

# 程序代写代做 CS编程辅导

Comp3620/Comp6320 Artificial Intelligence

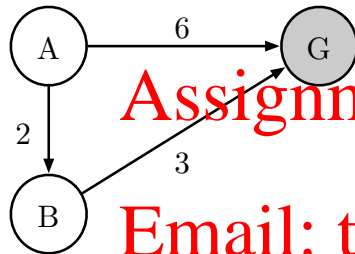
Tutorial 10: Heuristics, Game Tree Search



March 21–27, 2023

## Exercise 1 (heuristic function properties)

Consider the search problem shown on the left. It has only three states, and three directed edges.  $A$  is the start node and  $G$  is the goal node. To the right, four different heuristic functions are defined, numbered I through IV.



	$h(A)$	$h(B)$	$h(G)$
I	4	1	0
II	6	3	0
III	4	3	0
IV	5	2	0

Assignment Project Exam Help

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

For each heuristic, state whether it is admissible and whether it is consistent for the above search problem. Compare the informativeness of heuristics III and IV (i.e., state whether one of the heuristics dominates the other) and the informativeness of heuristics I and IV.

A heuristic  $h(n)$  is admissible if, at every node  $n$ , it does not overestimate the cost  $h^*(n)$  of the shortest path from  $n$  to the closest goal node, i.e.  $h(n) \leq h^*(n)$ . Note that this implies that  $h(n) = 0$  if  $n$  is a goal node. II is the only inadmissible heuristic, as it overestimates the cost from  $A$ :  $h(A) = 6$ , when the actual cost of the shortest path from  $A$  to  $G$  is 5. The other heuristics are admissible as can be easily verified from the table. For instance, for heuristic I,  $h(A) = 4 \leq 5$ ,  $h(B) = 1 \leq 3$  and  $h(G) = 0 \leq 0$ .

A heuristic is consistent if between any node  $n$  and its successor  $n'$  the heuristic doesn't decrease by more than the cost: i.e.  $h(n) - h(n') \leq c(n, n')$ . In this problem,  $h(G)$  is always 0, so making sure that the direct paths to the goal ( $A \rightarrow G$  and  $B \rightarrow G$ ) are consistent is the same as making sure that the heuristic value at the start of the path is admissible. Now for the path from  $A$  to  $B$ , the analysis is as follows:

- Heuristic I is not consistent:  $h(A) - h(B) = 4 - 1 = 3 \not\leq c(A, B) = 2$ .
- Heuristic II is not consistent: recall that consistency implies admissibility, and II is not admissible.
- Heuristic III is consistent:  $h(A) - h(B) = 4 - 3 = 1 \leq 2$
- Heuristic IV is not consistent:  $h(A) - h(B) = 5 - 2 = 3 \not\leq 2$

A heuristic  $h$  dominates another heuristic  $h'$  iff  $h(n) \geq h'(n)$  at all nodes  $n$ . We also say that  $h$  is more informed than  $h'$ . The table shows that there is no dominance relationship between III and IV, whilst IV dominates I.

## Exercise 2 (combining heuristics and heuristics for multiple goals)

There are green and red objects on a grid. An agent must collect *exactly one object of each color* to reach the goal. The actions are moving south, north, east or west, and are only applicable when they don't result in colliding with an obstacle (black) or exiting the grid. The agent collects an object when it reaches the cell at which this object is. The state of the problem is represented by a triple  $(a, G, R)$  where  $a$  is the location of the agent on the grid,  $G$  is the set of yet uncollected green objects, and  $R$  is the set of locations of yet uncollected red objects. Given two locations  $l_1$  and  $l_2$  on the grid,  $dist(l_1, l_2)$  returns the Manhattan distance between  $l_1$  and  $l_2$ .



Which of the following heuristics are admissible at any *non-goal* state  $s = (a, G, R)$  for this problem:

1. The sum of the Manhattan distances to the remaining objects?  

$$\sum_{o \in G \cup R} dist(a, o)$$

No. We only need to collect 2 objects (one of each kind). The sum of the Manhattan distances to all objects would greatly over-estimate the optimal cost of reaching just two of them. In fact, even if we had to collect all objects, it would be an over-estimate, as it assumes that we need to come back to the current location after collecting each object.

