National University of Computer & Emerging Sciences

(Karachi Campus)

Midterm Examination – Spring 2014

Artificial Intelligence (CS401)

Time Allowed: 60 Min. Max. Marks: 50

Dated: February 24, 2014

Instructions: Attempt all questions. Be to the point, there is a penalty for wild guesses. Draw neat and clean diagram/code where necessary.

Question No. 1	[30 Points] [Time: 40 Min.]

Consider our Robo is standing at some point in a grid consisting of n*m cells. Robo is standing at some start state and needs to find a path to the goal state. Our Robo can only move (UP, LEFT, RIGHT and DOWN) one cell at a time, with a cost of 1. There are some cells with a blockage (*) so the Robo has to detour around these cells. Consider the following example grid:

	G				
	*	*	*	*	
			*		
		S			

Each cell has a unique (i,j), considered as a node for exploration in search space.

a. If we use Breadth-first search for finding the path in the above example. Consider the frontier contains S as initial node, give the order of the cells explored if our Robo preferred to move UP, RIGHT, LEFT and DOWN as preferred directions (operators/actions). [7]

In BFS, we used a Queue to hold frontier, hence when the BFS start, it will only contains Frontier ={ [S,(2,0)]}, in state representation, we can hold the cell Number along with coordinates (x,y) assuming (0,0) as lower left corner of the given grid.
There are four operators (UP, RIGHT, LEFT, DOWN)
Step 1: we take S out of Frontier, it is not a goal state; we need to expand it, in preferred order. Hence the Frontier ={ [1,(2,1)], [2,(3,0)], [3, (1,0)]} the operator Down will generate a state [4,(2,-1)] which is not valid. If you want to maintain the explored list, you will have Explored ={[S, (2,0)]}, otherwise just hold another variable Visit in state representation.

	G				
	*	*	*	*	
		1	*		
	3	X	2		

Step 2: we take [1,(2,1)] out of frontier as it is not a goal state, we expand it as well. The sequence of move will be (UP, RIGHT, LEFT, DOWN), hence the expanded state will be [5, (2,2)], [6,(1,3)], [7,(1,2)] and [8,(2,0)] only [7,(1,2)] will goto Frontier as [5, (2,2)], and [6,(1,3)] are blocked. [8, (2, 0)] is already visited.

	G				
	*	*	*	*	
	7	X	*		
	3	X	2		

Step 3: Now, Frontier ={ [2,(3,0)], [3, (1,0)], [7,(1,2)] } , we take [2,(3,0)] so the change in grid will be

	G				
	*	*	*	*	
	7	X	*		
	3	X	X	10	

Step 4: Frontier ={ [3,(1,0)], [7,(1,2)], [10, (4,0)]

