Alan Concepcion

CSC 448

Midterm Report

Table of Contents

Abstract

My name is Alan Concepcion. My ID is 23310938, and my email is aconcep000@citymail.cuny.edu. The objective of this midterm report is to provide extra knowledge of topics we pick that were not discussed in class. We were asked to pick more than one topic and provide more information on these topics that weren't in the slides when discussed in class. To obtain more useful information I got more information from the textbook as when the topics were discussed in class they were extremely condensed but the textbook have a lot of information on them. The topics discussed in this report are: Uninformed search, Breadth First Search, Depth First Search, Heuristic search, Strong A.I, Weak A.I, and Deeper Blue. When talking about uninformed searches I discuss the 2 algorithms DFS and BFS and provide some information of the contrast; Heuristic search. After that I explain what Strong A.I, Weak A.I and then difference of both is. Finally, I wanted to end it in a topic I was interested in which is the chess computer system Deeper Blue.

Topic 1

Blind search

Blind search or also known as uninformed search is a way to traverse on problems such as state-space graphs with no knowledge of the search space. There are 2 known blind search algorithms that will be discussed later on. The point of blind search is to find a solution to a problem with no information, for example, traversing through a maze. In the class textbook it uses the maze example to say that in a blind search when you traverse a maze with no previous knowledge you will most likely hug the left side until the exit is found. Blind search algorithms have exponential time complexity because they need to search the entire state-space. Easy to implement for small state space problems however become less efficient the bigger the space.

Breadth First Search

One of the blind searches to examine, Breadth First Search or BFS. BFS traverses a tree level by level in its search for a problem solution however, it is hindered by its space demands. I remember when I took data structure and algorithm course in New York City College of Technology the professor stated that breadth first search can be seen as a breath, you blow on dust and see it all spread quickly. This search is like that as it checks every state space which is why it requires a lot of space. The time complexity is exponential in worst case, which results in O(n^d) where n is max nodes and d is max depth. In large state spaces you can see that this can become an extremely large number. One variation of BFS is called “branch and bound”. It is also called uniform cost search which is how it is referred to in our class slides. This variation is used for informed searches, but since the first topic is about blind/uninformed searches it won’t be talked about here. It was worth noting that BFS is used in both informed and uninformed searches. BFS is seen as complete and optimal as you are guaranteed to obtain a solution and it

provides the lowest cost solution, however it may be slow since it needs to check every node and its neighbor to find the shortest path. It accomplishes this by using a queue-based system to keep track of it all. A good visual example taken from class slides is:

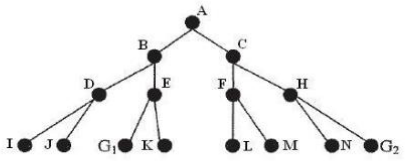

Figure 2.27: Search tree to illustrate depth first search. As dfs is a blind search, all heuristic estimates and node to node distances have been omitted.

Open = [A];	Closed = [ ]
Open = [B,C];	Closed = [A]
Open = [C,D,E];	Closed = [B,A]
Open = [D,E,F,H];	Closed = [C,B,A]
Open = [E,F,H,I,J];	Closed = [D,C,B,A]
Open = [F,H,I,J,G1,K];	Closed = [E,D,C,B,A]
Open = [H,I,J,G1,K,L,M];	Closed = [F,E,D,C,B,A]

... Until G_1 is at left end of Open list.

Algorithm stops once goal G_1 is reached.

Figure 1 Implementing BFS Slide 2 Blind Search page 37.

Pseudo code for BFS Looks like this:

```
function bfs
    begin
        open <- [Start]
        closed <- [];
        while open!= [] do
            begin
                remove leftmost state from open, call it x;
                    if x is a goal then return SUCCESS // goal found
                    else begin
                        generate children of X;
                        put x on closed;
                        discard children of x
                        if already on open or closed; //loop check
                        put remaining children on right end of open
//queue
                    end; // i.e. open is maintained as a queue!
                end;
            return FAIL; // no states left
        end
```

39

Figure 2 BFS Pseudo Code Slide 2 Blind Search page 39

Once again pulled from class slides BFS is preferred over Depth First Seach if, state-space is small, amount of branching factor(nodes) is small so less memory and if at least in bigger state space graphs a solution if found quick and not in a big depth level. To continue on this

discussion, we need to look at Depth First Search and the differences between the two.

Depth First Search

The next blind search algorithm to be discussed is Depth First Search or DFS. Depth First Search is different where rather than search everything it will take one path and try to get a solution from it and will only try other paths when a solution is not found in the path it took. Since it takes this approach, it uses less memory than breadth first search. However, since it takes this approach may not find a solution. Say if using DFS in the same maze example above. The exit is 2 steps to the right, but DFS starts its search walking left you see that DFS could miss solutions that are close to the start or also known as the root. The time complexity is the same as bfs for worse case which just means that it is just as slow in the worst case. A visual example of dfs would be:

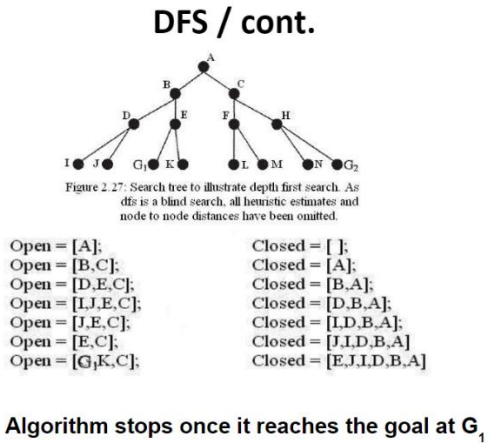

Figure 3 Implementing DFS Slide 2 Blind Search page 35.

