No request Text

Ass: 27%

Project: 12%

Midterm1: 18

Mid 2: 18

Final: rest

**Overview:**

Introduction AI

Search

Knowledge

Perception

What is Intelligent

. act rationally

**AI goals**

Strong AI:

Weak AI: designed to solve exactly one problem

$ diff perspecticves:

. Think vs Acting

. Human vs Rational

**Turning Test (Acting Humanly)**

**Cognitive Modeling (How human think process)**

. Both approaches are now distinct from AI

**Formalizing the “Laws of though” think Rationally**

.Logic

. Aristotle: arugument

**Acting Rationally: Rational Agent**

. Rational behavior: doing the right thing

. goal achievement

. Should serve rational action

**Building Intelligent Machine**

. I

. II Developing methods to match or exceed human performance

**Methodology of AI**

. Theoretical aspect: math

. Engineering aspect: The act of building

. empirical aspect: experiments

**AI leverages from diff disciplines**

. Philosophy: logic, methods

. Computer science and engineering: complex theory, algorithm, logic and interface, programming language

. Mathematics and physics: probability theory, statics modeling, continuous mathematics.

**AI: More Direct Influence**

. Obtaining an understanding of the human mind is one of the final frontier of modern science

**History of AI: Milestones The gestation of AI 1943 -1956.**

History of AI: early enthusiasm, greate expectations (1952 - 1969)

History of AI: a dose reality (1966 - 1978)

**Vison (Perception)**

Factor in Accelerated Progress.

What we can’t do Yet

Societal Issues

**Search**

Problem Solving as Search

. Search is a central topic in AI

**Define a search Problem**

. A Search problem consist: State space; A successor function (action + cost); and a state and a goal test.

. A Path is any sequence of states connected by a sequence of actions

. Path Cost: function that assigns a cost to a path; relevant if more than one path leads to the goal, and we want the shortest path.

. A solution is a sequence of actions (a plan)

**Example: the 8 = puzzle**

. State space? The location of each tile

. Initial State:? Any State can be the initial state

. Goal test:? Whether the state matches the goal state

. Successor Function: ? the movement of the blank space; ><^,down

. Path Cost:?Each step cost 1

**Cryptarithmetic**

. State space

. Goal Test

. Successor function

. Path Cost

**Ex: Traveling in Romania (Arad - Bucharest)**

. State space? cities

. Initial State:? arad

. Goal test:? Is state == Bucharest

. Successor Function: ? Road: go to adjacent cities with cost = distance

. Path Cost:? The cost of a path

**Solving a search problem: State Space Search**

**Input:**

. Initial state

. Goal test

. Sucessor function

. Path cost function

**Output:** path from initial state to goal. Solution quality is measured by the path cost function

. Expanding: apply each legal action to the current state.

Search procedure defines a search tree.

. root node – initial state

. children of a node – successor stats

. Leave of tree: states is not

. Search strategy- algorithm for deciding which leaf node to expand next

**Node data Structure.**

. Node data structure is used to keep track of the search tree.

. Node vs states:

. a Node is a bookkeeping data structure used to represent the search tree

. a state corresponds to configuration of the world.

**Evaluating a search strategy**

. Completeness: is the strategy guarantee to find a solution when there is one?

. Time complexity: how long does it take to find a solution.

. Space complexity: how much memory does it need?

. Optimally: doea strategy

**Generic Tree-Search Algorithm**

. Add initial state to the frontier

Loop:

Node = remove-frontier() – and save in order to return as part of path goal

If goal-tets(node) = true return path to goal

S = successors(node)

Add S to frontier

Until frontier is empty

Return failure

Tree Search vs Graph Search

.

.

Graph search

. idea: never expend a state stwice

. How

BFS:Example

**Time and memory requirement for BFS**

. Let b=Branching factor -> maximum b=number of sucessors of any node

. d = solution depth -> the shallowest goal node

. Then the maximum number of nodes generate is:

b + b2 + … + bd = O(bd)

. For graph search,

O(bd-1) in the closed set and O(bd) in the frontier.

**Uniform cost search**

. Use BFS, but always expand the lowest cost node on the frontier as measured by path cost g(n).

G(Successor(n)) > g(n) is a necessary condition for completeness and a sufficient condition for optimally

**Uninformed search: DFS**

. Use the last in, first out or LIFO queue to store the frontier

**DFS example:**

**Time and memory requirements for DFS**

. let b = branching factor -> maximum number of successor of any node

. m = maximum depth of any node.

.Time = for tree search, then the maximum number of nodes generates is O(bm).

. Space: for tree search, only need to store O(bm) node

Iterative Deeping {Korf 1985}

.

. Use artificial depth cut off c.

.To much space for BFS, choose DFS.

Iterative Deepening

.

Space requirement: same is DFS

. Time requirement would seem very very expensive! But not much different from single BFS or DFS to depth d

. Reason: almost

Example: d= 10, d= 5 . the number of nodes generates in a BFS

B + b2 + … + bn = 10 + 100 + …. 100000 = 111 110  
**Bidirectional Search**

. When is bidirectional search applicable?

. Generate predecessor is easy. (“no queen attacks another queen”).

Bidirectional Search

. Search forward from the start stae and backward from the goal state, simulanesouly and stop when the two seaches meet the middle.

. If braching factor = b fromboth directions, and solution exists at depth d, then need only O (2bd/2) = O(bd/2)

**Limitations**

. What are problem of all the methods? Slow!

. The blind in the sense that the inf of the goal sate is not used.

**. informed search:**

. With the guided of the goal state.

Informed Method: Heuristic Search

. We want to have some estimate of the distance from the states in the frontier to the goal state.

. Why estimate? Because the states we can reach are based on the action we can take.

Informed Methods: Heuristic (tự tìm tòi) Search

. We use an evalution function f(n) as our estimate

Best first search:

. **Nodes are selected for expansion** based on evaluations functions

. Expand the node with the lowest evalution

. The evaluation function can be complex: f(n) + f1(n) + f2(n)+…

Greedy Best first Search

. One natural component of f(n):

. Heuristic function:

. h(n) = estimate cost of the cheapest path from the state at node

. Greedy Bets -first Search

. Expands the node that is “Closet ” to the goal as measured by h(n)

. A common case:

. Best-first takes you straight to the (wrong) goal

. Worst case: like a badly-guided DFS

Example: 8 -puzzle problem

. Design of heuristic function is important.

. one possible h funciotn: the number tail misplace

Example: Find Path from Arad to Bucharest

. One possible heuristic function: The straight line distance to Buchasrtest

Greedy Best First Search

A\* search

. [https://towardsdatascience.com/intro-to-a-search-a3dfa444ad20#:~:text=A\*%20search%20is%20optimal%20if,to%20find%20the%20optimal%20path](https://towardsdatascience.com/intro-to-a-search-a3dfa444ad20#:~:text=A*%20search%20is%20optimal%20if,to%20find%20the%20optimal%20path).

Formal Definition of Admissibility

.

Example: Admissible Heuristic

. Path finding

Optimality A\* tree search

.