CSC 831: ARTIFICIAL INTELLIGENCE

UNINFORMED SEARCH

OGUNJINMI OYERONKE PRISCILLIA

179074021

UNINFORMED SEARCH

DEPTH FIRST SEARCH

The main strategy of depth-first search is to explore deeper into the graph whenever possible. Depth-first search explores edges that come out of the most recently discovered vertex, . Only edges going to unexplored vertices are explored. When all of ’s edges have been explored, the search backtracks until it reaches an unexplored neighbor. This process continues until all of the vertices that are reachable from the original source vertex are discovered. If there are any unvisited vertices, depth-first search selects one of them as a new source and repeats the search from that vertex. The algorithm repeats this entire process until it has discovered every vertex. This algorithm is careful not to repeat vertices, so each vertex is explored once. DFS uses a stack data structure to keep track of vertices .

Depth First Search Algorithm

```
visited[v]=1;
k=1;
while(k<n)
{
    if(v==root){
        return v;
    }
    else{

for(j=n;j>=1;j--)
if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
{
visit[j]=1;
stck[top]=j;
top++;
}
v=stck[--top];
```

```
cout<<v << " ";
k++;
visit[v]=0; visited[v]=1;
}

    }
}
```

BREADTH FIRST SEARCH

Breadth-first search starts by searching a start node, followed by its adjacent nodes, then all nodes that can be reached by a path from the start node containing two edges, three edges, and so on. Formally, the BFS algorithm visits all vertices in a graph that are edges away from the source vertex before visiting any vertex edges away. This is done until no more vertices are reachable from .

Breadth First Search Algorithm

```
visited[v]=1;
k=1;
while(k<n)
{
        if(v==root){
        return v;
    }
    else{
for(j=1;j<=n;j++)
if(cost[v][j]!=0 && visited[j]!=1 && visit[j]!=1)
{
visit[j]=1;
qu[rare++]=j;
}
v=qu[front++];
cout<<v << " ";
k++;
visit[v]=0; visited[v]=1;
```

```
}
    }
 }
```

DEPTH LIMITED SEARCH

Depth-Limited Search (DLS) is a modification of depth-first search that minimizes the depth that the search algorithm may go. In addition to starting with a root and goal node, a depth is provided that the algorithm will not descend below. Any nodes below that depth are omitted from the search. This modification keeps the algorithm from indefinitely cycling by halting the search after the pre imposed depth.

ITERATIVE DEEPENING SEARCH

Iterative Deepening Search (IDS) is a derivative of DLS and combines the of depth-first search with that of breadth-first search. IDS operates by performing DLS searches with increased depths until the goal is found.

UNIFORM COST SEARCH

Uniform Cost Search is the best algorithm for a search problem, which does not involve the use of heuristics. It can solve any general graph for optimal cost. Uniform Cost Search as it sounds searches in branches which are more or less the same in cost.

HOPFIELD ALGORITHM

A recurrent neural network has feedback loops from its outputs to its inputs. The presence of such loops has a profound impact on the learning capability of the network. After applying a new input, the network output is calculated and fed back to adjust the input. Then the output is calculated again, and the process is repeated until the output becomes constant.