**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Artificial Intelligence (BITS F444/ CS F407)**

**I Semester 2019-20**

**Programming Assignment-2**

**Coding Details**

**(September 27, 2019)**

*Instruction: Type the details precisely and neatly*

1. ID \_\_2016A7PS0150P\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name \_\_Patel\_Parth\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of Submitted files :
   1. utils.py
   2. heuristics.py
   3. algorithms.py
   4. GUI.py
   5. coding details.docx
   6. mine.jpg
   7. mine\_clicked.jpg
2. Total number of submitted files: \_\_7\_\_\_\_\_\_
3. Name of the folder :\_\_\_\_2016A7PS0150P\_\_\_\_\_\_\_\_
4. Have you checked that all the files you are submitting have your name in the top?(yes/no) Yes
5. Have you checked that all the files you are submitting are in the folder as specified in 4 (and no subfolder exists)?(yes/no) Yes
6. Problem formulation
   1. State representation: The state representation consists of following attributes:

1. Size of warfield , N and no. of mines, M.

2. orig\_board - A bytearray of size N\*N, representing the actual board state, where each byte is 0-8 (cells without mines) or 9 (mines).

3. curr\_board - A bytearray of size N\*N, representing the current board state, where each byte is 0-8 (explored cells) or 10 (unexplored cells).

4. Boolean variables - game\_over and game\_lost, denoting whether the game is over and whether the game is lost respectively.

* 1. How is the Initial state generated? M (input parameter) cells are randomly chosen to be mines out of N\*N cells in orig\_board. Initially, curr\_board consists only of unexplored cells, and game\_over is False.
  2. What is the goal state? Goal state means that all the non-mine cells have been explored by the agent i.e. all the mines have been flagged by the agent.
  3. Are there more than one goal states? If yes, then describe all the goal states. No; there’s a single goal state.
  4. Do you view the goal state as a state reaching its optimal heuristic value in a search landscape? Give details. Yes, I have viewed the goal state as a state reaching the optimal (in my case, minimal) heuristic value in the state space landscape generated by applying the heuristic value function on states.
  5. State representation in Python (name the construct and give one small example of a state):

class State:

def \_\_init\_\_(*self*, *N*, *M*, *board*):

“””Input arguments provided by mine\_generator(N, M) function”””

self.N = N *# NxN board*

self.M = M *# Number of mines*

self.orig\_board = board *# Actual state of board*

self.curr\_board = bytearray(N\*N) *# Current state of board*

*for* idx in range(N\*N):

self.curr\_board[idx] = 10 *# 10 means unexplored square*

self.game\_over = False *# Indicates if we can play further or not*

self.game\_lost = False *# Indicates if we have lost the game or not*

Eg: s = State(10, 10, board) # board is a 10\*10 bytearray denoting actual state of the warfield (with 10 mines).

1. NextState() function description

def next\_state(*state*, *x*, *y*):

"""Generates next state obtained by clicking on (x,y) in state""”

It takes as input the current state and action to be applied (i.e. (x, y) coordinates to be clicked upon) and returns the same Python state object changed as: If (x, y) contains a mine, it returns the next state with the mine stepped upon. Otherwise, it returns the next state obtained by opening up a convex region around (x, y).

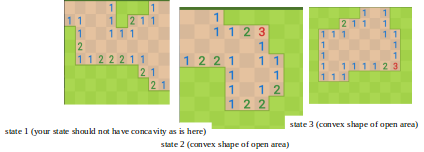
1. Heuristic functions
   1. Is the heuristic function applied on a cell or on a state? Its applied on a state.
   2. Define and explain the heuristics (in words) used in your program. Specify the input and output to each function in detail: Input to h1() and h2() is prev\_state and (x, y), so that it can return the heuristic value of the state generated by applying the move on (x, y) in prev\_state.
      1. h1 : Heuristic value of a state can be computed as 1-(no. of flagged mines i.e. no. of mines that can be detected by us)/M.
      2. h2 : Heuristic value of a state can be computed as ( [1-(no. of flagged mines i.e. no. of mines that can be detected by us)/M] + [1-(no. of explored cells)/(N\*N-M)] ) / 2
   3. Compute (manually) the heuristic values for the following three states state 1, 2 and goal state as given in the following three figures 1,2 and 3 respectively. Write the values below appropriately.

