cognitive foraging problem*, where patches represent a specific number set and individuals can travel between patches much like honeybees travel between flowers.

Cognitive foraging problems are a broad class of problems involving finding multiple solutions to a given problem (i.e., resources within patches) and travel between problems (i.e., travel time between patches). Similar tasks are the SCRABBLE task (Hills et al., 2007) and the verbal fluency task (Lezak, 1995). Cognitive foraging problems offer a unique way to study cognition because they expose multiple levels of cognitive processing. Bottom-up processes can be observed in terms of how solutions are dependent on the display of the problem. For example, in the SCRABBLE task, children are more likely to be influenced by the ordering of the SCRABBLE letters than are adults (Schneiderman et al., 1978). Cognitive foraging problems offer similar affordances for studying working memory and top down processes.

Tackling mathematics problems can involve similar processes of cognitive navigation, where movement in the solution space is potentially a process of moving outward from the familiar to the novel. Using the mathematical search task presented below, we can follow how individuals and groups navigate a mathematical solution space and begin to investigate underlying processes in new and interesting ways. To help facilitate this analysis, I also present a method for calculating the similarity between mathematical equations, which I present here as a cognitive hypothesis about how problem solving uses components of previous mathematical expressions to build new expressions. The remainder of this article focuses on a particular application of this methodology and focuses on a very basic question: are there general patterns, based on equation size and similarity with prior equations, for trajectories in a mathematical solution space?

Methods

Participants

77 English speaking undergraduate students at Indiana University participated in the experiment. All participants were recruited on a volunteer basis and there was no financial reward for their participation.

The Math Search Task

Participants were seated in front of a computer and asked to follow written instructions that appeared on the screen. Instructions guided participants through a series of tasks. The task that is relevant for the present work is the math search task. In the math search task participants saw the screen shown in Figure 1 and were asked to submit valid equations that would provide correct solutions. Order of precedence rules applied and divisions were calculated before multiplications. Instructions were as follows:

> "In this task, you will be presented with a number and you are to find as many mathematical solutions as you can. For example, you will see =*8* and you could enter *4X2* or *9-1* or *4X2X1*. Order of operations is important – division and multiplication take place before addition and subtraction. Also, the ordering of numbers is important. You

can enter *4X2* and *2X4* and get two solutions for *=8*. Each number can only be used once in each solution and numbers you have already used for this solution will turn to grey. To enter a solution, move the mouse over the number and click on it. Then move the mouse over the operation and click on it. When you are ready to enter the solution, click on 'enter'. To clear the solution, click on 'clear'. You should try to find as many solutions as you can for each set of numbers. When you can't find anymore, you can press 'NEXT' to go onto the next number set. When you press 'NEXT' you will have to wait for a short period (90 seconds) before the next number set is shown. To finish the session, you will need to find 60 solutions."

Participants were trained by asking them to submit 20 solutions using as many number sets as they needed. Then, later, participants were asked to submit 60 solutions, and the data I will look at here is from these 60 solutions.

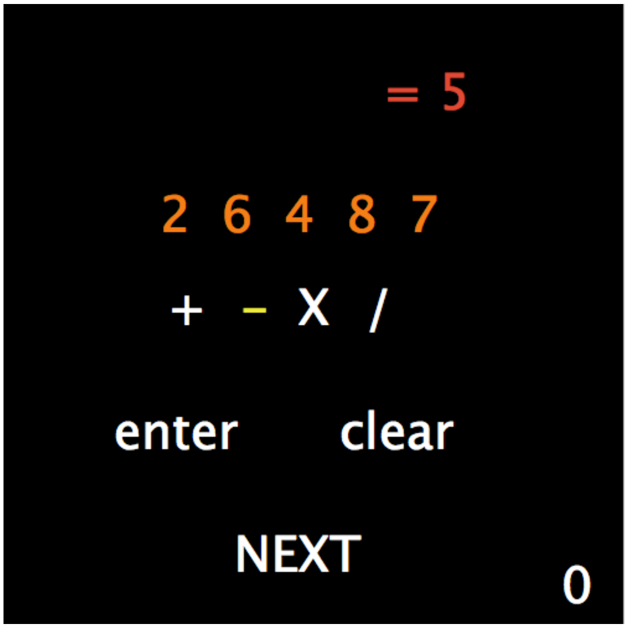

Figure 1. Computer screen where participants entered solutions to math problem.

