Solving Problems by Drawing Solution Paths

Steven L. Tanimoto
Dept. of Computer Science and Engineering
Univ. of Washington
Seattle, WA 98195, USA
tanimoto@cs.washington.edu

***Abstract*—When the classical theory of problem solving is put into practice, a sequence of operator applications must be found that transforms an initial problem state into a goal state. This is equivalent to finding a path through the graph of possible problem states from an initial node to a goal node. The search is usually conducted by an algorithm or by a human through an iterative process of selecting and applying operators to states found so far. Here, I demonstrate examples of searching in problem spaces via interactive drawing. The demonstrations help to elucidate issues in the design of new problem-solving affordances that may enhance the abilities of human solvers to understand and solve problems. This opens up new ways to conceptualize the process of solving problems, and it suggests new ways to teach a form of computational thinking.**

Keywords—problem solving, problem space graph, visualization, drawing, state-space search, computational thinking, Towers of Hanoi, Missionaries and Cannibals.

I. Introduction

This paper presents demonstrations of a method of problem solving that, while rooted in the classical theory of problem solving, engages human users in solving through an interactive drawing process. This technique was presented as one of seven forms of "livesolving" in a recent paper given at the first International Conference on Live Coding [7]. The current presentation documents the demonstrations, which were not included in the livesolving paper. It also raises questions about how to incorporate this technique into the teaching of problem solving.

II. Prior Work

A. Classical Theory of Problem Solving

While designing computer algorithms to solve problems, early researchers in artificial intelligence developed a theory of problem solving [3, 4] that we use as a methodological foundation. A problem can be specified formally as a tuple $\pi = (\sigma_0, \Phi, \Gamma)$, where σ_0 is the initial state, Φ is a set of operators, and Γ is a set of goal states. Each operator $\varphi \in \Phi$ has a name, a precondition predicate ρ_i, and a state-transformation function δ. A solution for π consists of a sequence of operators φ_i such that $\rho_0(\sigma_0)$, $\delta_0(\sigma_0) = \sigma_1$, $\rho_1(\sigma_1)$, $\delta_1(\sigma_1) = \sigma_2$, ... , $\delta_{n-1}(\sigma_{n-1}) \in \Gamma$. That is, the first operator is applicable to the initial state and its state transformation function produces a new state. The next operator is applicable to that state, etc., and the state produced by the last operator in the sequence is one of the goal states.

B. Interactive Systems

Lalanne and Pu studied interactive problem solving in the context of problems specified by operators and lists of constraints. They showed that visualizations of state-space search can help human solvers to understand the space of possible solutions.

C. LiveSolving

In [7], I described seven forms of "livesolving." These techniques could be used in the context of a CoSolve-like system to help increase the degree of cognitive "flow" [1] of people solving problems. Most of the methods involve reducing the latency associated with the steps required in using systems like CoSolve. The technique of drawing solution paths is the first of those forms of livesolving.

III. Drawing Solution Paths

In its simplest form, drawing a solution path means drawing an overlay on a displayed graph, such that the initial node is connected to a goal node through a continuous sequence of graph edges. The demonstration here allows this path to be drawn with retraction (backtracking), and the software also

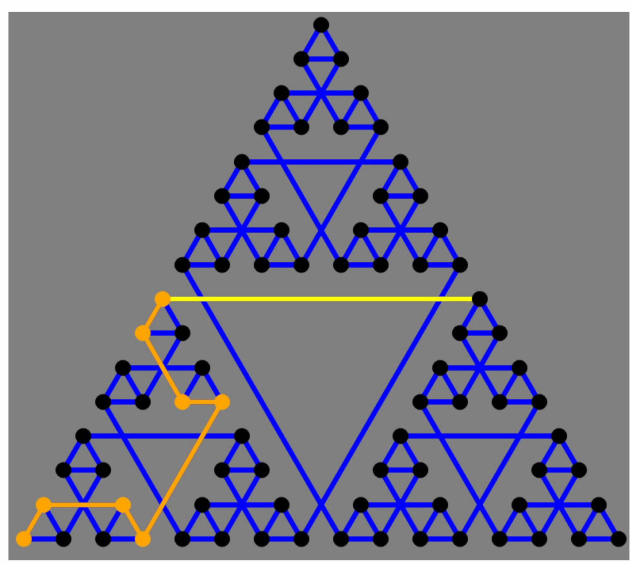

Figure 1. The problem-space graph associated with a 4-disk Towers-of-Hanoi problem. A path (in gold) is in progress and represents about half of a solution. The yellow edge is currently under consideration.

978-1-4673-7457-6/15/$31.00 ©2015 IEEE

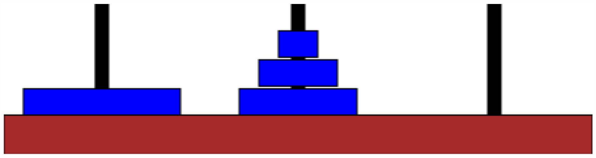

Figure 2. Current state of the Towers-of-Hanoi problem associated with the golden path shown in Fig. 1.

enforces the constraint that a drawn path be a continuous sequence of graph edges starting at the initial node.

