Reuben Tanner

CS457

Homework 3

For each of the six basic uninformed search strategies, describe a realistic search problem where that strategy would be more appropriate than the other 5, and state why.

1. BFS

Suppose we are faced with a problem where the path cost is a nondecreasing function of the depth of the node and our biggest concern is finding the optimal solution. For instance, we are digging a mine shaft and we have all the data about the geology of the surrounding rocks e.g. stability under pressure, compactness of the soil etc. Our highest concern is the safety of our workers which correlates to an optimal path through the most resilient parts of the earth in the land that we have purchased for the shaft. We are modeling our states by the various properties of the earth which are good for digging where the goal state is the state in which the highest amount of properties are conducive for drilling. We are renting the usage of Tianhe-2 from China’s National University of Defense Technology so time / space complexities are of little importance. The BFS would be ideal for this problem because time and space issues (which are the main problems with BFS) are trivial due to the ridiculous amount of resources we are using and the importance of finding the optimal path for our mineshaft is the primary concern and the step cost for the depth of our shaft increases steadily with distance drilled. BFS is more ideal than uniform cost for this problem because we are not concerned with the cost of each path and we mainly want to take a few large steps rather than whatever method would be the cheapest through several steps.

1. Uniform Cost

Assume we are a government agency during a national financial crisis and, unlike many of our counterparts, our director wants to save the most taxpayer money possible. A new road needs to be built across a rural area of our jurisdiction very soon and in an attempt to be as frugal as possible we want to know the absolute cheapest paths to build upon regardless of how meandering the road may be. As another part of the problem, we know that not all path costs are the same because there are certain areas of land which are quite expensive to build upon and others that are not. We have access to a decent super computer so time / space complexity isn’t an issue but finding the absolute cheapest path is. Uniform path would be the most ideal uninformed search strategy because the length of the path or strangeness of its curvature is of little importance as long as it is cheap. Also, there are no negative or zero weight paths in this search because every inch of road costs a significant amount of money.

1. DFS

Suppose we are writing an Android application that solves the common game “4 Pics 1 Word”, we want this application to take the least amount of time and space, produce correct results with the main emphasis on being fast and lightweight. With regard to the solution to this game, any solution that works is as good as any other, some may take more steps but the final answer is usually what the emphasis is placed on. In a situation like this, a DFS would be ideal for our application because there are no problems with infinite depth so one of the bigger issues with DFS is avoided, since the DFS will pull out the first correct solution found while using the least amount of memory it is ideal for the mobile environment where resources are limited.

1. Depth limited

Assume we are dealing with a problem where optimality is not an issue, there is a high chance of infinite depth branches and we have low space capacities and where we know that a solution can be found at a given depth. Consider that we are writing a mobile application for determining a restaurant of a given type that is within a certain radius from us. The user specifies a given distance (in miles or something) and the search provides him/her with the first restaurant that is within that given distance. (I know this one isn’t very good but I had a tough time thinking of something for this).

1. Iterative deepening

Suppose we are trying to solve the 8 puzzle problem on a mobile device. We are trying to solve this problem in a way so that we use as little memory as possible due to the mobile nature of the environment, we want an optimal solution but we also care about our space complexity. Iterative deepening would be an excellent solution to this problem because, like BFS it produces an optimal result because our path costs are nonnegative and nondecreasing and our branching factor is finite. While iterative deepening delivers this optimal and complete solution, it does so in a way that is perfect for the mobile environment due to the limited amount of memory that it ends up using.

1. Bidirectional

Consider a problem where the goal is known before the problem begins, such as the 8 puzzle. Since we know that, in the end, we desire to get to the single goal state that is part of this problem we can design our search that retains its explored states, perhaps by using a BFS, to begin searching predecessors from the goal state. This search would be ideal because, in the best case if we used iterative deepening as our non-memory retaining search for our search beginning at the initial state and a BFS as our memory-retaining search from our goal state then we could find an optimal solution in space/time complexity which is far less than breadth first or iterative deepening alone.