Problem Characteristics

- In order to choose the most appropriate method(s) for a particular problem, must analyse the problem along several dimensions.
- Some keywords:
  - decomposition, undo steps, predictability, obviousness of good solution, amount of knowledge required.

Problem Characteristics

1. Is the problem decomposable?
   - Can it be broken into a set of (nearly) independent smaller or easier sub problems?
   - Can then solve the smaller sets directly, or further break them down.
   - One example is the blocks world.
     - Given on the next slide with two operators:
       - CLEAR(x) [x has nothing on it] → ON(x, Table) [put it on the table]
       - CLEAR(x) and CLEAR(y) → ON(x, y) [put x on y]

Problem Characteristics

Given:

\[
\begin{array}{c}
\text{ON}(C, A) \\
\text{ON}(B, C) \text{ and } \text{ON}(A, B) \\
\text{ON}(B, C) \\
\text{ON}(B, C) \\
\text{CLEAR}(A) \\
\text{ON}(A, B)
\end{array}
\]

Goal is:

\[
\begin{array}{c}
\text{ON}(C, A) \\
\text{ON}(B, C) \text{ and } \text{ON}(A, B) \\
\text{ON}(B, C) \\
\text{ON}(B, C) \\
\text{CLEAR}(A) \\
\text{ON}(A, B)
\end{array}
\]

Problem Characteristics

- States not achieved are underlined.
- The solution above shows that the two sub problems are not independent.
  - They must be considered together in order to arrive at a solution for the entire problem.

Problem Characteristics

2. Can the solution steps be ignored or undone?
   - Three types of problem can be considered.
     - Solving a theorem.
       - If we proceed to prove a lemma which turns out to be no use, we can easily backtrack because all of the initial information is still true and in memory.
       - Any rules that could have been applied at any stage, still can.
     - Moving any tile to solve the problem may or not take us towards a solution.
     - A dumb move can be undone using backtracking. It requires more effort to undo than the theorem example, but it is possible.
     - Have to keep track of what moves were made.
   - Chess.
     - Making a dumb move here cannot be backtracked or restarted, must make the best of a bad situation.
### Problem Characteristics

- **Three types of problem:**
  - Ignorable in which steps can be ignored.
  - Recoverable in which solution steps can be undone.
  - Irrecoverable in which solution steps cannot be undone.
- **The recoverability of a problem influences the complexity of the control structure used.**

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>7</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>8</th>
</tr>
</thead>
</table>

### Problem Characteristics

- **Ignorable are easy.**
- **Recoverable requires a more complicated strategy.**
  - Backtracking will be necessary to recover, so the control structure must use a push down stack to record decisions.
- **Irrecoverable can use some sort of planning.**
  - Examine several steps before one is actually taken.

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>9</th>
</tr>
</thead>
</table>

### Problem Characteristics

- **3. Is the problem universe predictable?**
  - You can plan the 8 puzzle. You know what will happen every time you move a square.
  - Therefore a control structure that allows backtracking will be necessary.
  - However, this may not be so in other situations.
  - **Playing cards:**
    - which card to play first. Don’t know the location of all other cards.

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>10</th>
</tr>
</thead>
</table>

### Problem Characteristics

- **Playing cards**
  - will investigate several plans and try to use probabilities to choose a plan that may lead to a good score.
- **This illustrates certain outcome and uncertain outcome problems.**
- **Certain outcome problems can be easily planned, not so with uncertain.**

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>11</th>
</tr>
</thead>
</table>

### Problem Characteristics

- **To solve uncertain outcome problems,**
  - need to allow for plan revision as an initial plan is carried out and feedback is provided.
- **The characteristics ignorable, recoverable and irrecoverable and certain and uncertain interact.**
  - Already stated that to solve an irrecoverable problem, plan an entire solution.
  - Only possible with certain outcome problems.

<table>
<thead>
<tr>
<th>March 04</th>
<th>Problem Characteristics</th>
<th>12</th>
</tr>
</thead>
</table>

### Problem Characteristics

- **So one of the hardest problems to solve will be irrecoverable uncertain outcome.**
- **Example**
  - Playing cards.
  - Helping a lawyer to decide how to defend a client against a murder charge.
Problem Characteristics

4. Is a good solution relative or absolute?
   – How many ways can we prove Marcus is dead?
     • Once you find one solution, why bother evaluating another to see if he is dead another way?
   – Travelling Salesman
     • Requires that we find the best solution, so when you find one, you can only verify if it is a good solution by finding the other possible outcomes.
     – Illustrates any path problems and best path problems.

