

Course: Advanced Algorithm and Problem Solving

WEEK 12 Artificial Intelligence Algorithms

Lemlem Kassa (Ph.D.)

Addis Ababa Science and Technology University, Ethiopia

June, 2025

WEEK 12 : Artificial Intelligence Algorithms

Content

- Overview of Artificial Intelligence (AI)
- Types of Searching Algorithms
 - Uninformed search algorithms
 - Informed search algorithms

Lecture Learning Outcome

- Understand components of searching problem in Artificial Intelligence
- Differentiate AI Search & Optimization Algorithms
- Understand informed and Uninformed Search Algorithms
- Understand the heuristics approaches in informed search algorithms
- Understand applications of informed and Uninformed Search Algorithms

Overview of Artificial Intelligence (AI)

What is Artificial Intelligence (AI)

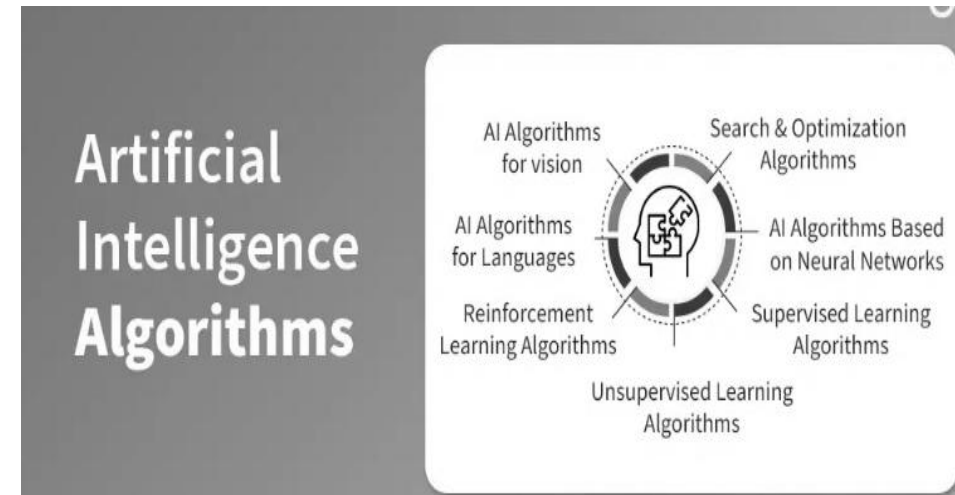
- The simulation of human intelligence in machines, allowing them to perform tasks that typically require human cognitive functions such as learning, reasoning, problem-solving, perception, and decision-making.
- AI systems analyze vast amounts of data, adapt to new information, and enhance their performance autonomously over time.
- The ultimate goal of AI is to develop machines that can think, reason, and act autonomously, and in some cases, surpass human capabilities across various domains

[1]. What is AI (Artificial Intelligence)? Definition, Types, Examples & Use Cases, Lev Craig, Nicole Laskowski, Linda Tucci, TechTarget, 2024. <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>

- AI algorithms encompass a diverse range of techniques, enabling machines to learn from data, identify patterns, make decisions, and solve complex problems autonomously.

