Models of Problem Solving LaLoCo, Fall 2013

Adrian Brasoveanu, Karl DeVries

[based on slides by Sharon Goldwater & Frank Keller]

Road map

Background

Motivation Early work Types of problems

Problem-solving strategies

Psychological Studies Selection without Search Goal-directed Selection Generalized Means-Ends Analysis

Discussion

Readings: [Cooper, 2002: Chapter 4]

Problem-solving

Many common tasks involve problem-solving of some sort:

- planning identifying what is needed to achieve some goal, usually under some constraints
- scheduling figuring out a "good" order in which to do things
- trouble-shooting figuring out what has gone wrong
- re-planning figuring out how to recover from what's gone wrong
- solving puzzles, playing games,

How do we solve these tasks?

Early psychological studies

- Early 1900s: Associationists explained problem-solving in terms of finding and strengthening stimulus-response patterns which would deliver solutions (or not): reproductive solutions.
- 1940s: Gestalt psychologists studied productive problem-solving, believed solution involved identifying the appropriate problem structure for a problem.
- Neither approach had much place for cognitive activity.
- Changed by work of Herbert Simon in 1970s.



(photo: http://en.wikipedia.org/wiki/Herbert_Simon)

Herbert A. Simon

Types of problems

Simon focussed on well-defined, knowledge-lean problems.

A problem is well-defined if

- all the relevant information
- all the problem-solver's options ("moves")
- the desired end state

can be specified completely and unambiguously.

A problem is knowledge-lean if

• the knowledge required to solve it can be specified completely and succinctly.

Types of problems

These are endpoints on two distinct dimensions:

	knowledge-lean	knowledge-rich
well-defined	Missionaries and cannibals Tower of Hanoi, chess	fixing computer problem diagnosing a patient
ill-defined	??	winning an election designing a fuel-efficient, large capacity aircraft

Example: Towers of Hanoi

- Starting point: all disks stacked on leftmost peg in order of size (largest on bottom); two other pegs empty.
- Legal moves: any move which transfers a single disk from one peg to another without placing it on top of a smaller disk.
- Goal: transfer all disks to the rightmost peg.

Using three disks:



Well-defined, knowledge-lean problems

Can be characterized by a state space

- Chess: state = configuration of pieces on the board
- Towers of Hanoi: state = configuration of disks and pegs
- Missionaries and cannibals: state = configuration of missionaries, cannibals, and boat.



http://www.aiai.ed.ac.uk/~gwickler/images/mc-init-state.png

Solving a problem involves figuring out how to transform the initial state into a desired state via a sequence of allowable moves.

Possible solution methods

Computers often solve similar problems by exploring the state space systematically, e.g., using depth-first or breadth-first search:



Could this be what people do?

Cognitive plausibility

DFS and BFS strategies don't match human behavior:

- People show greater difficulty at some points during solving than others not necessarily those with more choices.
- For complex tasks (e.g., chess), both methods can require very large memory capacity.
- People learn better strategies with experience: Novices may not find the best solution, but experts may outperform computers.

Anzai and Simon, 1979

Analyzed verbal protocols of one subject in four attempts at five-disk task (i.e., subject describes what she is doing and why).

- 1. Initially little sign of strategy, moving disks using simple constraints avoid backtracking, avoid moving same disk twice in a row.
- 2. By third attempt had a sophisticated recursive strategy, with sub-goals of moving disks of various sizes to various pegs.
- 3. Final attempt, strategy evolved further, with sub-goals involving moving pyramids of disks.

Developed adaptive production system to simulate acquisition and evolution of strategies.

Simple strategy: Selection Without Search

At each stage in the solution process:

- enumerate the possible moves;
- evaluate those moves with respect to local information;
- select the move with the highest evaluation;
- apply the selected move to the current state;
- if the goal state has not been achieved, repeat the process.

This approach can be applied to any well-specified problem (Newell and Simon, 1972).

