CONTENT

1.Problem space and search.

- Informed search techniques
- Uninformed search techniques

2. Knowledge.

- Definition of knowledge
- Importance of knowledge

WHAT IS PROBLEM SPACE ?

Refers to the entire range of components that exist in the process of finding a solution to a problem.

It is the environment in which the search takes place.

Set of all possible states.

It may consists:

- Starting state of problem.
- Goal state
- Operators.

EXAMPLE:

Start state:

3	10	13	7
9	14	6	1
4		15	2
11	8	5	12

Goal state:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

- The start state is some (almost) random configuration of the tiles
- The goal state is as shown
- Operators are
  - Move empty space up
  - Move empty space down
  - Move empty space right
  - Move empty space left
- Operators apply if not against edge

SEARCH

- **Search is about looking for something. Searching techniques are strategies that find solution to a problem.**
- **It is a problem solving technique.**
- **Solution to the search problem is a sequence of actions which achieve through Search algorithms.**

SEARCH STRATEGY

A search strategy is defined by picking the order of node expansion.

Strategies are evaluated along the following dimensions:

- **Completeness:** does it always find a solution if one exists?
- **Optimality:** does it always find a least-cost solution?

CONTD…

- **Time complexity**: number of nodes generated.
- **Space complexity**: maximum number of nodes in memory.

TYPES:

- **INFORMED.**
- **UNINFORMED.**

The informed search will provide the direction regarding the solution while in uninformed search has no suggestion regarding the solution. This makes an uninformed search more lengthy when the algorithm is implemented.

- Depth-first search, breadth-first search, and lowest cost first search are the algorithms come under the category of the uninformed search.
- The informed search covers the heuristic search algorithms such as A* algorithm, hill climbing algorithm.

INFORMED AND UN INFORMED SEARCH

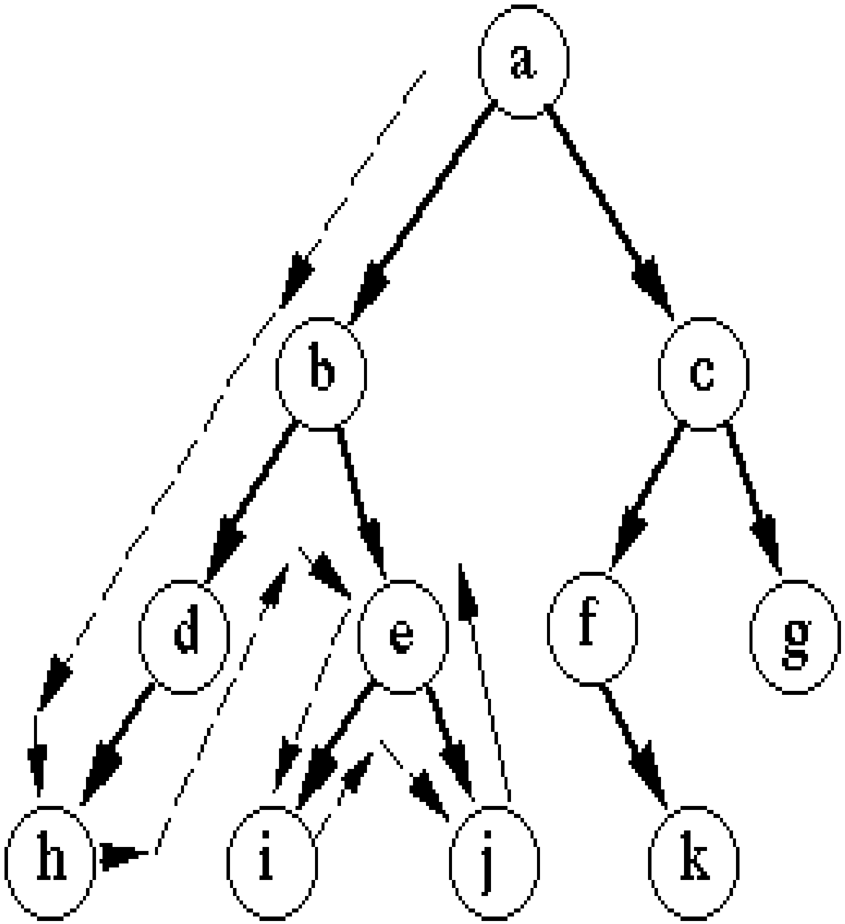
Basis of comparison	Informed search	Uninformed search
Basic knowledge	Uses knowledge to find the steps to the solution.	No use of knowledge
Efficiency	Highly efficient as consumes less time and cost.	Efficiency is mediatory
Cost	Low	Comparatively high
Performance	Finds the solution more quickly.	Speed is slower than the informed search.