	G				
	*	*	*	*	
	7	X	*		
	3	X	X	10	

Step 5: Frontier ={ [7,(1,2)], [10, (4,0), [15,(0,0)]}

	G				
	*	*	*	*	
	7	X	*		
15	X	X	X	10	

Step 6: Frontier ={ [10, (4,0), [15,(0,0)],[19,(0,1)] }

	G				
	*	*	*	*	
19	X	X	*		
15	X	X	X	10	

Step 7: Frontier ={ [15,(0,0)],[19,(0,1)], [21,(4,1)], [22, (0,5)] }

	G				
	*	*	*	*	
19	X	X	*	21	
15	X	X	X	X	22

Step 8: Frontier ={ [19,(0,1)], [21,(4,1)], [22, (0,5)] }

	G				
	*	*	*	*	
19	X	X	*	21	
X	X	X	X	X	22

Step 9: Frontier ={ [21,(4,1)], [22, (0,5)] , [25,(0,2)]}

	G				
25	*	*	*	*	
X	X	X	*	21	30
X	X	X	X	X	22

Step 10: Frontier ={ [25,(0,2)],[30,(5,1)]}

	G				
25	*	*	*	*	
X	X	X	*	X	30
X	X	X	X	X	X

Step 11: Frontier ={[30,(5,1)],[41,(0,3)]}

	G				
41					
X	*	*	*	*	
X	X	X	*	X	30
X	X	X	X	X	X

Step 12: Frontier ={[41,(0,3)], [45,(5,2)]}

	G				
41					
X	*	*	*	*	45
X	X	X	*	X	X
X	X	X	X	X	X

Step 13: Frontier ={ [45,(5,2)], [49,(0,4)], [50,(1,3)]}

49	G				
X	50				
X	*	*	*	*	45
X	X	X	*	X	X
X	X	X	X	X	X

Step 14: Frontier ={ [49,(0,4)], [50,(1,3)],[53,(5,3)]}

49	G				
X	50				53
X	*	*	*	*	X
X	X	X	*	X	X
X	X	X	X	X	X

Step 15: Frontier ={ [50,(1,3)],[53,(5,3)],[G,(1,4)]}

X	G				
X	50				53
X	*	*	*	*	X
X	X	X	*	X	X
X	X	X	X	X	X

Step 16: Frontier ={ [53,(5,3)],[G,(1,4)], [66,(2,3)]}

X	G				
X	X	66			53
X	*	*	*	*	X
X	X	X	*	X	X
X	X	X	X	X	X

Here we assume that node [50,(1,3)] will expand the node G again, as it is already in the Frontier there is no need to insert it again.

Step 17: Frontier ={ [G,(1,4)], [66,(2,3)], [69,(5,4)], [71, (4,3)]}

X	G				69
X	X	66		71	X
X	*	*	*	*	X
X	X	X	*	X	X
X	X	X	X	X	X

Step 18: Frontier ={ [66,(2,3)], [69,(5,4)], [71, (4,3)]}

When Node [G,(1,4)] extracted from the Frontier, goal test is performed and SUCCESS is reported from there.

X	G				69
X	X	66		71	X
X	*	*	*	*	X
X	X	X	*	X	X
X	X	X	X	X	X

b. If we use Depth-first search for finding the path in the above example. Consider the frontier contains S as initial node, give the order of the cells explored if our Robo preferred to move RIGHT, UP, LEFT, and DOWN as preferred operators. [7]

In DFS, we used a Stack as frontier, hence when the DFS start, it will only contains Frontier ={ [S,(2,0)]}, in state representation, we can hold the cell Number (in explored sequence) along with coordinates (x,y) assuming (0,0) as lower left corner of the given grid.
There are four operators (RIGHT, UP, LEFT, DOWN)

Step 1: we take S out of Frontier, it is not a goal state, we need to expand it, in preferred order. Hence the Frontier ={ [1,(1,0)], [2,(2,1)], [3,(3,0)]} the operator Down will generate a state [4,(2,-1)] which is not valid. If you want to maintain the explored list, you will have Explored ={[S, (2,0)]}, otherwise just hold another variable Visit in state representation.

	G				
	*	*	*	*	
		2	*		
	3	X	1		

Step 2: Now frontier will be { [1,(3,0)], [2,(2,1)], [3,(1,0)]} as it is a stack, the last inserted node will be extracted and goal test is performed. As it is not a goal state, the test will fail. We need to expand the node.

	G				
	*	*	*	*	
		2	*		
	3	X	1		

Step 3: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [7,(0,0)]}

	G				
	*	*	*	*	
	6	2	*		
7	X	X	1		

Step 4: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [10,(0,1)]}

	G				
	*	*	*	*	
10	6	2	*		
X	X	X	1		

Step 5: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [14,(0,2)]}

	G				
14	*	*	*	*	
X	6	2	*		
X	X	X	1		

Step 6: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [18,(0,3)]}

	G				
18					
X	*	*	*	*	
X	6	2	*		
X	X	X	1		

Step 7: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [21,(1,3)],[22,(0,4)]}

22	G				
X	21				
X	*	*	*	*	
X	6	2	*		
X	X	X	1		

Step 8: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [21,(1,3)],[G,(1,4)]}

X	G				
X	21				
X	*	*	*	*	
X	6	2	*		
X	X	X	1		

Step 9: { [1,(3,0)], [2,(2,1)], [6,(1,1)], [21,(1,3)],[G,(1,4)]}

X	G				
X	21				
X	*	*	*	*	
X	6	2	*		
X	X	X	1		

As G is a goal node, we report SUCCESS here.

c. Give the sequence of cells explored (search path) if our Robo perform Greedy-Best First search in this scenario. Assume that the greedy sense is the minimum Manhattan distance. Manhattan distance can be used as the heuristic function. That is, h(n) for any cell (i,j) is the Manhattan distance from cell(i,j) to cell(i*,j*)=G. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. [8]

