ECII/ECSI 3206:
Artificial Intelligence [and expert systems]
Topic 3: Search Algorithms and problem Solving

By: Edgar Otieno

Problem Solving

- Problem solving is an important concept in A.I
- Problem solving can broadly be classified into Search based techniques or Knowledge based techniques.
- **Weak A.I-**solve single problem or task
- **Strong A.I**-Self aware systems that solve complex problems

Definition of terms

- **Problem**-a situation consisting of a goal and a set of actions to be taken to achieve that goal
- **Searching**-A universal technique used in AI for problem solving that computers use to examine a problem space in order to find a goal
- **Problem space/search space**-environment within which a search takes place
- **Problem instance**-initial state +goal state
- **Initial state**-How we start a problem e.g starting to look for a person in a game of hide and seek
- **Goal state**-state that satisfies the set objective e.g finding the person who has hid themselves

Definitions cont.

- **Problem space graph**-it is a map representing the problem state. The states are shown by nodes and operations shown by arcs/edges
- **Depth of a problem**-shortest path from initial state to goal state
- **Admissibility of an algorithm-** the ability of an algorithm to always find an optimal solution

Factors to consider when choosing a Seach Algorithm

- **Completeness**-the guarantee to always find a solution
- **Optimality**-Always find a solution at the lowest cost
- **Time complexity**-max. amount of time required to reach a solution
- **Space complexit**y-Max. amount of space[memory] required to reach a solution

Searching strategies

1. Brute-force/Blind/Uninformed Search Strategies
2. Informed/Heuristic Search strategies
3. Optimization strategies
4. Game playing Strategies
5. Evolutionary Algorithms

1. Brute force/uniformed/blind search strategies

- They do not need domain specific knowledge
- Instead they need:
  1. State description/problem space state
  2. Set of valid operations
  3. Initial state
  4. Goal state

Examples of uninformed search techniques

a) Breadth First Search[BFS]

-Traverse from root to all nodes at same level(neighbors) before proceeding to the next level

-Implemented using Queue

-Optimality[YES]

-Completeness[YES]

Adv-It can check for duplicate nodes

Dis-Consumes lots of memory space

Cont....

b) Depth First search[DFS]

-traverse from root to last child of the leftmost leafnode before moving to the next child node.

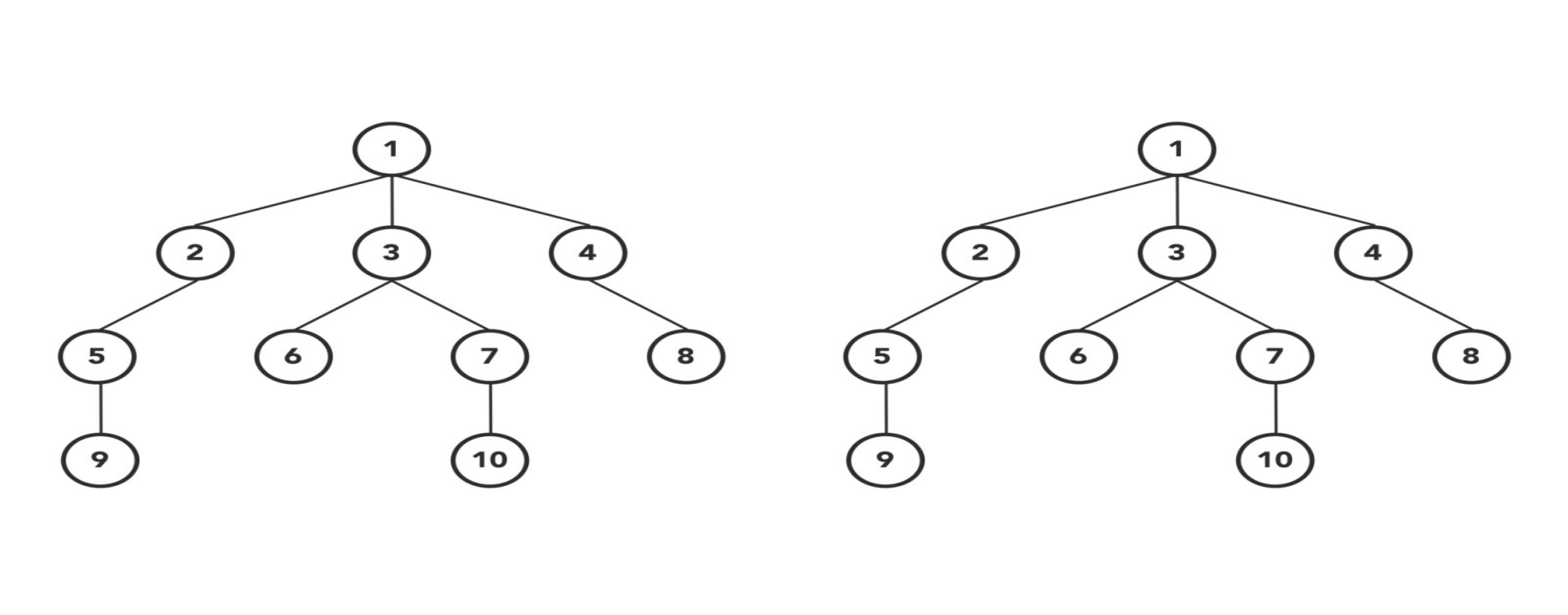
-Implemented using Stack

-optimality[NO]