2. The number of remaining objects?  
 $|G| + |R|$

No. Again we only need to collect two objects. If there are a large number of objects and the distance to them is short, this heuristic will over-estimate the optimal cost to the goal.

3. The smallest Manhattan distance to any remaining objects?  

$$\min_{o \in G \cup R} dist(a, o)$$

Yes. This is admissible because if we haven't reached the goal yet, then we still need to at least collect one object of a given color, avoiding obstacles. We can relax this into a simpler problem by requiring we collect exactly one object of any color, and assuming that we can go through obstacles. The optimal solution to that simpler problem is the smallest Manhattan distance to any remaining object. Hence it is a lower bound for the optimal cost of the original problem of reaching the goal from  $s$ .

4. The maximum Manhattan distance between any two remaining objects?  

$$\max_{o_1 \in G \cup R, o_2 \in G \cup R} dist(o_1, o_2)$$

No. This is inadmissible, since adding extra objects does not increase the optimal solution for the problem but increases the heuristic.

5. The minimum Manhattan distance between any two remaining objects of opposite colors?  

$$\min_{o_1 \in G, o_2 \in R} dist(o_1, o_2)$$

No. It could be that we have already collected one object, in which case it is possible for the heuristic to overestimate the minimal cost to the goal.

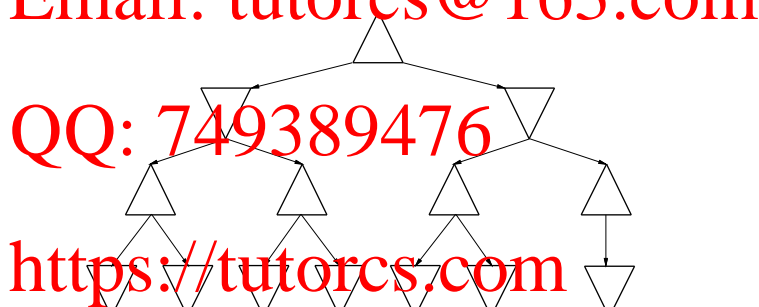
Some of the above heuristics, can be seen as the (often inadmissible) combination of several admissible heuristics for individual goals. This leads to the question of which combination of admissible heuristics are generally admissible. Let  $h(s)$ ,  $i(s)$  and  $j(s)$  be three admissible heuristics; indicate which combinations below are also guaranteed to be admissible:

1.  $\max(h(s), i(s), j(s))$
2.  $\min(h(s), i(s), j(s))$
3.  $\max(h(s), i(s))$
4.  $h(s) + i(s) + j(s)$
5.  $h(s) * i(s) * j(s)$
6.  $h(s)/3 + i(s)/3 + j(s)/3$

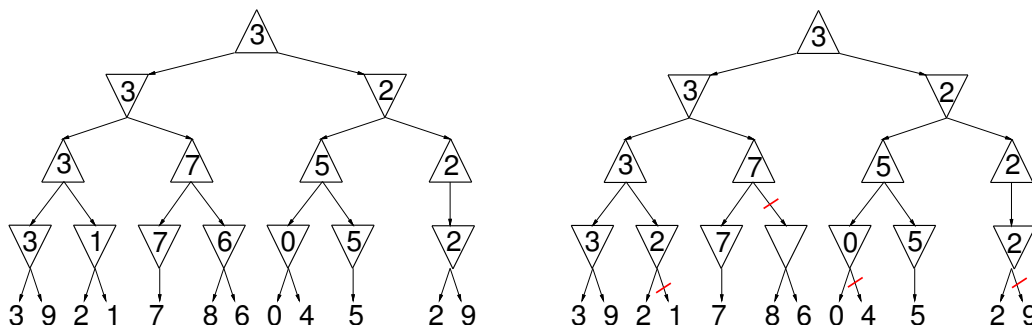
If a combined expression of heuristics is less than the maximum of the heuristics, then the combination is still admissible. Combinations 1. and 2. are admissible (min is always less than max). However, Combination 3.  $\max(h(s), i(s)) + j(s)$  could exceed the maximum of the 3 heuristics, and taking its maximum with  $j(s)$  would exceed the maximum of the three heuristics (note that replacing max with min would have been OK however). Sums aren't less than the maximum in general, except in special cases (e.g. if the three heuristics represent disjoint pattern database heuristics), so Combination 4. is not admissible. Products do not guarantee this either, unless again in special cases like if the heuristics are always less than 1, so Combination 5. is also not admissible. Combination 6. is admissible because  $h(s) + i(s) + j(s) \leq 3 \times \max(h(s), i(s), j(s))$  and the division by 3 guarantee that the combination is lower than the max.