Although the demonstrations use precomputed graphs, the technique of drawing solution paths can also work with dynamically computed graphs, where the relevant parts of the graph are drawn as the path grows. This permits solving in large and even infinite problem spaces.

In order to permit this method of problem solving, the problem to be solved must not only be formulated with the classical theory's approach, but a means must be provided for computing the screen coordinates of the nodes of the problem-space graph. Two methods are presented in [7] for such layouts. One involves defining a distance metric on states and then situating a set of landmark states. These then determine the locations of all nodes of the graph. The Towers-of-Hanoi example in Fig. 1 was laid out with this method. The other involves identifying two independent heuristic functions and using their values as x and y coordinates of nodes. The Missionaries-and-Cannibals examples in Figs. 3 and 4 use this method.

Complex problems typically lead to graph layouts that are dense. These graphs can be difficult to draw on top of. Additional interactive affordances are required in order to select particular nodes or node subsets from overlapping nodes or dense groups. Dynamic projections of higher-dimensional graph layouts may also be helpful, so that users can adjust their views of the graph as they draw their solution path. Intelligent prompting can also help by putting user preferences into action, dynamically showing and hiding nodes as their potential relevance to the path continuation changes.

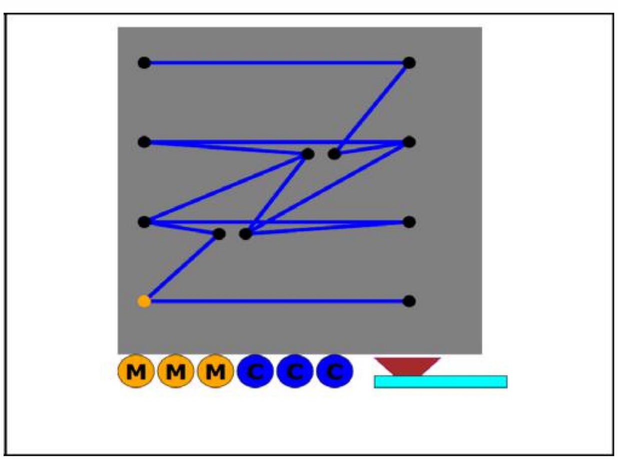

Figure 3. Problem- space graph for the Missionaries and Cannibals problem. The initial state is shown, and the solution path is empty.

IV. Educational Issues

This method of drawing solution paths has potential uses in teaching problem solving. First, the concept of solving by tracing is powerful. The solver gets a sense that the solution is close at hand because the graph is there or could be computed there by the system in response to solving actions. Second, the solver is empowered by the technology to solve by drawing, and although the path may not be simple, the sense of empowerment may enhance motivation. Finally, the difficulties of (a) finding paths in complex graphs, and (b) creating the graphs to begin with, are contextualized; our meta-solving design gives structure and definition to both.

The key question that arises in applying the method in education is this: How should teachers of problem solving best leverage these advantages? Also interesting is What are good ways to make this kind of solving collaborative?

Acknowledgment

Thanks to Alan Blackwell and Rob Thompson for discussions related to collaborative solving.

References

[1] Bederson, Benjamin. 2004. "Interfaces for Staying in the Flow." *Ubiquity*. ACM. (September).

[2] Fan, Sandra B., Robison, Tyler, and Tanimoto, Steven L., 2012. "CoSolve: A System for Engaging Users in Computer-supported Collaborative Problem Solving." *Proc. VL/HCC '12*., pp.205-212.

[3] Newell, Allen; and Simon, Herbert. 1972. *Human Problem Solving.* Englewood-Cliffs, NJ: Prentice-Hall.

[4] Nilsson, Nils J. 1971. *Problem Solving Methods in Artificial Intelligence*, New York: McGraw-Hill.

[5] Pearl, Judea. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison Wesley.

[6] Pu, Pearl, and Lalanne, Denis. 2002. "Design Visual Thinking Tools for Mixed Initiative Systems." *Proc. 7th Int'l Conf. on Intelligent User Interfaces* (IUI '02). ACM, New York, NY, USA, 119-126.

[7] Tanimoto, Steven L. 2015. "Livesolving: Enabling Collaborative Problem Solvers to Perform at Full Capacity." *Proc. Int'l Conf. on Live Coding.* Leeds, U.K.

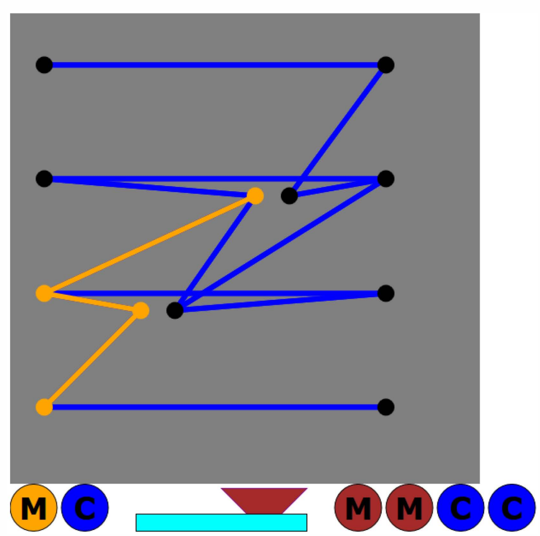

Figure 4. Partial solution path, leading to a promising intermediate state.