Week 4 T3

Wednesday, 10 August 2022 4:07 PM

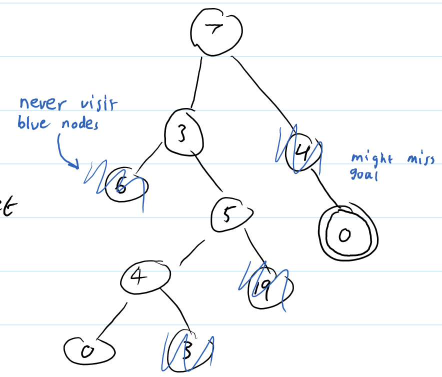
IDA*

- ID algorithm with different limit
- Initial limit = $f(s_0)$
- Perform DFS ignoring all nodes with f value greater than limit
- Next iteration, update limit to be min f value of ignored nodes
- Space advantage of ID
- Node efficiency + complete + optimal of A* (with admissible heuristic)

Hill Climbing

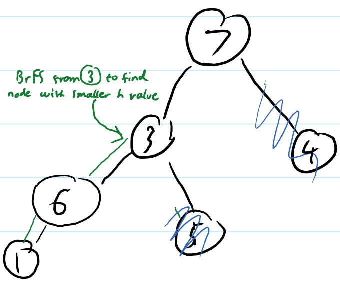
- Local search algorithm
- Expand node with smallest h value
- Remove all other nodes from openset

Not Complete



Enforced Hill Climbing

- Perform BrFS on graph rooted at current node with goal of finding any node with smaller h value.
- Found node becomes new current node (keep track of path between nodes)



Classical Planning Model

State Model:

- finite and discrete state space S
- a known initial state $s_0 \in S$
- a set $S_G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each $s \in S$
- a deterministic transition function $s' = f(a, s)$ for $a \in A(s)$
- positive action costs $c(a, s)$

Conformant Planning Model

- finite and discrete state space S
- a set of possible initial state $S_0 \in S$
- a set $S_G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each $s \in S$
- a non-deterministic transition function $F(a, s) \subseteq S$ for $a \in A(s)$
- uniform action costs $c(a, s)$

- Solution is a sequence of actions which will allow you to reach the goal state regardless of initial state or transitions. Without feedback!
- Idea is that you could close your eyes, perform the actions, and always end up at the goal state

Markov Decision Process MDP

- a state space S
- initial state $s_0 \in S$
- a set $G \subseteq S$ of goal states
- actions $A(s) \subseteq A$ applicable in each state $s \in S$
- transition probabilities $P_a(s'|s)$ for $s \in S$ and $a \in A(s)$
- action costs $c(a, s) > 0$

- Solution is a Policy: Function which tells you which action to take when you are in any given state With Feedback!
- Optimal = minimize expected cost

Partially Observable Markov Decision Process POMDP

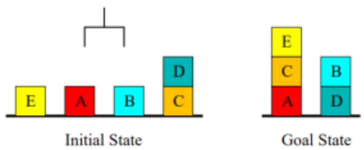
- states $s \in S$
- actions $A(s) \subseteq A$
- transition probabilities $P_a(s'|s)$ for $s \in S$ and $a \in A(s)$
- initial belief state b_0
- final belief state b_f
- sensor model given by probabilities $P_a(o|s)$, $o \in Obs$

- Belief states are probability distributions
- Solution is a Policy that gives actions with probabilities for each belief state which allows us to reach final belief state distribution.
- Sensor model provides known probabilistic information on each state

STRIPS

- A **problem** in STRIPS is a tuple $P = \langle F, O, I, G \rangle$:
  - F stands for set of all atoms (boolean vars) FACTS
  - O stands for set of all operators (actions) ACTIONS
  - $I \subseteq F$ stands for initial situation Initial Facts that are True
  - $G \subseteq F$ stands for goal situation Facts that must be True
    ≠ not a full set
- Operators $o \in O$ **represented** by
  - the Add list $Add(o) \subseteq F$ Make Facts True
  - the Delete list $Del(o) \subseteq F$ Make Facts False
  - the Precondition list $Pre(o) \subseteq F$ Required Facts True

(Oh no it's) The Blocksworld



- Propositions: $on(x, y)$, $onTable(x)$, $clear(x)$, $holding(x)$, $armEmpty()$.
- Initial state: $\{onTable(E), clear(E), \ldots, onTable(C), on(D, C), clear(D), armEmpty()\}$.
- Goal: $\{on(E, C), on(C, A), on(B, D)\}$.
- Actions: $stack(x, y)$, $unstack(x, y)$, $putdown(x)$, $pickup(x)$.
- $stack(x, y)$?

Unstack(x, y)

Pre : { On(x, y), armEmpty(),
clear(x) }

Add : { holding(x), clear(y) }

Stack(x, y) : { Add : {on(x, y), armEmpty(), clear(x)}
Delete : { holding(x), clear(y) }
Pre : { holding(x), clear(y) }
}

del tends to be similar or subset of Pre

Pre : at(x, y) connected(x, y, x', y')
Del : at(x, y)