Artificial Intelligence – Course Overview

Welcome to the foundational course on Artificial Intelligence (AI). This presentation outlines the core topics, learning objectives, and compelling applications of AI that you will master.

Prepare to explore how machines think, learn, and make decisions.

What Is Artificial Intelligence?

AI is the science and engineering of making intelligent machines, especially intelligent computer programs. The central goal is to build systems that can **reason**, **learn**, and **act** autonomously.

Thinking Humanly

Cognitive modeling approach.

Acting Humanly

The Turing Test approach.

Thinking Rationally

Laws of thought approach.

Acting Rationally

Rational agent approach (focus).

The State of the Art in AI

Modern AI capabilities drive innovations across multiple sectors. These sophisticated applications are the culmination of decades of research and exponential increases in computing power.

Natural Language Processing (NLP)

Enabling machines to understand and generate human language, powering chatbots and automated translation.

Computer Vision

Allowing systems to interpret and understand visual information from the world, essential for autonomous vehicles and image recognition.

Robotics

The physical embodiment of AI, used in complex manufacturing, logistics, and exploration tasks.

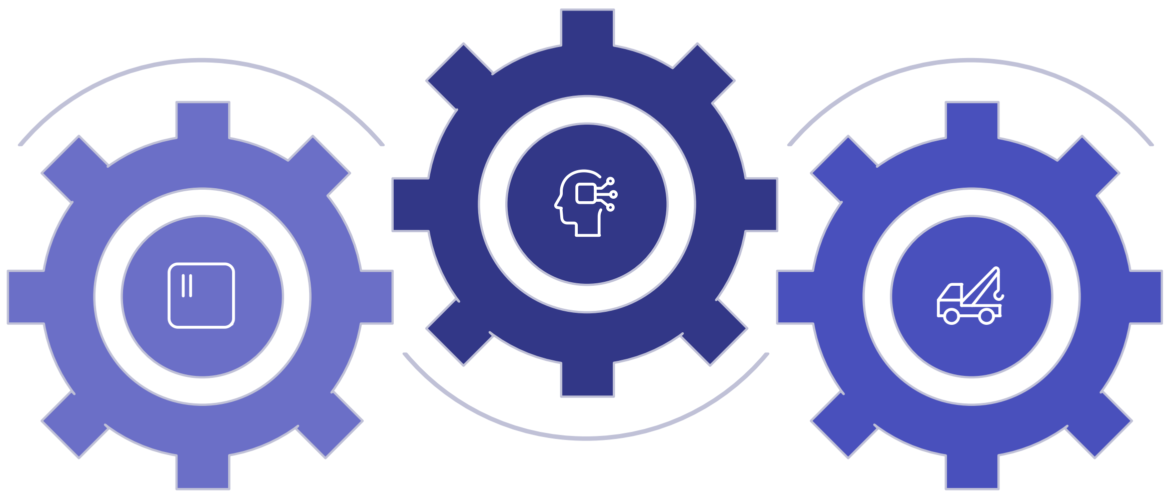
We will examine the underlying algorithms that make these breakthroughs possible.

Intelligent Agents: The Core Concept

An intelligent agent is anything that perceives its **environment** through sensors and acts upon that environment through actuators. The goal is to design **rational agents** that do the right thing—the action that maximizes expected performance.

Agent Program

Processes percepts and decides on an action.



Percepts (Sensors)

The agent perceives its environment through sensors.

Action (Actuators)

The agent acts upon the environment using actuators.

> Rationality is about **expected success**, given the available information.

