Problem Spaces & Search

CSE 473

473 Topics

Agents & Environments
Problem Spaces
Search & Constraint Satisfaction
Knowledge Repr'n & Logical Reasoning
Machine Learning
Uncertainty: Repr'n & Reasoning
Robotics

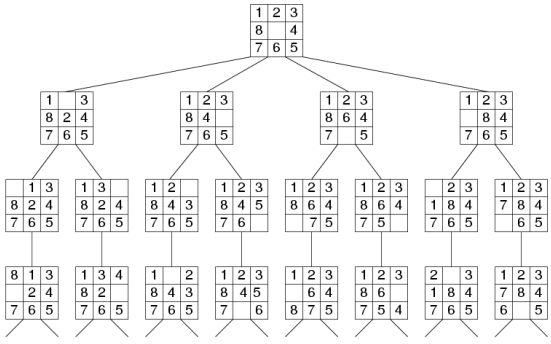
Weak Methods

"In the knowledge lies the power…"
[Feigenbaum]

But what if no knowledge????

Generate & test: As weak as it gets

Example: Fragment of 8-Puzzle Problem Space



Search thru a Problem Space / State Space

Input:

- Set of states
- Operators [and costs] (successor function)
- Start state
- Goal state [test]

Output:

- Path: start at a state satisfying goal test
- [May require shortest path]

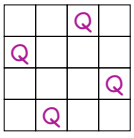
Example: Route Planning

Input:

- Set of states
- Operators [and costs]
- Start state
- Goal state (test)

Output:

Example: N Queens



Input:

- Set of states
- Operators [and costs]
- Start state
- Goal state (test)

Output:

Search Strategies

Blind Search

- Depth first search
- Breadth first search
- Iterative deepening search
- Iterative broadening search

Informed Search

Constraint Satisfaction

Adversary Search

Search Strategies

A search strategy is defined by picking the order of node expansion

Strategies are evaluated along the following dimensions:

- completeness: does it always find a solution if one exists?
- time complexity: number of nodes generated
- 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

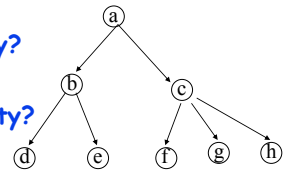
- *b:* maximum branching factor of the search tree
- *d:* depth of the least-cost solution
- *m:* maximum depth of the state space (may be ∞)

Breadth First Search

Maintain queue of nodes to visit

Evaluation

- Complete?
  Yes
- Time Complexity?
  O(b^d)
- Space Complexity?
  O(b^d)



Memory a Limitation?

Suppose:

2 GHz CPU
1 GB main memory
100 instructions / expansion
5 bytes / node

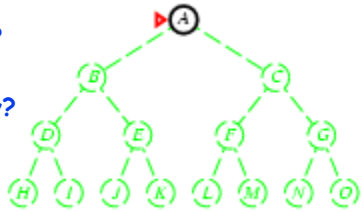
200,000 expansions / sec
Memory filled in 100 sec … < 2 minutes

Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

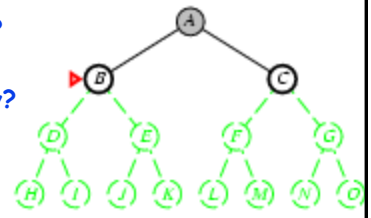


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

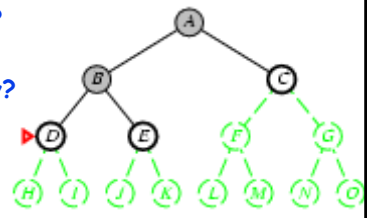


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

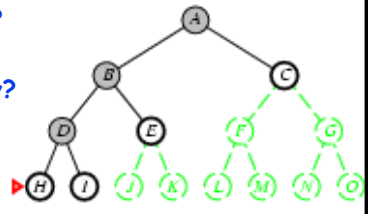


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

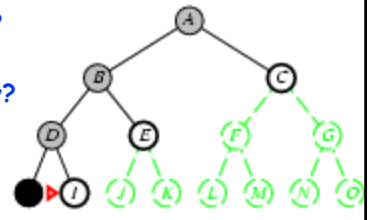


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

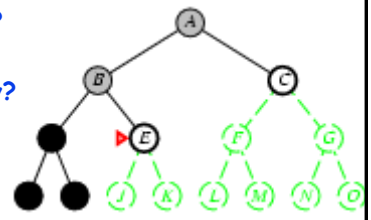


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

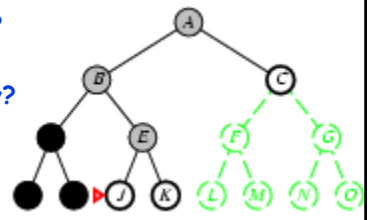


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

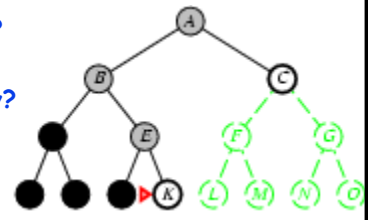


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

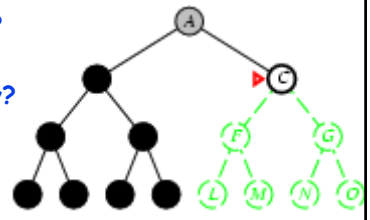


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

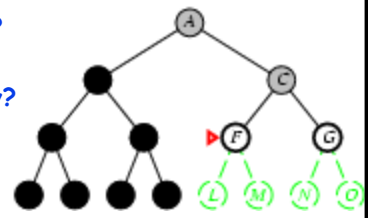


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

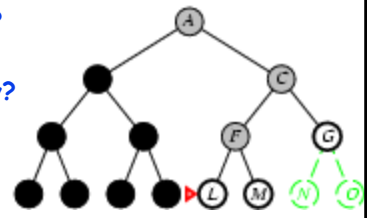


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

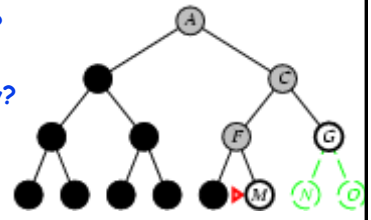


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
- Time Complexity?
- Space Complexity?

