CS 461

ARTIFICIAL INTELLIGENCE

Lecture # 04

March 11, 2021

SPRING 2021

FAST – NUCES, CFD Campus

Dr. Rabia Maqsood

rabia.maqsood@nu.edu.pk

Today’s Topics

■ Search strategies

– *Uninformed search algorithms*

■ Uniform Cost Search

■ Depth Limited Search

■ Iterative Deepening Search

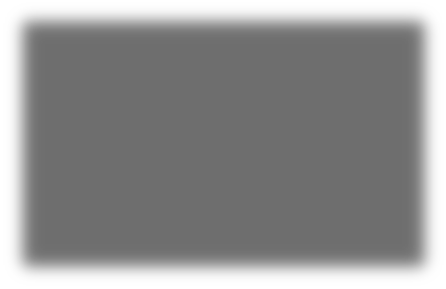
■ Bidirectional Search

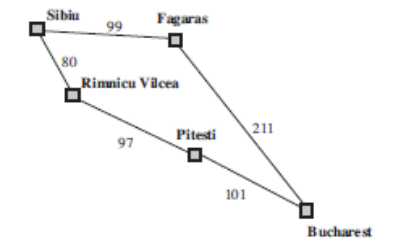
CS 461 - SPRING 2021 2

Uniform-Cost Search (UCS)

Recall that, BFS is optimal if all step costs are equal

■ Basic idea

– *Expand the least-cost unexpanded node* 

– *Works well for any cost function* 

■ More insights on UCS vs. BFS

– *Goal test is applied to a node when*

A

M

I

*it is selected for expansion*

A

:

e

c

r

– *A test is added in case a better path*

u

o

S

*is found to a node currently on the*

*frontier*

CS 461 - SPRING 2021 4

Uniform-Cost Search (UCS) Recall that, BFS is optimal if all step costs are equal

■ Basic idea

– *Expand the least-cost unexpanded node*

– *Works well for any cost function*

■ Implementation

■ Properties

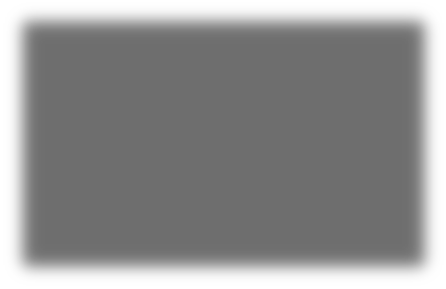
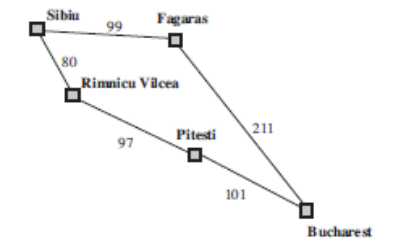
– *Complete?*

– *Optimal?*

– *Time?*

– *Space?*

In UCS, the frontier is a priority queue

A 

M

I

A

:

e

c

r

u

o

S

CS 461 - SPRING 2021 5

Uniform-Cost Search (UCS)

■ Complete: Yes (if step cost ≥ ε)

■ Optimal: Yes (node with least-cost is always expanded)

■ Time: *No. of nodes with g <= cost of optimal solution*

■ Space: *No. of nodes with g <= cost of optimal solution*

CS 461 - SPRING 2021 6

Depth-limited Search

■ Basic idea

– *DFS with depth limit k, i.e., nodes at depth k have no successors* ■ Implementation

■ Properties

– *Complete: NO (if k < d)*

– *Optimality: NO (if k > d)*

– *Time: O(bk)*

– *Space: O(bk)*

CS 461 - SPRING 2021 8

Iterative Deepening Search (IDS)

▪ Basic idea

▪ Use DFS to look for solutions at depth 1, then 2, then 3, etc.

▪ For depth D, ignore any paths with longer length

▪ Depth-bounded depth-first search

CS 461 - SPRING 2021 9

Iterative Deepening Search (IDS)

depth = 1 depth = 2

If no goal re-start from scratch and get to depth 2

If no goal re-start from scratch

and get to depth 3

depth = 3. . .

If no goal re-start from scratch and get to depth 4

CS 461 - SPRING 2021 10

Iterative Deepening Search (IDS)

■ Properties

– *Complete: Yes*

– *Optimal: Yes, if step cost = 1*

– *Time: (d)b + (d-1)b2 + … + (1)bd = O(bd)*

– *Space: O(bd)*

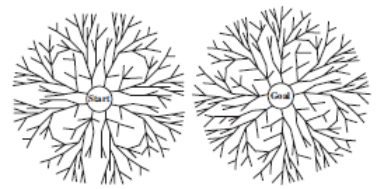
■ Repetition is wasteful – isn’t it?

CS 461 - SPRING 2021 11

Bidirectional Search

■ Basic idea

– *Run two simultaneous searches, one forward from the initial state and other backward from the goal state, hoping that the two searches meet in the middle*

■ Implementation 

■ Properties

– *Complete: Yes*

– *Optimal: No* – *Time: O(bd/2)*

A

MI

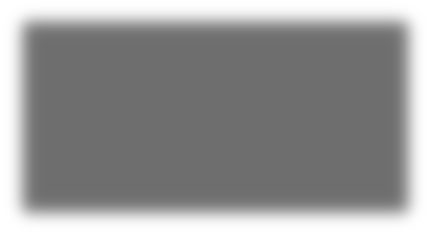
A

:

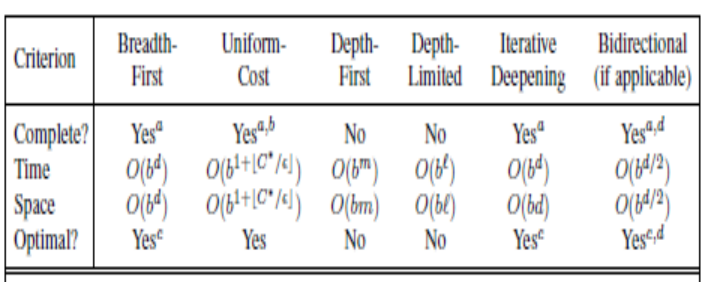
e

c

r

– *Space: O(bd/2)*SouCS 461 - SPRING 2021 13

Comparison: Uninformed Search Strategies

A 

M

I

A

:

e

c

r

u

o

S

CS 461 - SPRING 2021 14

Summary

■ Problem formulation usually requires abstracting away real-world details to define a state space that can feasibly be explored

■ Variety of uninformed search strategies

■ Iterative deepening search uses only linear space and not much more time than other uninformed algorithms

CS 461 - SPRING 2021 15

Graph search: basic idea

Input:

- a graph

- a set of start nodes

- Boolean procedure goal(n) testing if n is a goal node

frontier:= [<s>: s is a start node]; While frontier is not empty:

select and remove path

<no,….,nk> from

frontier;

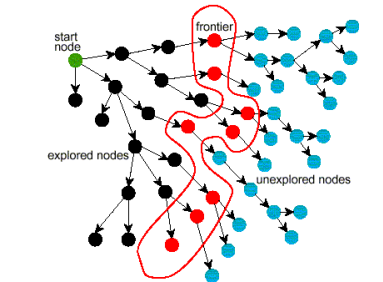
If goal(nk)

return <no,….,nk>;

For every neighbor n of nk,

add <no,….,nk, n> to frontier;

The way in which the frontier is expanded defines the search strategy



end CS 461 - SPRING 2021 16

Reading Material

■ Russell & Norvig: Chapter # 3

■ David Poole: Chapter # 3

CS 461 - SPRING 2021 17