5. Is the solution a state or a path to a state?
   – For natural language understanding, the interaction among the interpretations of the constituents of a sentence may cause ambiguity.
   – To solve the problem of finding the interpretation required, we need to produce only the interpretation itself, the workings are not necessary.

6. What is the role of knowledge?
   – However, for the jug problem, the final state of (2,0) is not sufficient. What is required is the path to the solution.
   – Playing chess. Even with unlimited computing power, the only knowledge required is the legal moves, and an appropriate search engine.
     • Additional knowledge will help, but is not essential.

7. Does the task require interaction with a person?
   – Can distinguish two types of problem here:
     – Solitary
       • In which the computer is given a problem description and produces an answer with no intermediate communication and no demand for an explanation of the reasoning process.
     – Conversational
       • In which there is intermediate communication between a person and the computer, either to provide additional assistance to the computer or to provide additional information to the user, or both.
Problem Characteristics

- Looking at these questions, it is clear that there are several broad classes of problems.
- These classes can be associated with generic control strategies appropriate to solving the problem.
  - Classification Problem - examine input and classify - medical diagnosis.
  - Propose and refine - design and planning problems.

Problem Characteristics

- There is no one single way of solving all problems.
- However, each new problem need not be considered in isolation.

Production System Characteristics

- Having looked at these problem types and bearing in mind that production systems are a good way to describe operations to be performed in search of a solution, there are two questions.
  - 1. Can production systems like problems be described by a set of characteristics that shed some light on how they can easily be implemented. (yes they can)

Production System Characteristics

- monotonic production systems - the application of a rule never prevents the later application of another rule which was an option at this time.
- Partially commutative systems - the application of a set of rules transforms state x into state y. Any permutation of those rules also transforms state x into state y.
- Commutative system is both monotonic and partially commutative.

Production System Characteristics

- 2. What relationships are there between problem types and the types of production systems best suited to solving the problems.
  - In theory, there is no relationship as any type of production system can solve any type of problem.
  - In practice however, there is a definite relationship between the kinds of problems and the systems that lend themselves naturally to describing those problems.

Production System Characteristics

<table>
<thead>
<tr>
<th>Partially</th>
<th>Monotonic</th>
<th>Partially</th>
<th>Nonmonotonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutative</td>
<td>Theorem Proving</td>
<td>Not Partially</td>
<td>Chemical synthesis</td>
</tr>
<tr>
<td></td>
<td>Robot navigation</td>
<td>Commutative</td>
<td>Bridge (cards)</td>
</tr>
</tbody>
</table>

- Partially comm. Monotonic
  - useful for ignorable problems
  - can be implemented with backtracking to optimise the search
Production System Characteristics

- Because the database does not need to be restored, the decisions made and the positions of changes in the search process are not recorded.

  - Partially comm. Nonmonotonic
    - useful for problems in which change occurs but can be reversed and the order of operation is not important.
    - Common in physical manipulation problems.

Production System Characteristics

- Non partially comm.
  - Useful in situations where irreversible change occurs

- Partially comm systems can produce the same individual states during a search.
- Non Partially comm systems are less likely to produce the same node many times.

Production System Characteristics

- Issues in design of search programs
  - the search process must find a path through the tree that connects an initial state with a goal state.
  - In theory
    - the tree to be searched could be constructed in its entirety from the rules that define allowable moves in the state space.

Production System Characteristics

- In Practice
  - the theoretical tree is too large.
  - Rather than building the tree explicitly and then building it, most systems represent the tree implicitly in the rules and generate explicitly only those parts which are being explored.
  - There are many search types, but some important issues arise in all, no matter which is used.

Production System Characteristics

- the direction in which to conduct the search
  - forward or backward reasoning.
- How to select the applicable rules (matching).
  - Critical to be efficient as a lot of time is spent matching.
- How to represent each node
  - knowledge representation problem and the frame problem