State1 : h1 = 0.5 , h2 = 0.485

State2 : h1 = 0.5 , h2 = 0.485

State3 : h1 = 0.6 , h2 = 0.54

Goal state : h1 = 0 , h2 = 0

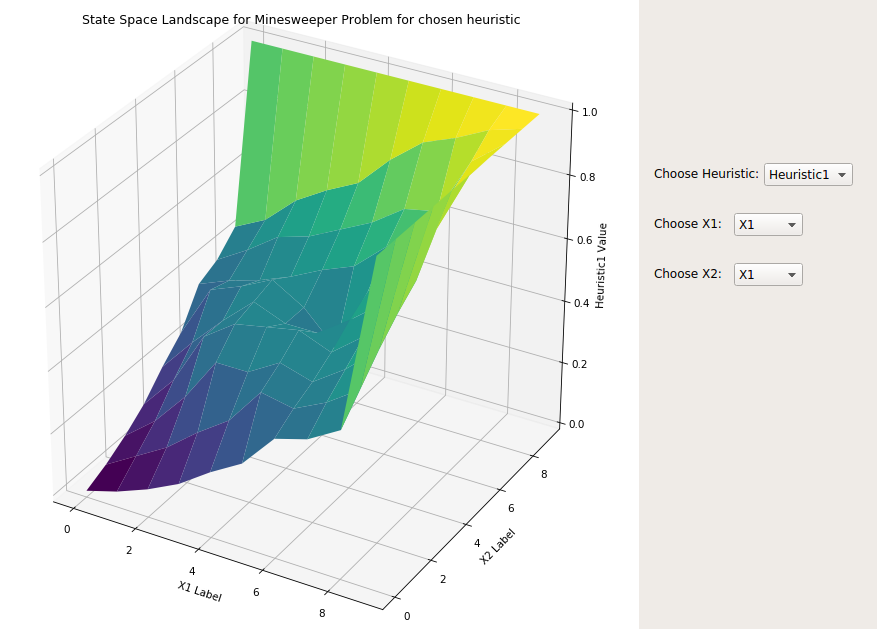
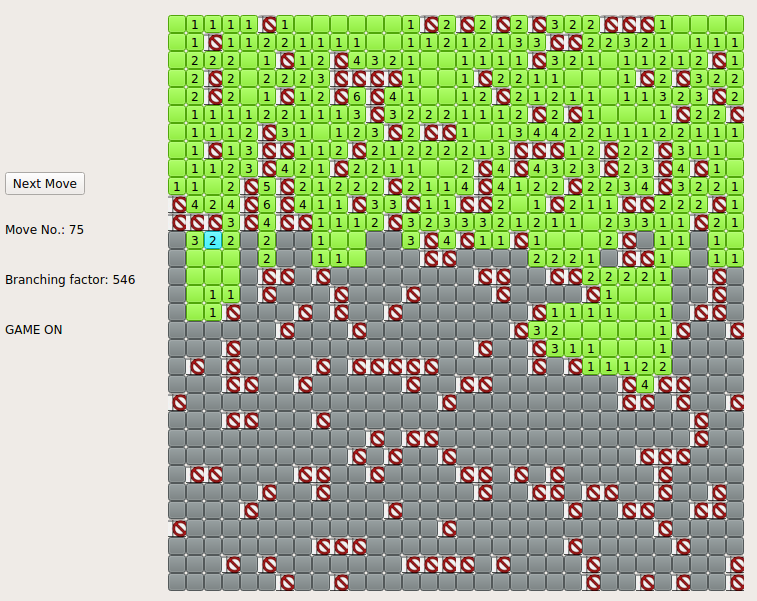


