Programming Assignment 1 : Problem Solving Agent

Problem Description

A 10 by 10 tiled floor is cleaned by an intelligent vacuum cleaner (Fig.1). The vacuum cleaner has two sensors to perceive the position and dirt of the tile. The actuator actions are MR, ML, MU and MD respectively for movements in right, left, up, and down directions. The fifth action is 'Suck the dirt' (S) and the sixth action is 'Do nothing'(N). The action 'Do nothing' is meant to indicate the vacuum cleaner resting action. While the vacuum cleaner can move in the any one of the four directions one step at a time, its cost is 2 units per step. The dirt is cleaned by the vacuum cleaner by its actuator brush which sucks the dirt completely from the tile at the cost of 1 unit. The vacuum cleaner cleans the floor and can rest at any one corner of the room.

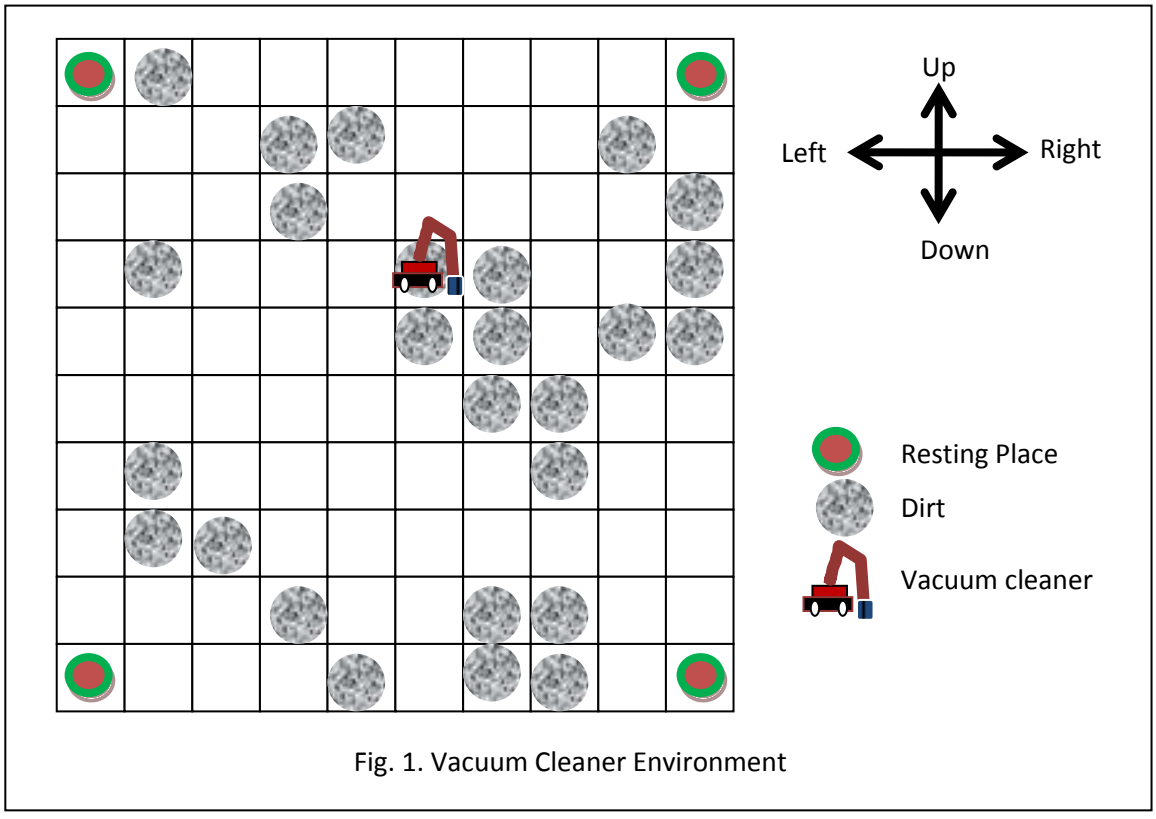

Fig. 1. Vacuum Cleaner Environment

Write a program in Python programming language to implement the above intelligent agent using appropriate techniques discussed in the class. Represent the given problem as a search problem, define a state appropriately, define the goal state and obtain the sequence of actions to clean the floor in minimum cost. An example of a path travelling optimally is shown in Fig. 2 costing 112 units (=43*2+26). The dirt is generated randomly in some tiles and the environment can easily be represented as a two dimensional array of size 10 x10. Approach the problem initially with one resting point, say the upper left corner, and later try to modify the search to include all four resting points.

