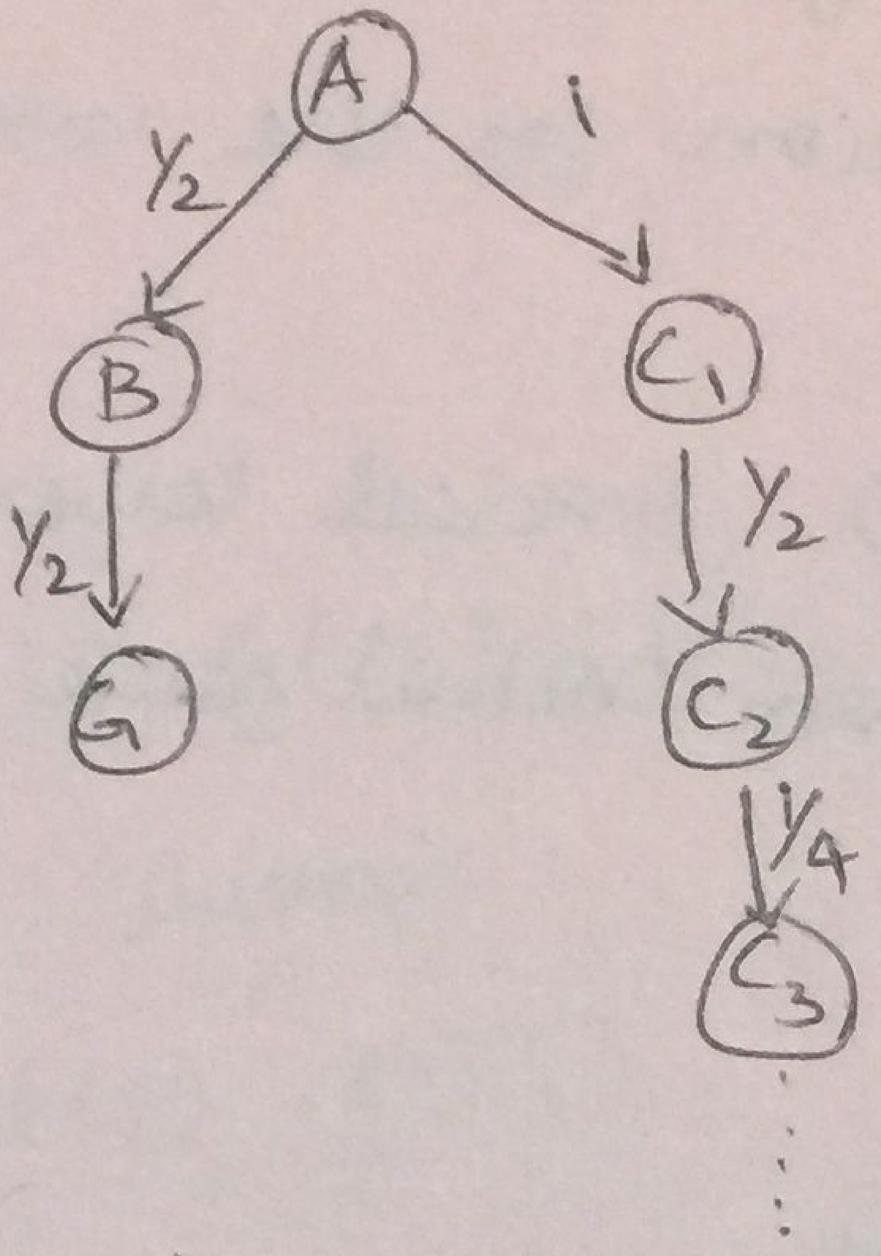
CS 540: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

HOMEWORK ASSIGNMENT #4

QUESTION 1: Inadmissible heuristic affects completeness.



For all states s,

$h(s) = 0$, where $s \neq B$

$h(B) = 100$.

1) For any state s, an admissible heuristic function $h(s)$ should be a lower bound of the true cost ($h^*(s)$) of reaching the goal from state s (i.e.) $h(s) \leq h^*(s)$

when $s = B$, $\quad h(B) \leq h^*(B)$ for $h(B)$ to be admissible.

True cost of reaching the goal G from B, $h^*(B) = 1/2$

$\therefore h(B) \leq 1/2$

Also, admissible heuristic functions are non-negative

$\therefore h(B) \geq 0$

$\therefore$ Admissible range of $h(B)$ is $[0, 1/2]$

$\Rightarrow 0 \leq h(B) \leq 1/2$

2) Let each state s be represented as $s^b < f, g, h >$ where

s → actual state

b → backpointer denoting the parent state of 's' that generated 's'.

g → Cost of reaching 's' from initial state

h → Heuristic function for the cost of reaching the goal state from 's'.

f: $f(s) = g(s) + h(s)$ (Overall lower bound of the cost path for reaching a goal state from initial state when the path passes through state 's').

