Assignment

1. State Space description for Multipeg towers of Hanoi problem

Say n pegs would be with x disks:

$\langle x, 0, \ldots, 0 \rangle$ – Initial state

$\langle 0, 0, \ldots, x \rangle$ – Final state

Any state in the game can be represented as

$$\langle x_1, \ldots, x_k \rangle \text{ where } \left\{ \begin{array}{l} x_1 + x_2 + \ldots + x_k = x \\ k = n \end{array} \right\}$$

2 a) Matrix Chain Multiplication on 4 matrices

Let A be 10×30

B be 30×5

C be 5×60

For 3 matrices we can have (AB) C or A (BC) and compare the number of multiplications

Similarily for four matrices, we can extend the above as:

let A be 40×20, B be 20×30, C be 30×10 and D be 10×30

So calculate A (BCD), (AB) (CD) and (ABC) D based on these three more subproblems can be created and finally we get an optimum state of least number of multiplications with the state:

(A (BC)) D

$= 20 \times 30 \times 10 + 40 \times 20 \times 10 + 40 \times 10 \times 30$

$= 26000$

So we can have all the following configuration of a n matrix chain multiplication and have a subproblems broken into smaller sub problems.

b) State Space Definition of Matrix Chain Multiplication:

$$< M_m : M_n \,,\, M_i : M_n \,,\, min(M_m : M_n) \,,\, M_{n-1} M_n >$$

$$m <= n$$

Here we have specified states, set of start states, transition f^n c
Final state set.

c) DFS does not always gurantee a sol^n always because if
branching factor of the tree (search) is more then if the st
Space is infinite and finite branching, then DFS might go dou
never terminate while the goal node might be at some level
than it (above it & towards right).

d) Iterative deepening search is just like DFS with BFS It re
linear memory like DFS and searches for the goal state with s
Solution. The computation involves generation of tree at all all le
iteratively.

Iterative deepening search number of expansions are with depth = d
branching factor = b are:

$$\sum_{i=0}^{d} (d+1-i) \times b^i$$

Average no of expansions DFS = $\left(d + 1 + \frac{b^d - 1}{b-1}\right)/2$

Average no of expansions BFS = $\left(1 - b^d + \frac{b^d - 1}{b-1}\right)/2$

b = branching factor

d = depth

3 (a) A^{α} optimal sol^n proved by contradiction

Assume g is an optimal state with path cost $f(g)$ S is a sub optimal state with path cost $g(S) > f(g)$ n is node on an optimal path to g. Assume A^{a} selects S instead of g from Open list.

Since h is admissible $f(g) \geq f(n)$

n not chosen over S for expansion then

$f(n) \geq f(S)$

Thus ~~[illegible]~~ $\rightarrow f(g) \geq f(S)$

Since S is goal state, $h(S)=0$

So $f(S) = g(S)$

$f(g) \geq g(S)$ contradicts statement that S is suboptimal so A^{α} only selects expansion node sol^n as optimal

(b) If a node in closed state is reopened, then it would lead the search to inconsistency.

As the nodes in the closed state are found out by best possible heuristic computations, the system is consistent and reopening of the node is not required

(c) Node removed from open the cost of the node from path is

$f(n) = g(n) + h(n)$

So if we get $g(n)$ edge cost from shortest path so:

$f(n') = g(n') + h(n)$ (through n')

$f(n') < f(n)$

So if our heuristic selects shortest path then node from open is moved to closed.

(d) $f(n) = g(n) + h(n)$

$f(n)$ less than optimal cost

$f(n) < h^{a}(n)$

$h(n)$ admissible heuristic fn.

$f(n)$ increases along path to goal, $h(n)$ never overestimates goal..

descendants of node n, n_i has $f(n_i) = f(n)$

descendants of node n, n_i having $f(n_i) > f(n)$

$f(n_i) \leq h^{a}(n_i)$

Node m has to verify if $f(m) \leq h^{a}(m)$

(e) A^{a} heuristic more accurate then bands will get focussed / stretc towards goal state and focus on optimal path is less.

$f(n) = g(n) + h(n)$

With accurate heuristics, expansion of nodes can be minimal as focu optimal path is narrow.