1. AI Search & Optimization Algorithms

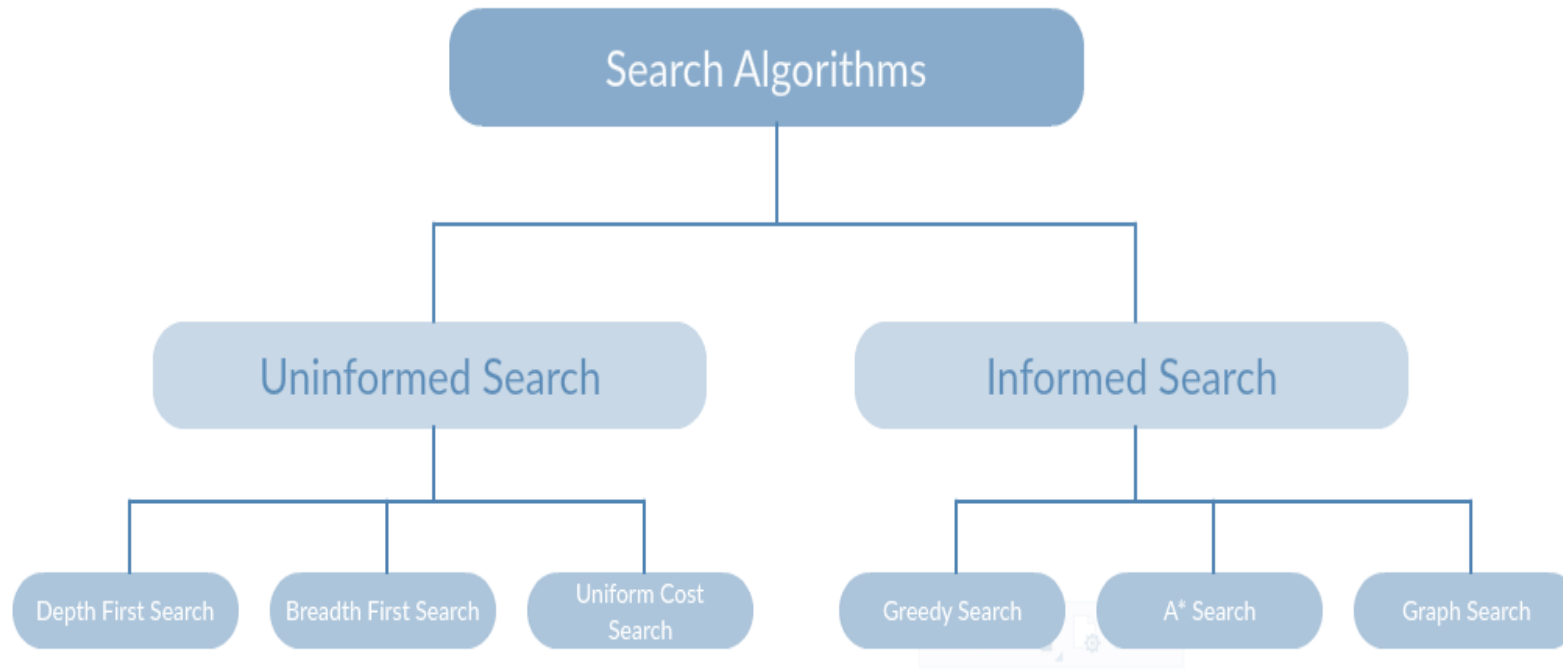
- AI search and optimization algorithms are fundamental tools in artificial intelligence (AI) for solving complex problems.
- These algorithms are designed to explore vast search spaces and find optimal solutions or make well-informed decisions.



- A search problem consists of:
 - A **State Space**. Set of all possible states where we can be.
 - A **Start State**. The state from where the search begins.
 - A **Goal State**. A function that looks at the current state returns whether or not it is the goal state.
- The **Solution** to a search problem is a sequence of actions, called the **plan** that transforms the start state to the goal state. This plan is achieved through search algorithms.

[2]. Artificial Intelligence (AI) Algorithms, GeeksforGeeks, GeeksforGeeks,2024.
<https://www.geeksforgeeks.org/ai-algorithms/>

Searching Algorithms in AI



[3]. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023.
<https://www.geeksforgeeks.org/search-algorithms-in-ai/>

Types of Searching Algorithms

Uninformed Search Algorithms

- The search algorithms in this section have no additional information on the goal node other than the one provided in the problem definition.
- The plans to reach the goal state from the start state differ only by the order and/or length of actions.
- Uninformed search is also called **Blind search**. These algorithms can only generate the successors and differentiate between the goal state and non goal state.

[3]. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023.
<https://www.geeksforgeeks.org/search-algorithms-in-ai/>

Types of **uninformed search** algorithms

- Depth First Search
- Breadth First Search
- Uniform Cost Search

Each of these algorithms will have:

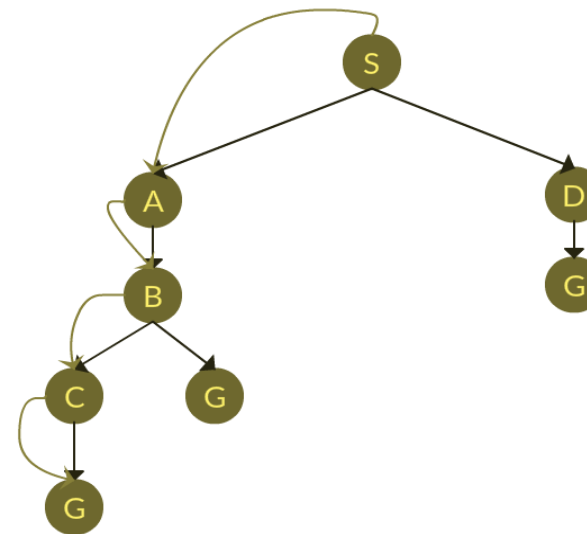
- A problem **graph**:- containing the start node S and the goal node G.
- A **strategy**:- describing the manner in which the graph will be traversed to get to G.
- A **fringe**:- which is a data structure used to store all the possible states (nodes) that we can go from the current states.
- A **tree**:- that results while traversing to the goal node.
- A solution **plan**:- which the sequence of nodes from S to G.

