

# Homework 2

## Heuristic Search

*Due: 7:00pm on 2/20/08*

### Problem 2.1

Before you begin, read the slide handout (or at least the parts that explain the game, A\*, and Manhattan distance). Now perform A\* by hand on the following sliding tiles problem. If there is a tie as to which node on the “fringe” to expand next (they have the same f value), pick the node you get to by moving the lower numbered tile. You may use the demo to check your work on this problem, but if you only use the demo you will be missing a large part of the search tree.

1	2	3	4
5	7	15	8
9	6	11	0
13	10	14	12

Your handin should consist of a fairly large tree where each node has

- A sketch of the current state of the board, similar to the one above
- The current G value
- The current H value

This can either be done by hand or with a program like Dia, but be sure the end result is readable and clear

You only need to go to a depth of 4 (the initial state is depth 0).

**Solution:**

**Care of Lee Young Ho**

The solution can also be viewed (in more readable form) at

<http://www.cs.brown.edu/courses/cs141/asgn/sol/02slide.pdf>



**Problem 2.2**

Which of the following are admissible, given admissible heuristics  $h_1, h_2$ ? Which of the following are consistent, given consistent heuristics  $h_1, h_2$ ?

- $h(n) = \min\{h_1(n), h_2(n)\}$
- $h(n) = wh_1(n) + (1 - w)h_2(n)$ , where  $0 \leq w \leq 1$
- $h(n) = \max\{h_1(n), h_2(n)\}$

Please provide proofs.

**Solution:**

Care of Ethan Schreiber and Shrivaths Iyenga

**Admissibility**

By definition, if  $h^*(n)$  represents the true cost of getting from a node  $n$  to a goal node, then a heuristic  $h$  is admissible iff for all nodes  $n$ ,  $h(n) \leq h^*(n)$ . Thus, we are given that for all nodes  $n$ ,  $h_1(n) \leq h^*(n)$  and  $h_2(n) \leq h^*(n)$ .

a and c) **Given:**  $h(n) = \min\{h_1(n), h_2(n)\}$ .

**Given:**  $h(n) = \max\{h_1(n), h_2(n)\}$ .

$h(n) = \min\{h_1(n), h_2(n)\} \leq \max\{h_1(n), h_2(n)\} \leq h^*(n)$

**Thus,  $h(n) \leq h^*(n)$  i.e  $h(n)$  is admissible.**

b) **Given:**  $h(n) = wh_1(n) + (1 - w)h_2(n)$ .

$h_1(n) \leq h^*(n) \Rightarrow wh_1(n) \leq wh^*(n)$

$h_2(n) \leq h^*(n) \Rightarrow (1 - w)h_2(n) \leq (1 - w)h^*(n)$

$wh_1(n) + (1 - w)h_2(n) \leq wh^*(n) + (1 - w)h^*(n) = h^*(n) \Rightarrow h(n) \leq h^*(n)$

**$\Rightarrow h(n)$  is admissible.**

**Consistency**

By definition,  $h(n)$  is a consistent heuristic iff  $\forall n, n'$ , such that  $n$  is the parent of  $n'$ ,  $h(n) \leq c(n, n') + h(n')$  where  $c(n, n')$  is the cost of going from node  $n$  to  $n'$ .

Thus, we are given that  $\forall n, n'$ , such that  $n$  is the parent of  $n'$ ,  $h_1(n) \leq c(n, n') + h_1(n')$  and  $h_2(n) \leq c(n, n') + h_2(n')$ .

a) **Given:**  $h(n) = \min\{h_1(n), h_2(n)\}$ .

$$\begin{aligned} h(n) &= \min\{h_1(n), h_2(n)\} \\ &\leq \min\{h_1(n') + c(n, n'), h_2(n') + c(n, n')\} \\ &= \min\{h_1(n'), h_2(n')\} + c(n, n') \end{aligned}$$