Pseudo code for DFS Looks like this:

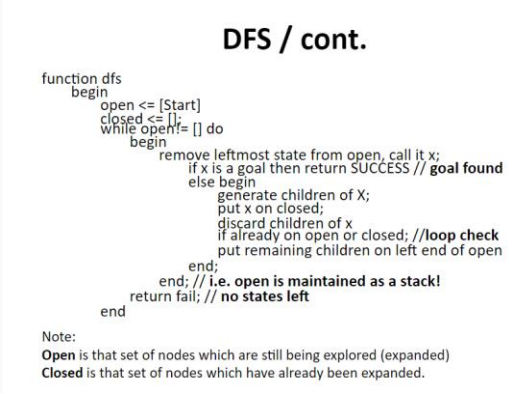

Figure 4 DFS Pseudo Code Slide 2 Blind Search page 34

Heuristic search

Heuristic search looks forward into state-space graphs to determine the path it will take to the goal. All because it does this doesn't mean that the algorithm just knows how far the goal is. It does, however, have an estimate as to how far from the goal it is. 3 search algorithms that follow the look forward method is hill climbing, beam search, and best first search. Heuristic search wants to reduce the number of nodes needed to get to the goal, which means it just wanted to reduce the number of nodes it needs to visit. Since it does this it is used because it is effective for complex problems, and gives the solution with the best path while ignoring the ones that may require more memory.

Topic 2

Strong A.I

Strong A.I is the idea of being able to provide a computer system with software that will allow it to think, hear, and act like humans. Examples given in the slides, class and textbooks are movies like A.I and Terminator. Etc. As of the time of writing this we have not achieved this level of A.I. This belief depends on a computer system being knowledge about its world and the problems that it might encounter

Weak A.I

Weak A.I is the idea that we can model our way of thinking into a computer system to help solve complex problems. This way of thinking does not believe that a computer is intelligent as a human is intelligent. This belief system employs systems such as logic, automated reasoning, and other general structures that can be applied to a wide range of problems. This system does not incorporate any real knowledge about the world and the problem that is being solved as apposed to strong A.I which we stated it does.

Strong A.I vs Weak A.I

In theory Strong A.I is capable of bigger, more complex problems so the scope and problem-solving capabilities are much higher than Weak A.I. Strong A.I can adapt to any problem as its suppose to mimic human thinking while Weak A.I is more limited to what was programmed to solve. Since Strong A.I has not been implemented as of yet there are no examoles for it just theory crafting but for Weak A.I we have assistant tools such as apple's siri, samsung's bixby, and even more recently chatgpt.

Topic 3

Deeper Blue

This was an interesting topic I wanted to write about since I heard about it after watching this interesting video on the topic: https://youtu.be/HwF229U2ba8?si=nOL6KPBPKCjdEUAp . In short, Deeper blue is upgraded version of deep blue, a chess playing computer created by IBM. This machine became one of the most notable A.I systems due to the fact that it actually defeated chess world champion Garry Kasparov. Deep Blue was created by Feng-Hsiung Hsu and it was a machine built just to play chess. This could be an example of Weak A.I. In the first iteration known as Deep Blue it did lose to Garry Kasparov, this was as explained by him that he noticed that the computer would default to certain moves, and he just exploited that fact. This is mentioned in the video linked above. After losing to Kasparov Feng-Hsiung Hsu and his team decided to figure out ways to prevent this exploit and decided to call this upgraded version of deep blue, deeper blue. Before the rematch against Kasparov they tested it against high level chess players while working on deeper blue, feeding it any and every move possibility. This was so it would have a solution for any move in any position for any piece. After defeating Kasparov the project was shelved as they finally achieved what they wanted, creating a machine good enough to beat a chess champion.

Kasparov vs Deep Blue. Image from https://theworld.org/stories/2018-01-05/garry-kasparov-and-game-artificial-intelligence

Conclusions

In this report more information was provided for the selected topics. With the extra information we can see the differences in; for example, the differences between DFS and BFS, the differences in Weak A.I and Strong A.I. There were limitations to what I could find as while we have seen lots of advancements in Weak A.I, Strong A.I is still just a theory and most information found online and even in the class textbook it's all just theory crafting what Strong A.I can possibly do. It would be nice to relook into how far Strong A.I advances more later down the road.

References

Most Information pulled from textbook and class powerpoints.

Deep Blue | Down the Rabbit Hole by Fredrik Knudsen:
https://youtu.be/HwF229U2ba8?si=nOL6KPBPKCjdEUAp

Garry Kasparov and the game of artificial intelligence:

https://theworld.org/stories/2018-01-05/garry-kasparov-and-game-artificial-intelligence