SEARCH TREE

Node: State in state tree

Root node: Top of state tree

Children: Nodes that can be reached from a given node in 1 step (1 operator)

Expanding: Generating the children of a node

Open: Node not yet expanded

Closed: Node after expansion

Queue: Ordered list of open nodes

SEARCH

BLIND SEARCH: *Systematic Search*

Depth 1st: Continue along current path looking for goal

Breadth 1st: Expand all nodes at current level before progressing to next level

Depth–limited Search: Depth-1st + depth-limit

Iterative Deepening Search: limit=0,limit=1, ...

USING COST: $g(n)$=*cost from start to n*

Uniform Cost Search: Select best $cost(n)$

Branch-and-bound: Select best $g(n)$.

USING HEURISTIC: $h(n)$=*Estimate cost to a goal*

Greedy Search: Select best $h(n)$

A*: Select best $f(n) = g(n) + h(n)$

IDA*: Depth-1st + f-cost limit.

Hill-climbing: Depth-1st exploring best $h(n)$ first

Simulated Annealing: Hill climbing + random walk

Beam Search: Breadth-1st pruning worse $h(n)'s$