DEPTH FIRST SEARCH

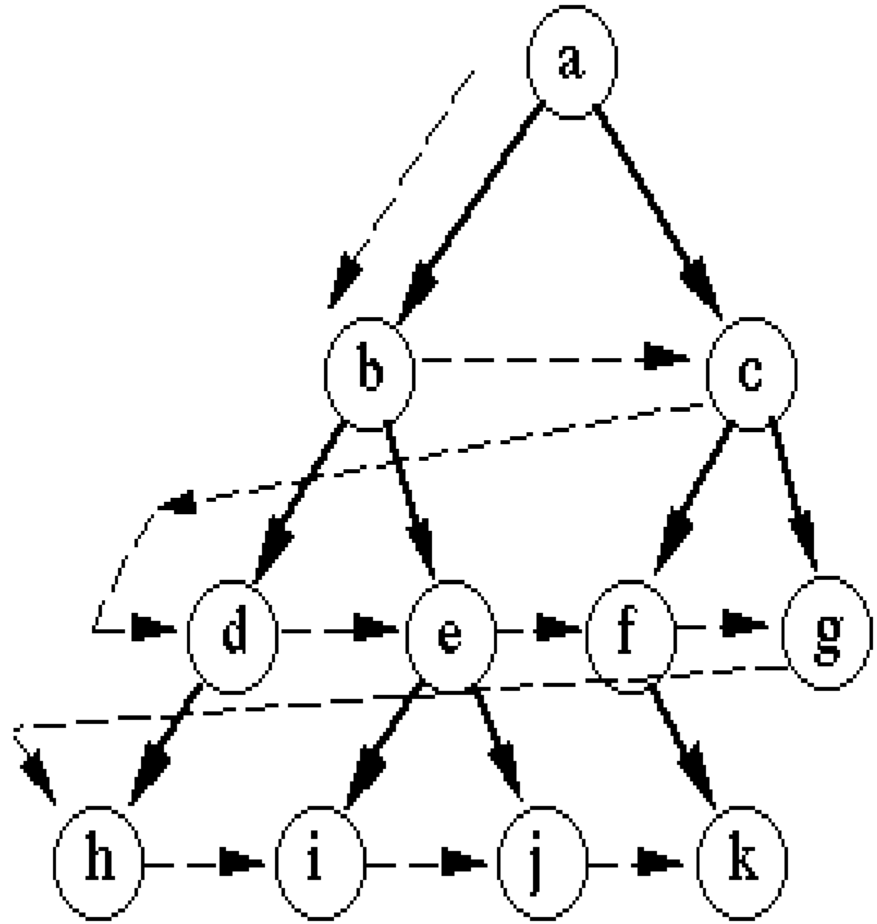
Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking

BREADTH-FIRST SEARCH

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a ‘search key’), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.



Depth-first search



Breadth-first search

BFS:

Pros: No dead end, Complete and optimum.

Cons: time & memory consuming.

DFS:

Pros: small amount of memory, accidentally can find solution in short time.

Cons: incomplete possibility (tree-search allow redundant path.)

HEURISTIC SEARCH

A Heuristic is a technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot.

This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.

At each branching step, it evaluates the available information and makes a decision on which branch to follow. It does so by ranking alternatives.

A* ALGORITHM

IDEA: Avoiding expanding paths that are expensive.

EVALUATION FUNCTION :

