Artificial Inelligence : State Space Search

Text book

- Artificial Intelligence by Elaine Rich and Kevin Knight, 2nd edition, TMH

Steps to solve a problem

1. Define the problem precisely : include precise specifications of initial situations and acceptable final solutions

2. Analyze the problem

3. Isolate and represent the task knowledge that is necessary to solve the problem

4. Choose the best problem-solving technique and apply it to the particular problem

Problem Characteristics

1. Is the problem decomposable into a set of independent smaller or easier subproblems ?
2. Can solution steps be ignored or undone if they prove unwise ?
3. Is the problem's universe predictable ?
4. Is a good solution to the problem obvious without comparison to all other possible solutions ?
5. Is the desired solution a state of the world or a path to the state ?
6. Is a large amount of knowledge absolutely required to solve the problem, or is knowledge important only to constrain the search ?
7. Can a computer that is simply given the problem return the solution, or will the solution of the problem require interaction between the computer and a person ?

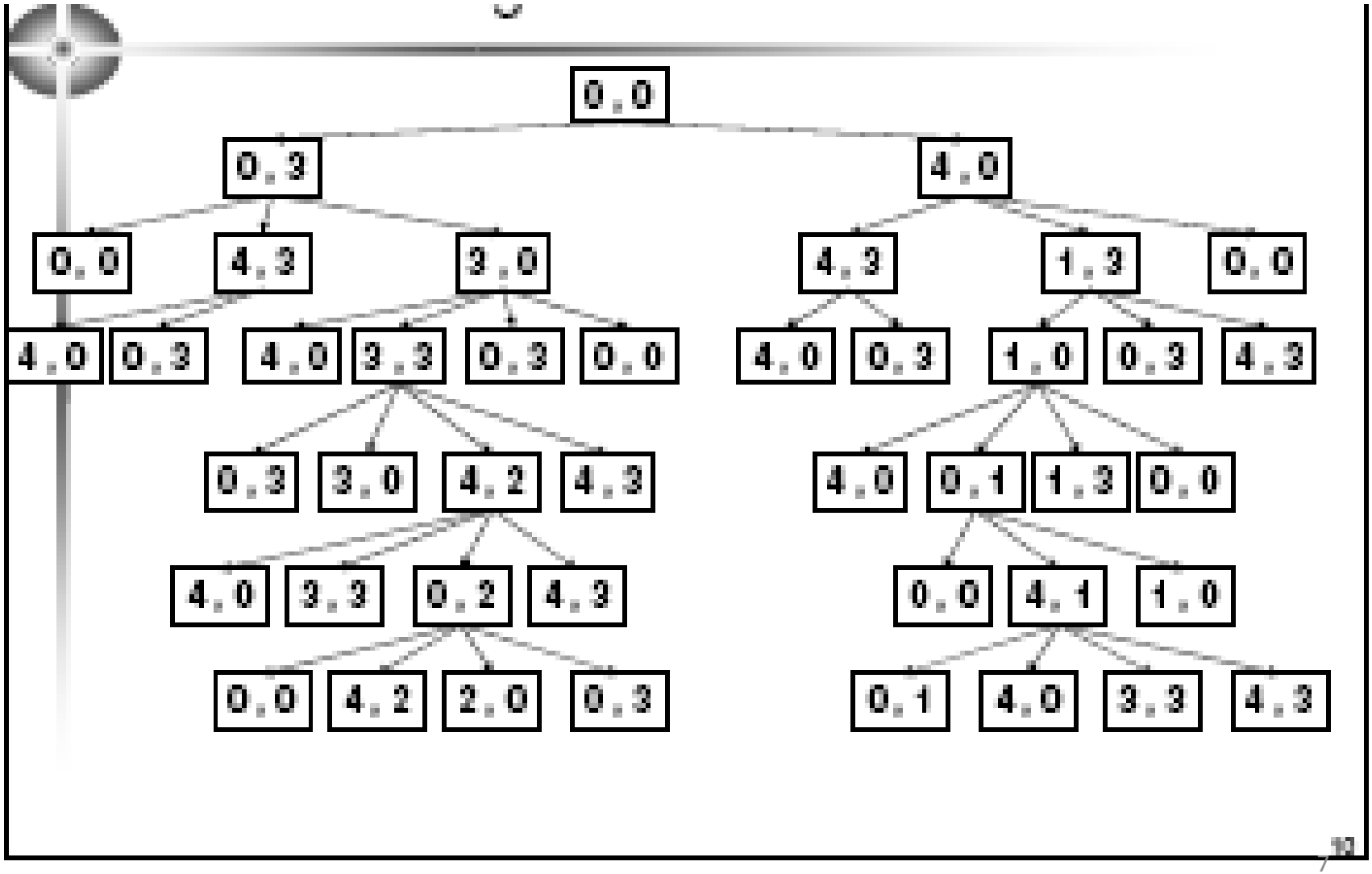
Production System

- A Production System consists of
  - A set of rules
  - Knowledge base
  - Control Strategy : order in which to apply rules
  - A Rule Applier

The WaterJug Problem

- You are given two jugs, a 4-gallon one and a 3-gallon one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into the 4-gallon jug ?
- Rule set for the Water Jug Problem
- A solution for the Water Jug Problem

Water Jug Problem : State Space Tree



Characteristics of a good control strategy

- It should cause motion
- It should be systematic

What is a heuristic ?

- A heuristic is a technique that improves the efficiency of a search process, possibly by sacrificing claims of completeness.

Search Techniques

- Depth First Search
  - Requires less memory since only the nodes on the current path are stored.
  - DFS may find a solution without examining much of the search space at all.

Search Techniques

- Depth First Search
  - Requires less memory since only the nodes on the current path are stored.
  - DFS may find a solution without examining much of the search space at all.
  - May get caught in a blind alley.

