Revision + Broad Picture

Informatics 2D: Reasoning and Agents
The basics of search

Adapted from slides provided by Dr Petros Papapanagiotou
Revision
Intelligent Agents and their Environments

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents
- Learning agents

- Properties of environments
  - Partially vs. fully observable
  - Deterministic vs. stochastic
  - Episodic vs. sequential
  - Static vs. dynamic
  - Discrete vs. continuous
  - Single vs. multi-agent
Problem Solving by Searching

- Problem formulation usually requires abstracting away real-world details to define a state space that can feasibly be explored.

- Variety of uninformed search strategies:
  - breadth-first, depth-first, iterative deepening

- Iterative deepening search uses only linear space and not much more time than other uninformed algorithms.
Evaluating search strategies

Completeness: does it always find a solution if one exists?

Time complexity: number of nodes generated / expanded

Space complexity: maximum number of nodes in memory

Optimality: does it always find a least-cost solution?

Time and space complexity are measured in terms of:

- $b$: maximum branching factor of the search tree
- $d$: depth of the least-cost solution
- $m$: maximum depth of the state space (may be $\infty$)
### Summary of Base Algorithms

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Breadth-First</th>
<th>Uniform-Cost</th>
<th>Depth-First</th>
<th>Depth-Limited</th>
<th>Iterative Deepening</th>
<th>Bidirectional (if applicable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete?</td>
<td>Yes</td>
<td>Yes&lt;sup&gt;a,b&lt;/sup&gt;</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes&lt;sup&gt;a,d&lt;/sup&gt;</td>
</tr>
<tr>
<td>Time</td>
<td>O(b&lt;sup&gt;d&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;d&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;m&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;l&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;d&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;d/2&lt;/sup&gt;)</td>
</tr>
<tr>
<td>Space</td>
<td>O(b&lt;sup&gt;d&lt;/sup&gt;)</td>
<td>O(b&lt;sup&gt;1+\frac{C^*}{\epsilon}&lt;/sup&gt;)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes&lt;sup&gt;c,d&lt;/sup&gt;</td>
</tr>
<tr>
<td>Optimal?</td>
<td>Yes</td>
<td>Yes&lt;sup&gt;a,b&lt;/sup&gt;</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes&lt;sup&gt;a,d&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

If finite

If cost = 1 per step
Informed Search

- Smart search based on heuristic scores
  - Best-first search
  - Greedy best-first search
  - A* search
  - Admissible heuristics and optimality.
3. Consider the following search tree in which the nodes represent states and the arcs represent the moves connecting these states. Each node is labelled by a letter. The numbers on the arcs represent the true cost of the associated move. The numbers on the nodes represent the estimated cost of reaching the goal state from that node.

In which order would the A* algorithm explore this search tree?

(a) A, B, C, D, I, J, G, H.
(b) A, B, C, D, I, J.
(c) A, C, H.
(d) A, D, I, J.
(e) A, C, G, H.
A* search

- Evaluation function $f(n) = g(n) + h(n)$
  - $g(n) =$ cost so far to reach $n$
  - $h(n) =$ estimated cost from $n$ to goal
  - $f(n) =$ estimated total cost of path through $n$ to goal

- Avoid expanding paths that are already expensive
Example

4 + 4 = 8  
5 + 2 = 7  
4 + 2 = 6

A B C D
Example

A*: Search

INF2D: REASONING AND AGENTS
A* Search

Example

We're done as we've expanded a node containing a goal state.
3. Consider the following search tree in which the nodes represent states and the arcs represent the moves connecting these states. Each node is labelled by a letter. The numbers on the arcs represent the true cost of the associated move. The numbers on the nodes represent the estimated cost of reaching the goal state from that node.

In which order would the A* algorithm explore this search tree?

(a) A, B, C, D, I, J, G, H.
(b) A, B, C, D, I, J.
(c) A, C, H.
(d) A, D, I, J.
(e) A, C, G, H.
Smart Searching Using Constraints

- **Constraint Satisfaction Problems (CSPs):**
  - states defined by values of a fixed set of variables
  - goal test defined by constraints on variable values

- **Backtracking** = depth-first search with one variable assigned per node.

- **Variable ordering and value selection** heuristics help significantly.

- **Forward checking** prevents assignments that guarantee later failure.

- **Constraint propagation** (e.g., arc consistency) does additional work to constrain values and detect inconsistencies.
Adversarial Search

- Minimax assumes that both players play optimally
- Informally: Each agent is making its decision for the next move based on the assumption that the other agent is playing as well as it can.
Adversarial Search (Contd)

- $\alpha$-$\beta$ Pruning and its properties
- Reasoning about relevant computations only enables search space to be pruned.
- How to deal with deep trees: need for evaluation functions.
Broad Picture
Key issues:
• Utility of logic
• Bounded reasoning
Multiple agents, each with different logics but (limited) consistency across logics

Key issues:
• Equivalence between logics
• Monotonicity of inference
Agents too complex to specify
Logic applied to their interactions

Key issues:
- Specification of interaction
- Semantics across interactions
### In healthcare:
- Data sources are created by experiments and care pathways.
- Precision medicine drives to global scale.