In GBFS, we used a priority queue with priority on h(n) as frontier, hence when the GBFS start, it will only contains Frontier ={ [S,(2,0),5]}, in state representation, we can hold the cell Number (in explored sequence) , coordinates (x,y) assuming (0,0) as lower left corner of the given grid and the value of h(n) for each node.
There are four operators (UP, LEFT, RIGHT, DOWN)

Step 1: we take S out of Frontier, it is not a goal state; we need to expand it, in preferred order (UP, LEFT, RIGHT, DOWN). Hence the Frontier ={ [1,(2,1),4], [2,(1,0),4], [3,(3,0),6]} the operator Down will generate a state [4,(2,-1)] which is not valid. If you want to maintain the explored list, you will have Explored ={[S, (2,0),5]}, otherwise just hold another variable Visit in state representation.

	G				
	*	*	*	*	
		1	*		
	2	X	3		

Step 2: Hence the Frontier ={ [1,(2,1),4], [2,(1,0),4], [3,(3,0),6]} , we next expand [1,(2,1),4], hence [6,(1,1),3] is added to Frontier

	G				
	*	*	*	*	
	6	X	*		
	2	X	3		

Step 3: Hence the Frontier ={ [6,(1,1),3], [2,(1,0),4], [3,(3,0),6]} , we next expand [6,(1,1),3] , and we add [10, (0,1),4] to the Frontier.

	G				
	*	*	*	*	
10	X	X	*		
	2	X	3		

Step 4: Hence the Frontier ={ [10, (0,1),4], [2,(1,0),4], [3,(3,0),6]} , we next expand [13,(0,2),3] , and [16, (0,0),5] are added to the Frontier.

	G				
13	*	*	*	*	
X	X	X	*		
16	2	X	3		

Step 5: Hence the Frontier ={ [13,(0,2),3], [2,(1,0),4], [16, (0,0),5], [3,(3,0),6]} , we next expand [13,(0,2),3] and add [17,(0,3),2]

	G				
17					
X	*	*	*	*	
X	X	X	*		
16	2	X	3		

Step 6: Hence the Frontier ={ [17,(0,3),2], [2,(1,0),4], [16, (0,0),5], [3,(3,0),6]} , we next expand [17,(0,3),2], we will add [22,(0,4),1] and [23,(1,3), 1]

21	G				
X	23				
X	*	*	*	*	
X	X	X	*		
16	2	X	3		

Step 7: Hence the Frontier ={ [21,(0,4),1], [23,(1,3), 1] , [2,(1,0),4], [16, (0,0),5], [3,(3,0),6]} , we next expand [21,(0,4),1],

21	G				
X	23				
X	*	*	*	*	
X	X	X	*		
16	2	X	3		

We will add [G, (1,4),0] to the Frontier.

Step 8: Hence the Frontier ={ [G, (1,4),0], [23,(1,3), 1] , [2,(1,0),4], [16, (0,0),5], [3,(3,0),6]} , we next expand [G,(1,4),0],

21	G				
X	23				
X	*	*	*	*	
X	X	X	*		
16	2	X	3		

As G is a goal node, we report SUCCESS here.

d. Give the sequence of cells explored (search path) if our Robo perform A* search in this scenario. Assume that the Manhattan distance can be used as the heuristic function- h(n) for any cell. The cost to reach a cell (i,j) = i+j that is the sum of indexes i and j. Therefore, A* uses f(n)=g(n)+h(n); [8]

In A*, we used a priority queue with priority on f(n) as frontier, hence when the A* start, it will only contains Frontier ={ [S,(2,0),2+5]}, in state representation, we can hold the cell Number (in explored sequence) , coordinates (x,y) assuming (0,0) as lower left corner of the given grid and the value of f(n) which is g(n)+h(n); for each node. There are four operators (UP, LEFT, RIGHT, DOWN)

Step 1: we take S out of Frontier, it is not a goal state; we need to expand it, in preferred order (UP, LEFT, RIGHT, DOWN). Hence the Frontier ={ [2,(2,0), 2+4], [1,(2,1),3+4], [3,(3,0),3+6]} the operator Down will generate a state [4,(2,-1)] which is not valid. If you want to maintain the explored list, you will have Explored ={[S, (2,0),2+5]}, otherwise just hold another variable Visit in state representation.

