Problem Solving by Searching

Russell and Norvig, chapter 3

Outline
- Problem-solving agents
  - A kind of goal-based agent
- Formulating problems
- Example problems
- Search strategies

Puzzles!

The missionaries and cannibals problem
- Goal: transport the missionaries and cannibals to the right bank of the river.
- Constraints:
  - Whenever cannibals outnumber missionaries, the missionaries get eaten
  - Boat can hold two people and can’t travel empty

Formulating the problem

- A state description that allows us to describe our state and goal:
  \[(M_L, C_L, B)\]
- Initial state: \((3,3,L)\) Goal: \((0,0,R)\)

Problem solving agents
- Problem Formulation
  - States and actions (successor function).
- Goal Formulation
  - Desired state of the world.
- Search
  - Determine the possible sequence of actions that lead to the states of known values and then choosing the best sequence.
- Execute
  - Given the solution, perform the actions.
- Assumptions:
  - Environment is fully observable, deterministic
  - Agent knows the effects of its actions
Graph formulation of the problem

- Nodes: all possible states.
- Edges: edge from state $u$ to state $v$ if $v$ is reachable from $u$ (by an action of the agent).
- Edges for missionaries and cannibals problem?
- Problem is now to find a path from $(3,3,L)$ to $(0,0,R)$.
- In general, paths will have costs associated with them, so the problem will be to find the lowest cost path from initial state to the goal.

Stating a Problem as a Search Problem

- State space $S$ (nodes)
- Successor function: the states you can move to by an action (edge) from the current state
- Initial state
- Goal test: is a state a goal?
- Cost

Back to our problem

Actions (operators):
- CCR - transport two cannibals to the right bank
- MCL - transport a missionary and a cannibal to the left bank

The (partially expanded) search graph

Actions:
- CCR - transport two cannibals to the right bank
- MCL - transport a missionary and a cannibal to the left bank

Repeated states

The search graph is not necessarily a tree!

Searching the State Space

- Often it is not feasible (or too expensive) to build a complete representation of the state graph
- A problem solver must construct a solution by exploring a small portion of the graph
Searching the State Space

Search tree

Searching the State Space

Search tree

Searching the State Space

Search tree

Searching the State Space

Search tree

Searching the State Space

Search tree

Searching the State Space

Search tree
The 8 puzzle

- States?
- Initial state?
- Actions?
- Goal test?
- Path cost?

(n²-1)-puzzle

- States? Integer location of each tile
- Initial state? Any state
- Actions? (tile, direction) where direction is one of {Left, Right, Up, Down}
- Goal test? Check whether goal configuration is reached
- Path cost? Number of actions to reach goal
- Is the search graph a tree?

15-Puzzle

Sam Loyd offered $1,000 of his own money to the first person who would solve the following problem:

But no one ever won the prize!!
Solution to the Search Problem

- A solution is a path connecting the initial node to a goal node (any one)

Path Cost

- An edge cost is a positive number measuring the "cost" of performing the action corresponding to the edge, e.g.:
  - \( c = 1 \) in the 8-puzzle example
- We will assume that for any given problem the cost \( c \) of an arc always satisfies:
  - \( c \geq \epsilon > 0 \), where \( \epsilon \) is a constant
- Why? Has to do with the cost of arbitrarily long paths

Goal State

- It may be explicitly described:
- or partially described:
- or defined by a condition, e.g., the sum of every row, of every column, and of every diagonal equals 30
- ("a" stands for "any" other than 1, 5, and 8)

Another example: the 8 queens problem

- Incremental vs. complete state formulation:
  - Incremental formulation starts with an empty state and involves operators that augment the state description
  - A complete state formulation starts with all 6 queens on the board and moves them around

8 queens problem: representation is key

- Incremental formulation
  - States? Any arrangement of 0 to 8 queens on the board
  - Initial state? No queens
  - Actions? Add queen in empty square
  - Goal test? 8 queens on board and none attacked
  - Path cost? None
  - 64 x 63 x \( \ldots \) 57 \( \sim 3 \times 10^{19} \) states to investigate
- Is the search graph a tree?
**A better representation**

Another Incremental formulation:
- States? n (between 0 and 8) queens on the board, one in each of the n left-most columns; no queens attacking each other.
- Initial state? No queens
- Actions? Add queen in leftmost empty column such that it does not attack any of the queens already on the board.
- Goal test? 8 queens on board

---

**n-Queens Problem**

- A solution is a goal node, not a path to this node (typical of design problem)
- Number of states in state space:
  - 8-queens $\rightarrow$ 2,057
  - 100-queens $\rightarrow$ $10^{42}$
- But techniques exist to solve n-queens problems efficiently for large values of n

---

**Path Planning**

What is the state space?

---

**Assumptions in Basic Search**

- The world is static
- The world is discretizable
- The world is fully observable
- The actions are deterministic

But many of these assumptions can be removed, and search still remains an important problem-solving tool

---

**Search and AI**

- Search methods are ubiquitous in AI systems. They often are the backbones of both core and peripheral modules
- An autonomous robot uses search methods:
  - to decide which actions to take and which sensing operations to perform,
  - to quickly anticipate collision,
  - to plan trajectories,
  - etc...

---

**Applications**

- Search plays a key role in many applications, e.g.:
  - Route finding (mapquest, internet, airline)
  - VLSI Layout
  - Robot Navigation.
  - Pharmaceutical drug design, protein design
  - Video games