Introduction to Artificial Intelligence and Search Algorithms

Introduction to Artificial Intelligence

- **What is Intelligence?**
- Intelligence is the ability to learn about, to learn from, to understand about, and interact with one's environment.
-
- **What is Artificial Intelligence (AI)?**
- **A.I:-** Is simply a way of making a computer think.
- **A.I:-** Is the part of computer science concerned with designing intelligent computer system that exhibit the characteristic associated with intelligent in human behavior.
- **This requires many processes:-**
- **1- Learning: -** acquiring the knowledge and rules that used these knowledge.
- **2- Reasoning:-** Used the previous rules to access to nearly reasoning or fixed reasoning.

Introduction to Artificial Intelligence

- **A.I Principles:-**
- 1- The data structures used in knowledge representation.
- 2- The algorithms needed to apply that knowledge.
- 3- The language and programming techniques used their implementation.

- **What are the goals of AI research?**
- The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects.

- **What is problem reduction meaning?**
- Problem Reduction means that there is a hard problem may be one that can be reduced to a number of simple problems. Once each of the simple problems is solved, then the hard problem has been solved.

Applications of AI

- • Game playing
- • Speech recognition
- • Understanding natural language
- • Computer vision
- • Expert systems
- • Heuristic classification

Characteristics of AI

- • High societal impact (affect billions of people)
- • Diverse (language, vision, robotics)
- • Complex (really hard)

Search Algorithms

To successfully design and implement search algorithms, a programmer must be able to analyze and predict their behavior.

Many questions needed to be answered by the algorithm these include:

- Is the problem solver guaranteed to find a solution?
- Will the problem solver always terminate, or can it become caught in an infinite loop?
- When a solution is found, is it guaranteed to be optimal?
- What is the complexity of the search process in terms of time usage? Space search?
- How can the interpreter be designed to most effectively utilize a representation language?

State Space Search

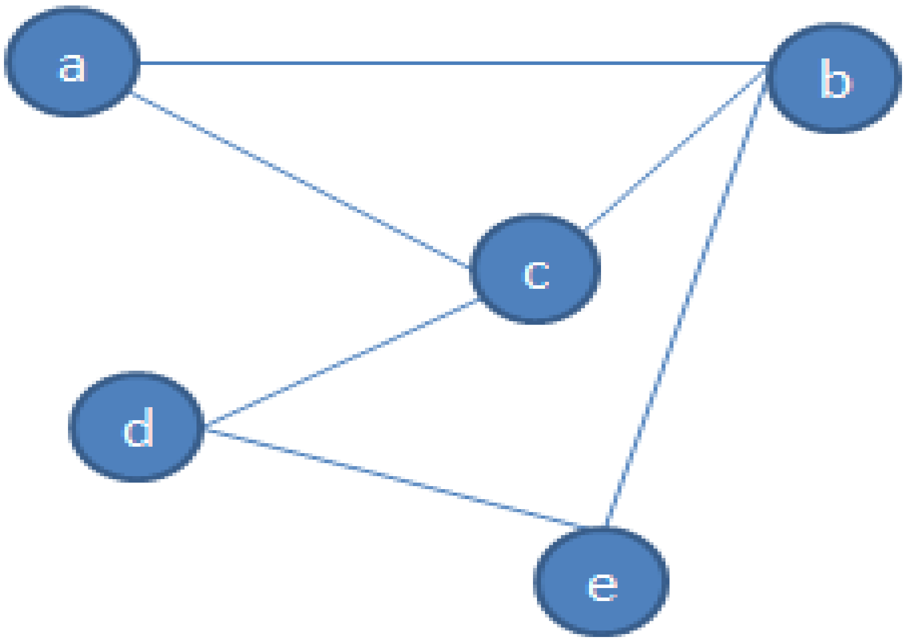
- The theory of state space search is our primary tool for answering these questions, by representing a problem as state space graph, we can use graph theory to analyze the structure and complexity of both the problem and procedures used to solve it.

Graph Theory:-

- A graph consists of a set of a nodes and a set of arcs or links connecting pairs of nodes. The domain of state space search, the nodes are interpreted to be stated in problem solving process, and the arcs are taken to be transitions between states.

