Intelligent Agents
Human Problem Solving and Planning

Ute Schmid

Cognitive Systems, Applied Computer Science, Bamberg University

last change: June 15, 2010
Cognitive Perspective in Problem Solving

- Behaviorism: reinforcement of solutions found by trial and error
- Cognitivism: trial-and-error cannot explain the systematicity and productivity of thinking (finding new solutions)
- Most influential researchers: Newell and Simon (Human Problem Solving, 1972)
- Assumption: search in problem space
- Weak strategies (uninformed but complete)
  Strong strategies (informed but incomplete)
- Research question: What types of strong strategies are used by human problem solvers?
Greedy Strategies

- Humans often use greedy strategies for solving problems ("bounded rationality", Herbert Simon)
- The means-end (MEA) strategy which is the search algorithm used in the GPS (General problem solver) is such a greedy strategy
- Like the original STRIPS planner, MEA uses a linear strategy and therefore is not complete! (Sussman Anomaly)
Subjects have problems with the transformation from state (6) to (7). Here 2 and not only 1 passenger must be transported back to the left river bank. That is, there must be created a situation which is further removed from the goal state than the situation before.
Means-End-Analysis

**Transform**: Compare current state with goal state

- **IF** the current state fulfills the goal
- **THEN** stop and announce success
- **ELSE** Reduce the difference between the current state and the goal.

**Reduce**: Find operator which reduces the difference between current state and goal

- **IF** there is no such operator
- **THEN** stop and announce failure
- **ELSE** Apply the operator to the current state.

**Apply**: Apply an operator to the current state

- **IF** the operator is applicable to the current state
- **THEN** apply it and transform the resulting state into the goal.
- **ELSE** Reduce the difference between the current state and the application conditions of the operator.
MEA Example: Tower of Hanoi

Transform: initial state (discs 1, 2, 3 on peg A)
to goal state (discs 1, 2, 3 on peg C)

Reduce: 3 is not on C
Apply: Move 3 to C

Reduce: 3 is not free, because of 2
Apply: Remove 2

Reduce: 2 is not free, because of 1
Apply: Remove 1;
1 can be moved to C
2 is free
2 can be moved to B
3 is free

Reduce: 3 cannot be moved to C, because of 1
Apply: Remove 1;
1 can be moved to B
3 can be moved to C
MEA Example cont.

Transform: State (1 and 2 on B, 3 on C) to goal state
   Reduce: 2 is not on C
   Apply: Move 2 to C
       Reduce: 2 is not free because of 1
       Apply: Remove 1;
       1 can be moved to A
       2 is free
   2 can be moved to C
Transform: State (1 is on A, 2 and 3 on C) to goal
   Reduce: 1 is not on C
   Apply: Move 1 to C;
   1 can be moved to C
   1 is on 3
Transform: State (1, 2 and 3 on C) to goal
success
Cognitive Architectures

- **Cognitive Architecture**: “unified theory of cognition”
  - Explicit definition of basic mechanisms of information processing
  - Assumption that these mechanisms are constant over all domains (problem solving, language understanding, pattern recognition etc.)
  - Basic mechanisms: control of interaction with environment, representation of information in memory, strategy to select rules
  - Advantage: different models realized in the same architecture get comparable

- Alternative: special purpose cognitive models (such as SME for analogical reasoning, see below)
Prominent Architectures: The ACT-family (J.R. Anderson et al.), Soar (based on GPS)

ACT and Soar are production systems

ACT: long-term memory is divided in a declarative memory (“know what”, activation net) and a procedural memory (“know how”)

Example strategies for selecting rules: most specific first, most recently used, priority values (updated in dependence of success)
Production System

INTERPRETER

Match
Select
Apply

DATA
("working memory")

Production Rules
("long term memory")
Finding a good representation

- In human problem solving, there is an interaction between constructing a suitable representation and solving the problem.
- In AI systems, typically the representation needs to be fixed before problem solving (see Kaplan & Simon, 1990). Exceptions: approaches to solving proportional analogies using re-representation (Copycat, Hofstadter et al. 1995, PAN, O’Hara 1992, Indurkhya 1992)
- Empirical studies: Pltzner & van Lehn, 1997
- Examples: Mutilated checkerboard, nine-dots problem
Re-Representation

\begin{align*}
\begin{tikzpicture}
\node (A) at (0,0) {\text{GLUE}};
\node (B) at (1,0) {\text{GLUE}};
\node (C) at (2,0) {\text{GLUE}};
\node (D) at (0,-1) {\text{ROT}[p,60]};
\node (E) at (1,-1) {\text{ROT}[p,60]};
\node (F) at (2,-1) {\text{ROT}[p,60]};
\node (G) at (0,-2) {\text{ROT}[p,60]};
\node (H) at (1,-2) {\text{ROT}[p,60]};
\node (I) at (2,-2) {\text{ROT}[p,60]};
\draw (A) -- (B);
\draw (B) -- (C);
\draw (C) -- (A);
\draw (D) -- (E);
\draw (E) -- (F);
\draw (F) -- (D);
\draw (G) -- (H);
\draw (H) -- (I);
\draw (I) -- (G);
\end{tikzpicture}
\end{align*}
Context Effects