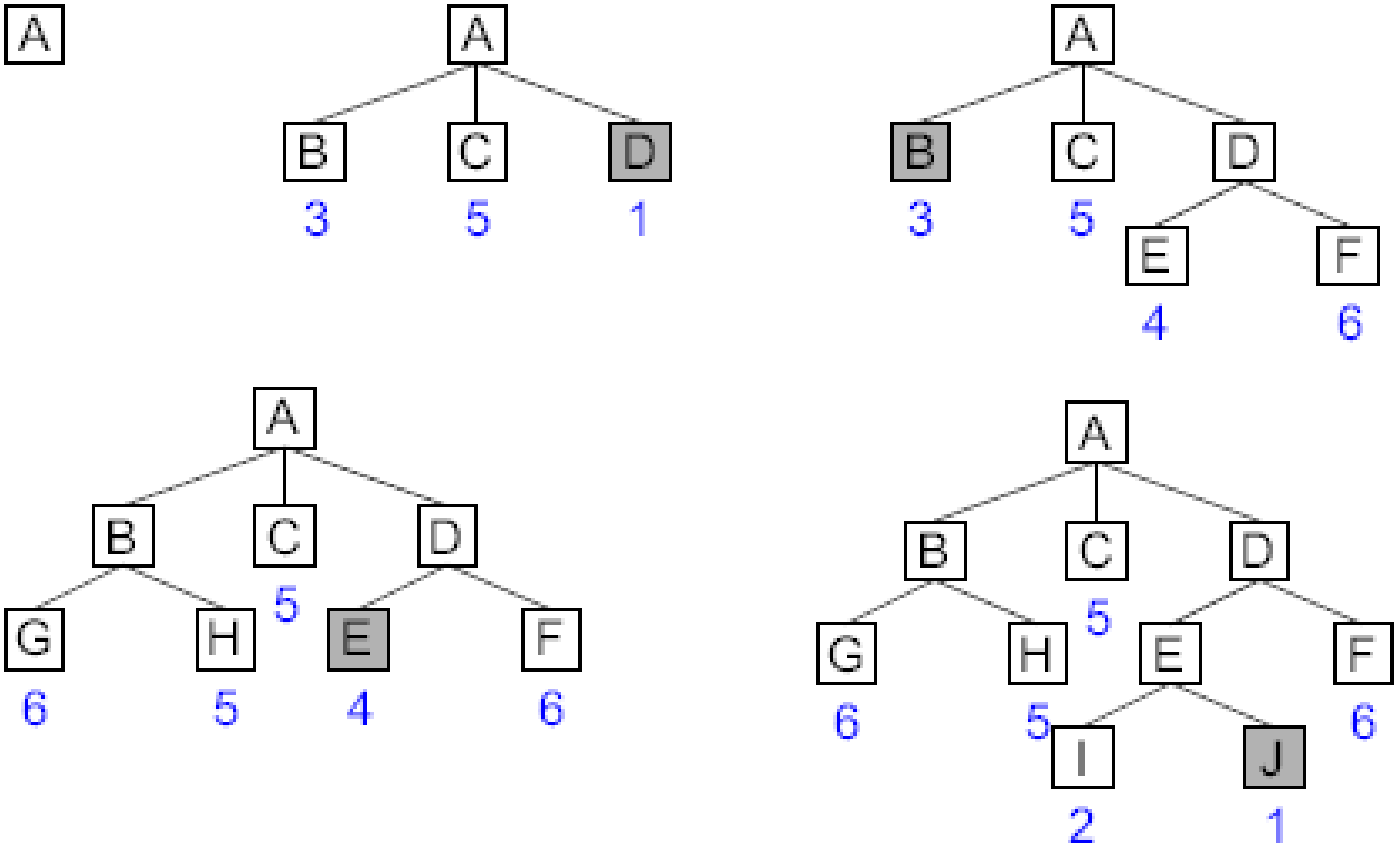
Search Techniques

- Breadth First Search
  - Will not get trapped exploring a blind alley.
  - If there is a solution, BFS is guaranteed to find it.
  - If there are multiple solutions, then, the minimal one will be found. (Because longer paths are not explored until all shorter ones have been examined).

Heuristic Search Technique

- Best First Search
  - Combines advantages of DFS and BFS. (i.e. possible to find solution without exploring all paths as in DFS *and* not get trapped on dead-end paths as in BFS.)

Best First Search….Example



Some Terms

- OPEN - Nodes that have been generated and have had the heuristic function applied to them but which have not yet been examined.
- CLOSED – Nodes that have already been examined. (We need to keep these nodes in memory if we want to search a graph rather than a tree, since whenever a new node is generated, we need to check whether it has been generated before.)

Algorithm: Best-First Search

1. Start with OPEN containing just the initial state.
2. Until a goal is found or there are no nodes left on OPEN do:
   - Pick the best node on OPEN.
   - Generate its successors.
   - For each successor do:
     - If it has not been generated before, evaluate it, add it to OPEN, and record its parent.
     - If it has been generated before, change the parent if this new path is better than the previous one. In that case, update the cost of getting to this node and to any successors that this node may already have.

Heuristic Search Techniques..continued