Equation Edit Distance

There are a number of unique and interesting ways to characterize the solution sets that individuals move through when they submit solutions in the math search task. Because the number of unique solutions can be computed for a given problem, if we have a hypothesis for how individuals move between one solution and the next, then we can visualize the solution space and begin to speak quantitatively about how math learners move around in a math solution space. The method I use here is analogous to string edit distance (Sankoff & Kruskal, 1983), in which the distance between two strings is defined as the minimum number of insertions, deletions, or substitutions that are needed to move from one sequence to another. For example, with string edit distance, the distance between BOAT and BLOT is two, one insertion (the L) and one deletion (the A), or two substitutions (replace the O with L and the A with O).

Equation edit distance calculates the distance between two equations as the minimum number of insertions, deletions, or substitutions of nested mathematical units required to move from one equation to another. *Mathematical units* are the minimal divisions within a mathematical expression that, when permuted, still generate the same solution. For example, in the equation *4+5-2* there are 3 mathematical units-- *+4*, *+5*, and *-2*. Any permutation of these mathematical units still has the same solution = *7*. In the following equation, *4+3X1,* there are two additive units, *+4* and *+3X1,* but there are also two multiplicative units within the additive unit, *+3* and *+1*. Permutations within a multiplicative unit or among the additive units all generate the same solution (e.g., *1X3+4=7).*

Finding the distance between two equations involves finding the number of shared mathematical units. This is done by searching for shared units at each operational level, beginning with the operation level of highest precedence. Set one equation as the match and the other as the target. After a mathematical unit with the highest precedence operation is identified in the match equation, it is compared with all units in the target equation. If it is present in both, a shared unit is tallied, and both are simplified. If it is not present in both, no shared unit is tallied, and the unit is simplified in the match equation. Then the next unit of highest precedence is found in the match equation and searched for in the other equation. After all units at a given operational level are simplified in the match, then units at the same operational level are simplified in the target.

The highest possible distance between two equations is the number of mathematical units in the longest equation. To calculate the similarity, I take the total number of shared units and divide by the total number of units in the longest equation. Any permutation of another equation therefore has a similarity of 1 (e.g., *1X3+4* and *4+3X1*), whereas two completely novel expressions have a similarity of 0 (e.g., *1X3+4* and *5+1*). As with string edit distance, there are a number of ways to tune the subtleties of distance by allocating different costs to different kinds of alterations between match and target. In the analyses I perform here, I let *3X1* and *1X3* have a similarity of 1, but it is perfectly reasonable to assume there is some similarity cost associated with the transposition. It is not the focus of the work described here to determine these more subtle cognitive differences, but it is nonetheless an interesting focus for future work.

The Network

For each problem and number set posed in the math search task, there are a predetermined number of solutions. These were calculated by creating a program that ran through all possible equations that could be generated from the number set and evaluating these for their equivalency to the problem. For the problem 2, 1, 6, 3, 9 and = 3, there are 490 solutions. Between each pair of solutions, the equation edit distance generates a similarity measurement. For the above problem with 490 solutions, this creates a 490 X 490 matrix with the similarity between equations *i* and *j* at position *i,j* in the matrix. Letting each equation be a node in a network, with edge weights between nodes *i,j* set to the similarity between equations *i* and *j*, we can construct a visual description of the solution space (Figure 2).

