Introduction to Artificial Intelligence Homework 2 report

Andres Ponce 彭思安

May 1, 2020

1 Background

Playing games has been an interesting research area in Artificial Intelligence for several decades, and has led to some of the greatest accomplishments in Artificial Intelligence with Deep Blue. Several algorithms can be used to model certain types of games by modifying a search algorithm. One such an example is the game of Minesweeper. This game can be modeled by certain definite rules and thus lends itself very well to being playable by an algorithm.

In this assignment, there was a 6x6 board that consisted of three types of cells: hints, variables, and mines. Hint nodes contain a numeric value that describes how many bombs there are in its 8 possible surrounding cells. Given a limited number of bombs, the objective was to find a configuration of the board in which all bombs are used and every hint node has the amount of adjacent mines described in its value. The objective was to employ a backtracking search to carry out this search for a valid configuration.

2 Solution Description

The backtracking approach used in this assignment involved a simple main algorithm:

```
Algorithm 1: solve_main(root)
Result: valid board configuration
if reject(node) then
    return false;
if accept(node) then
    return true;
for children in node do
    solve_main(child);
    child = node->next;
```

Following is a description of those methods.

2.1 Rejection

The `reject` method returns true if the valid board configuration is invalid and should be rejected, and returns false otherwise. This method takes in a node from the board (i.e. a cell in the board with some additional info) and performs a simple check on hint nodes, to see if there are more mine nodes than the hint's value allows for. After that, we move on to the accept method.

2.2 Accepting

The accpeting method will check whether the current board configuration is a valid one according to the constraints given. For a configuration to be valid, we need both all the mines to be used up, and the hint nodes should all have the exact amount of adjacent nodes described in their values. When this happens, then we know we have achieved a valid configuration. Otherwise, this method will return false.

2.3 Recursion

Here is where the tree like aspect of the problem unfolds. Since we maintain a data structure with the yet unassigned nodes, we get the one from the list to the closest distance. We set a limit to the amount of children that each node can have (4 in this assignment),