Uninformed/Blind Search Algorithm

a. Depth First Search (DFS)

- 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. It uses last in- first-out strategy and hence it is implemented using a stack.
- As DFS traverses the tree "deepest node first", it would always pick the deeper branch until it reaches the solution (or it runs out of nodes, and goes to the next branch).

• **Path:** S -> A -> B -> C -> G



[3]. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023.
<https://www.geeksforgeeks.org/search-algorithms-in-ai/>

Uninformed/Blind Search Algorithm

Applications of DFS in AI

- **Maze generation:** The Maze generation is comprised of designing a layout of passages and walls within a maze.
- This maze generation makes use of a randomized approach of the Depth-first search algorithm because it leverages the recursive method and stack.
- The DFS performs by selecting any random neighbor at first that has not been visited. It removes the wall between the two cells that are not connected and then it adds the new cell to the stack.
- This process continues until there is no more solution can be generated, resulting in a complete maze.

.....Uninformed Search Algorithms

Applications of DFS in AI

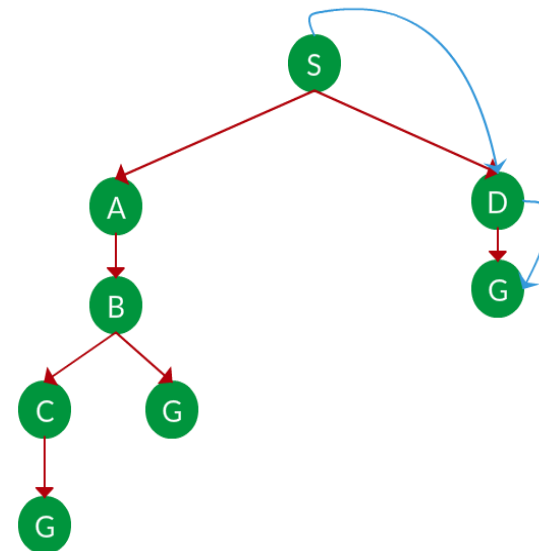
- **Puzzle-solving:** Puzzle-solving including Japanese nonograms can employ Depth-first search as a method for systematically exploring possible solutions.
- **Pathfinding in robotics:** DFS can be employed for pathfinding in robotics, especially in scenarios where simplicity, memory efficiency, and adaptability are important considerations

[3]. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023.

<https://www.geeksforgeeks.org/search-algorithms-in-ai/>

b. 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.
- It is implemented using a queue.
- As BFS traverses the tree "shallowest node first", it would always pick the shallower branch until it reaches the solution (or it runs out of nodes, and goes to the next branch).
- **Path: S -> D -> G**



c. Uniform Cost Search

- Unlike breadth-first and depth-first algorithms, uniform cost search considers the expense.
- When there are multiple paths to achieving the desired objective, the optimal solution of uniform cost algorithms is the one with the lowest cost.
- So uniform cost search will check the expense to go to the next node. It will choose the path with the least cost if there are multiple paths.
- The goal is to find a path where the cumulative sum of costs is the least.

[4]. Search Algorithms in Artificial Intelligence, Eshika Shah, Scaler, 2023.

