**CMPE 480 INTRODUCTION TO ARTIFICIAL INTELLIGENCE**

**MIDTERM ANSWERS**

1. The problem types are “formal tasks”, “mundane tasks”, and “expert tasks”. See the lecture notes for the complete answer to this question.
2. State space: The set of ordered pairs (x,y), where x ϵ [0,5] and y ϵ [0,3].

Start state: (0,0)

Goal state: (2,n), for any n.

Rules:

1. (x,y) 🡪 (5,y)

if x<5

1. (x,y) 🡪 (x,3)

if y<3

1. (x,y) 🡪 (0,y)

if x>0

1. (x,y) 🡪 (x,0)

if y>0

1. (x,y) 🡪 (min(5,x+y), max(0,y-(5-x)))

if y>0

1. (x,y) 🡪 (max(0,x-(3-y)), min(3,x+y))

if x>0

Heuristic: h′(x,y) = |x-2|

First step:

(0,0)

R1 R2

(5,0) (0,3)

h′=3 h′=2

Since h′ for state (0,3) is better, we continue with state (0,3).

(0,0)

R1 R2

(5,0) (0,3)

h′=3 h′=2

R1 R4 R5

(5,3) (0,0) (3,0)

h′=3 h′=2 h′=1

We have four states not yet expanded. State (3,0) is the best one, we continue with state (3,0).

(0,0)

R1 R2

(5,0) (0,3)

h′=3 h′=2

R1 R4 R5

(5,3) (0,0) (3,0)

h′=3 h′=2 h′=1

R1 R2 R3 R6

(5,0) (3,3) (0,0) (0,3)

h′=3 h′=1 h′=2 h′=2

Now, state (3,3) is the best one and we continue with it. The search continues in this way.

1. Label the rows from top to bottom as 1,2,...,6. Label the columns from left to right as 0,1,...,8, where 0 and 8 are used, respectively, to denote the initial and final positions. Denote a state as (x,y), where x is the row number and y is the column number. The initial state is (4,0) and the final state is (4,8).
2. (4,0) distance=8

(4,1) distance=7

(5,1) distance>7 (4,2) distance=6

(4,1) distance=7 (3,2) distance>6

Then the algorithm stops

From the initial state (4,0), there is only one possible move (4,1). The robot moves to (4,1) since it is better than the current state. From (4,1), there are two possible next states, of which one ((4,2)) is better than the current state. From (4,2), there are no better next states. Thus, the algorithm terminates without finding a solution. The robot is at a local maximum.

1. Suppose that g=0 and we use only the h function which is the given distance heuristic. Also suppose that we check the duplicate states and do not expand duplicate states again (shown with X below).

(4,0) distance=8

(4,1) distance=7

(5,1) distance=7.07 (4,2) distance=6

(4,1) distance=7 (3,2) distance=6.08

X

(4,2) distance=6 (2,2) distance=6.32

X

(1,2) distance=6.71 (3,2) distance=6.08

X

(1,1) distance=7.62 (2,2) distance=6.32 (1,3) distance=5.83

X

...

From the initial state (4,0), there is only one possible move (4,1). The robot moves to (4,1). From (4,1), there are two possible next states, of which (4,2) is better than (5,1). From (4,2), there are two possible moves, where (4,1) is a duplicate state and removed. Now we have two alternatives, (5,1) and (3,2). Since (3,2) is better, we continue from (3,2). It has one non-duplicate next state, (2,2). Again we have two alternatives to continue, (5,1) and (2,2). We continue from (2,2), which has one next state which is (1,2). From (5,1) and (1,2), we choose (1,2). Now we have (5,1), (1,1) and (1,3). We continue from (1,3). The process continues in this way and it finds a solution. If we accept the optimality criterion as the number of moves, this is not an optimal solution.

1. h′2 may not be admissible, because h′2 ≥ h′ when h′ ≥ 1, so it may exceed the optimal distance to goal. ≤ h′ for h′ ≥ 1, so it is admissible (assuming integer values for the heuristic, which is typical). It will likely work worse than h′, though, because its estimate is farther from the optimal value (so it is a worse estimate of the cost-to-go).
2. 8-puzzle problem.

h′ = number of tiles not in the correct position.

Clearly, to reach from any state s to the goal state, at least h′ number of steps will be necessary. Because, this number of tiles are in incorrect positions and for each tile at least one step will be necessary to move it to the correct position. In fact, the number of steps required to solve the problem will be much more than this heuristic value.

1. Consider the path s1-s2-...-sn-1-sn. Suppose that h′ is a monotonic heuristic.

By the monotonicity property,

h′ (s1) – h′ (s2) ≤ h (s1) – h (s2)

h′ (s2) – h′ (s3) ≤ h (s2) – h (s3)

...

h′ (sn-1) – h′ (sn) ≤ h (sn-1) – h (sn)

where h is the actual (unknown) heuristic function.

When we simplify the terms on the left and on the right, we have

h′ (s1) – h′ (sn) ≤ h (s1) – h (sn)

h′ (sn) = 0 and h (sn) = 0, since sn is the goal state. So, we have

h′ (s1) ≤ h (s1).

That is, for any state s1, the estimated distance from s1 to the goal is less than or equal to the actual distance from s1 to the goal. That is, h′ underestimates h. This is the definition of admissibility. So, h′ is admissible.