9.65 December 10, 2001 Problem Solving Handout

Outline:

Approaches to understanding problem solving:

1. Different styles of problem-solving

2. How and why people get stuck

3. Prescriptive approaches

4. Simulations of problem-solving behavior

Introduction: Try this problem (from W. Kohler). A circle divided into 4 sections. One radius is labelled "y". A rectangle is drawn in another section, in which a right triangle is embedded:

Given d, x, and y, how long is L?

Scientists and mathematicians have sometimes classified themselves as symbol-pushers versus visualizers. Which are you?

Now, classify yourself into one of 4 categories:

	Visualizer	Symbol-pusher
Quick solution to circle problem		
Slower (or no) solution to the circle..		

1. Different styles of problem-solving:

The idea of classifying people as visualizers versus symbol-pushers is an example of this approach.

Wertheimer, Productive thinking, 1923:

The Altar window problem.

SLIDE

Wertheimer proposed that there are 3 styles of problem-solving:

a) Use deductive logic: standard algorithms; math.; logical syllogisms.

b) Retrieve the answer from memory: learn the steps in a proof. Improve performance by repetition, habit, frequency, recency, trial and error.

c) Use productive thinking: analogies, application of knowledge from another field, and especially "structural analysis."

Another example that distinguishes among these styles or approaches:

Area of a parallelogram:

SLIDE

These are examples in which a **visual** representation may help one to pick out the appropriate transformations to make the problem into a more familiar one: "insight" (or, in Reisberg, "illumination").

However, visualization can sometimes be misleading: as in folding a sheet of paper 1/100 inches thick over, repeatedly: how thick will it be after 50 folds?

What is "insight"? A one-step recognition that a new problem is equivalent to a familiar problem? If so, insight will depend on the knowledge you already have.

The 13 problem of Duncker: Why are all 6-place numbers of the form 267,267 or 591,591, etc., all divisible by 13? (8% of subjects in Duncker's experiment solved with no further information; hints led to a higher level of solutions)

Can’t have a feeling of "insight" unless the underlying principle is fully familiar and intuitively correct, once it comes to mind.

2. How people get stuck.

(a) Set or Einstellung: Luchin’s water jars.

(b) Duncker: "Functional fixedness." (See Reisberg.) Problems with real objects; tacks in a box made it less likely the subject would think of using the box as part of the solution.

Other problems in which an object with a standard use is not seen as useable in another role (like pliers as a weight for a pendulum).

More abstract problems can be regarded as cases of "functional fixedness":

Big and little Indians on a log. The little Indian is the big Indian’s son, but the big Indian isn’t the little Indian’s father.

Related issue: **transfer of training: recognition that another problem is analogous to one you already know the answer to**.

It is often difficult to realize that two problems have the same underlying structure, so that if you've solved one, you can easily solve the other. See the discussion in Reisberg, 459-463.

3. Prescriptive approaches

-make inferences & associations

-look for analogies

-figure out subgoals

-hill-climb

-if stuck, look for new routes

-if necessary, leave the problem and return later

Question of **incubation**: Does your brain continue to work on the problem while you're not consciously thinking about it? See Reisberg, pp. 480-83 for a possible answer.

4. Simulations of problem-solving:

Newell & Simon (1972): The General Problem Solver (GPS) [see Reisberg, pp. 448-453] They proposed the following.

A. There are just a few **general characteristics** of human problem solving:

-serial processing

-certain elementary processes: information transfer, elementary operations: they take of the order of 50-300 ms each

-Inputs and outputs are held in a small-capacity STM, measured in chunks that depend on experience

-storing in LTM takes about 5 s per chunk

--retrieval from LTM is fairly fast: e.g., 200 ms.

B. A task environment is represented as a **problem space** in which problem solving take place: a space of possible situations to be searched.

How do you set up the problem space? Task instructions and display; memory for similar or analogous tasks; metaprograms for setting up space-general task programs.

C. **The task structures the space**.

D. The **structure determines** the possible problem-solving program.

You start with an **initial state**, and you work toward a **goal state**, using **operators**. In setting up the **problem space**, you may have some **path constraints**. (See Reisberg, p. 448.]

At any one time you are **at a given node or knowledge state in the problem space**; the knowledge state is what you have in STM. You can either choose an operator to apply to take you to a new node--or can retreat to a different node (recover a different set of STM info--or look at one's external memory).

Heuristics include: **hill-climbing** (which doesn't always work, as you may have to go away from the goal at some point in the solution), **means-end analysis**, and **working backward from the goal**--see Reisberg, pp. 450-453.

Computer simulation consistent with protocol analysis of subjects solving problems like the DONALD + GERALD = ROBERT problem: Each letter is a different digit: e.g., all D's are one digit.

```
  D O N A L D
+ G E R A L D
-------------
  R O B E R T
```