Iteration Number	OPEN	CLOSED
Before 1st iteration	$A^{\phi} < 0, 0, 0 >$	{ }
End of Iteration 1	$B^{A} < 100.5, 0.5, 100 >$ $C_1^{A} < 1, 1, 0 >$	$A^{\phi} < 0, 0, 0 >$
End of Iteration 2	$B^{A} < 100.5, 0.5, 100 >$ $C_2^{C_1} < 1.5, 1.5, 0 >$	$A^{\phi} < 0, 0, 0 >$ $C_1^{A} < 1, 1, 0 >$
End of Iteration 3	$B^{A} < 100.5, 0.5, 100 >$ $C_3^{C_2} < 1.75, 1.75, 0 >$	$A^{\phi} < 0, 0, 0 >$ $C_1^{A} < 1, 1, 0 >$ $C_2^{C_1} < 1.5, 1.5, 0 >$
End of Iteration 4	$B^{A} < 100.5, 0.5, 100 >$ $C_4^{C_3} < 1.875, 1.875, 0 >$	$A^{\phi} < 0, 0, 0 >$, $C_1^{A} < 1, 1, 0 >$ $C_2^{C_1} < 1.5, 1.5, 0 >$ $C_3^{C_2} < 1.75, 1.75, 0 >$

Iteration Number	OPEN	CLOSED
End of Iterations 5	$B^{A}\langle 100.5, 0.5, 100\rangle$ $C_5^{C_4}\langle 1.9375, 1.9375, 0\rangle$	$A^{\phi}\langle 0,0,0\rangle$, $C_1^{A}\langle 1,1,0\rangle$ $C_2^{C_1}\langle 1.5, 1.5, 0\rangle$ $C_3^{C_2}\langle 1.75, 1.75, 0\rangle$ $C_4^{C_3}\langle 1.875, 1.875, 0\rangle$

3) $\lim_{i \to \infty} f(C_i)$

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

For all i, $h(C_i) = 0$ (Given)

$$\therefore f(C_i) = g(C_i)$$

$f(C_1) = 1$; $f(C_2) = f(C_1) + c(C_1, C_2)$ where $c(s, s')$ is the path cost from s to s'.

$$= 1 + 1/2$$

$$= 1.5$$

$$f(C_3) = f(C_2) + c(C_2, C_3)$$

$$= (1 + 1/2) + 1/4$$

$$\therefore f(C_i) = 1 + \frac{1}{2} + \dots + \frac{1}{2^{i-1}}$$

As $i \to \infty$, $f(C_i)$ is an infinite ~~arithmetic~~ geometric progression with ratio, $r = 1/2$. Since $|r| < 1$

$$\lim_{i \to \infty} f(C_i) = \frac{1}{1 - 1/2} = 2$$

$$\therefore \lim_{i \to \infty} f(C_i) = 2$$

4) Inorder for A* to find goal state G, the state G has to be placed in the OPEN list and be visited. In order for G to be added to the OPEN list, the state B has to be visited (and its successor G has to be added to OPEN). In the given graph, A* will never visit state B. The reason is as follows:

A* search picks the state with the smallest $f(s)$ value from the OPEN list. $f(B) = g(B) + h(B) = 0.5 + 100 = 100.5$.

But we just found that

$$\lim_{i \to \infty} f(c_i) = 2.$$

The maximum 'f' value on the right branch is 2. Therefore, B will never be visited before any of the C_i states. Since the C_i states are infinite in number, B will never be visited, G will never be generated and hence A* will not find G.

5) From 4) we know that inorder to find the state G,

Given $g(B) = 1/2$

$$f(B) < \max(f(C_i)) \quad i = 1, 2, 3, \ldots \infty$$

$$\therefore f(B) < 2 \Rightarrow g(B) + h(B) < 2 \Rightarrow h(B) < 1.5 \quad \text{—— ①}$$

From 1), we know the admissible range of $h(B)$ is

$$0 \leq h(B) \leq 1/2 \quad \text{—— ②}$$

i.e. An admissible 'h' is a non-negative value and less than the true cost of reaching the goal from the state.

From equations ① and ②, the range of $h(B)$ that allows A* to find G is

$$0 \leq h(B) < 1.5$$

But $0 \leq h(B) \leq 0.5$ is an admissible range for $h(B)$

$\therefore$ Inadmissible range of $h(B)$ that allows A^* to find G is $0.5 < h(B) < 1.5$ i.e. $h(B) \in (0.5, 1.5)$

6) From 5) we know that A^* search can find the optimal goal even with an inadmissible 'h' value. So, admissible 'h' is NOT a necessary condition for A^* search to find the optimal goal.