- Since human problem solving is typically guided by knowledge, search for a solution might be misled by preconceptions
- Gestalt-Theory: functional fixation (Duncker 1945)
  Examples: Candle, matches, and box with pushpins; pendulum problem
- A related phenomenon: set-effect (Luchins & Luchins, 1950)
  Water jug problems
A problem solving strategy alternative to heuristic search is using analogical reasoning.

- Retrieve a suitable source problem.
- Map the entities of the source with the entities of the target problem in a structure preserving way.
- “Carry-over” known parts of the source to target (possibly perform necessary adaptations)

- Gentner (1983)
- Cognitive Models: SME (Falkenhainer et al. 1989), LISA (Hummel & Holyoak, 1998)
- Empirical investigation of analogical transfer (Schmid et al., 1999)
Learning by Doing

- A problem solving system has no memory. Therefore, it might recalculate solutions which it already had achieved in another problem solving episode.
- The power law of learning (Anzai & Simon, 1979): learning curve, speed-up effect
- Humans acquire skills (procedural knowledge) when solving problem
- Cognitive Models, based on production systems: ACT (Anderson et al.), SOAR (Newell et al.): Declarative knowledge is “compiled” into rules
- But: these models do not cover strategy learning/control rule learning (see Schmid et al. 2000)
Hierarchical Planning

- Humans typically exploit knowledge about a domain when solving a problem.
- Often, it is known which sub-goals must be fulfilled to reach a given goal.
- This idea is realized in goal-driven production systems (such as ACT-R).
- This idea is also realized in hierarchical planning.
- Hierarchical planning is a special case of domain-dependent planning.
Besides using heuristics, problem solving can be guided by knowledge about the problem structure.

Problem decomposition: Dividing a problem in sub-problems

→ More complex production rules, goal-directed systems

Advantage: dealing with smaller sub-problems and generating the solution by composition (“divide and conquer”)

Representation: **AND-OR Trees**

standard tree: each arc which exits a node represents an alternative (“or”); extension: specially mark edges which lead to sub-trees which must be all fulfilled for the current node to be fulfilled (“and”)  

Special heuristic search algorithm for AND-OR trees: **A***
Example: MOVER

(Winston, 1992)

PUT-ON
(put block1 on block2)

GET-SPACE
(create space on goal block)

MAKE-SPACE
(move away blocks)

GRASP
(grasp a block)

CLEAR-TOP
(remove all blocks on top of a block)

GET-RID-OF
(put blocks on table)

MOVE
(move block)

UNGRASP
(let go of block)
AND-OR Tree for MOVER

GET-SPACE A B
MAKE-SPACE A B
GET-RID-OF D
PUT-ON D Table
GET-SPACE D Table
GRASP D
MOVE D Table
UNGRASP D

PUT-ON A B
"UND"
GRASP A
CLEAR-TOP A
GET-RID-OF C
PUT-ON C Table
GET-SPACE C Table
GRASP C
MOVE C Table
UNGRASP C

MOVE A B
UNGRASP A

GET-SPACE C Table
PUT-ON A B
PUT-ON C Table
GET-RID-OF C
Example: Route Finding Problems

- "OR"-node: X - F (go from X to F), X - F is primitive, if F can be reached from X in one step (there exists an applicable operation) primitive sub-problems are leafs in the tree
- "AND"-node: X - Z via Y (go from X to Z via Y) "constraint"
- Problem solving: extracting an (optimal) AND-Tree
- Using costs: Each leaf is associated with its cost, the costs are propagated upwards in the tree, the AND-tree with the lowest costs is returned
- Algorithm: AO* (Nilsson)
- In planning: hierarchical planning
Example cont.

Two of many possible decompositions.
Question: How many AI people does it take to change a lightbulb?

Answer: At least 67.

2nd part of the solution: **The Problem Space Group (5)**

- One to define the goal state
- One to define the operators
- One to describe the universal problem solver
- One to hack the production system
- One to indicate about how it is a model of human lightbulb-changing behavior

(“Artificial Intelligence”, Rich & Knight)