<https://www.scaler.com/topics/artificial-intelligence-tutorial/search-algorithms-in-artificial-intelligence/>

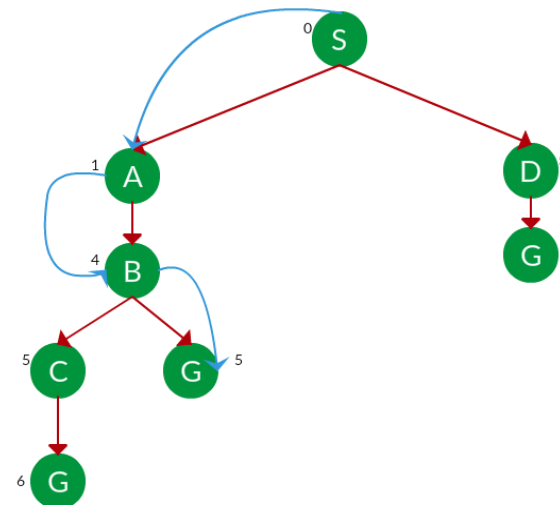
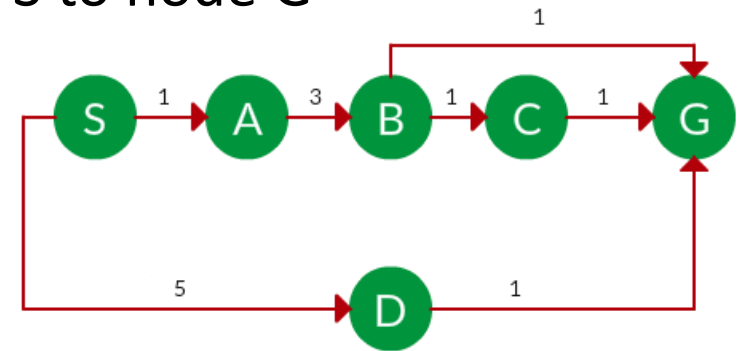
Example: Uniform Cost Search

Question. Which solution would UCS find to move from node S to node G

- The cost of each node is the cumulative cost of reaching that node from the root.
- Based on the UCS strategy, the path with the least cumulative cost is chosen.

Note

- Due to the many options in the fringe, the algorithm explores most of them so long as their cost is low, and discards them when a lower-cost path is found.



Advantages of Uniform Cost Search

- UCS is complete only if states are finite and there should be no loop with zero weight.
- UCS is optimal only if there is no negative cost.

Disadvantages of Uniform Cost Search

- Explores options in every "direction".
- No information on goal location.

[3]. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023.
<https://www.geeksforgeeks.org/search-algorithms-in-ai/>

Informed search algorithm in AI

- An informed search algorithm, also known as a heuristic search, is a type of search algorithm used in artificial intelligence that leverages additional information about the state space of a problem to efficiently find a solution.
- This additional information, typically in the form of a heuristic function, helps estimate the cost or distance from a given node in the search space to the goal node.
- The use of heuristics distinguishes informed search algorithms from uninformed search algorithms, which do not use any domain-specific knowledge.

Key Characteristics of Informed Search Algorithms

- **Heuristic Function:** Informed search algorithms use a heuristic function $h(n)$ that provides an estimate of the minimal cost from node n to the goal.
 - This function helps the algorithm to prioritize which nodes to explore first based on their potential to lead to an optimal solution.
- **Efficiency:** By focusing on more promising paths, informed search algorithms often find solutions more quickly than uninformed methods, especially in large or complex search spaces.

...Key Characteristics of Informed Search Algorithms

- **Optimality and Completeness:**

- Depending on the heuristic used, informed search algorithms can be both optimal and complete.
- An algorithm is complete if it is guaranteed to find a solution if one exists, and it is optimal if it always finds the best solution.
- For instance, the A^* search algorithm is both complete and optimal when the heuristic function is admissible (i.e., it never overestimates the true cost).

Informed Search Algorithms

- Informed search algorithms use heuristics to make the search process more efficient by focusing on more promising paths. These algorithms include:
 - Beam Search
 - *Greedy Best First Search Algorithm*
 - *A* Search Algorithms*
 - AO* Search algorithm
 - *Iterative Deepening A* algorithm (IDA*)*
 - Memory-bounded search

[5]. Informed Search Algorithms in Artificial Intelligence, GeeksforGeeks, GeeksforGeeks, 2024.
<https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/>

a) Greedy Best-First Search in AI

- Greedy Best-First Search is a heuristic-driven algorithm that prioritizes the exploration of nodes based on their estimated cost to the goal.
- The algorithm selects the node that appears most promising according to a heuristic function $h(n)$, which estimates the cost from node n to the goal.

How It Works?