### Table: Millions of population for 250 incident cancer patients a year with biomarker at -

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Incidence (per M)</th>
<th>Mortality</th>
<th>20% frequency</th>
<th>5% frequency</th>
<th>1% frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Breast</td>
<td>653</td>
<td>23%</td>
<td>1.9</td>
<td>7.7</td>
<td>38.3</td>
</tr>
<tr>
<td>2</td>
<td>Prostate</td>
<td>548</td>
<td>26%</td>
<td>2.3</td>
<td>9.1</td>
<td>45.6</td>
</tr>
<tr>
<td>3</td>
<td>Lung</td>
<td>530</td>
<td>83%</td>
<td>2.4</td>
<td>9.4</td>
<td>47.1</td>
</tr>
<tr>
<td>4</td>
<td>Colon</td>
<td>340</td>
<td>37%</td>
<td>3.7</td>
<td>14.7</td>
<td>73.5</td>
</tr>
<tr>
<td>5</td>
<td>Melanoma</td>
<td>167</td>
<td>17%</td>
<td>7.5</td>
<td>29.9</td>
<td>149.4</td>
</tr>
<tr>
<td>10</td>
<td>Pancreas</td>
<td>111</td>
<td>94%</td>
<td>11.3</td>
<td>45.1</td>
<td>225.6</td>
</tr>
<tr>
<td>14</td>
<td>Ovary</td>
<td>87</td>
<td>62%</td>
<td>14.4</td>
<td>57.5</td>
<td>287.7</td>
</tr>
<tr>
<td>20</td>
<td>Liver</td>
<td>56</td>
<td>89%</td>
<td>22.4</td>
<td>89.5</td>
<td>447.3</td>
</tr>
<tr>
<td>25</td>
<td>Cervix</td>
<td>36</td>
<td>33%</td>
<td>34.5</td>
<td>138.2</td>
<td>690.9</td>
</tr>
<tr>
<td>30</td>
<td>Larynx</td>
<td>29</td>
<td>32%</td>
<td>43.3</td>
<td>173.4</td>
<td>866.9</td>
</tr>
</tbody>
</table>

### Key:
- **Green** - achievable in a large region like Scotland or East of England
- **Amber** - achievable across the whole of the UK
- **White** - requires international cooperation
If Interactions are Specifications then They Can be Distributed to Autonomous Agents (if the Agents can Interpret Them)

Numerous peer to peer infrastructures were built for this (similar to the idea of smart contracts in distributed ledger systems)
Science Drivers

**Past**
- Sensors: Sensor is simply a data source
- Robotics: Single component
- Natural language: Small corpus and brittle
- Human factors: Uncanny valley
- Machine learning: Domain specific
- Data linkage: Single database
- Architectures: AI on top of conventional CS
- Social computation: Engineering of individual agents
- Security: AI to expedite security

**Future**
- Sensors: AI on the sensor = edge compute
- Robotics: Full humanoid
- Natural language: Large corpus and robust
- Human factors: Realistic human representations
- Machine learning: Transportable theory
- Data linkage: Knowledge graphs
- Architectures: AI as basis for new architectures
- Social computation: Engineering of social interactions
- Security: Security for and against AI systems

Confluence through applications
Examples of Confluence

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Al on the sensor = edge compute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotics</td>
<td>Full humanoid</td>
</tr>
<tr>
<td>Natural language</td>
<td>Large corpus and robust</td>
</tr>
<tr>
<td>Human factors</td>
<td>Realistic human representations</td>
</tr>
<tr>
<td>Machine learning</td>
<td>Transportable theory</td>
</tr>
<tr>
<td>Data linkage</td>
<td>Knowledge graphs</td>
</tr>
<tr>
<td>Architectures</td>
<td>AI as basis for new architectures</td>
</tr>
<tr>
<td>Social computation</td>
<td>Engineering of social interactions</td>
</tr>
<tr>
<td>Security</td>
<td>Security for and against AI systems</td>
</tr>
</tbody>
</table>

- Optimising the virtual machine
- Task-specific (AI) chips
- Parkinsons detection via audio
- Robot as a platform
- Emergent behaviours
- Predictable autonomy
- Deep modelling of speech
- Speech to dialogue
- Authentication
- Data federation for analytics
- Reproducible machine learning
- Privacy in a federated system
- Distributed protocols
- “Smart contracts”
- Interaction security