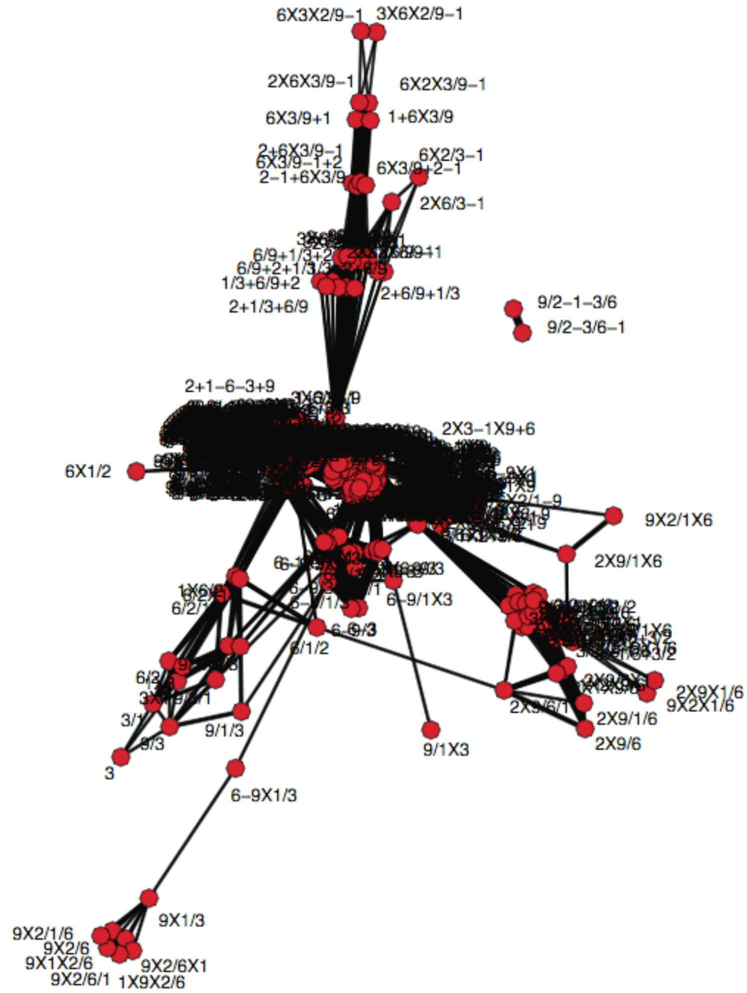

Figure 2. The solution similarity network computed using equation edit distance on the problem, *9 6 1 3 2   = 3*. There are approximately 490 possible solutions. Only edges with a similarity value greater than 0.35 are shown. Thicker edges represent greater similarity. Solutions with high numbers of similar permutations are clustered in the center.

Network structures are quite variable and can produce a wide variety of configurations based on the similarity properties of the underlying space. Figure 2 represents a structure with a fairly complicated layout with more sparsely connected equations on the periphery and more densely connected nodes towards the center. Figure 3 presents the solution similarity network for a smaller solution space, for the problem *8 7 9 1 4   =1*, with approximately 110 solutions.

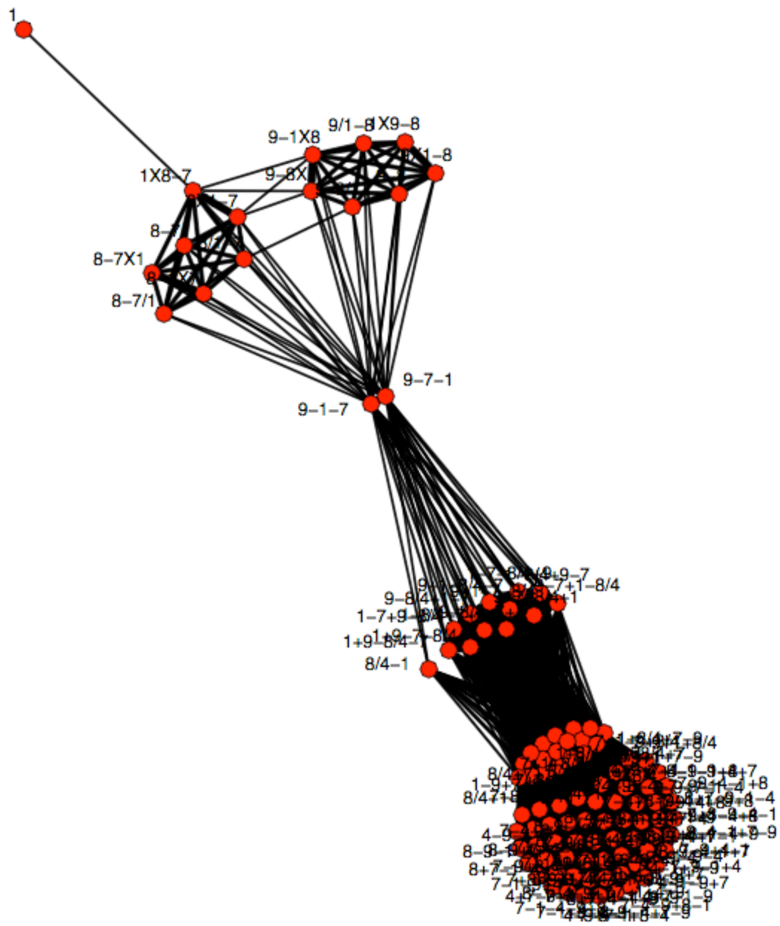

Figure 3. The solution similarity network computed using equation edit distance on the problem, *8 7 9 1 4 = 1*. There are 108 possible solutions. Only edges with a similarity value greater than 0.3 are shown. Thicker edges represent greater similarity. The ball of solutions in the lower right corner is largely made up of permutations of the mathematical units in the expression *7−1+8−9−4*.

