# AI Searching Techniques

Informed Search Techniques

Dr.Sherif Saad

# Learning Objectives

- Introduce the students to different search techniques and algorithms
- Present a general approach to model and represent problems as search problems
- Demonstrate how to implement AI search techniques and algorithms

# Last-Time

- Breadth First Search Algorithm
- Lowest-Cost-First Search

# Outlines

- Introduction to Informed Search Techniques
- Design Heuristics Functions (Heuristics Engineering)
- Best-First Search

# Informed Search Techniques: Definition

What is informed search?

- Also known as heuristic search
- A family of search techniques that focus on searching the most promising path or state first.
- It uses a problem-specific information to solve the search problem and avoid exploring the entire search space.

# Informed Search Techniques: Main Idea

- The main ideas of informed or heuristic search can be summarized in the following points
  - Use domain or problem specific knowledge to guide the search operation
  - Expand the most promising state first
  - If there is no information to guide the search fall back to uninformed search.

# Informed Search Techniques: Why??

Why informed (heuristic) search?

- If the search space is too big, an uninformed search can just become impractical (useless). Either too much time to find the solution, or stuck in a loop, or become limited to the storage space.
- In general, all search problems have some domain specific knowledge that we can use to guide the search operation and avoid searching (expanding) the entire search space.
- For some problems good enough solutions are sufficient

# Informed Search Techniques: Characteristics

- In general, locate the goal state more quickly.
- Can work with time constraints (e.g. limited time available to find the solution)
- While it does not search the entire search space, it often finds the near optimal solution. Why??
- Let A and B two informed search techniques if A can identify the most promising path or branch of the search space more than B. Then A is **more informed** than B
- Many search problem are NP-complete with exponential time complexity, informed search is useful for these problem.

# Informed Search Techniques: Challenges

In your opinion what are the key challenges of applying heuristic search to solve search problems?

# Heuristic Function: Definition

- A heuristic function h(n) is a function that estimates the cost of reaching the goal (finding solution) by expanding node n.
- A heuristic function h(n) estimates how far n from the goal.
- In a search graph G if we are currently at node v and we want to get to node u, then h(v) is a function that estimates how v is away from u
- Design a good heuristic function require understanding of the problem domain and the goal

# Heuristic Function: Admissible Heuristics

If a heuristic function h(n) never overestimates the distance to the goal, then h(n) is admissible.

For example h(v) is the euclidean distance between current node and the goal

$$d(a,b) = \sqrt{(a_x - b_x)^2 + a_y - b_y}^2$$

The estimated cost to reach the goal is not higher than the lowest possible cost from the current node in the path.

# Heuristic Function: Consistent Heuristics

If a heuristic function h(n) estimation is always less than or equal to the estimated cost  from any neighboring node to the goal, plus the step cost of reaching that neighbor, then h(n) is consistent heuristic.

Consistent heuristic is admissible but not every admissible heuristic is consistent.

*Russel and Norvig point out in Artificial Intelligence: A Modern Approach (the most commonly used AI textbook) it is challenging to come up with a heuristic that is admissible but not consistent*

# Heuristic Function: Design

How we design a good heuristic function?

- We must only use the information that we can collect about the node and the problem domain. (it depend on the problem domain or context)
- The common practice to design (derive) h(n) is to solve a simple (smaller) problem of the actual problem and use the cost of the smaller problem as heuristic of the actual problem.
- A simple problem means a relaxed version of the actual problem (e.g. drop constraints, reduce the search space)

# Heuristic Function: Design

How we design a good heuristic function?

- Identify constraints which, when we dropped make the problem easy to solve.
- Combine multiple heuristic when applicable (e.g. $h_1(n)$ and $h_2(n)$ are both admissible heuristic then we can define $h_3(b) = \min(h_1(n), h_2(n))$
- Heuristics could be extracted from patterns, goals stored in a database
- Examples
  - Euclidean Distance
  - Manhattan Distance
  - Hamming Distance

# Heuristic Function: Select a good Heuristic

**H1:** One possible h(n) is the number of misplaced blocks (tiles), this is the hamming distance.

**H2:** Another possible h(n) is the sum of number of moves between each block in the current state and its goal position.

Which heuristic is more informed and why?

# Best First Search: Description

- It is a general algorithm to implement informed search techniques.
- It combines the advantages of DFS and BFS; this means the goal can be found without expanding all nodes and it does not get trapped in a loop or deep branch.
- The idea is very simple as each step during the search the most promising node is chosen for expansion.
- If one of the selected nodes generates nodes that are less promising, we can switch to another promising node on the same level. If the nodes on the same level are not as promising as we expected, then we backtrack.

# Best First Search: Main Idea

- Use a heuristic function to estimate the desirability of each node.
- Always expand the most desirable and unexpanded node
- What data structure we should use to store the nodes?
- There are many forms of the Best First Search:
  - A* Search Algorithm
  - Beam Search
  - Greedy Best First Search
  - What about the Lowest-Cost-First-Search?

# Best First Search: Intelligence Properties

Complete: NO|Yes ??

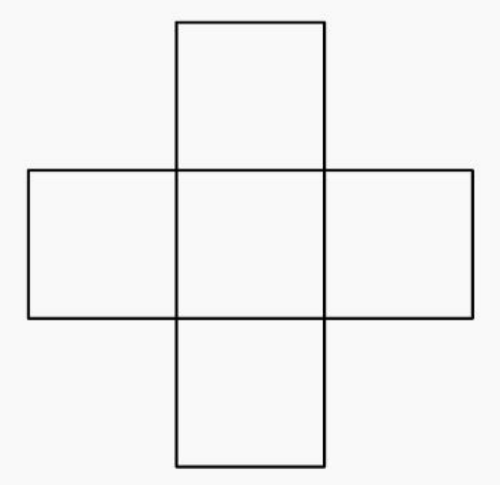
Optimality: NO|Yes ??

Time Complexity: O(b^d)

Space Complexity: O(b^d)

# Heuristic Function: Question

Each of the five numbers 1, 4, 7, 10, and 13 is placed in one of the five squares so that the sum of the three numbers in the horizontal row equals the sum of the three numbers in the vertical column.



# Next-Time

- A*  Search Algorithm
- Greedy Algorithm
- Travel Salesman Problem

# Questions