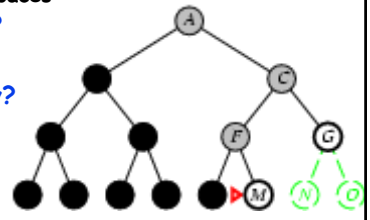


Depth First Search

Maintain stack of nodes to visit

Evaluation

- Complete?
  Not for infinite spaces
- Time Complexity?
  O(b^m)
- Space Complexity?
  O(bm)

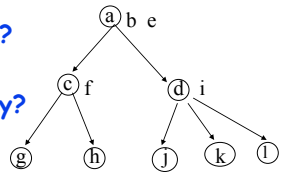


Iterative Deepening Search

DFS with limit; incrementally grow limit

Evaluation

- Complete?
  Yes
- Time Complexity?
  O(b^d)
- Space Complexity?
  O(bd)



Iterative deepening search

Number of nodes generated in a depth-limited search to depth d with branching factor b:

$$N_{DLS} = b^0 + b^1 + b^2 + \ldots + b^{d-2} + b^{d-1} + b^d$$

Number of nodes generated in an iterative deepening search to depth d with branching factor b:

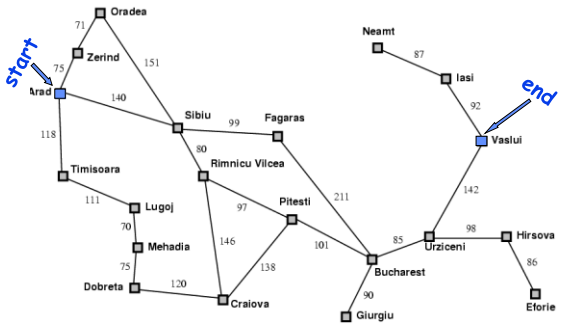
$$N_{IDS} = (d+1)b^0 + d\,b^{\wedge 1} + (d-1)b^{\wedge 2} + \ldots + 3b^{d-2} + 2b^{d-1} + 1b^d$$

For $b = 10$, $d = 5$,

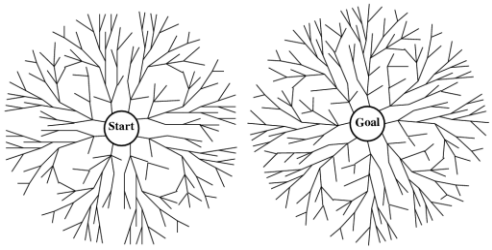
- $N_{DLS} = 1 + 10 + 100 + 1{,}000 + 10{,}000 + 100{,}000 = 111{,}111$
-
- $N_{IDS} = 6 + 50 + 400 + 3{,}000 + 20{,}000 + 100{,}000 = 123{,}456$
-

Overhead = (123,456 - 111,111)/111,111 = 11%

Forwards *vs.* Backwards

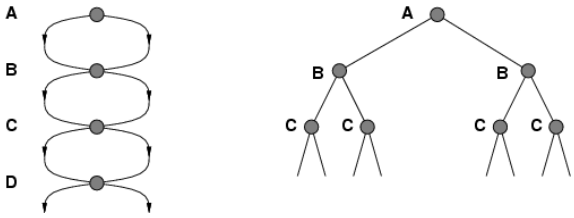


vs. Bidirectional



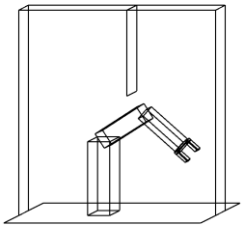
Repeated states

Failure to detect repeated states can turn a linear problem into an exponential one!



Graph search: Stores expanded nodes in a set called Closed and only adds new nodes to the fringe

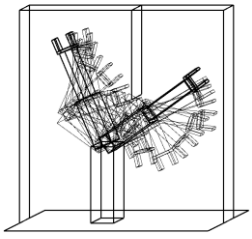
Manipulator Control



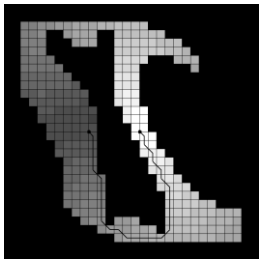
Arm with two joints

Configuration space

Manipulator Control Path

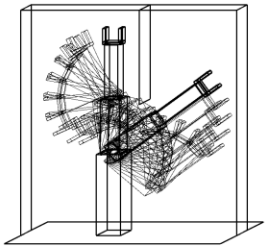


State space

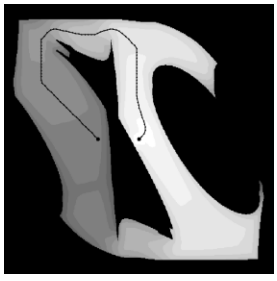


Configuration space

Manipulator Control Path



State space



Configuration space

Problem

All these methods are slow (blind)

Solution → add guidance ("heurstic estimate")
→ "informed search"