- Generate and Test
  1. Generate a possible solution.
  2. Test if solution is acceptable.
  3. If solution is found, quit. Otherwise return to step 1.

Hill Climbing

- Searching for a goal state = Climbing to the top of a hill

Hill Climbing..continued

- Hill Climbing
  - A variant of generate and test in which feedback is used to make decision in which direction to move in the search space.
  - In pure generate and test procedure, the test function is augmented with a heuristic function that provides an estimate of how close a given state is to a goal state.

Simple Hill Climbing

Algorithm

1. Evaluate the initial state.
2. Loop until a solution is found or there are no new operators left to be applied:
   - Select and apply a new operator
   - Evaluate the new state:

     goal → quit

     better than current state → new current state

Simple Hill Climbing

Algorithm

1. Evaluate the initial state.
2. Loop until a solution is found or there are no new operators left to be applied:
   - Select and apply a new operator
   - Evaluate the new state:

     goal → quit

     better than current state → new current state

not try all possible new states!

Steepest Ascent Hill Climbing

Considers all the moves from the current state.

Selects the best one as the next state.

Steepest Ascent Hill Climbing

Algorithm

1. Evaluate the initial state.
2. Loop until a solution is found or a complete iteration produces no change to current state:
   - Apply all the possible operators
   - Evaluate the best new state:

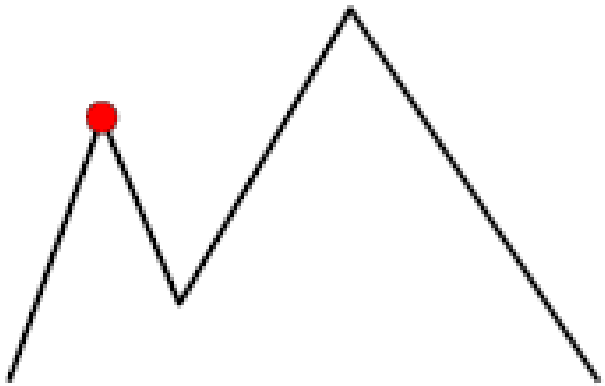
     goal → quit

     better than current state → new current state

Hill Climbing : Disadvantages

Local maximum

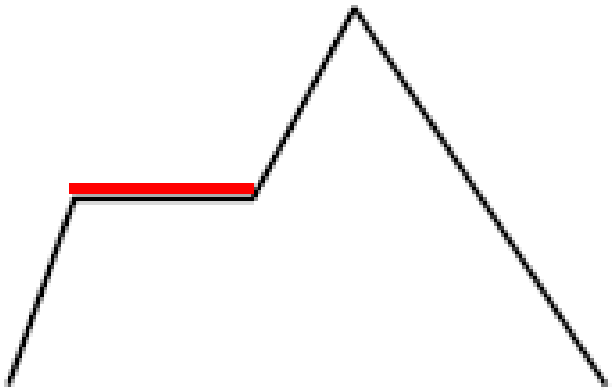
A state that is better than all of its neighbours, but not better than some other states far away.



Hill Climbing : Disadvantages

Plateau

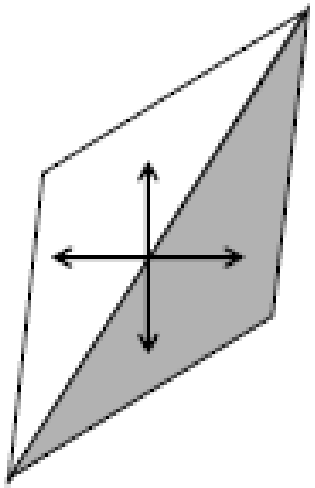
A flat area of the search space in which all neighbouring states have the same value.



Hill Climbing : Disadvantages

Ridge

The orientation of the high region, compared to the set of available moves, makes it impossible to climb up. However, two moves executed serially may increase the height.

