Programming Project 1

Solving the 8-puzzle using A* algorithm

1. Introduction

1.1 8-puzzle problem

In the 8-puzzle problem we have 9 tiles - one blank tile and 8 tiles numbered from 1 to 8. The goal of the puzzle is to reach a particular configuration of the 8 tiles (goal state) by swapping the blank tile with a numbered tile.

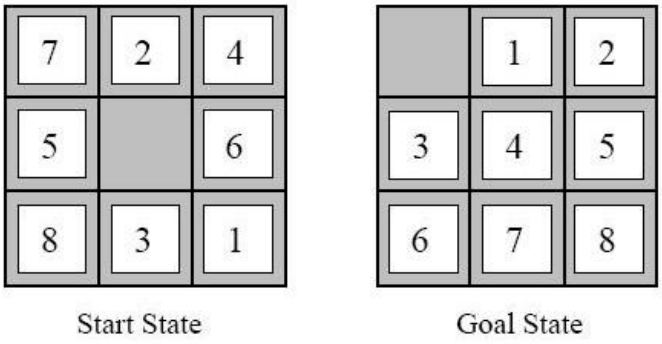

Fig 1.1 Example of 8-puzzle problem

1.2 A* Algorithm

In this algorithm we decide to explore the node having the least f(n) value where f(n) is the sum of two parameters g(n) the step cost and h(n) the heuristic value and n being a particular node.We are using two heuristics here:

a. **Misplaced Tiles**

   It gives the number of tiles that are not in the same position as in the goal state

b. **Manhattan Distance**

   It gives the sum of number of steps needed for each tile to reach its position as in the goal state.

Min(f(n)), where f(n)= g(n)+h(n)
n= the node that is being explored

2. Program Structure

The project has been implemented using Python 3.7.0. The program consists of a class called A_Star. The blank tile is represented using '0'.

Class Variables

a. i_zero - Stores row value of zero in the state being explored
b. j_zero - Stores column value of zero in the state being explored
c. step_cost -Stores the g(n) value i.e. the step cost
d. nodes_generated - Stores the count of total number of nodes generated
e. initial_state - Stores the state being explored
f. initial_state_Copy - Stores a copy of initial state to use for different heuristic
g. goal_state - Stores the goal state
h. unexplored_states - Stores all the nodes that are yet to be explored
i. unexplored_states_node -Stores new nodes that have been generated in exploring_nodes function
j. explored_states - Stores list of nodes that has been explored
k. misplaced_tiles_list - Stores f(n) value corresponding to nodes in unexplored_states list
l. manhattan_distance_list - Stores f(n) value corresponding to nodes in unexplored_states list

Functions

a. __init__():

It is used to initialize the flag variable.

- **Local Variable**

  Flag – It is used to decide which heuristic is being used misplaced tiles (flag =0) or manhattan distance (flag=1).

b. finding_zero():

This function helps in finding the row and column of '0' in the state being explored and also checks if goal state is found. Also, few variables are initialized again so that can be used for a different heuristic.

- **Local Variable**

  a – It is used to display the state as a 3X3 array.

c. exploring_nodes():

This function finds all the possible nodes that can be generated by swapping the zero with numbers on top, right, bottom and left to create a new node and appends the new node to the unexplored_states list

- **Local Variable**

  new_state – It stores the node generated by swapping the zero with any number on top, left, right and bottom.

d. misplaced_tiles():

This function gives the number of tiles that are not in the same position in the state being explored as in goal state.

- **Local Variable**

  i,j,k – They are used to traverse the unexplored_ states_node and goal_state.
  count – It is used to store the h(n) value.
  min_misplaced – It stores the minimum f(n) value from the misplaced_tiles_list.

e. manhattan_distance():

This function gives the number of steps that are needed by each tile to reach the position as in goal state.

- **Local Variable**

  i, j, k – They are used to access the numbered tile and traverse the unexplored_ states_node.
  gi, gj - They are used to access the numbered tile in the goal state.
  distance – It is used to store the h(n) value.
  min_manhattan – It stores the minimum f(n) value from the manhattan_distance_list.

2.1 User Input

The user has to give the initial state and the goal state as input. The input is being taken row-wise. It is taken as a group of three numbers, each number separated by a space.

```
Enter Initial state

Enter 1st row
1 2 3
Enter 2nd row
4 5 6
Enter 3rd row
0 7 8

Enter the goal state

Enter 1st row
1 2 3
Enter 2nd row
4 5 6
Enter 3rd row
7 8 0
```

2.2 Functionality

- We create a variable called A_Star_1 for the class A_Star and call finding_zero() function using A_Star_1 and passing flag as 0 which starts with misplaced tiles heuristic.

- The user is asked for input and the input is stored as a list of list.

- The finding_zero() checks if the initial state is same as goal state if not finds the position of zero and calls the exploring node function

- The exploring_node() function generates all possible nodes by swapping the ‘0’ with adjacent tiles.

- The generated nodes are added to unexplored_states_node which is appended to unexplored_states list.

- The misplaced tile heuristic of every state in unexplored_states is generated, added to step cost and appended to misplaced_tiles_list.

- The index of the minimum f(n) value is selected from the misplaced_tiles_list and the corresponding state from unexplored_states is made as the initial state to repeat the above process.

- The flag is changed to 1 and all necessary variables are initialized in the finding_zero() function.

- Once the goal state is found the control goes back to the beginning and A_Star_1 is used to call the finding_zero() function again but this time with flag value as 1.

- The finding_zero() checks if the initial state is same as goal state if not finds the position of zero and calls the exploring node function

- The exploring_node() function generates all possible nodes by swapping the '0' with adjacent tiles.

- The generated nodes are added to unexplored_states_node which is appended to unexplored_states list.

- The manhattan distance heuristic of every state in unexplored_states is generated, added to step cost and appended to manhattan_distance_list.

- The index of the minimum f(n) value is selected from the misplaced_tiles_list and the corresponding state from unexplored_states is made as the initial state to repeat the above process.

3. Sample Output

3.1 Sample 1

Initial state:	Goal State:
1 2 3	1 2 3
7 4 5	8 6 4
6 8 0	7 5 0

Misplaced Tiles
Number of nodes generated 210 nodes
Number of nodes explored 119 nodes

Manhattan Distance
Number of nodes generated 18 nodes
Number of nodes explored 9 nodes

3.2 Sample 2

Initial state:	Goal State:
2 8 1	3 2 1
3 4 6	8 0 4
7 5 0	7 5 6

Misplaced Tiles
Number of nodes generated 22 nodes
Number of nodes explored 13 nodes

Manhattan Distance
Number of nodes generated 12 nodes
Number of nodes explored 6 nodes