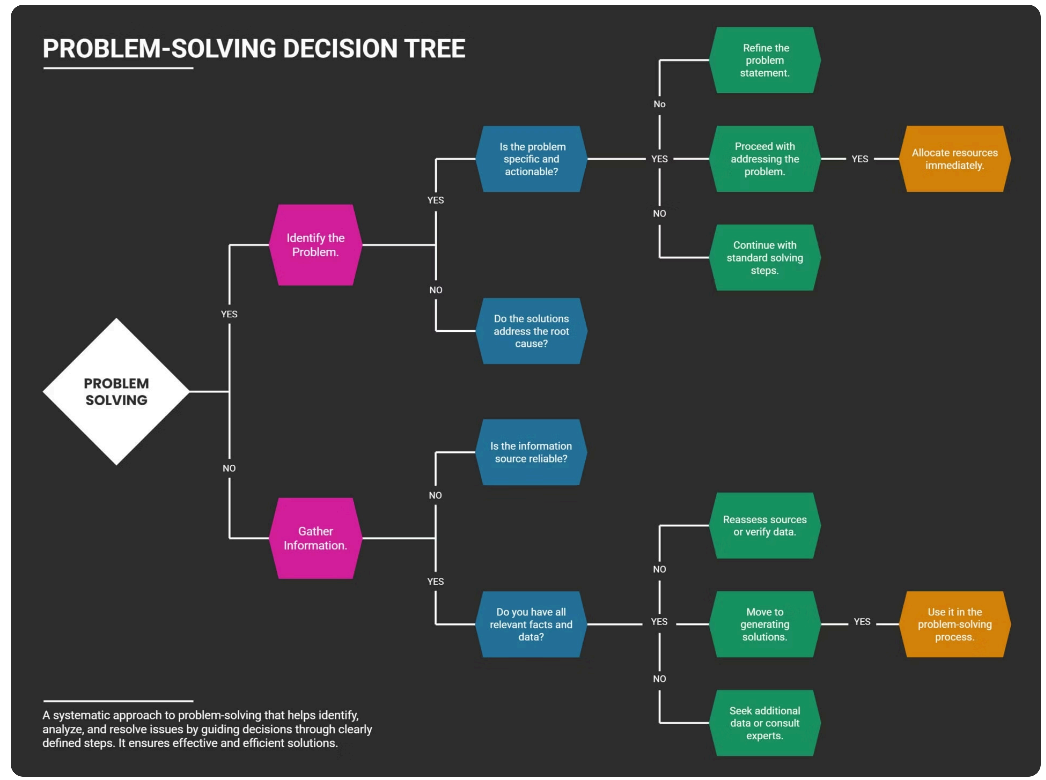
Problem-Solving by Searching

Many AI challenges, from route planning to logic puzzles, can be formulated as **search problems**. An agent starts in an initial state and searches through a state space to find a sequence of actions that leads to a goal state.

Key Components:

- Initial State
- Successor Function (actions)
- Goal Test
- Path Cost

Example problems include the **8-puzzle**, **route-finding**, and the **Travelling Salesperson Problem (TSP)**.



Uninformed Search Strategies

Uninformed (or blind) search algorithms explore the state space without any domain-specific knowledge about the location of the goal. We analyze them based on **completeness**, **optimality**, and **time/space complexity**.

Breadth-First Search (BFS)

Explores layer by layer. Complete and optimal, but can be memory intensive.

Depth-First Search (DFS)

Explores deeply before backtracking. Faster, but neither complete nor optimal for infinite spaces.

Depth-Limited Search (DLS)

DFS with a predefined depth cutoff. Avoids infinite paths but may miss a goal beyond the limit.

Iterative Deepening DFS (IDS)

Combines the benefits of BFS and DFS. Optimal and complete, with low memory use.

Informed Search Strategies

Informed search uses **heuristic functions**—an estimate of the cost from the current state to the goal—to guide the search. This domain knowledge dramatically improves efficiency.

A* Search

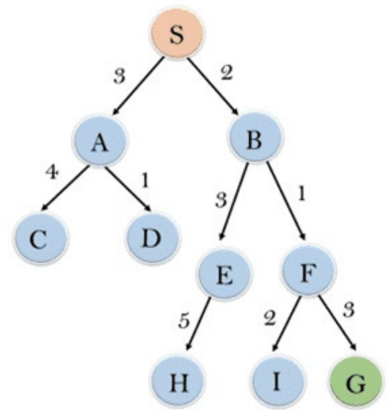
Optimal and complete. Uses **f(n) = g(n) + h(n)** where g is the cost so far and h is the heuristic estimate.

Best-First Search

Greedy approach that expands the node closest to the goal, according to h(n).

Beam Search

A refinement of BFS that limits the number of nodes at each depth to a fixed size (the "beam width").