	G				
	*	*	*	*	
		1	*		
	2	X	3		

Step 2: The Frontier ={ [2,(2,0), 2+4], [1,(2,1),3+4], [3,(3,0),3+6]} we will take out [2,(2,0), 2+4], as it is not a goal state, we expand it. The Frontier { [5, (1,1), 2+ 3] [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}

	G				
	*	*	*	*	
	5	1	*		
6	X	X	3		

Step 3: The Frontier { [5, (1,1), 2+ 3], [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, now we expand [5, (1,1), 2+ 3], we will get Frontier { [10, (0,1), 1+ 4], [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]},

	G				
	*	*	*	*	
10	X	1	*		
6	X	X	3		

Step 4: The Frontier { [10, (0,1), 1+ 4], [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, we will expand [10, (0,1), 1+ 4], the next [13,(0,2), 2+3] is added.

	G				
13	*	*	*	*	
X	X	1	*		
6	X	X	3		

Step 5: The Frontier { [13,(0,2), 2+3] , [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, we will expand [13, (0,2), 2+3], the next [17,(0,3), 3+2] is added.

	G				
17					
X	*	*	*	*	
X	X	1	*		
6	X	X	3		

Step 6: The Frontier { [17,(0,3), 3+2] , [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, we will expand [17, (0,3), 3+2], the next [21,(0,4), 4+1] is added.

21	G				
X					
X	*	*	*	*	
X	X	1	*		
6	X	X	3		

Step 7: The Frontier { [21,(0,4), 4+1] , [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, we will expand [21,(0,4), 4+1] , and [G,(1,4), 5+0] is added.

21	G				
X					
X	*	*	*	*	
X	X	1	*		
6	X	X	3		

Step 7: The Frontier { [G,(1,4), 5+0] , [6, (0,0), 0+5], [1,(2,1),3+4], [3,(3,0),3+6]}, we will expand [G,(1,4), 5+0] as it is a goal state , we report SUCCESS.

Question No. 2	[20 Points] [Time: 20 Min.]

1. Considering the task environment where an agent performs shopping for used AI books on the Internet. Outline the PEAS Description of the task environment. [4]

Performance Measure	Obtained search results on query string; Minimize search time Fetch most relevant results.
Environment	Internet – WWW, Http request / response.
Actuators	Follow links, submit data on web-forms, display contents of web pages;
Sensors	Web pages, users requests

2. Illustrate the difference between Reflexive Agent vs. Model-Based Agent. [4]

Simple Reflexive Agent	**Model Based Agent**
Simplex Reflexive Agent performs action based on the precept that it received and programmed for. It has very limited intelligence	Agent can have model information about the evolving environment. The model gives the agents predicted information about the world. Agent becomes more informed.

3. Consider the random-restart hill climb. Given many enough restarts, this algorithm is likely with a probability approaching 1 to be optimal. Would it, and how would it if it did, affect the optimality of random-restart hill climb if you started it n times in parallel instead of n times serially? [4]

It would not affect the search as both have needed finitely many tries.

4. Considering a task environment of a “Part Picking Robot”, classify this environment as (full/partial observable, deterministic /stochastic, sequential/episodic, static/dynamic, discrete/continuous, single/multi agent). [4]

   The environment of “Part Picking Robot” is partially observable, stochastic, episodic, dynamic, continuous and single agent type.

5. Since hill climbing is extremely sensitive to initialization, you have decided to smarten it up with a way to be in the better bits of the state space at the time of initialization. The hypothesis is that if you had an idea of what the state space landscape looked like, you could design a guided-restart hill climb. In order to do this, you decide to explore the landscape by sampling it and fitting a polynomial sheet to it. Since you cannot assume the nature of a search landscape, this sheet is at least piecewise continuous.

   (a) Please suggest from this point on how would your method select (re-)start locations for the hill climb. [2]

   When we sample few points from the one-dimensional state space of any problem, we can establish a polynomial for approximating the landscape.

   (b) If the fitted sheet p(X) is continuously differentiable all over, suggest a way to reach all maxima/minima without having to climb or descend to them. [2]

   The p(X) is continuous and differentiable at all points, at some point where the value of the differential is zero, our maxima/minima situated at that point.

<Best of Luck>