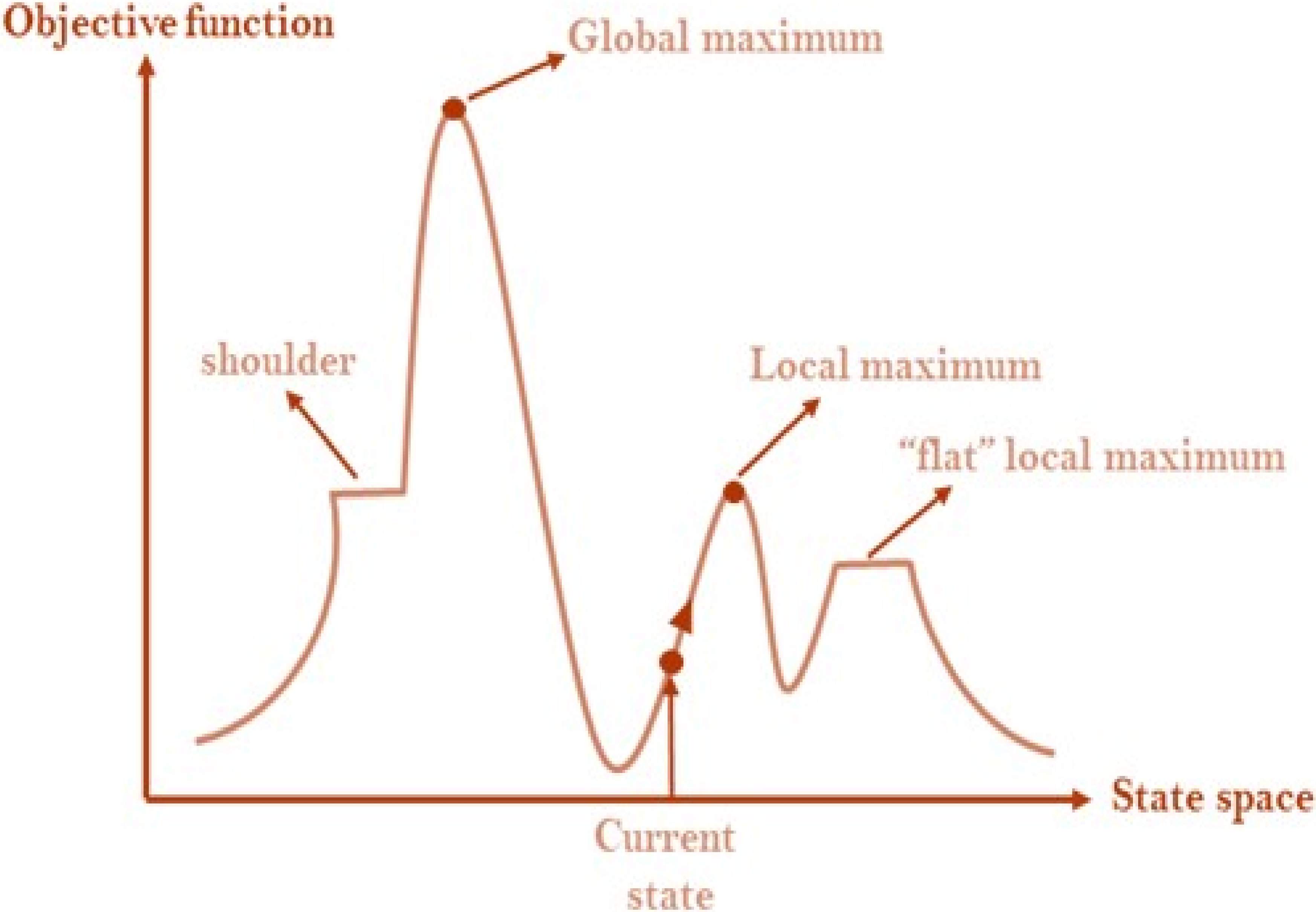
$$f(n)=g(n)+h(n)$$

- **g(n)** : cost so far to reach n.
- **h(n):**estimated cost from n to goal node.
- **f(n):**estimated total cost of path from n to goal node.

(where n is a current node)

HILL CLIMBING

- **Hill climbing algorithm moves up in the direction of increasing value i.e. uphill.**
- **Breaks its moving up loop when it reaches a peak when no neighbor has higher value.**
- **It does not maintain search tree.**
- **Only look for immediate neighbor of current node.**


Objective function
Global maximum
shoulder
Local maximum
"flat" local maximum
State space
Current
state

TYPES

- **Simple Hill climbing:** It examines the neighboring nodes one by one and selects the first neighboring node.
- **Steepest-Ascent Hill climbing:** It first examines all the neighboring nodes and then selects the node closest to the solution state as of next node
- **Stochastic hill climbing :** It just selects a neighboring node at random and decides whether to move to that neighbor or to examine another.

PROBLEMS WITH HILL CLIMBING

Local maximum : At a local maximum all neighboring states have a values which is worse than the current state. Since hill-climbing uses a greedy approach, it will not move to the worse state and terminate itself. The process will end even though a better solution may exist.

Plateau : **On plateau all neighbors have same value . Hence, it is not possible to select the best direction.**

Ridge : **Any point on a ridge can look like peak because movement in all possible directions is downward. Hence the algorithm stops when it reaches this**

KNOWLEDGE

- **In artificial intelligence knowledge concerned with AI agents thinking and how thinking process an information into knowledge.**
- **Machines do all these things comes under knowledge representation and reasoning.**
- **Knowledge is the body of facts and principles. Knowledge can be language, concepts, procedures, rules, ideas, abstractions, places, customs, and so on.**

IMPORTANCE OF KNOWLEDE

- **Knowledge plays a major role in building intelligent systems.**
- **Intelligence requires knowledge. That is, to exhibit intelligence, knowledge is required.**