### Exercise 3 (game tree search)

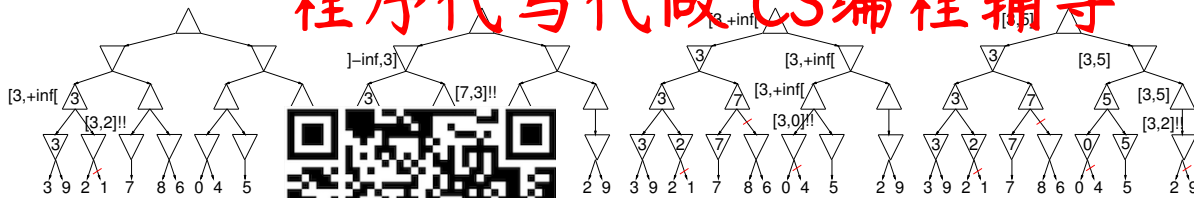
Apply Minimax without and with alpha-beta pruning to the following game tree. The root at the top of the tree is a max-node.



Minimax with and without alpha-beta pruning yields the following respective trees:



The points at which pruning occurs are the following:



1. The left-most min node has a leaf child with value 2.  $2 \leq \alpha = 3$ . The min-node just on the right has a leaf child (1).
2. The next bottom mid node has a leaf child with value 0. Since at that point  $\alpha = 3$ , we can prune the sibling of that leaf (4).
3. The next bottom mid node has a leaf child with value 2. Again,  $\alpha = 3$  at that point and the sibling of that leaf (9) can be pruned.
4. Finally the last bottom mid node has a leaf child with value 2. Again,  $\alpha = 3$  at that point and the sibling of that leaf (9) can be pruned.
5. The next bottom mid node has a leaf child with value 0. Since at that point  $\alpha = 3$ , we can prune the sibling of that leaf (4).
6. The next bottom mid node has a leaf child with value 2. Again,  $\alpha = 3$  at that point and the sibling of that leaf (9) can be pruned.
7. The sibling of that min-node can be pruned because its parent max node has a better child.
8. The next bottom mid node has a leaf child with value 0. Since at that point  $\alpha = 3$ , we can prune the sibling of that leaf (4).
9. Finally the last bottom mid node has a leaf child with value 2. Again,  $\alpha = 3$  at that point and the sibling of that leaf (9) can be pruned.

In general, which of the following assertions are correct about alpha-beta pruning?

1. It can reduce computation time by pruning portions of the game tree.  
True: In the worst case, no node is pruned, but in the best case (with an optimal node ordering) it could enable us to solve game trees with double depth as classic minimax.
2. It is generally faster than minimax but loses the guarantee of optimality.  
False: given a tree explored completely, minimax does not prune nodes that would lead to changing the decision of the player at the top of the tree, and returns the same value as minimax at the top of the tree. (However, in practice, alpha-beta pruning might enable us to explore more of a very deep tree than classic minimax, in which case the value at the top of the tree could be either lower or larger than the minimax value).
3. It always returns the same value as minimax for all nodes on the leftmost edge of the tree, assuming successor nodes are expanded from left to right.  
True: the leftmost branch is always completely evaluated in that case.
4. It always returns the same value as minimax for all nodes in the tree.  
False: the example above shows a case where the second min-node at the bottom has a higher value (2) with alpha-beta pruning than the minimax value (1). Similarly, a max node could have a lower value with alpha-beta pruning.