Results

Many different kinds of questions can be asked using the math search task and the solution similarity network. In the present paper I want to focus on a very specific set of analyses to demonstrate how the math search task and solution networks might be used to address a specific problem. The problem is the following: is there a general pattern, based on size and similarity, for trajectories in a mathematical solution space? A simple null hypothesis is that solutions generated for a specific problem are randomly chosen from the space of all possible solutions. Equation edit distance allows us to develop an alternative kind of hypothesis, in which we propose that solutions are generated based on their similarity to prior solutions. This similarity hypothesis is generally consistent with ideas of constructivism, and more specifically with Piaget's statements about "the gap between the new and the familiar" made in the introduction, a still more rigorously stated in work focusing on cognitive search in general (Hills & Stroup, 2004; Hills, 2006).

An alternative hypothesis to similarity, one based simply on the concept of cognitive load, is that participants in the math search task will generate short solutions first and longer solutions later. Cognitive load refers to the natural constraints on working memory during problem solving and has been a useful tool for thinking about instructional design (Miller, 1956; Sweller, 1994). The basic idea is that is easier to generate a solution like *9-8*, than it is to generate a solution like *7−1+8−9−4*. But we can add a further element of rigor to this question by asking if participants in the task are providing the shortest possible solutions. In other words, if there are 20 solutions using one operation (e.g., *3X2* and *7-1*), do subjects submit all 20 solutions with one operation before moving on to solutions with two operations (e.g., *3X2X1*)?

Figure 3 shows how the number of operations in a solution is related to the entry number (the order in which a solution was submitted) for the problem 9 6 1 3 2 = 3, with approximately 490 solutions. There is one entry with zero operations, 3, and approximately 10 entries with 1 operation. Several things are evident from the figure. First, solution size does start out small and grow in size. Second, by around entry 20, participants have reached a plateau of approximately three operations. Finally, subjects appear to everywhere be submitting entries longer than the shortest possible entry length. This suggests that our initial interpretation of cognitive load theory as a simple rule for generating the shortest possible solution is incorrect. If we simply assume that participants compute the easiest set of solutions (based on length) for a given problem, we make incorrect inferences about the solutions they are likely to submit.

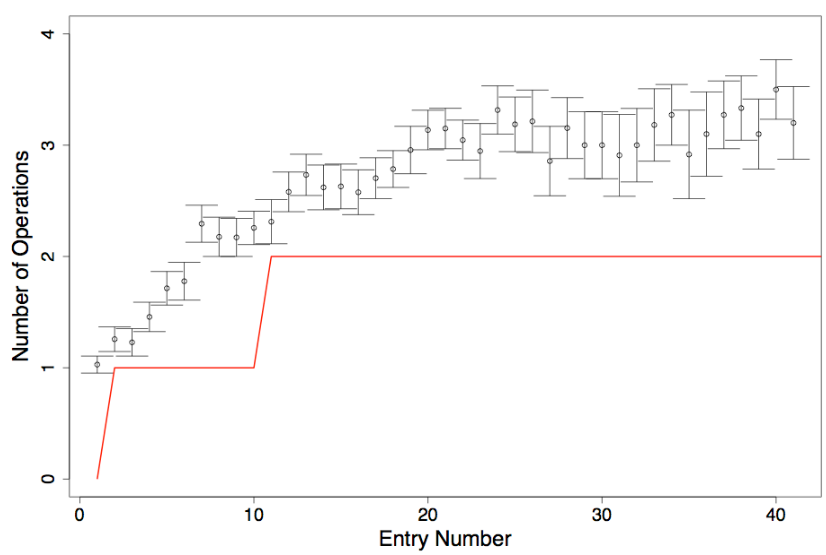

Figure 3. Number of operations in a solution versus the entry number (the order of solution entry) for the problem 9 6 1 3 2 = 3, with approximately 490 solutions. The observed number of operations are shown with unfilled circles. Error bars are standard error the mean. The solid line shows the entry size that would have been submitted if participants always chose the shortest possible equation size.