If  $h_1(n) \leq h_2(n)$  then  $h(n) = h_1(n)$

If  $h_1(n') \leq h_2(n')$ ,  $h(n') = h_1(n')$  and since  $h_1(n)$  is consistent:

$$h(n) = h_1(n) \leq h_1(n') + c(n, n')$$

If  $h_1(n') \geq h_2(n')$ ,  $h(n') = h_2(n')$  and since  $h_2(n)$  is consistent:

$$h(n) = h_1(n) \leq h_2(n) \leq h_2(n') + c(n, n')$$

So regardless of whether  $h_1(n') \leq h_2(n')$  or  $h_2(n') \leq h_1(n')$ , we can say that:

$$h(n) \leq h(n') + c(n, n')$$

which is the definition of consistency.

By symmetry, we can provide the same proof for  $h_1(n) \geq h_2(n)$  and hence our proof is complete.

b) **Given:**  $h(n) = wh_1(n) + (1-w)h_2(n)$ .

By the definition of consistency, we can say:

$$h_1(n) \leq h_1(n') + c(n, n') \tag{1}$$

$$h_2(n) \leq h_2(n') + c(n, n') \tag{2}$$

$$\begin{aligned} h(n) &= wh_1(n) + (1-w)h_2(n) \\ \Rightarrow h(n) &\leq w[c(n, n') + h_1(n')] + (1-w)[c(n, n') + h_2(n')] \\ \Rightarrow h(n) &\leq \underbrace{wh_1(n') + (1-w)h_2(n')}_{h(n')} + c(n, n') \\ \Rightarrow h(n) &\leq h(n') + c(n, n') \end{aligned}$$

Thus,  $h(n)$  is consistent.

c) **Given:**  $h(n) = \max\{h_1(n), h_2(n)\}$ .

$$\begin{aligned} h(n) &= \max\{h_1(n), h_2(n)\} \\ &\leq \max\{h_1(n') + c(n, n'), h_2(n') + c(n, n')\} \\ &= \max\{h_1(n'), h_2(n')\} + c(n, n') \end{aligned}$$

If  $h_1(n) \geq h_2(n)$  then  $h(n) = h_1(n)$

Since  $h_1(n)$  is consistent:

$$h(n) = h_1(n) \leq h_1(n') + c(n, n')$$

Since  $h(n') = \max\{h_1(n'), h_2(n')\}$ , we have two choices. If  $h_1(n') \geq h_2(n')$ ,  $h(n') = h_1(n')$  and we have already shown that  $h(n) \leq h_1(n') + c(n, n')$ . Otherwise, if  $h_1(n') \leq h_2(n')$ ,  $h(n') = h_2(n')$  and then:

$$\begin{aligned} h(n) = h_1(n) &\leq h_1(n') + c(n, n') \\ &\leq h_2(n') + c(n, n') \\ &= h(n') + c(n, n') \end{aligned}$$

So regardless of whether  $h_1(n') \leq h_2(n')$  or  $h_2(n') \leq h_1(n')$ , we can say that:

$$h(n) \leq h(n') + c(n, n')$$

which is the definition of consistency.

By symmetry, we can provide the same proof for  $h_2(n) \geq h_1(n)$  and hence our proof is complete.

### Problem 2.3

Consider the algorithm  $wA^*$ , which is a variant of  $A^*$  search that uses the following weighted cost function: for some  $w \geq 1$ ,

$$f_w(n) = g(n) + wh(n)$$

As usual,  $g$  is the cost from the root node to  $n$  and  $h$  is an admissible heuristic.

- a) Prove that the goal node  $m^*$  returned by  $wA^*$  search on trees is within a factor of  $w$  of the optimal goal  $n^*$ : i.e.,  $g(m^*) \leq wg(n^*)$ .
- b) The  $wA^*$  algorithm uses a weighted cost function that increases the value of the heuristic function  $h$ , hoping to proceed more directly towards a goal. An alternative is to ignore  $g$  entirely, simply letting  $f = h$ . Give two advantages of  $wA^*$  over this alternative.