-completeness[NO]

Dis-may continue indefinitely. Also cant check for duplicates.

DFS vs BFS traversals



Cont…

c) Bi-Directional Search

-It searches forward from an initial state and backwards from a goal state until both meet to identify a common state

-Each search is done only upto approximately half of the total path

Cont..

d) Uniform Cost search

-Sorting is done in increasing cost of the path to a node. It always expands the least cost node.

-When each transaction has the same cost, it works the same as BFS

-implemented using a Queue

-Optimality[YES]

-Completeness [Yes]

Dis-It may end up taking too long since it explores all paths.

Cont..

e) **Iterative Deepening DFS**

It performs a DFS at the first level, Starts again and performs a DFS at level 2 and continues that way until it reaches a solution.

-Optimal[YES]

-Completeness[YES]

2. Informed/ Heuristic Search strategies

- They calculate the optimal path between two states
- Problem specific knowledge need to be added to increase the efficiency if these algorithms
- Heuristic search means that the search algorithm may not find the optimal solution, however, it will always give a good enough solution in reasonable time.

Examples of Informed search techniques

a) Greedy Best First Search

-Expands node that is estimated to be closest to the goal

-implemented using priority Queue

Optimality[NO]-it may get stuck in loops

Competeness[NO]

Cont..

b) A* Search

-It is a form of BFS

-Expands the promising path first i.e avoids expanding path that are already expensive

-Implemented using Priority Queue

F(n)=g(n)+h(n)

where f(n) is the estimated total cost of a path to the desired goal/ the cheapest solution, g(n) represents greedy cost and f(n) represents uniform cost

3. Optimization Search techniques

- These are algorithms used for optimization of heuristic search techniques
- Given a large set of input, they try to find a sufficiently good enough solution to a problem
- Think of the travelling salesperson problem who has to travel through the various towns in Kenya starting from Nairobi and achieve this at the most optimal cost.

Example of optimization Search Algorithms

a) Hill climbing/Gradient descent

-It works on the principle of always choosing the next best successor.

-uses generate and test technique i.e. an iterative algorithm that starts with an arbitrary solution to a problem and attempts to find a better solution to the problem by changing a single element of the solution incrementally.

It also uses the greedy approach i.e. If the change produces a better solution, this incremental change is taken a the new solution and we repeat the process again until no further improvement to the solution can be made.

-optimality[NO]-problems of Global maxima Vs. Local maxima

-Completeness[NO]

Hill climbing cont.

- Types of Hill climbing
  - Simple hill climbing[select neighbors which optimize current cost]
  - Steepest Hill climbing[Select neighbor closest to the solution]
  - Stochastic hill climbing[There is no examining of nodes. Select of neighbor node to explore is done at random]

4. Game playing Algorithms/ Adversarial Search techniques

-Are modelled along games

-Both players try to win a game .To achieve this, both players try to make the best possible move that will maximize their benefit and Minimize their opponents benefit. E.g MiniMax algorithm

Example of Game playing Search Algorithms

a) MiniMax Algorithm

It uses two main functions

MOVEGEN-all possible moves that can be generated/ are possible from the current position

STATICEVALUATION-returns value depending on its goodness from the two players.