But, however if 'h' is admissible, we are assured of finding the optimal goal. Therefore admissibility of 'h' is SUFFICIENT for A^* search to find the optimal goal.

Admissible 'h' is SUFFICIENT BUT NOT NECESSARY.

QUESTION 2: Simulated Annealing.

Given, state $x \in Z$ (set of integers)

$$\text{successor}(x) \in [x-10, x+10]$$

Score function, $f(x) = \max\{4-|x|, 2-|x-6|, 2-|x+6|\}$

Probability of moving from x to y when $f(y) \leq f(x)$,

$$p = \exp\left\{\frac{-|f(x) - f(y)|}{T}\right\}$$

where $T(i) = 2(0.9)^i$ $i \rightarrow$ iteration number.

Given $x=2$ is the starting point.

i	x	y	f(x)	f(y)	T(i)	P	Z	Move ?
1	2	3	2	1	1.8	$e^{-1/1.8} = 0.5737$	0.102	YES $(z \le p)$
2	3	1	1	3	1.62	1 $(f(y) > f(x))$	0.223	YES
3	1	1	3	3	1.458	$e^{-0/1.458} = 1$	0.504	YES $(z \le p)$
4	1	4	3	0	1.312	$e^{-3/1.312} = 0.1016$	0.493	NO $(z > p)$
5	1	2	3	2	1.181	$e^{-1/1.181} = 0.4288$	0.312	YES $(z \le p)$
6	2	3	2	1	1.063	$e^{-1/1.063} = 0.3903$	0.508	NO $(z > p)$
7	2	4	2	0	0.957	$e^{-2/0.957} = 0.1237$	0.982	NO $(z > p)$
8	2	3	2	1	0.861	$e^{-1/0.861} = 0.3130$	0.887	NO $(z > p)$

QUESTION 3: CITY OF MADISON - Open data : Street trees.

1) Every state is represented as a permutation 'n' trees.

Total no. of states $= n!$ Factorial function $(n! = n(n-1)! \ ; \ 0! = 1)$

For Madison dataset, $n = 112511$

$\therefore$ Total no. of states $= (112511)!$

2) Neighbourhood of a state = Successors of the state

Given a state $s \ (t_1, t_2, \ldots t_n)$, successors are generated by swapping t_j, t_{j+1} for $j \in [1, n-1]$.

There are $(n-1)$ choices for 'j'. Therefore for every state there are $(n-1)$ successors.

Size of the neighbourhood $= (n-1)$

Fraction of the state space the neighbourhood covers $\Big\}$ = $\dfrac{\text{Size of neighbourhood}}{\text{Size of state space}}$

$$= \frac{(n-1)}{n!} = \frac{1}{n * (n-2)!}$$

For the city of Madison, the fraction of state space that the neighbourhood covers $\Big\}$ = $\dfrac{112510}{(112511)!} = \dfrac{1}{(112509)! * 112511}$

3) $n = 112511$

Number of states (for Madison dataset) $= (112511)!$

$$= 1.0403 \times 10^{519455}$$

$$\simeq 1 \times 10^{519455}$$

4) Worst case estimate for total distance

$$f(t_1, \ldots, t_n) = d(0, t_1) + \sum_{i=1}^{n-1} d(t_i, t_{i+1}) + d(t_n, 0)$$

Given $d(0, t_1) = d(t_n, 0) = d(t_i, t_{i+1}) = k = 10\text{ km}$ for $i = 1, \ldots, n-1$

$$f(t_1, \ldots t_n) = k + \sum_{i=1}^{n-1} k + k$$

$$f(t_1, \ldots t_n) = k + (n-1)k + k = (n+1)k$$

Since $n = 112511$, $k = 10\text{km}$

$$f(t_1, \ldots t_n) = (112512) * 10\text{ km}$$

Since, $1\text{ LD} = 384400\text{ km}$

$$f(t_1, \ldots t_n) = 2.9269\text{ LD} \simeq 3\text{ LD}$$

5) $f(t_1, \ldots t_n) = (n+1)\,k$ From (4)

where $n = 112511$ and $k = 10\,m$ for best case distance

$$f(t_1, \ldots t_n) = 112512 * 10\,m$$

$$f(t_1, \ldots t_n) = 1125.12\,km \approx 699.117 \text{ miles}$$

6) Speed of the inspector = 25 mph.

Distance covered by the inspector in 1 day (i.e 24 hrs) = 25 * 24 miles

= 600 miles.

<u>NO</u>. The inspector can drive for only 600 miles in a day and hence cannot finish the job even if the best case distance (of 10 m) is considered.

metres