ARTIFICIAL INTELLIGENCE

Huỳnh Tấn Đạt
Email: htdat@cse.hcmut.edu.vn
Home Page: http://www.cse.hcmut.edu.vn/~htdat/
Ch2: Problems & Search

- State space search
- Search strategies
- Problem characteristics
- Design of search programs
State Space Search

- To solve a particular problem, we need:
  - Define the problem precisely:
    - Initial situation
    - Final situation
  - Analyze problem to have various possible techniques for solving the problem
  - Represent the task knowledge that is necessary to solve the problem
  - Choose the best problem-solving technique

- Solving Problem = Searching for a goal state
Playing Chess

- Each position can be described by an 8-by-8 array.
- Initial position is the game opening position.
- Goal position is any position in which the opponent does not have a legal move and his or her king is under attack.
- Legal moves can be described by a set of rules:
  - Left sides are matched against the current state.
  - Right sides describe the new resulting state.
State Space Search

- State space is a set of legal positions.
- Starting at the initial state
- Using the set of rules to move from one state to another
- Attempting to end up in a goal state.

Tic-tac-toe

A state  A state  \( \ldots \)  A state

O

X
State Space Search

Tic-tac-toe

A rule

Start state

Goal state
Water Jug Problem

“You are given two jugs, a 4-litre one and a 3-litre one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 litres of water into 4-litre jug.”

– State space search:

• A state: \((x, y)\)
  
  \[ x = 0, 1, 2, 3, \text{ or } 4 \quad y = 0, 1, 2, \text{ or } 3. \]

• Start state: \((0, 0)\).

• Goal state: \((2, n)\) for any \(n\), \(n = 0, 1, 2, 3\).

• Attempting to end up in a goal state by using rules.
### State Space Search

<table>
<thead>
<tr>
<th>No.</th>
<th>Left</th>
<th>Right</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(x, y) &amp; x &lt; 4</td>
<td>(4, y)</td>
<td>Fill the 4–litre jug</td>
</tr>
<tr>
<td>2</td>
<td>(x, y) &amp; y &lt; 3</td>
<td>(x, 3)</td>
<td>Fill the 3–litre jug</td>
</tr>
<tr>
<td>3</td>
<td>(x, y) &amp; x &gt; 0</td>
<td>(0, y)</td>
<td>Empty the 4–litre jug</td>
</tr>
<tr>
<td>4</td>
<td>(x, y) &amp; y &gt; 0</td>
<td>(x, 0)</td>
<td>Empty the 3–litre jug</td>
</tr>
<tr>
<td>5</td>
<td>(x, y) &amp; x + y &gt;= 4 &amp; y &gt; 0</td>
<td>(4, y - (4 - x))</td>
<td>Pour water from 3–litre jug into 4–litre jug until the 4–litre jug is full</td>
</tr>
<tr>
<td>6</td>
<td>(x, y) &amp; x + y &gt;= 3 &amp; x &gt; 0</td>
<td>(x - (3 - y), 3)</td>
<td>Pour water from 4–litre jug into 3–litre jug until the 3–litre jug is full</td>
</tr>
</tbody>
</table>
## State Space Search

<table>
<thead>
<tr>
<th>No.</th>
<th>Left</th>
<th>Right</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>((x, y) &amp; x + y \leq 4 &amp; y &gt; 0)</td>
<td>(\Rightarrow ((x + y), 0))</td>
<td>Pour all the water from 3–litre jug into 4–litre jug</td>
</tr>
<tr>
<td>8</td>
<td>((x, y) &amp; x + y \leq 3 &amp; x &gt; 0)</td>
<td>(\Rightarrow (0, (x + y)))</td>
<td>Pour all the water from 4–litre jug into 3–litre jug</td>
</tr>
</tbody>
</table>
State Space Search

1. current state = (0, 0)

2. Loop until reaching the goal state (2, 0)
   - Apply a rule whose left side matches the current state
   - Set the new current state to be the resulting state

(0 0) 2\textsuperscript{nd} rule
(0 3) 7\textsuperscript{th} rule
(3 0) 2\textsuperscript{nd} rule
(3 3) 7\textsuperscript{th} rule
(4 2) 3\textsuperscript{rd} rule
(0 2) 7\textsuperscript{th} rule
(2 0)
State Space Search

Water Jug Problem

- The role of the condition in the left side of a rule
  \[ \Rightarrow \text{restrict the application of the rule} \]
  \[ \Rightarrow \text{more efficient} \]
- Special-purpose rules to capture special-case knowledge that can be used at some stage in solving a problem
  - \((x, y) \& x < 4 \Rightarrow (4, y)\)
  - \((x, y) \& x > 0 \Rightarrow (0, y)\)
State Space Search

To provide a formal description of a problem:
- Define a state space that contains all the possible configurations of the relevant objects.
- Specify the initial states.
- Specify the goal states.
- Specify a set of rules:
  - What are unstated assumptions?
  - How general should the rules be?
  - How much knowledge for solutions should be in the rules?