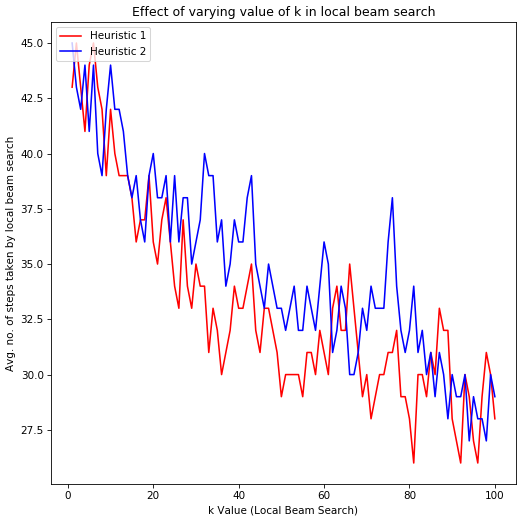
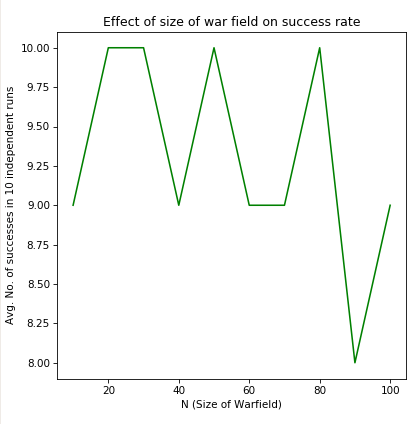
1. Hill Climbing (T1) technique
   1. Code status (implemented fully/ partially/ not done) : implemented fully
   2. Write the sequence of steps followed by you (choose words from - First click, next state, all next states, compute heuristic value, cell, state, open area, closed area, mine etc.) : First click, compute heuristic value for all next states, choose next state with minimum heuristic value, open area (done by internal agent as only it has knowledge about mines).
   3. Print the pre-computed values