If similarity, as defined by equation edit distance, is playing a role in the types of solutions that are being generated, then it should take longer to generate novel solutions than it does to generate solutions of similarity one. Figure 4 shows the difference in the amount of time it takes to submit answers that are strict permutations of the previous answer (equation edit distance similarity of one) versus answers that are not strict permutations, for a variety of solution sizes. The results suggest that strict permutations take less time to generate than novel solutions. Moreover, it takes no more time to generate a permutation of the last equation if it had three operations than it does to find a novel solution with one operation.

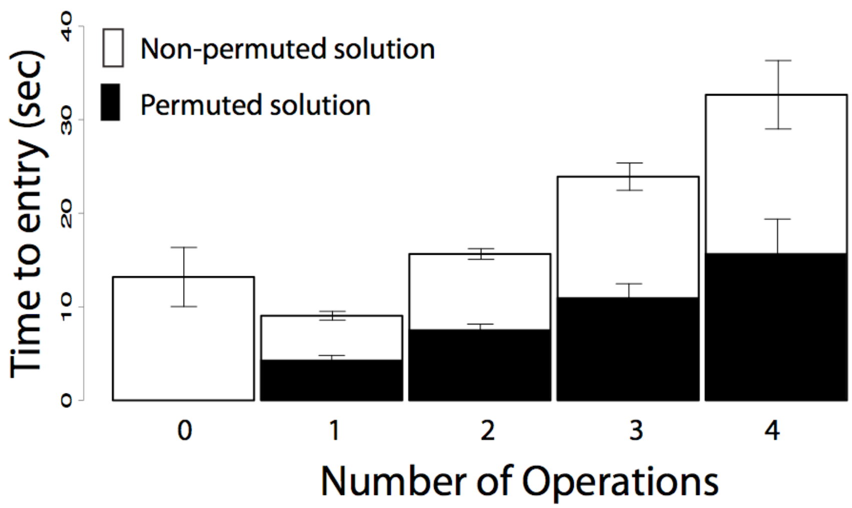

Figure 4. The time taken to submit an equation using a particular number of operations, separated by whether or not the equation submitted is a strict permutation of the last equation (similarity equal to one) or not. Error bars are standard error of the man. This figure uses all equation entries taken over all problem spaces.

Because longer equations also share high similarity with more equations than shorter equations (compare the clusters of high density with those of low density in Figures 2 and 3), it may be that participants are using equation similarity to leap-frog through the solution space towards longer equations. While shorter equations act as points of entry into the solution space, longer equations are the attractors in this space. When subjects find these equation clusters they tend to stay in them. If this were the case, then longer equations entries would, on average, be followed by equation entries of higher similarity than shorter equation entries. But that result alone is not sufficient to determine that subjects are actually using this similarity to guide there working memory processes. A random submission following a longer equation will have a higher similarity to that equation on average than a random submission following a shorter equation. Therefore, we must extend this question by asking if equations that follow longer entries are of higher similarity to the prior entry than would be expected if random submissions followed every entry. In other words, are submissions more similar to longer equations than would be expected by chance? The answer is that subsequent equation entries are more similar to longer equations than would be expected by chance (Figure 5). This turns out to be true for all equation sizes except sizes of length one (with zero operators). Zero operator equations tend to be followed by equations that share no similarity with that expression—meaning, they don't include the number that was in the prior equation. However, longer equations are all more similar than expected and as they get longer, they become increasingly similar to the previous equation. This result supports the hypothesis that participants are using similarity to leap-frog through the space and that they are doing so with increasing frequency as they move to longer equations.