Solving a problem = search in the state space by using the rules + control strategy
Search Strategies

- Requirements of a good search strategy:
  - It causes motion.
    - Otherwise, it will never lead to a solution.
  - It is systematic.
    - Otherwise, it may use more steps than necessary.
  - It is efficient.
    - Find a good, but not necessarily the best, answer.
Search Strategies

- **Uninformed search** (blind search)
  - Use only the information available in the problem definition.
  - Having no information about the number of steps from the current state to the goal.

- **Informed search** (heuristic search)
  - Use problem-specific knowledge beyond the problem definition.
  - More efficient than uninformed search.
Search Strategies: Blind Search
Search Strategies: Blind Search

- **Blind Search**
  - **Breadth-first search:** Expand all the nodes of one level first.
  - **Depth-first search:** Expand one of the nodes at the deepest level.
## Search Strategies: Blind Search

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Breadth-First</th>
<th>Depth-First</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$b^d$</td>
<td>$b^m$</td>
</tr>
<tr>
<td>Space</td>
<td>$b^d$</td>
<td>$b^m$</td>
</tr>
<tr>
<td>Optimal?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Complete?</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

$b$: branching factor  
$d$: solution depth  
$m$: maximum depth
Search Strategies: Heuristic Search

- Heuristic is a technique that improves the efficiency of a search process, possibly by sacrificing claims of completeness or optimality.

- Heuristic is for combinatorial explosion.

- Optimal solutions are rarely needed.

- Heuristic:
  - General-purpose heuristic:
    - Is for combinatorial problems.
  - Special-purpose heuristic:
    - Exploit domain-specific knowledge to solve particular problems.
Search Strategies: Heuristic Search

- General-purpose heuristic

The Travelling Salesman Problem
“A salesman has a list of cities, each of which he must visit exactly once. There are direct roads between each pair of cities on the list. Find the route the salesman should follow for the shortest possible round trip that both starts and finishes at any one of the cities.”
Search Strategies: Heuristic Search

Nearest neighbour heuristic:

1. Select a starting city.
2. Select the one closest to the current city.
3. Repeat step 2 until all cities have been visited.

\( O(n^2) \) vs. \( O(n!) \)
Search Strategies: Heuristic Search

- Special-purpose heuristic:
  - Exploit domain-specific knowledge to solve particular problems
  - Domain-specific, heuristic knowledge can be incorporated into a rule-based search procedure in 2 ways:
    - In the rules themselves
    - As a heuristic function: evaluate individual problem states and determine a desirable successor state
Search Strategies: Heuristic Search

- Special-purpose heuristic:
  - Example:
    - Chess: the advantage of our side over the opponent
    - 8 puzzle: The differences between current state and goal state
  - There is a trade-off between the cost of evaluating a heuristic function and the savings in search time that the function provides
Problem Characteristics

To choose an appropriate method for a particular problem:

- Is the problem decomposable?
- Can solution steps be ignored or undone?
- Is the universe predictable?
- Is a good solution absolute or relative?
- Is the solution a state or a path?
- What is the role of knowledge?
- Does the task require human-interaction?
Is the problem decomposable?

- Can the problem be broken down to smaller problems to be solved independently?

- Decomposable problem can be solved easily.

\[
\int (x^2 + 3x + \sin^2 x \cdot \cos^2 x) dx
\]

\[
\int x^2 dx \quad \int 3x dx \quad \int \sin^2 x \cdot \cos^2 x dx
\]

\[
\int (1 - \cos^2 x) \cos^2 x dx
\]

\[
\int \cos^2 x dx \quad -\int \cos^4 x dx
\]
Problem Characteristics

Is the problem decomposable?