R1 = 11.2 KB R2 = 0.053 secs. R3 = 0.35 R4 = 19 R5 = 17

R6 = 41 R7 = 10 R8 = 8 R9 = 35 R10= 37

* 1. Cut and paste the images of graphs G1, G2, G3 and G4 below



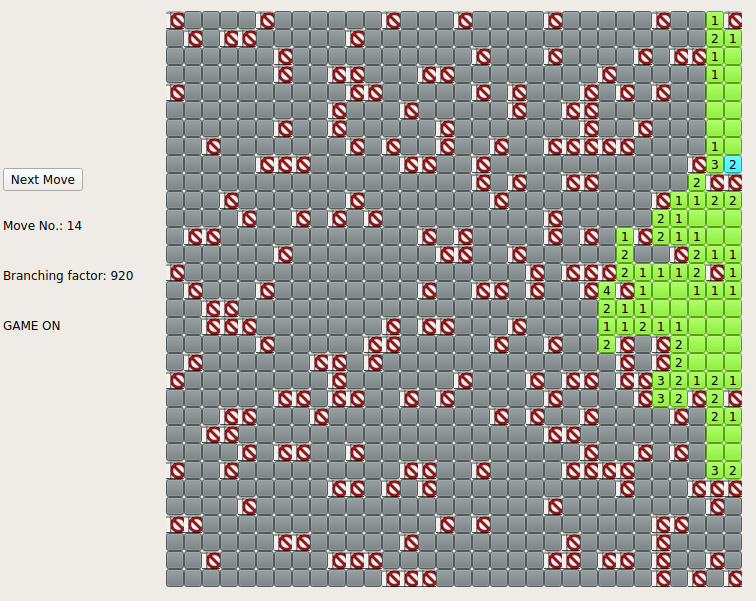
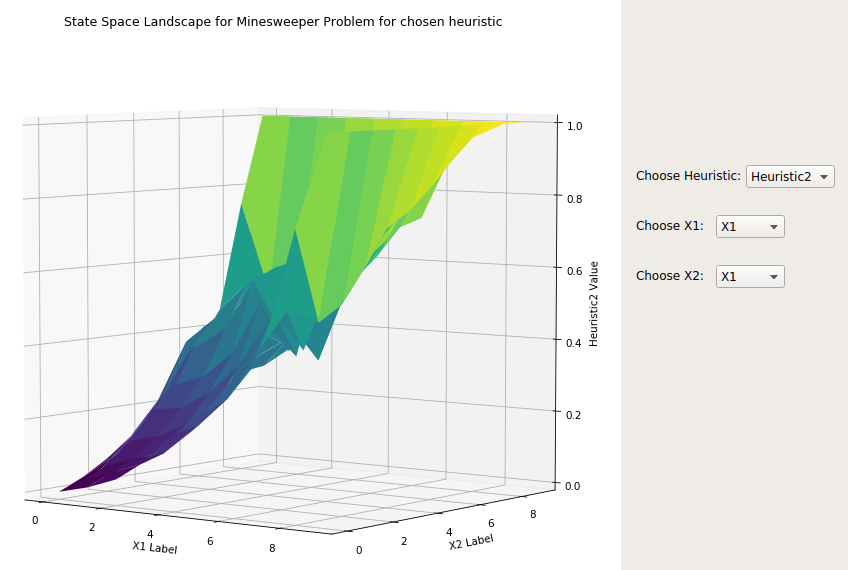
 

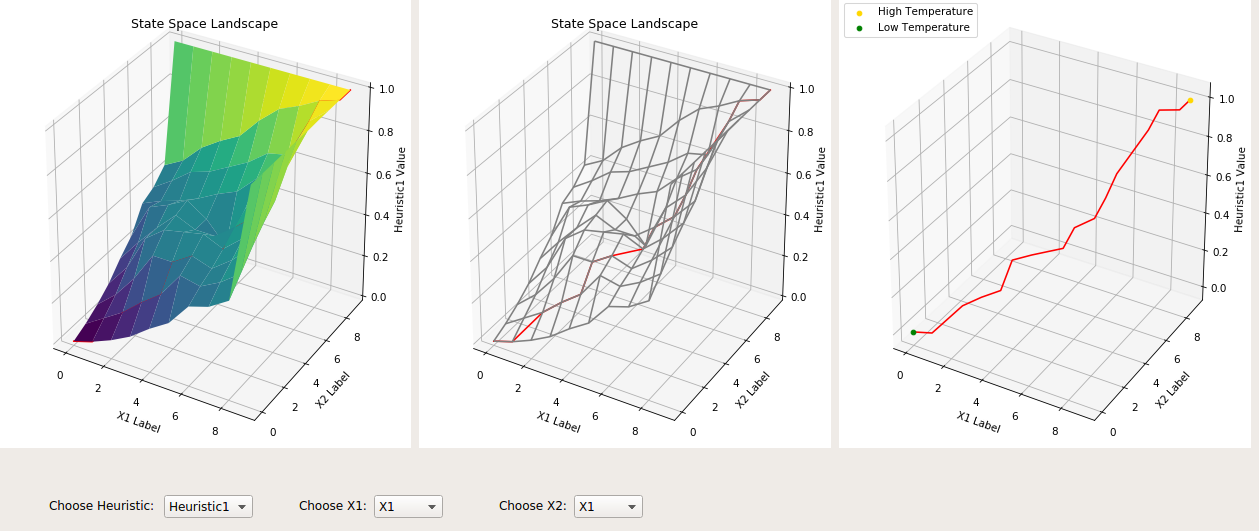
* 1. Are you posing the problem as maximization problem or minimization problem? Discuss why. I have posed it as a minimization problem, because I am trying to visualise heuristic of a state as its distance from the goal state. That’s why the heuristic value of the goal state is zero.
  2. Discuss how you view the changing values of heuristics as you proceed. The heuristic value should decrease as we proceed in hill climbing (for minimization problem).
  3. Discuss the state which represented suboptimal solution? Why? The state representing suboptimal solution (i.e. local minima) is when there is a mine among the explored cells.