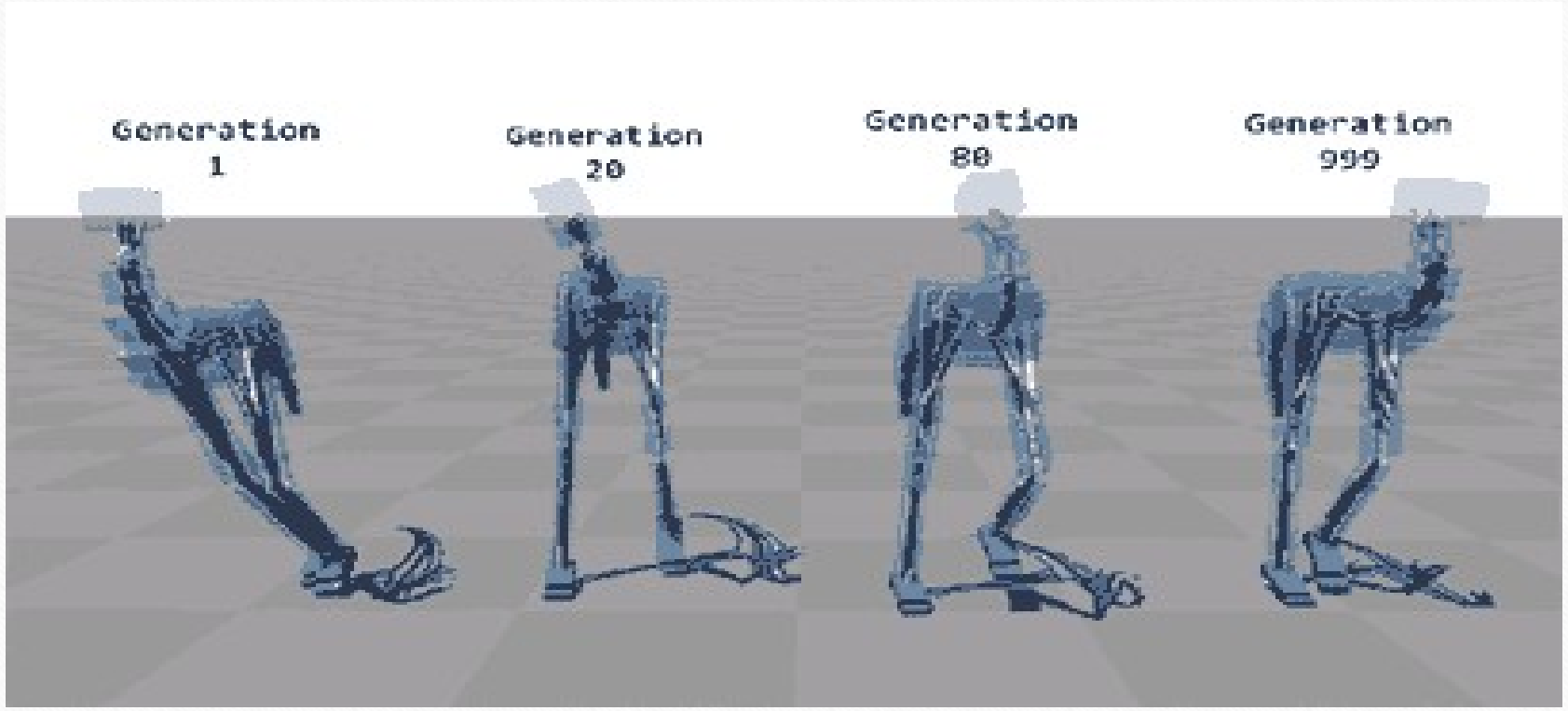
5. Evolutionary Algorithms[E.A]

- Evolutionary algorithms are a heuristic-based approach to solving problems
- These are a subset of Heuristic/ Informed algorithms that are based on biological evolution theories of survival for the fittest.
- A ‘population’ of possible solutions to the problem is first created with each solution being scored using a ‘fitness function’ that indicates how good they are. The population evolves over time and (hopefully) identifies better solutions. Of the various types of evolutionary algorithm, the **genetic algorithm** is the most well

Example…Evolutionary Algorithms[E.A]

- To illustrate the result of this process I will show an example of an EA in action. The gif shows several generations of dinosaurs learning to walk by optimizing their body structure and applied muscular forces. From left to right the generation increases, so the further right, the more optimized the walking process is. Despite the fact that the early generation dinosaurs were unable to walk, the EA was able to evolve the dinosaurs over time through mutation and crossover into a form that was able to walk…….

Example…. Evolutionary Algorithms



Other algorithms related to Evolutionary algorithms

- **Ant colony optimization** is based on the ideas of ant foraging by pheromone communication to form paths. Primarily suited for combinatorial optimization and graph problems.
- **The runner-root algorithm** (RRA) is inspired by the function of runners and roots of plants in nature
- **Artificial bee colony algorithm** is based on the honey bee foraging behavior. Primarily proposed for numerical optimization and extended to solve combinatorial, constrained and multi-objective optimization problems.
- **Bees algorithm** is based on the foraging behavior of honey bees. It has been applied in many applications such as routing and scheduling.
- **Particle swarm optimization** is based on the ideas of animal flocking behavior. Also primarily suited for numerical optimization problems.

Applications of Searching Techniques in A.I. problem solving

- Finance-Bots for investing in stocks, forex ,crypto currency, automated auctions e.t.c [but be weary of online scams because it ain’t easy to take bots to court for prosecution]
- Medicine-D.S.S
- Telecommunications-network maintenance scheduling
- Data mining