Start

\[
\begin{array}{c}
C \\
A \\
\end{array}
\quad \begin{array}{c}
B \\
\end{array}
\]

Goal

\[
\begin{array}{c}
A \\
B \\
C \\
\end{array}
\]

Blocks World

- CLEAR(x) $\rightarrow$ ON(x, Table)
- CLEAR(x) and CLEAR(y) $\rightarrow$ ON(x, y)
Problem Characteristics

ON(B, C) and ON(A, B)

ON(B, C)          ON(A, B)

CLEAR(A)          ON(A, B)

A
C
B

A
B
C
Problem Characteristics

- Can solution steps be ignored or undone?

  Theorem Proving: A lemma that has been proved can be ignored for next steps.
  
  ⇒ Ignorable

  The 8-Puzzle: Moves can be undone and backtracked.
  
  ⇒ Recoverable

  Playing Chess: Moves cannot be retracted.
  
  ⇒ Irrecoverable
Problem Characteristics

- Can solution steps be ignored or undone?
  - **Ignorable problems** can be solved using a simple control structure that never backtracks.
  - **Recoverable problems** can be solved using backtracking.
  - **Irrecoverable problems** can be solved by recoverable style methods via planning. (Chapter 9)
Problem Characteristics

- Is the universe predictable?

  The 8-Puzzle: Every time we make a move, we know exactly what will happen.

  ➔ Certain outcome!

  Playing Bridge: We cannot know exactly where all the cards are or what the other players will do on their turns.

  ➔ Uncertain outcome!
Problem Characteristics

- Is the universe predictable?
  - For certain-outcome problems, planning can be used to generate a sequence of operators that is guaranteed to lead to a solution.
  - For uncertain-outcome problems, a sequence of generated operators can only have a good probability of leading to a solution.
  - Plan revision is made as the plan is carried out and the necessary feedback is provided.
Problem Characteristics

- Is a good solution absolute or relative?
  1. Marcus was a man
  2. Marcus was a Pompeian
  3. Marcus was born in 40 A.D
  4. All men are mortal.
  5. All Pompeians died when the volcano erupted in 79 A.D
  6. No mortal lives longer than 150 years.
  7. It is now 2008 A.D

Question: Is Marcus alive?
Problem Characteristics

- Is a good solution absolute or relative?

  8. Marcus is mortal (1, 4)
  9. Marcus’ age is 1964 years. (3, 7)
  10. Marcus is dead. (6, 8, 9)

  ________ OR ________

  11. All Pompeians are died in 79 AD
  12. Marcus is dead.

Different reasoning paths lead to the answer. It does not matter which path we follow.
Is a good solution absolute or relative?

- The Travelling Salesman Problem: We have to try all paths to find the shortest one.

- Any-path problems can be solved using heuristics that suggest good paths to explore.

- For best-path problems, much more exhaustive search will be performed.
Is the solution a state or a path?

- Finding a consistent interpretation: “The bank president ate a dish of pasta salad with the fork”.
  - “bank” refers to a financial situation or to a side of a river?
  - “dish” or “pasta salad” was eaten?
  - Does “pasta salad” contain pasta, as “dog food” does not contain “dog”?
  - Which part of the sentence does “with the fork” modify? What if “with vegetables” is there?

No record of the processing is necessary.
Is the solution a state or a path?

- The Water Jug Problem
  - The path that leads to the goal must be reported.

- A path-solution problem can be reformulated as a state-solution problem by describing a state as a partial path to a solution.

- The question is whether that is natural or not.
What is the role of knowledge?

- **Playing Chess**
  - Knowledge is important only to constrain the search for a solution.

- **Reading Newspaper**
  - Knowledge is required even to be able to recognize a solution.
Problem Characteristics

Does the task require human-interaction?

– Need to distinguish between 2 types of problems:
  
  • Solitary problem, in which there is no intermediate communication and no demand for an explanation of the reasoning process.
  
  • Conversational problem, in which intermediate communication is to provide either additional assistance to the computer or additional information to the user.
Problem Classification

- There is a variety of problem-solving methods, but there is no one single way of solving all problems.

- Not all new problems should be considered as totally new. Solutions of similar problems can be exploited.
Homework

- Analyze the following problems with respect to 7 problem characteristics:
  - Chess
  - Water jug
  - 8-puzzle
  - Travelling salesman
  - Tower of Hanoi
  - Cryptarithmetic

- Perform breadth-first and depth-first traversals on graphs
The missionaries and cannibals problem (Amarel, 1968)

Three missionaries and three cannibals are on side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side, without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.

- Draw a diagram of the complete state space
- Implement and solve the problem optimally using an appropriate search algorithm