1. Simulated annealing (T1) technique
   1. Code status (implemented fully/ partially/ not done) : implemented fully
   2. Write the sequence of steps followed by you (choose words from - First click, next state, all next states, compute heuristic value, cell, state, open area, closed area, mine etc.): First click, select a next state at random, compute heuristic value for the chosen next state, choose next state if its heuristic is lower than current state or if random no. generation yields a random no. in favour of the computed probability (i.e. probability of choosing a bad move), open area.
   3. Print the pre-computed values

R11 = 8.3 KB R12 = 0.032 secs. R13 = 0.4 R14 = 9 R15 = 7 R16 = 39

* 1. Cut and paste the images of graphs G5, G6 and G7 below



* 1. Discuss the temperature range used: Temperature is computed as : 0.1 /((no. of moves)^0.1), so range is (0.1, 0.093, 0.089,...)
  2. Discuss the probability computation: In case of negative deltaE, probability is computed as exp(deltaE/temp.), and the move is chosen if a random no. generated between 0 and 1 is less than the computed probability.
  3. How are you selecting the bad moves? A bad move is chosen in case of negative deltaE, when a random no. generated between 0 and 1 is less than the computed probability.
  4. Are you posing the problem as maximization problem or minimization problem? Discuss why. I have posed it as a minimization problem, because I am trying to visualise heuristic of a state as its distance from the goal state. That’s why the heuristic value of the goal state is zero.
  5. Discuss how you view the changing values of heuristics as you proceed: The heuristic value should generally decrease as we proceed, except in case of bad moves.

1. GUI details
   1. Created the GUI (yes/ No): Yes
   2. Have created it according to the specifications?(yes/No) Yes
   3. Which module of Python used for creating graphics? PyQt5
   4. Is this under the standard Python library/ Matplotlib/ PyQT or not? Yes
   5. If not, why? N/A
2. Compilation Details:
   1. Code Compiles (Yes/ No):\_\_Yes\_\_\_\_\_\_\_
   2. Mention the .py files that do not compile:\_\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. Any specific function that does not compile:\_\_N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. Ensured the compatibility of your code with the specified Python version(yes/no)\_\_\_Yes\_\_\_\_\_
   5. Instructions for compilation of your files mentioning the multi file compilation process used by you (We may use the replica of these for compiling your files while evaluating your code): **python GUI.py**
3. Driver Details: Does it take care of the options specified earlier(yes/no):\_Yes\_
4. Execution status (describe in maximum 2 lines) - All the submitted code works. The GUI is as specified - R1 to R16 are displayed, and buttons are provided to view G1-G7. You can run my code for hill climbing and simulated annealing for N, M, and heuristic (h1 or h2) chosen by you through the GUI.
5. Any other detail: **For 3D plotting in matplotlib, matplotlib's plot3D() function was not accepting Python lists as input. So, NumPy arrays have been used for this purpose. I have NOT used NumPy arrays at any other place in the code.**
6. Declaration: I, \_\_\_\_\_Patel\_Parth\_\_\_\_\_\_\_\_\_ (name) declare that I have put my genuine efforts in creating the python code for the given programming assignment and have submitted only the code developed by me. I have not copied any piece of code from any source. If the code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

ID\_\_\_\_2016A7PS0150P\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_Patel\_Parth\_\_\_\_\_\_\_\_\_

Date: \_\_\_27/09/2019\_\_\_\_\_

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Should not exceed six pages\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*