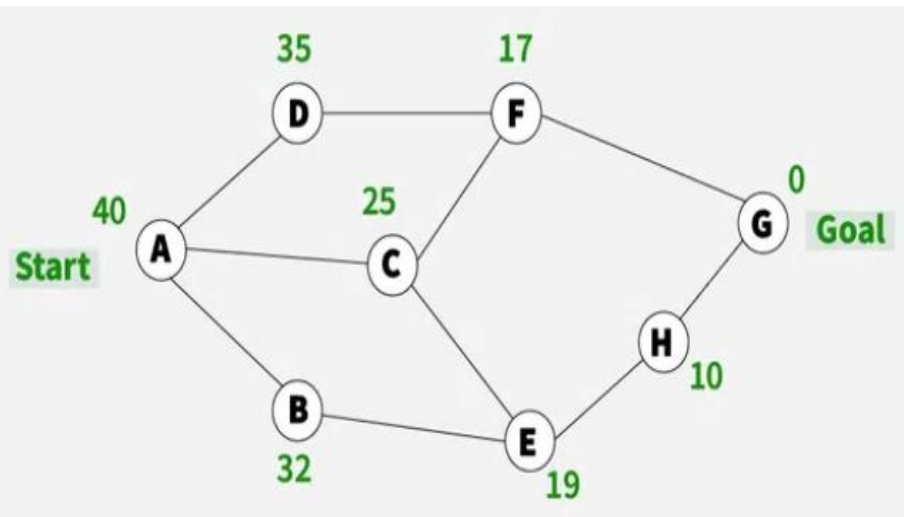
- The algorithm starts at the initial node and evaluates its neighbors.
- It chooses the neighbor with the lowest heuristic value $h(n)$ and continues the process.
- Greedy Best-First Search does not guarantee finding the optimal path, as it can get trapped in local optima by always choosing the path that looks best at the moment.

[6]. Greedy Best First Search Algorithm, GeeksforGeeks, GeeksforGeeks, 2024.

<https://www.geeksforgeeks.org/greedy-best-first-search-algorithm/>

Example : Greedy Best-First Search in AI

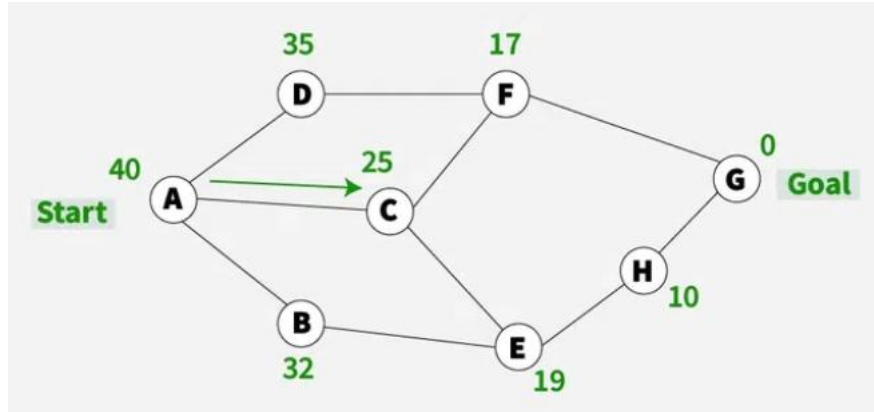
- The values in red color represent the heuristic value of reaching the goal node G from current node



- We are starting from A , so from A there are direct path to node B(with heuristics value of 32) , from A to C (with heuristics value of 25) and from A to D(with heuristics value of 35)*
- So as per best first search algorithm choose the path with lowest heuristics value , currently C has lowest value among above node . So we will go from A to C.*

[6]. Greedy Best First Search Algorithm, GeeksforGeeks, GeeksforGeeks, 2024.

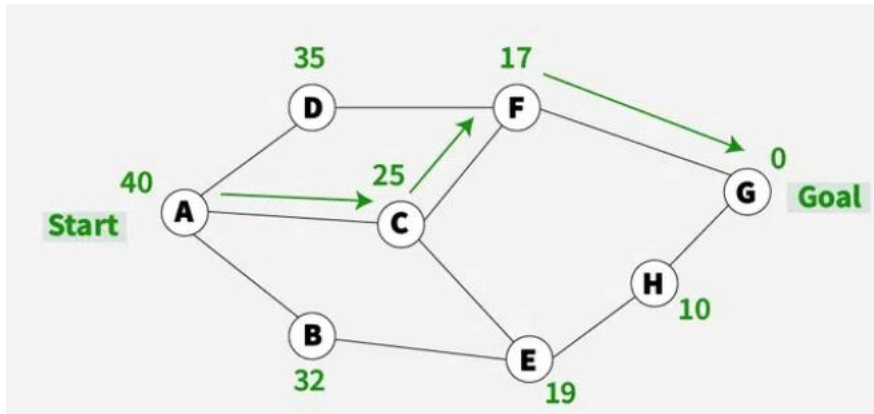
<https://www.geeksforgeeks.org/greedy-best-first-search-algorithm/>