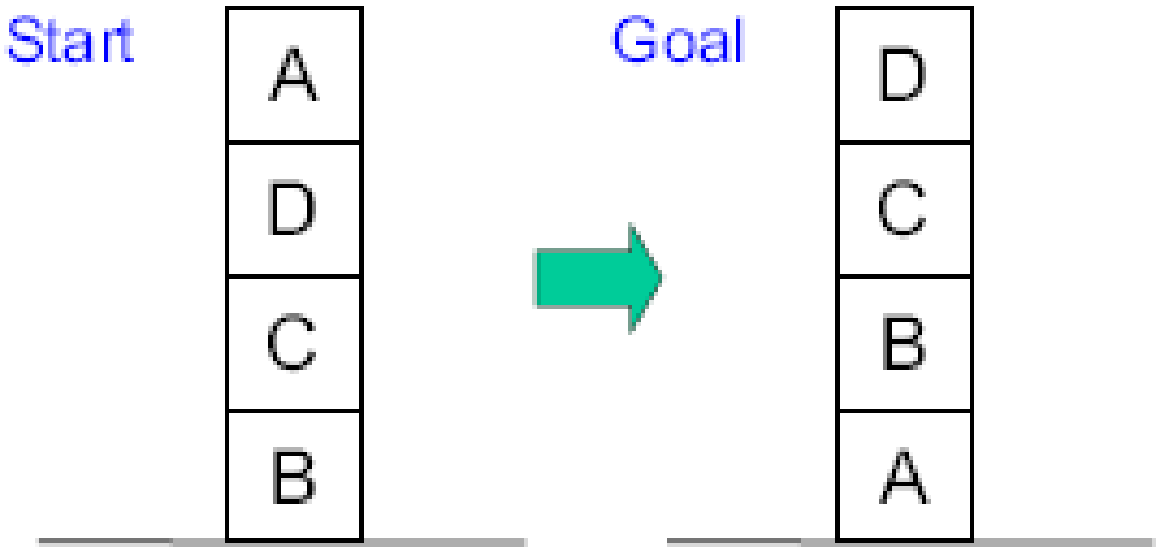


Hill Climbing : Disadvantages

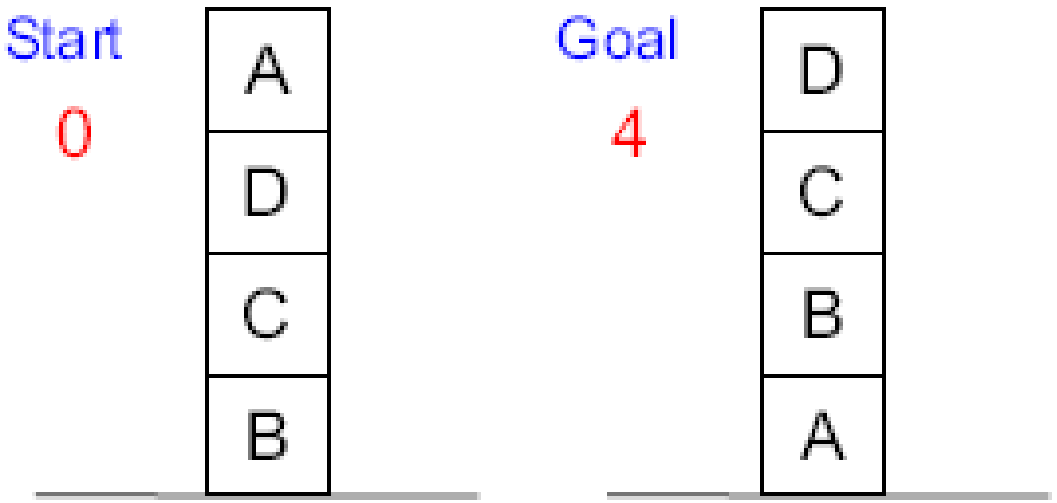
Ways Out

- Backtrack to some earlier node and try going in a different direction.
- Make a big jump to try to get in a new section.
- Moving in several directions at once.

Hill Climbing : Blocks World



Hill Climbing : Blocks World

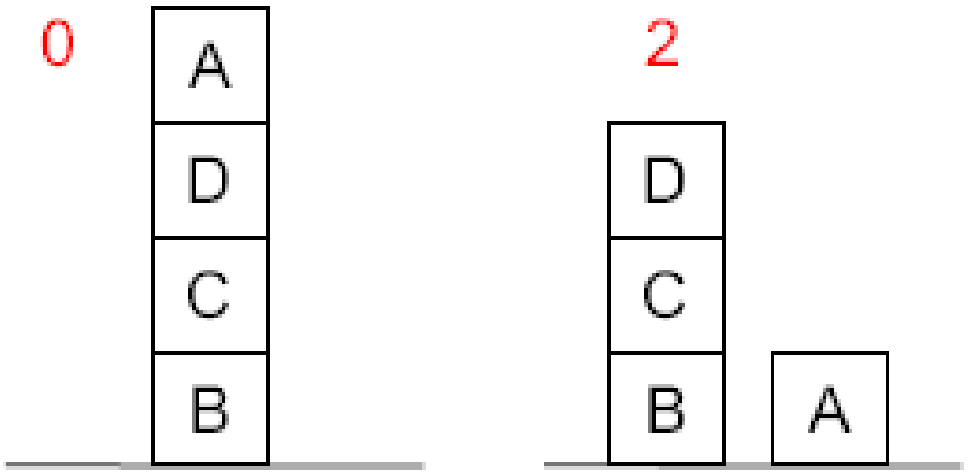


Local heuristic:

