**Searching for Gold:**

**An Analysis of Search as a Means to Solve Wumpus World**

**Mike Silva-Breen, Trevor Boone, Alex Cole**

**Abstract**

The research explores different ways of solving the Wumpus World problem. Wumpus world is a simple game where a hero tries to get gold and escape a cave without dying. The hero while traversing the world doesn’t have full knowledge of the world and only perceives the world through different percepts. We went away from the traditional knowledge base application and we tried to do a search function with a heuristic and some very basic inferences.

**1 Introduction**

**1.1 Wumpus World**

Wumpus world is a simple grid based world for an agent to explore. The world consists of the wumpus, pits, gold and a hero, which is the agent. The hero makes decisions based on information gathered from the neighboring squares (north, south, east, west). There are five basic percepts that the hero needs to consider when making a decision. If the hero is in a square that neighbors the wumpus, a stench is detected. Similarly, if the hero is in a square that neighbors a pit, a breeze is detected. A glitter is detected if the hero is in the same room as the gold, a bump is detected if the hero moved forward into a wall, and a scream is heard if the hero killed the wumpus.

Based on these precepts the hero can decide to do the following: turn left or right, move forward, shoot an arrow, grab, or climb out of the cave. Usually the agent would make these decisions with a knowledge-based approached using the precepts it has at a given square. We, however, chose to use a simplified version of the wumpus world to see how searching algorithms performed when applied to the wumpus world. The decisions made by the hero are extremely simplified since now the hero decides which square to go to and does not have to worry about about the direction he is facing. We have also removed trying to kill the wumpus in our version of Wumpus World in order to simplify moving from square to square. In our search algorithm that uses a heuristic, the only percepts that the hero considers are stench and breeze since our hero is more consider about if a square is dangerous or not. He will also detect the gold when in the same square as the gold and then retrace his path back to the start to climb out of the cave.

**1.2 Changes to the Wumpus World**

To facilitate the change to a search based solution, we created a relaxed version of Wumpus World. We removed the hero’s ability to shoot an arrow. It would be possible to detect the wumpus and kill it in the heuristic search, but BFS and DFS would be much more difficult to do it with.

**2 Implementation**

**2.1 Implementing Heroes Based on Search**

We decided to implement the solution to the Wumpus World by searching the graphs of the square for a path to and from the goal square. The lack of knowledge of the grid makes it so it is impossible to go straight for the goal. We decided to use a search and use breadth first search and depth first search as role models. To make the implementation simpler we implemented an AbstractHero class which removes possible movements from the front of a list. From here is was easy to change the position of items in the list to implement different search algorithms. Essentially it will just pop the node off the front of the list then paths to that node via already visited and known safe nodes.

**2.1.1 DFS & BFS Heroes**

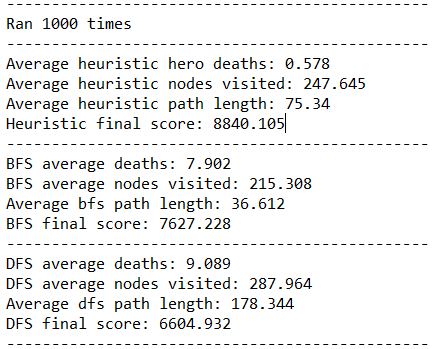
Building off of the AbstractHero class made the implementations of heroes using a modified depth first search or breadth first search. Adding nodes to the front of the list will make the hero search the graph by a standard depth first approach and adding it to the end of the list will make the hero search in a classic breadth first approach. It is possible over the course of the search for there to have the hero die and in that situation the hero just continues to the next node in the list. With this approach it is guaranteed that they will eventually visit every node that they can visit

**2.1.2 Heuristic Search Hero**

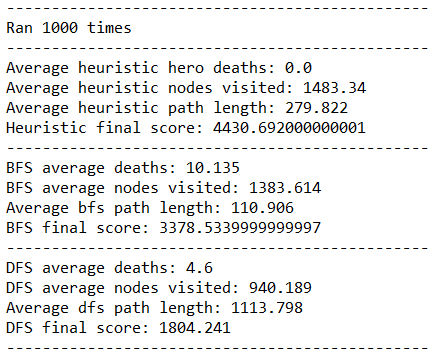
The final hero type uses a heuristic to improve performance. The only knowledge about the board is through the percepts that the hero has so our hero leverages those percepts to avoid hazards. If a square has the percepts stench or breeze it means that the square around that square are dangerous. In addition if multiple square adjacent to a square have those percepts then it is even more likely that that square has a hazard on it. Therefore for our heuristic we keep track of a danger value for all of the nodes and when we reach a square with any of the dangerous percepts we increment the danger counter for all of the adjacent squares. We the then keep the list of possible squares to go to sorted so the hero searches the least dangerous squares first. We also made an addition an addition to this algorithm based on a simple rule, if a square has no dangerous percepts we set all squares’ danger level to zero and make it so the danger of those cells cannot be changed again. Essentially this means that squares adjacent to a square with no dangerous percepts is then considered to be safe no matter what. It follows this algorithm until it finds the gold and then it backtracks through known safe squares back to the goal.

**3 Evaluation Data**

**20x20 Grids**



**50x50 Grids**



**4 Data Analysis**

As can be seen from the results of our tests, the Heuristic Hero is much, much better at avoiding death than either of the other two heroes, but explores the most squares, as he flees from any amount of danger. In large environments, the hero can run away from danger much more easily, but this causes him to potentially flee into areas of empty space, which he may then mill around in. The hero tries to explore any available space that is the least hazardous that is also close and not previously explored. This means that he could potentially cross the board to an early turn if he encounters a hazard late into his path finding, which causes him to remain safe as long as possible, means he spends a long time on each floor. So while our scoring method was favorable to this type of search pattern, other defined scoring methods may rate speed more highly than safety, and our hero could become less useful.

**5 Conclusions**

As shown in the data analysis a basic danger heuristic with some slight inferences is enough to make the hero seldom die. While it didn’t always get to a solution in less moves than the other types of search it did die significantly less than the heroes without any sort of danger analysis. This shows that even some very quick and basic knowledge greatly enhanced the search functionality. More inferences could potentially be made, such as eliminating past danger levels based on further gathered data, but at that point, we would have essentially been creating a knowledge base/agent approach to the problem, which was not the intent. It is due to this fact, that we cede to knowledge base, and admit that it is a superior approach to the problem.

**References**

[*http://www.cis.temple.edu/~giorgio/cis587/readings/wumpus.shtml*](http://www.cis.temple.edu/~giorgio/cis587/readings/wumpus.shtml)

*Artificial Intelligence: A Modern Approach, 3rd Edition. Russell and Norvig.*