Simple strategy: Selection Without Search

Modelling this in Cogent requires:

- a buffer to hold the current state;
- a buffer to hold the representation of operators;
- a process to manipulate buffer contents.

ToH: Selection without Search: Tue Jan 20 16:	29:00 2004*		
File Edit View Run Help Done	🏘 🐝 l い Ox 🕒 💙 📕		
Name: ToH: Selection without Search	Type: Box/Compound/Generic		
Brief Description: A simple model that is not goal directed. It selects moves	s based mainly on what is possible.		
Diagram Description Properties Message Matrix Messages			
Current P Select State	Possible P Operators		

Representing the Current State

Each disk may be represented as a term:

```
disk(Size, Peg, Position)
```

with the initial state of the five disk problem represented as:

disk(30,left,5)
disk(40,left,4)
disk(50,left,3)
disk(60,left,2)
disk(70,left,1)

(N.B. Position 1 at the bottom and 5 at the top of a 5-disk stack.

Representing Moves

Each move can be represented as a term:

```
move(Size,Peg1,Peg2)
```

and each possible operator and evaluated operator as other terms:

```
operator(Move, possible)
operator(Move, value(Value))
```

Operator Proposal

In the operator proposal phase, we propose moving the top-most disk on some peg to some other peg:

IF not operator (AnyMove, AnyState) is in Possible Operators
 top_disk_on_peg(Size, Peg1)
 other_peg(Peg1, Peg2)
THEN add operator (move (Size, Peg1, Peg2), possible) to
 Possible Operators

Some possible operators may violate task constraints.

Operator Evaluation

In the operator evaluation phase, we assign numerical evaluations to all possible operators:

IF operator (Move, possible) is in Possible Operators
 evaluate_operator (Move, Value)
THEN delete operator (Move, possible) from Possible Operator
 add operator (Move, value (Value)) to Possible Operators

Possible operators that violate task constraints should be assigned low evaluations by

```
evaluate_operator(Move,Value).
```

Other operators should receive high evaluations.

Evaluation Function

In addition, recall that Anzai and Simon (1979) found that on the subject's first attempt,

- she avoids backtracking and moving the same disk twice;
- she never moves the small disk back to the peg it was on two moves previously.

These too should be incorporated into the evaluation function. Consider how you would do this.

Operator Selection

In the operator selection phase, we select the rule with the highest value:

IF not operator (AnyMove, selected) is in Possible Operators
 operator (Move, value (X)) is in Possible Operators
 not operator (OtherMove, value (Y)) is in Possible Operators
 Y is greater than X

THEN add operator (Move, selected) to Possible Operators

Once an operator has been selected, others can be deleted:

IF exists operator (Move, selected) is in Possible Operators
 operator (AnyMove, value (V)) is in Possible Operators
THEN delete operator (AnyMove, value (V)) from
 Possible Operators

Operator Application

Applying an operator involves changing the current state:

disk(Size,FromPeg,FromPosition) is in Current State
 get_target_position(ToPeg,ToPosition)
THEN delete disk(Size,FromPeg,FromPosition) from

Current State

add disk (Size, ToPeg, ToPosition) to Current State clear Possible Operators

Properties

Properties of selection without search:

- selection of the first move is random;
- if the model selects the wrong first move, it can go off into an unproductive region of the problem space;
- the model will find a solution eventually, but it can be very inefficient.

Nevertheless, Anzai and Simon (1979) found that subjects appeared to use this strategy first.

Goal Directed Selection

Strategy: set intermediate goals (aka sub-goals) and move disks to achieve them.

- subgoal: move the largest out-of-place disk to the middle peg.
- maintain a stack on which further subgoals can be pushed, in case initial subgoal is not directly achievable.
- when completed, pop the top subgoal from the stack.

Revised Diagram

For goal-directed search, **Possible Operators** must be a stack buffer (**Goal Stack**).