3) Now from C we have direct paths as C to F (with heuristics value of 17) and C to E (with heuristics value of 19) , so we will go from C to F

4) Now from F we have direct path to go to the goal node G (with heuristics value of 0) , so we will go from F to G.

5) So now the goal node G has been reached and the path we will follow is **A->C->F->G** .



b) A Search*

- Is one of the most widely used informed search algorithms. It combines the strengths of both uniform-cost search and Greedy Best-First Search by using a composite cost function

$f(n) = g(n) + h(n)$, where:

- $g(n)$ is the exact cost from the start node to node n .
- $h(n)$ is the heuristic estimate of the cost from n to the goal.

How It Works?

- A* evaluates nodes based on the sum of the path cost and the heuristic estimate.
- The algorithm ensures that the path found is both the shortest and most cost-effective, provided that the heuristic function is admissible (it never overestimates the actual cost).

[5]. Informed Search Algorithms in Artificial Intelligence, GeeksforGeeks, GeeksforGeeks, 2024.
<https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/>

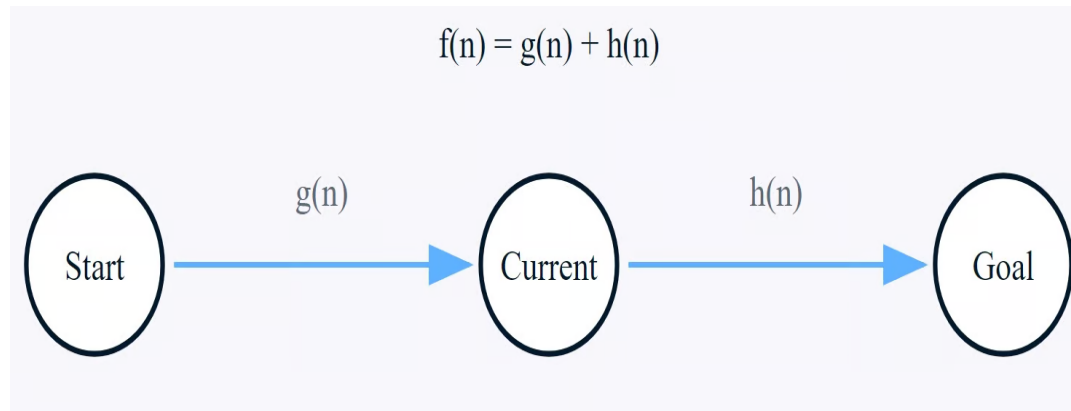
How Does the A* Algorithm Work?

- Combines the best aspects of two other algorithms:
 - **Dijkstra's Algorithm:** This algorithm finds the shortest path to all nodes from a single source node.
 - **Greedy Best-First Search:** This algorithm explores the node that appears to be closest to the goal, based on a heuristic function.
- To find the shortest route between two cities on a map. While Dijkstra's algorithm would explore in all directions and Best-First Search might head straight toward the destination (potentially missing shortcuts).
 - A* considers both: *The distance already traveled from the start and, a smart estimate of the remaining distance to the goal*
 - This combination helps A* make informed decisions about which path to explore next, making it both efficient and accurate.

Key Concepts in A* Search

- The A* algorithm's efficiency comes from its smart evaluation of paths using three key components: $g(n)$, $h(n)$, and $f(n)$.
- These components work together to guide the search process toward the most promising paths.

A algorithm Cost Function*



Understanding the cost functions

i. Path cost $g(n)$

- The path cost function $g(n)$ represents the exact, known distance from the initial starting node to the current position in our search. Unlike estimated values, this cost is precise and calculated by adding up all the individual edge weights that have been traversed along our chosen path.
- Mathematically, for a path through nodes n_0 (start node) to n_k (current node), we can express $g(n)$ as:

$$g(n_k) = \sum_{i=0}^{k-1} w(n_i, n_{i+1})$$

Where:

- $w(n_i, n_{i+1})$ represents the weight of the edge connecting node n_i to node n_{i+1} .
- As we move through the graph, this value accumulates, giving us a clear measure of the actual resources (whether that's distance, time, or any other metric) we've expended to reach our current position.

