
LAB5 Uninformed Search – II

1.1 Uniform Cost Search:

Instead of expanding the shallowest node as in BFS, **uniform-cost search** expands the node n with the *lowest path cost* $g(n)$. This is done by storing the frontier as a priority queue ordered by g . The algorithm is shown below. At each stage, the path that has the lowest cost so far is extended. In this way, the path that is generated is likely to be the path with the lowest overall cost.

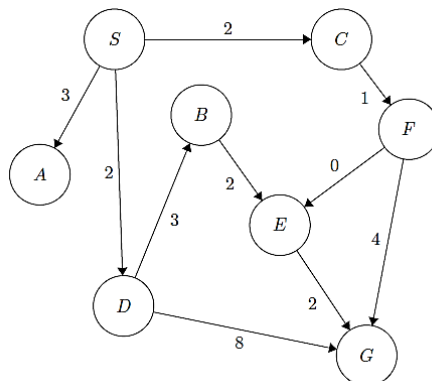
```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node  $\leftarrow$  a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier  $\leftarrow$  a priority queue ordered by PATH-COST, with node as the only element
  explored  $\leftarrow$  an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node  $\leftarrow$  POP(frontier) /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child  $\leftarrow$  CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        frontier  $\leftarrow$  INSERT(child, frontier)
      else if child.STATE is in frontier with higher PATH-COST then
        replace that frontier node with child
```

Figure 0-1. Uniform-cost search on a graph use of a priority queue and the addition of an extra check in case a shorter path to a frontier state is discovered.

1.2 Lab Tasks

Exercise 5.1.

Modify the code of Breadth First Search implemented in LAB 04 to Uniform Cost Search. Display the order in which nodes are added to the explored set, with **start state S** and **goal state G**. Path costs are mentioned on the arcs. Break ties in alphabetical order. What is the total cost of path found using UCS?



Exercise 5.2.

An alternative method that many people know for traversing a maze is to start with your hand on the left side of the maze (or the right side, if you prefer) and to follow the maze around, always keeping your left hand on the left edge of the maze wall. In this way, you are guaranteed to find the exit. Implement the UCS algorithm to find the shortest path in the following maze.

