Ashutosh A. Sharma
Class: TE - COMP C
Roll no: 13

Experiment 3(a):
Uninformed Search

Learning Objective: Students should be able to solve a given problem using uninformed search technique

Tools: Python under Windows or Linux Environment

Theory: Study and implement DLS or DFIDS uninformed search techniques.

Depth limited search: Depth first search has some desirable properties as space, but if wrong branch is expanded with no solution on it then it may not terminate. Thus, introduce limit on branches to be expanded. Hence, not expanded a branch below a particular depth. Hence, DLS will always terminate with solutions if one exists in the limit previously set before running the program.
Too small bounds misses on the solution and too large bound may find poor solution when there are better ones. It may also run for a very long time thus remaving it's advantage over DFS.

eg. Romania problem - only 20 cities on the map. Therefore, no path longer than 19 units.

Achutosh A. Sharma
COMPC -13

Depth first iterative deepening search: Choosing depth bound provides incomplete or poor solution. It may also give no solution. This variation is complete and finds the best possible solution

Algorithm (DFIDS)

```
// returns true if target is reachable
// from src within max-depth.
bool DFIDS (src, target, max-depth)
    for limit from zero to max_depth
        if DLS (src, target, limit) == True
            return True
    return false

bool DLS (src, target, limit)
    if (src == target)
        return true
    // if reached the maximum depth
    // stop recursing

    if (limit <= 0)
        return false

    for each adjacent i of src
        if DLS (i, target, limit-1)
            return true
    return false
```

Ashutosh A. Sharma
COMPC-13

Advantages:

1. DFIDS gives us the hope to find the solution if it exists in the tree
2. When the solutions are found at lower depths, then the algorithm proves to be efficient in time.
3. Though the work done here is more, performance of DFIDS is better than single BFS and DFS operating exclusively

Disadvantages:

1. The time taken is exponential to reach the goal node and it increases greatly as the depth increases.
2. The main problem with DFIDS is the time and wasted calculations at

Properties:

Complete: Depth first iterative deepening search algorithm is complete if the branching factor is finite

Time : Let's suppose 'b' is the branching factor and depth is 'd' then the worst case time- complexity of algorithm is $O(b^d)$

Space : The space complexity of DFIDS will be $O(bd)$

Optimal : DFIDS algorithm is optimal if path cost is a non-decreasing function of the depth of the node

Applications: DFIDS is used when we do not know the depth of our solution and have to search a very large state space. It may also be used as a slightly slower substitute for BFS if we are constrained by memory or space

Design:

Program and Output:

Ashutosh A. Shama
COMPC-13

Subject :- IIS Experiment / Tutorial / Assignment No. :- 3 Page :-

Experiment 3b:
Informed Search

Learning objectives: Students should be able to solve a given problem using informed search technique

Tools : Python under Windows or Linux environment / Online Platform

Theory: Study and implement Best First search or S* search under informed search techniques

Algorithm:

1. Create an empty Priority Queue
   Priority Queue pq;

2. Insert "start" in pq
   pq.insert(start)

3. Untill priority queue is empty
   u = PriorityQueue.deletemin
   If u is the goal
   Exit
   Else
   For each neighbour v of u
   If v "unvisited"
   Mark v "visited"
   pq.insert(v)
   Mark as examined
   End.

Ashutosh A. Sharma
COMP C-13

Subject :- IIS Experiment / Tutorial / Assignment No. :- 3 Page :-

Advantages:

1. Best First search can switch between BFS and DFS by gaining the advantage of both the algorithms
2. This algorithm is more efficient than BFS and DFS algorithms

Disadvantages:

1. It can behave as an unguided depth first search in the worst case scenario.
2. It can get stuck in a loop like DFS
3. It is not an optimal algorithm wrt time and space

Properties:

Complete: Best first search is incomplete even if the state space is finite.

Time: The worst case time complexity of best first search is $O(b^m)$ where b is the branching factor and m is the maximum depth of the search space

Space: The worst case space complexity of Best first search is $O(b^m)$

Optimal: Best First search is not an optimal algorithm

Ashutosh A. Sharma
COMPC 43

Subject :- IIS Experiment / Tutorial / Assignment No. :- 3 Page :-

Applications: Best First search or A$\S$* algorithm is used to predict the closeness of the end of the path and it's solution. It is used to decide which adjacent branch is most promising and then explore

Design:
Code and output

Result and Discussion: (i) Uninformed search algorithms do not know about the goal state. (ii) Informed search algorithms have some information such as distance of nodes wrt goal node to calculate the minimum distance in minimum time

Learning Outcomes: Students should have the ability to
LO1: identify a problem which can be solved using uninformed search methods
LO2: implement uninformed search methods
LO3: describe properties of uninformed search algorithm
LO4: identify advantage and disadvantage of the algorithm.

Course outcomes: Upon completion of the course students will be able to evaluate various problem solving methods for an agent to find a sequence of actions to reach the goal state

Conclusion: In this experiment different informed and uninformed algorithms were understood and implemented. Thus, the experiment was successfully completed.