.... Understanding the cost functions

ii. Heuristic function $h(n)$

- The heuristic function $h(n)$ provides an estimated cost from the current node to the goal node, acting as the algorithm's "informed guess" about the remaining path.
- In grid-based or map-like problems, common heuristic functions include the ***Manhattan distance*** and ***Euclidean distance***.
- For coordinates (x_1, y_1) of the current node and (x_2, y_2) of the goal node, these distances are calculated as follow:-

Manhattan distance

$$h(n) = |x_1 - x_2| + |y_1 - y_2|$$

Euclidean distance

$$h(n) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

....Understanding the cost functions

Total estimated cost $f(n)$

- The total estimated cost $f(n)$ is the cornerstone of A* algorithm's decision-making process, combining both the actual path cost and the heuristic estimate to evaluate each node's potential. For any node n , this cost is calculated as:

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

Where:

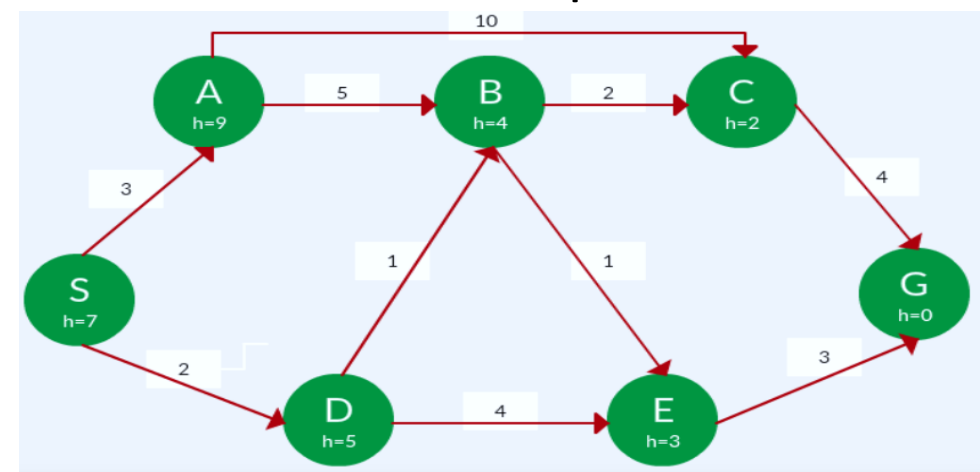
- $g(n)$ represents the actual cost from the start to the current node
- $h(n)$ represents the estimated cost from the current node to the goal.
- The algorithm uses this combined value to strategically choose which node to explore next, always selecting the node with the lowest $f(n)$ value from the open list.

Types of Searching Algorithms

...cont'd

Example: A* Search

- Question.** Find the path to reach from S to G using A* search.



Path: S -> D -> B -> E -> G
Cost: 7

- Solution.** Starting from S, the algorithm computes $g(x) + h(x)$ for all nodes in the fringe at each step, choosing the node with the lowest sum. The entire work is shown in the table below.
- Note that in the fourth set of iterations, we get two paths with equal summed cost $f(x)$, so we expand them both in the next set. The path with a lower cost on further expansion is the chosen path.

Path	$h(x)$	$g(x)$	$f(x)$
S	7	0	7
S -> A	9	3	12
S -> D	5	2	7
S -> D -> B	4	2 + 1 = 3	7
S -> D -> E	3	2 + 4 = 6	9
S -> D -> B -> C	2	3 + 2 = 5	7
S -> D -> B -> E	3	3 + 1 = 4	7
S -> D -> B -> C -> G	0	5 + 4 = 9	9
S -> D -> B -> E -> G	0	4 + 3 = 7	7

Applications of A* Search Algorithm

Navigation systems

- Route planning in GPS applications
- Traffic-aware navigation services
- Public transport route optimization
- Indoor navigation systems

Robotics and automation

- **Autonomous vehicle path planning:** Self-driving cars use A* to navigate roads, making decisions in real time about how to move from point A to point B while avoiding collisions and adhering to traffic rules.