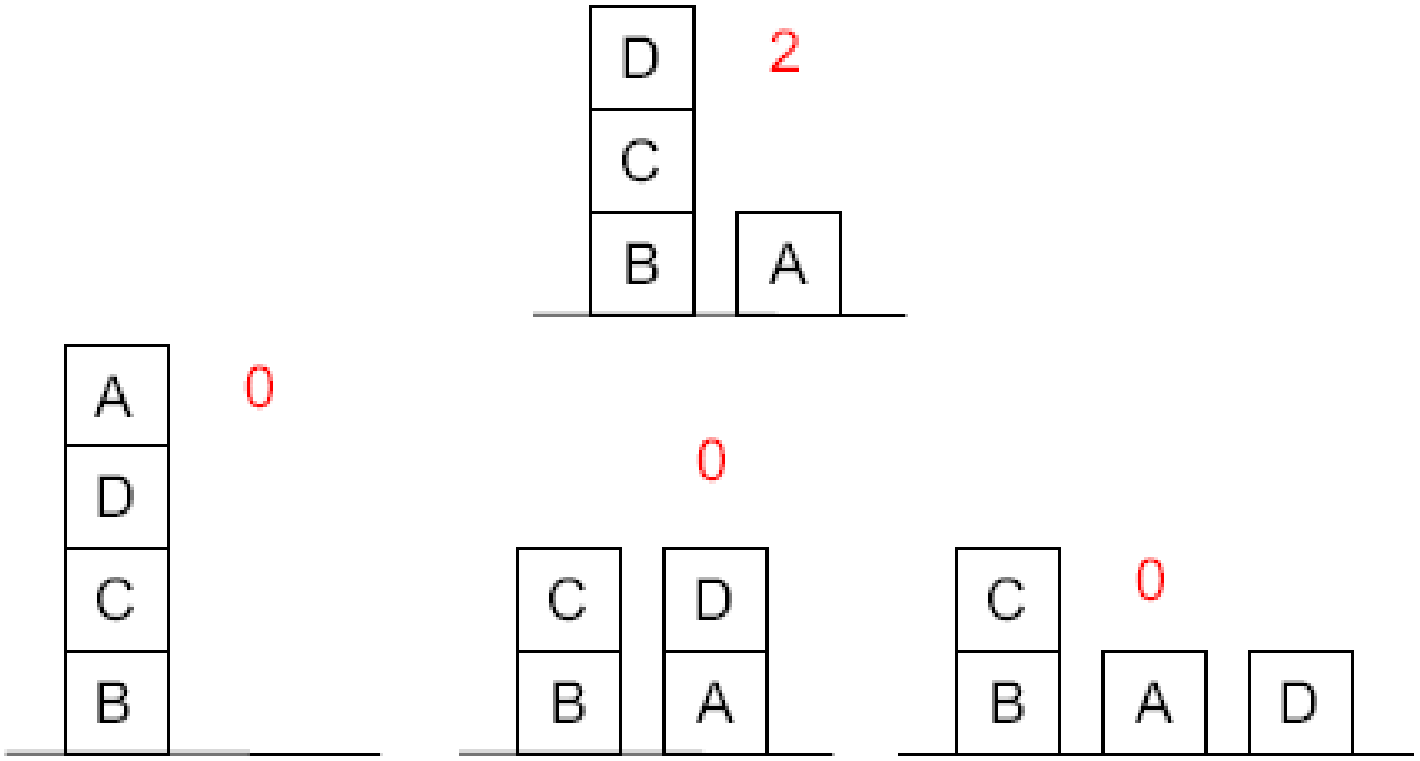
+1 for each block that is resting on the thing it is supposed to be resting on.

–1 for each block that is resting on a wrong thing.

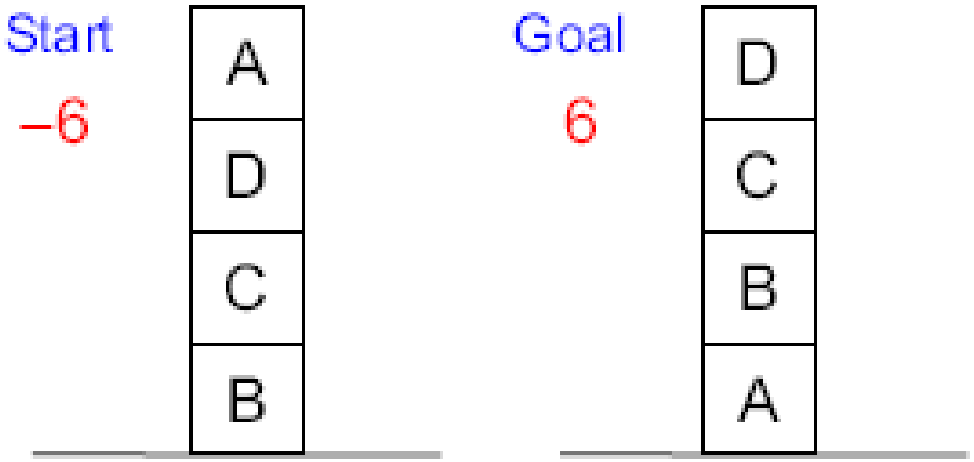
Hill Climbing : Blocks World



Hill Climbing : Blocks World



Hill Climbing : Blocks World



Global heuristic:

For each block that has the correct support structure: +1 to every block in the support structure.

For each block that has a wrong support structure: –1 to every block in the support structure.

Hill Climbing : Blocks World