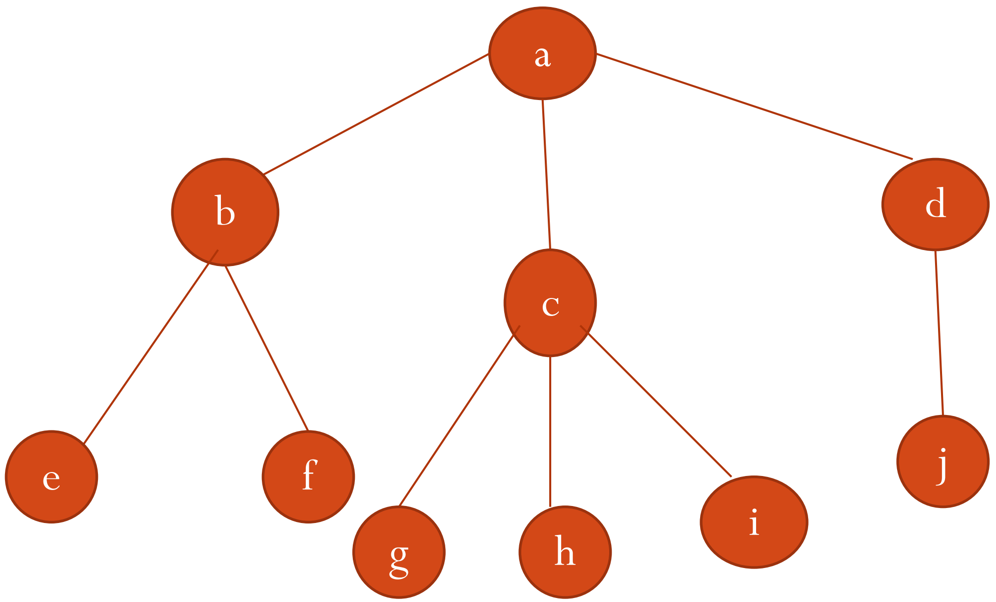
Graph theory is our best tool for reasoning about the structure of objects and relations.

Graph Theory



Nodes={a,b,c,d,e}

Arcs={(a,b), (a,c),(b,c),(b,e),(d,e),(d,c),(e,d)}



Nodes={a,b,c,d,e,f,g,h,i,j}
Arcs={(a,b),(a,c),(a,d),(b,e),(b,f),(c,g),(c,h),(c,i),(d,j)}

State Space Representation

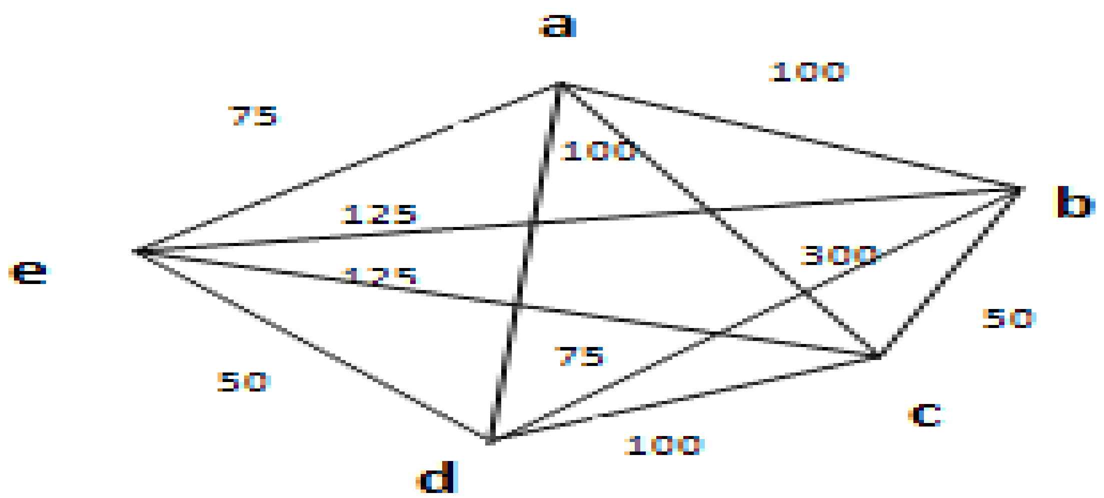
A state space is represented by four tuple [N,A,S,GD], where:-

- **N** is a set of nodes or states of the graph. These correspond to the states in a problem –solving process.
- **A** is the set of arcs between the nodes. These correspond to the steps in a problem –solving process.
- **S** a nonempty subset of N , contains the start state of the problem.
- **G** a nonempty subset of N contains the goal state of the problem.

- **A solution path:- Is a path through this graph from a node S to a node in GD.**

Example:- Traveling Salesman Problem

- Starting at A , find the shortest path through all the cities , visiting each city exactly once returning to A.



The complexity of exhaustive search in the traveling Salesman is **(N-1)!**, where N is the No. of cities in the graph. There are several technique that reduce the search complexity.