Applications of A* Search Algorithm

- **Warehouse robot navigation:** In automated warehouses, robots rely on A* to navigate efficiently between storage racks to pick and place items, minimizing delays and avoiding collisions with other robots.
- **Drone flight path optimization:** A* helps drones plan efficient flight paths, whether for delivery, surveying, or recreational use, ensuring they avoid obstacles and follow optimal routes.
- **Manufacturing robot movement planning:** In factory settings, A* is used to ensure robots can move seamlessly between workstations, avoiding collisions with other machinery and maintaining productivity

.....Applications of A* Search Algorithm

- A* is also applied in optimizing network operations, where efficiency in resource utilization and routing is paramount.
 - Network packet routing:
 - Resource allocation in distributed systems:
 - Circuit board path design:
 - Network cable routing optimization

Applications of Informed Search Algorithms

- Informed search algorithms are extensively used in various applications, such as:
 - **Pathfinding in Navigation Systems:** Used to calculate the shortest route from a point A to point B on a map.
 - **Game Playing:** Helps in determining the best moves in games like chess or Go by evaluating the most promising moves first.
 - **Robotics:** Utilized in autonomous navigation where a robot must find the best path through an environment.
 - **Problem Solving in AI:** Applied to a range of problems from scheduling and planning to resource allocation and logistics.

[5]. Informed Search Algorithms in Artificial Intelligence, GeeksforGeeks, GeeksforGeeks, 2024.
<https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/>

Advantages of Informed Search Algorithms

- **Improved Efficiency:** Informed search algorithms can dramatically reduce the search space, leading to faster solutions.
- **Domain-Specific Optimization:** The use of heuristics allows these algorithms to be tailored to specific problem domains.
- **Balance Between Optimality and Efficiency:** Algorithms like A* offer a balance, providing optimal solutions with reasonable resource usage.

[5]. Informed Search Algorithms in Artificial Intelligence, GeeksforGeeks, GeeksforGeeks, 2024.
<https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/>

Challenges and Considerations in Informed Search Algorithms

- **Heuristic Design:** The effectiveness of informed search algorithms heavily depends on the quality of the heuristic function. A poorly designed heuristic can lead to suboptimal performance.
- **Memory Constraints:** While informed search algorithms like A^* are powerful, they can require significant memory resources, especially for large search spaces.
- **Trade-offs:** Algorithms like beam search introduce a trade-off between memory usage and solution optimality.

Summary

- Search algorithms are essential in AI, facilitating problem-solving and decision-making in a wide range of applications.
- Uninformed search algorithms, such as BFS and DFS, provide a systematic exploration of the problem space without domain-specific knowledge.
- Informed search algorithms, like A* Search and Greedy Best-First Search, leverage domain-specific heuristics to guide intelligent exploration.
- Each algorithm has its strengths and limitations, making it important to choose the right tool for the problem at hand.
- Real-world applications of search algorithms include GPS navigation, game-playing agents, web search engines, and machine learning.
- Challenges faced by search algorithms include scalability, uncertainty, and memory constraints.
- The evolving landscape of search algorithms involves advancements like parallel search, hybrid approaches, and the potential impact of quantum computing.

References

1. What is AI (Artificial Intelligence)? Definition, Types, Examples & Use Cases, Lev Craig, Nicole Laskowski, Linda Tucci, TechTarget, 2024. <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>.
2. Artificial Intelligence (AI) Algorithms, GeeksforGeeks, GeeksforGeeks, 2024. <https://www.geeksforgeeks.org/ai-algorithms/>
3. Search Algorithms in AI, GeeksforGeeks, GeeksforGeeks, 2023. <https://www.geeksforgeeks.org/search-algorithms-in-ai/>
4. Search Algorithms in Artificial Intelligence, Eshika Shah, Scaler, 2023. <https://www.scaler.com/topics/artificial-intelligence-tutorial/search-algorithms-in-artificial-intelligence/>
5. Informed Search Algorithms in Artificial Intelligence, GeeksforGeeks, GeeksforGeeks, 2024. <https://www.geeksforgeeks.org/informed-search-algorithms-in-artificial-intelligence/>
6. Greedy Best First Search Algorithm, GeeksforGeeks, GeeksforGeeks, 2024. <https://www.geeksforgeeks.org/greedy-best-first-search-algorithm/>

Thank You!

For your attention