Simple MEA Strategy: Tue Jan 20 16:41:51 2	2004 🛛 🗖
File Edit View Run Help Done 🔲 🕼 🎒 🏄	🖓 🗛 🖕 💙 💌 🔳
Name: Simple MEA Strategy	Type: Box/Compound/Generic
Brief Description: A simple (task-specific) MEA model	
Diagram Description Properties Message Matrix Messages	
Current State Select Operators	Goal Stack
$\longrightarrow \bigcirc \bigcirc$	

Setting Primary Goals

 IF the goal stack is empty there is a difference between the current and goal states
 THEN find the biggest difference between the current and goal states (the largest disk that is out of place) set a goal to eliminate that difference (move that disk to its goal location)

Setting Subgoals and Making Moves

Setting Subgoals:

IF the current goal is not directly achievable THEN set a subgoal to achieve current goal's preconditions

Moving Disks and Popping Subgoals:

IF the current goal is directly achievable THEN achieve the current goal (move the disk to its goal location) pop the goal off the goal stack

Properties

Properties of goal-directed selection:

- selection of moves is no longer random;
- selection is guided by the goal of moving the largest disk that is in an incorrect position;
- if the goal is not directly achievable, it is recursively broken down into subgoals;
- efficient strategy that avoids unproductive regions of the search space.

Goal-directed selection seems to be used by experienced players. (Anzai and Simon (1979) provide evidence that this is learned.)

Generalized Means-Ends Analysis

The problem solving strategy in the previous model is known as means-ends analysis (MEA).

In general, MEA involves:

- locating the largest difference between current and
- goal state, and selecting an operator to eliminate this difference.

Using MEA on a specific problem requires:

- identifying an appropriate distance measure for differences;
- identifying operators that can eliminate differences.

Most people seem to have access to this general strategy.

Switching to MEA

Why didn't the subject use MEA from the outset?

- She may have assumed a simpler solution strategy (selection without search) was sufficient.
- She may have lacked the knowledge of the problem space needed to perform MEA (operators and differences that they can be used to eliminate).

Hypothesis: During her first attempt, the subject acquired an understanding of how to decompose the problem into subgoals.

Evidence: Explicit mention of subgoals became more common as she gained experience with the task.

Learning to Solve

At least three types of learning were seen in the subject:

- switching from Selection without Search to Goal-directed strategy;
- changing the goal type from moving disks to moving pyramids;
- chunking subtasks (treating movement of top 3 disks as a single move).

How could these be modeled?

Summary

- Tower of Hanoi case study shows how people's problem-solving strategies change over time. Strategies include:
 - selection without search: enumerate all solutions, select the best one;
 - goal-directed selection: decompose the problem into subgoals and solve those;
 - generalized means-ends analysis: find the largest difference between the current state and the goal state and select an operator to eliminate it.
- The model posed by Anzai and Simon (1979) captures individual stages well, but provides a less satisfactory explanation of transitions between stages.

Problem decomposition strategy

The Towers of Hanoi problem can be decomposed into a sequence of sub-problems

- 1. move the largest disk to the right peg;
- 2. move intermediate-sized disk to the right peg;
- 3. move the smallest disk to the right peg.

which need to be solved in order:

 Achieve a state in which the largest disk can be moved to the right peg in a single move (i.e., no other disks on it, no disks on right peg).

Now move two-disk tower from left to middle peg: easier version of initial problem; the same principles used to solve it.

Simon, 1975

Simon, 1975 analyzed possible solution strategies and identified four classes of strategy:

- problem decomposition strategy (see above);
- two simpler strategies that move disks (rather than towers), moves triggered by perceptual features of the changing state;
- a strategy of rote learning.

Strategies have different properties in terms of generalization to larger numbers of disks and processing requirements.

References I

Anzai, Y. and H.A. Simon (1979). "The theory of learning by doing". In: *Psychological Review* 86, pp. 124–140.
Newell, Alan and Herbert A. Simon (1972). *Human Problem Solving*.
Englewood Cliffs, NJ: Prentice-Hall.
Simon, H. A. (1975). "The Functional Equivalence of Problem Solving Skills".
In: *Cognitive Psychology* 7, pp. 268–288.