Remember that the environment is partially observable and the intelligent agent is able to receive the percepts from all its 8 neighbors (Fig.2).

The PEAS descriptions of the vacuum cleaner agent are as follows

Performance (P):	Path cost
Environment (E):	Room having 10x10 tiles with dirt on randomly selected tiles, also has 4 resting places for the vacuum cleaner
Actuator (A):	Wheels and cleaning brush implemented through Graphics
Sensors (S):	Position and Dirt sensors (percepts can be simulated using a 2D array)

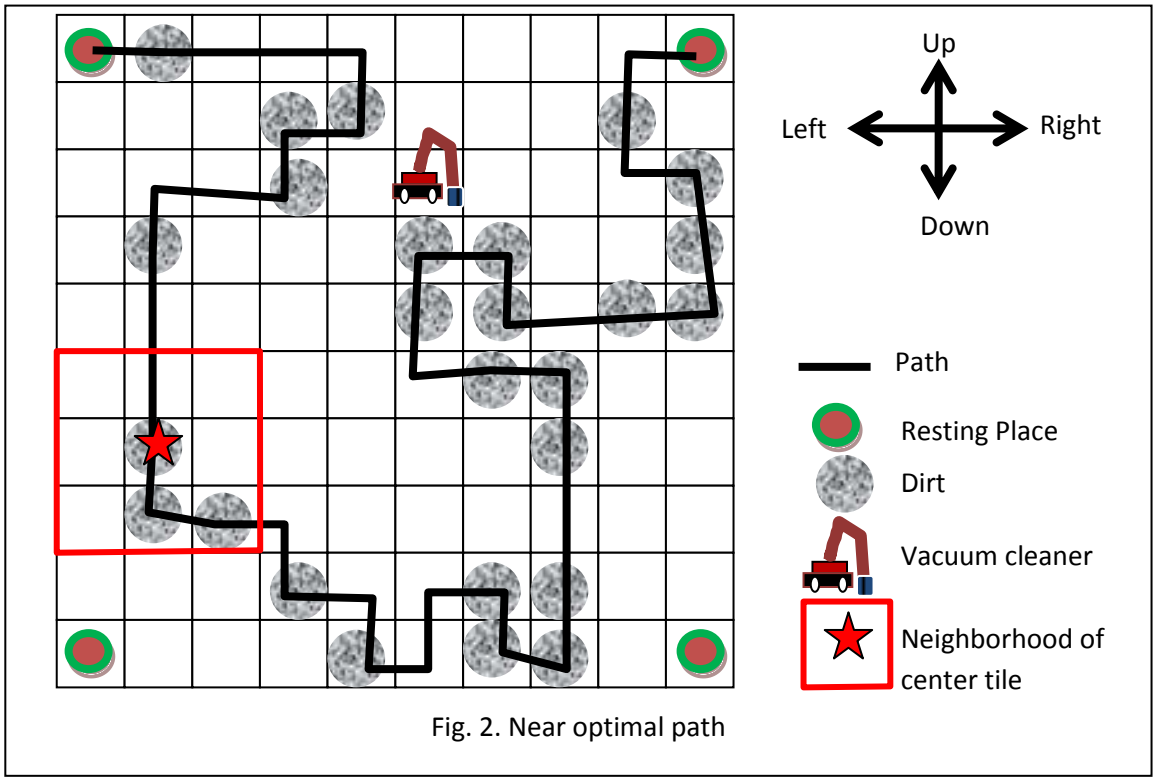

Fig. 2. Near optimal path

Implementation

Use Python version 2.7.13 (Windows 10) for implementing your solution. Only standard Python libraries should be used. Support from external sources or libraries such as github will not be accepted in your submissions. Each student must design own solution and write own code. [Refer handout to understand the malpractice policies.]

Modules

You must implement the following in your program.

1. **Dirt Generator**
   The amount of dirt is fixed and is same for each tile. Randomly select the tiles to introduce the dirt within. Dirt generator introduces dirt by taking as input the value 'p' where 'p' is the percentage of tiles to get the dirt. The dirt generator returns the initial state of the room.

