Basic Search Problem

- All search techniques so far have a worst case exponential time complexity.
- A full search may take too long!
  - Chess has a possible $10^{120}$ game paths
  - Checkers has a possible $10^{40}$ game paths

Examples

- Trees for common 2-player games
  - Tic-tac-toe has a possible $9!$ game paths (this is why tic-tac-toe tends to be used in brute force search examples)
  - The Eight Puzzle is another game example

Heuristic Search

- "Intelligence for a system with limited processing resources consists in making wise choices of what to do next..."
  - Newell and Simon, 1976, Turing Award Lecture

Consider Two Problems

- Medical Diagnosis
- Chess

Two Basic Situations

- A problem may not have an exact solution because of inherent ambiguities in the problem statement or available data. Medical diagnosis is an example of this. Heuristics are used to choose the most likely diagnosis and formulate a plan of treatment.
- A problem may have an exact solution, but the computational cost may be prohibitive. An example here is chess.

Comments on the Nature of Heuristics

- Heuristics have limited information and are seldom able to predict the exact behavior of the state space farther along in the search – can lead to suboptimal solutions
- Heuristic algorithms have two parts:
  - The heuristic measure
  - An algorithm that uses it to search the state space
### Hill Climbing

- The simplest thing to do is to climb a hill the steepest path possible
- Hill climbing strategies search and evaluate only the best children's children. They ignore other siblings.
- What could go wrong here?

### Local Maxima

- What if you have to take a winding path (it goes down and sometimes up) to get to the top of the hill?

> Many algorithms in AI suffer from the problem of getting stuck in local maxima.

### Best-first Search

- Local maxima are hopefully avoided by backtracking
- Uses two lists:
  - An open list: Keeps track of current fringe of the search
  - A closed list: keeps track of states already visited
  - The algorithm orders states on open according to their "closeness" to a goal
  - This amounts to a priority queue for the search

### Admissibility

- Admissibility: Heuristics that find the shortest path to a goal whenever it exists are admissible.
- Breadth-first search is admissible as it examines all possible states at level n, before going to the next level

### Let’s Say

- The evaluation function f for our search is made up of the sum of:
  - The actual length of the path from any state n to the current start point
  - The estimated distance from state n to the goal
- This is represented as: $f(n) = g(n) + h(n)$
  - n is a node in the state space graph
  - $g(n)$ measures the depth as which that state has been found in the graph
  - $h(n)$ is the estimate of the distance from n to a goal
**f*(n) Definition**

- **f*(n)** is the optimal path consisting of:
  - g*(n): the shortest path from the start node to node n
  - h*(n): the actual cost of the shortest path from n to the goal
- This is defined as: f*(n) = g*(n) + h*(n)

**Algorithms A and A***

- **Algorithm A:**
  - May find a less than optimal cost to the current node
  - g(n) is a reasonable estimate of g*(n) or g(n) >= g*(n)
- **Algorithm A***:
  - Uses Algorithm A AND
    - Uses a heuristic estimate of the minimal cost to a goal state
    - h(n) is a reasonable estimate of h*(n) or h(n) <= h*(n)

**A***

- All A*** algorithms are admissible
- Breadth-first search is A***
  - f(n) = g(n) + 0

**Monotonicity**

- A*** algorithms don’t require that g(n) = g*(n), so they may initially reach non-goal states along a suboptimal path, as long as the alg. Eventually finds an optimal path.
- Are there heuristics that are “locally admissible” – they consistently find the minimal path to each state they encounter in the search?
  - This property is called monotonicity.

**Informedness**

- Informedness: When one heuristic is better than another, it is more informed.
- The more informed an algorithm is, the less states need to be expanded when searching
- Example of the eight puzzle:
  - Using breadth-first search
  - Using the number of tiles out of place