**Solution:**

Care of Payman Yadollahpour and Jie Mao

a) Suppose  $wA^*$  returns goal node  $m^*$  before it returns optimal goal node  $n^*$ . It follows that there exists node  $n$  on the priority queue that is also on the path to  $n^*$  s.t.  $f_w(m^*) = f_w(n)$ . But then  $g(m^*) \leq wg(n^*)$  by the following reasoning:

$$\begin{aligned}
 g(m^*) &= g(m^*) + wh(m^*) && \text{by } w \geq 1 \text{ and } h(m^*) = 0 \\
 &= f_w(m^*) && \text{by definition} \\
 &\leq f_w(n) && \text{by assumption} \\
 &= g(n) + wh(n) && \text{by definition} \\
 &\leq g(n) + h^*(n) && \text{by admissibility} \\
 &= g(n) + h^*(n) + (w - 1)h^*(n) \\
 &= g(n^*) + (w - 1)h^*(n) && \text{by } g(n) + h^*(n) = g(n^*) \\
 &\leq g(n^*) + (w - 1)g(n^*) && \text{by } h^*(n) \leq g(n^*) \\
 &= wg(n^*)
 \end{aligned}$$

b) **First advantage:**  $wA^*$  guarantees that the cost of the goal node is bounded by  $g(m^*) \leq wg(n^*)$  (as we just proved in part a), while the alternative ( $f = h$ ) is not bounded. It can return a goal node with much larger cost than the optimal goal node.

**Second advantage:**  $wA^*$  guarantees completeness (given all edge costs greater than some positive constant  $\epsilon$ ) because there do not exist infinitely many nodes  $n$  s.t.  $f(n) < f^*$  ( $g$  will grow along the depth of the path and eventually make  $f(n) > f^*$ ). However, the alternative ( $f = h$ ) is not complete because when there exists cycles in the path, it is possible for  $f(n) < f^*$  for all node on the cycle.

### Problem 2.4

Connect Four is a classic struggle of red ( $X$ ) against black ( $O$ ) for domination in a gridded world. It is also a fun game (if you are unfamiliar with the rules, ask google). You are red ( $X$ ). What is your next move? Argue why. Describe a system for determining your next move.

				$O$		
$X$				$X$	$X$	
$O$	$O$		$O$	$X$	$O$	
$O$	$X$		$X$	$O$	$O$	$X$

**Solution:**

For convenience, label the columns from left to right as  $A, B, C, D, E, F,$  and  $G$ .

We want to move in  $D$ . This guarantees that no matter where our opponent moves, we can still win the game (the best move given our opponent makes his best move...

minimax). If  $O$  moves in  $C$ , then we can move in  $C$  and win. If  $O$  moves anywhere else, we can move in  $G$ . This threatens 4-in-a-row horizontally, forcing  $0$  to move in  $G$ . Unfortunately for  $O$  but fortunately for us, we can respond by going in  $G$ , making 4-in-a-row diagonally.

### Problem 2.5

Given a finite set  $S$  and a binary operation  $*$  :  $S \times S \rightarrow S$ , the tuple  $(S, *)$  is called a group if and only if the following properties hold:

- Neutral Element:  $\exists n \in S \forall a \in S : n * a = a = a * n$
- Associative:  $\forall a, b, c \in S : (a * b) * c = a * (b * c)$
- Inverse:  $\forall a \in S : \exists b \in S : a * b = n = b * a$

Prove the following:

- a. The neutral element is unique.
- b.  $\forall a \in S$ , it's inverse element  $b$  with  $b * a = n = a * b$  is unique.

Each of these can be done in less than 5 lines.

#### Solution:

**suppose**  $\exists w \in S$  s.t.  $w * a = a = a * w \forall a \in S$

**then**  $w = w * n = n$

**so**  $w = n$  and **n is unique**

**Suppose**  $a * b = n = b * a$  and  $a * c = n = c * a$  **then**

$b = b * n = b * (a * c) = (b * a) * c = n * c = c$  **so**  $b = c$  and **the inverse is unique**