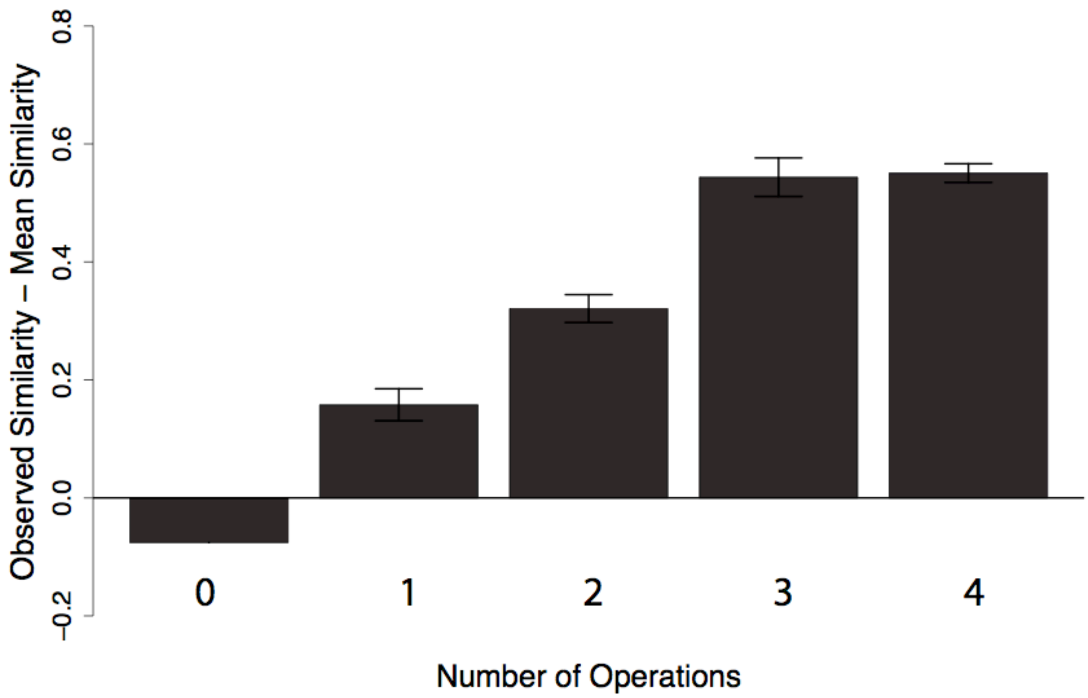

Figure 5. The difference between the observed similarity subsequent equation entries and that mean similarity of all equations to the prior equation, separated by the size of the prior equation (measured in number of operators). This is for the problem *9 6 1 3 2 = 3*. Error bars are standard error the mean. The results show that longer equations are followed by more similar equations than would be expected by chance.

Discussion

Search in math solution networks represents a novel way to study mathematical thinking and learning. Here, I have demonstrated how to pose problems with mathematical solution networks and also how one might address questions with these networks in a specific context. I also introduce the tool of equation edit distance and show how this can be used to structure and think about solutions in a mathematical space. The basic problem I posed here was what contributions do size and similarity play in the trajectories that our undergraduate students take in these mathematical solution spaces. The preliminary results that I provide suggest that size does play a role, but less so than one would think—students are everywhere submitting longer equations than they could be submitting if they were really looking for the simplest expressions. On the other hand, similarity to the last expression, based on equation edit distance, offers predictive power with respect to both time of entry and what the next expression will look like. With equations using two or more numbers, subsequent equations tend to borrow mathematical units from prior equations and they do more so than we would expect based on random solution generation. While I only show results for a few specific networks, the quantitative results generalize to other networks in the task. Overall, the results are consistent with ideas presented in prior work (Hills & Stroup, 2004; Hills, 2006) and follows along nicely with the inferences one may take from a constructivist view of working memory and problem solving. But again, the main goal here is to offer a new way of thinking about and studying mathematical processing, based on solution trajectories in math networks.

I believe there are a wide variety of uses for this tool in educational contexts, which range from evaluation purposes to research on the kinds of environments that influence trajectories in a mathematical space. We may want to know how a particular problem influences the trajectories that students are likely to take, and how the duration of time they spend in a particular network can influence the kinds of solutions they are likely to come up with. By analyzing the kinds of operations that students submit, we can gain an understanding of their familiarity with the utility of particular operators. For example, Figure 6 shows how the operations that students submit changed over the course entries for the *9 6 1 3 2 = 3* problem. Information presented in this way allows us to see that subtractions are more common than additions (solid thin lines), even though addition and subtraction operations are equally frequent over all possible equations (dotted thin lines).

The math search task can also be posed easily to groups of networked students simultaneously, which can offer new avenues of study for the effect of social interaction in mathematical thinking and learning. This work can then be easily juxtaposed with individual solution trajectories to determine how social interaction influences group trajectories in these space relative to when individual work on these problems alone. Similarly, one can use the math network task and equation edit distance to study how novices differ from experts in mathematical problem solving. Further applications may include different kinds of formative assessment or methods for mapping out groups of students approach and meet curricular goals.