2. **Uninformed Search technique (T1)**
   Implement any ***one*** of the following techniques
   - Breadth first search
   - Depth First Search
   - Iterative Deepening Search

3. **Informed (Heuristic based) Search Technique (T2)**
   Define at least two heuristics h_1 and h_2 to implement any ***one*** suitable technique discussed in the class. Create following functions to compute these values for a given state.
   compute_heuristic1(state s)
   compute_heuristic2(state s)

4. **Use Turtle graphics**
   Use Python based Turtle graphics to display the actions taken and the movements of the vacuum cleaner in parallel. Display the initial room environment as soon as the dirt generator generates the dirt. Once the action path is computed, keep showing the movements and actions graphically. A user friendly comprehensive Graphics User Interface (GUI) is required to be created. This GUI must be divided into two partitions (P1 and P2) in the ratio 1:2 vertically. P1 is used for the results in text form and P2 is used to display the graphs. The right side bigger partition (P2) must be further divided into 4 quadrants. It must have on its upper left quadrant the graphs for T1 (i.e. G1), upper right quadrant, the graph for T2 (i.e. G2) . Also, the lower left quadrant must have the graph G3 and the lower right quadrant must have the graph G4. The details (R1-R11) must appear in one rectangular box in P1 area on the extreme left.

5. **Analysis Module**
   Produce the following analyses and display the resultant values/path etc. on the GUI.
   (a) T1 based analysis
      i. Compute the number of nodes generated till the problem is solved. **[R1]**
      ii. Compute the amount of memory allocated to one node. **[R2]**
      iii. Compute the maximum growth of the auxiliary stack or queue used with the search tree. **[R3]**

iv. Produce the action path in text and in graphics both. For graphics, use RED color to show the path obtained using T1. **[G1]**
v. Compute total cost to clean the room. **[R4]**
vi. Compute the total time to compute the path. **[R5]**

(b) T2 based analysis

i. Compute the number of nodes generated till the problem is solved. **[R6]**
ii. Compute the amount of memory allocated to one node. **[R7]**
iii. Produce the action paths corresponding to the two heuristics separately. You should use BLUE for path obtained using T2's h_1 heuristic and use GREEN color to show the path obtained using T@'s h_2 heuristic. **[G2]**
iv. Compute the cost of both these paths. **[R8]**
v. Compute the total time to compute the paths. **[R9]**

(c) Comparative analysis

i. Compare the memory used in T1 and T2. **[R10]**
ii. Run both the techniques 10 times each with randomly generated initial state of the room. Compute average path cost of the path obtained using T1 and T2 respectively. **[R11]**
iii. Plot a graph with two curves displaying the time taken to compute the path by h_1 and h_2 heuristics of T2 against the room size varying from 3 x 3 to 20 x 20 in step size of 1 in each direction. **[G3]**
iv. Similarly plot a curve displaying the time taken to compute the path using h_1 heuristic against the level of dirt in the room varying from 10% to 100% in step size of 5%. **[G4]**

6. Driver

The driver must integrate all functionalities and execute the functions appropriately using these options

Option 1: Display the room environment
Option 2: Find the path (action sequence) and path cost using T1
Option 3: Find the path (action sequence) and path cost using T2
Option 4: Show all results and graphs in the GUI.

Option 1 uses the dirt generator and displays the room environment graphically.
Options 2 and 3 use the appropriate functions and display the result on the console.
Option 4 uses all functions and computes the path using T1 and T2 both and displays all results on the GUI as specified earlier.

Write up

A two page write up illustrating the solution, approach, selected techniques etc in MS word format (*.docx) is required to be typed. The contents must display the depth of the concepts gained.

Evaluation

The assignment is of 15 marks (5% weight) . The evaluation will be done giving maximum weight to the space and time efficiency carefully imbibed in the code. The carefully selected heuristics resulting in the least cost path will fetch credit. No marks will be given for the code alone, if the code does not compile and execute. The vacuum cleaner displaying intelligent behavior is expected. GUI will have a great impact on presenting your findings, analyses and results. This being the first assignment will be given four days extra time to get acquainted with the Python programming language and for experimenting with graphics. Hence the total time given to complete this assignment is 14 days. Remaining four assignments will be of 15 marks each and will be given only 10 days each for completion. As mentioned in the handout, each student will individually work on the assignments and submit through own Nalanda account.

Errata

Students are advised to read the document carefully and inform me any discrepancy, if exists, immediately. Few corrections, if required will be informed through announcement section of Nalanda.

Submission

Instructions will follow on Nalanda as submission guidelines two days before the deadline.

Vandana
August 31, 2017