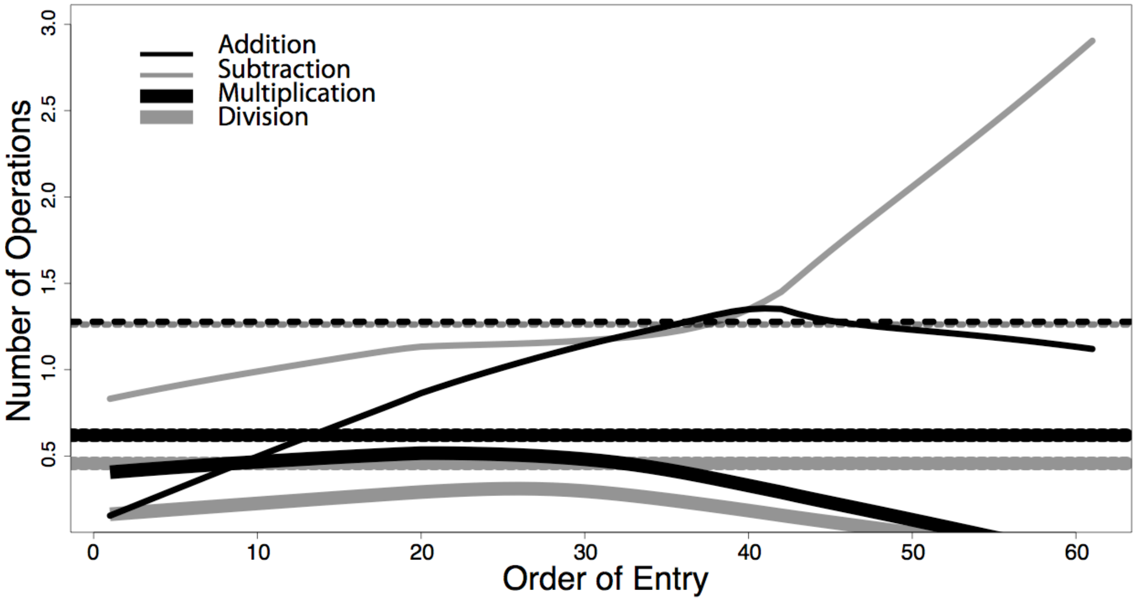

Figure 6. Number of operations by order of entry separated by type of operation. Thick lines represent the average number of operations submitted at a particular point in the order of entry. Dotted lines represent the average number of occurrences of an operation over all possible equation entries.

References

Anderson, J. R., Reder, L. M., & Simon, H. A. (1996). Situated learning and education. *Educational Researcher,* 25(4), 5-11.

Bransford, J. D. & Schwartz, D. L. (1999). Rethinking transfer: A simple proposal with multiple implications. In A. Iran-Nejad and P. D. Pearson (Eds.), *Review of Research in Education*, 24, 61-100. Washington, D.C.: American Educational Research Association.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and Instruction,* 10, 105-225.

Flavell, J. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, 34, 906-911.

Lezak M.D. (1995) *Neuropsychological assessment*. (3rd ed.) Oxford University Press, New York, Oxford.

Miller, G.A. (1956). "The magic number seven plus or minus two: some limits on our capacity to process information". *Psychological Review,* 63, 81-97.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *The Journal of the Learning Sciences,* 13(3), 423-451.

Sankoff, D., & Kruskall, J. (Eds.) *Time Warps, String Edits, and Macromolecules: TheTheory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA., 1983.

Schoenfeld, Alan H. (1992a). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In Douglas Grouws (Ed.), *Handbook for research on mathematics teaching and learning* (pp. 334-370). NewYork: Macmillan.

Schneiderman, M. H., Reber, A. S., & Hainline, L. (1978) Children's anagram solutions as an index of linguistic and cognitive development. *Child Development*, 49, 765-772.

Smith, J., diSessa, A., & Roschelle, J. (1993). Misconceptions reconceived: Aconstructivist analysis of knowledge in transition. *The Journal of the Learning Sciences*, 3 (2), 115-163.

Sweller, J. (1994). "Cognitive Load Theory, learning difficulty, and instructional design". *Learning and Instruction*, 4, 295-312.

Vygotsky, L.S. (1978). *Mind and society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wood, D., Bruner, J., Ross, G. (1976) The role of tutoring in problem solving. *Journal of Child Psychology & Psychiatry & Allied Disciplines,* 17, 89-102