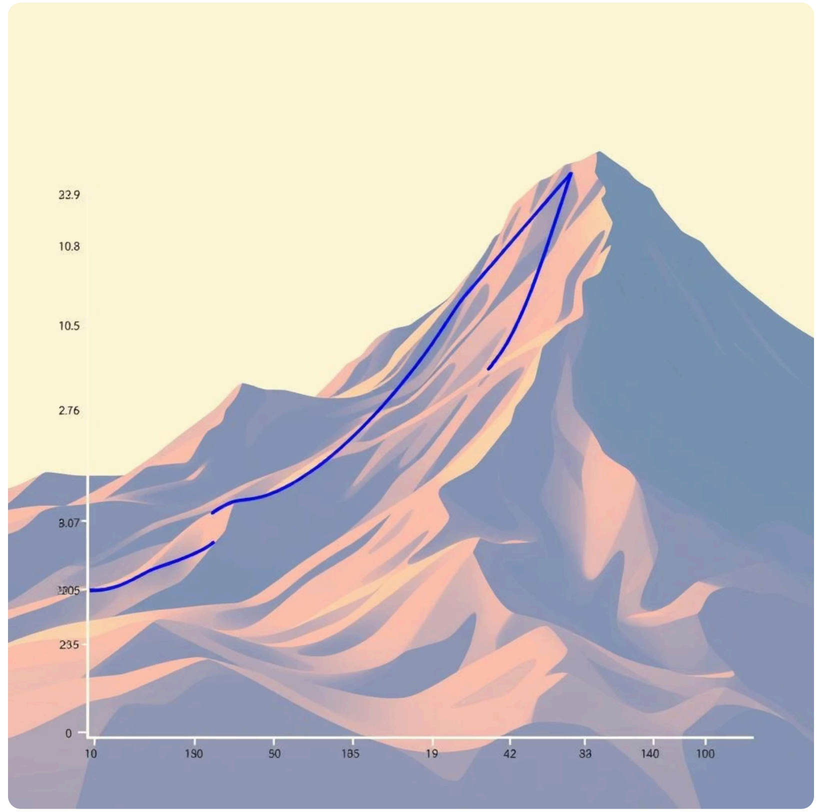


Node	H(n)
A	10
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

Local Search and Optimization

For problems where the path is irrelevant—only the final state matters—**local search** algorithms are highly effective. They focus on finding an optimal solution in a vast state space, often used for optimization.

- **Hill-Climbing Search**: Iteratively moves towards states with better values, stopping at a peak (local optimum).
- **Simulated Annealing**: Uses randomness to escape local optima by occasionally accepting worse moves, particularly at the start.
- **Genetic Algorithms**: Inspired by biological evolution, maintains a population of solutions and combines them to generate new, better solutions.



Game Playing & Adversarial Search

Games present a unique challenge: the agent must account for the actions of an opponent. Adversarial search involves planning multiple steps ahead to anticipate the best and worst possible outcomes.

Tic-Tac-Toe(MiniMaxAlgorithm)

Determines the optimal move for the player assuming the opponent plays optimally to minimize the agent's score.

Chess(Alpha-Beta Pruning+MinMax)

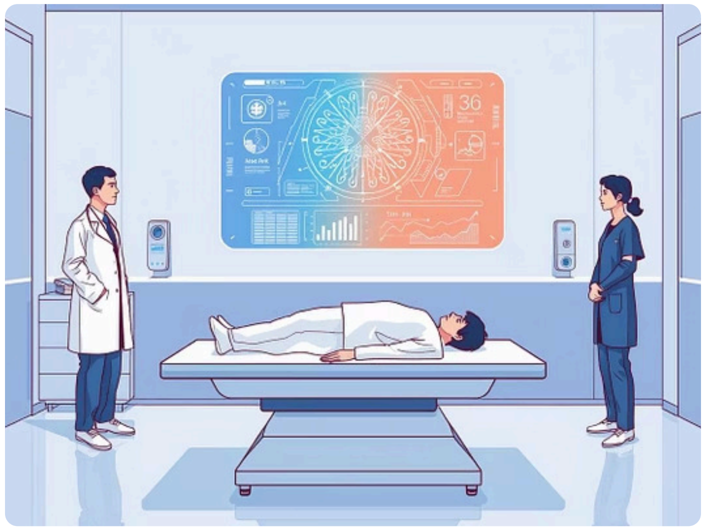
A technique to eliminate branches in the game tree that do not affect the final decision, significantly speeding up Minimax.

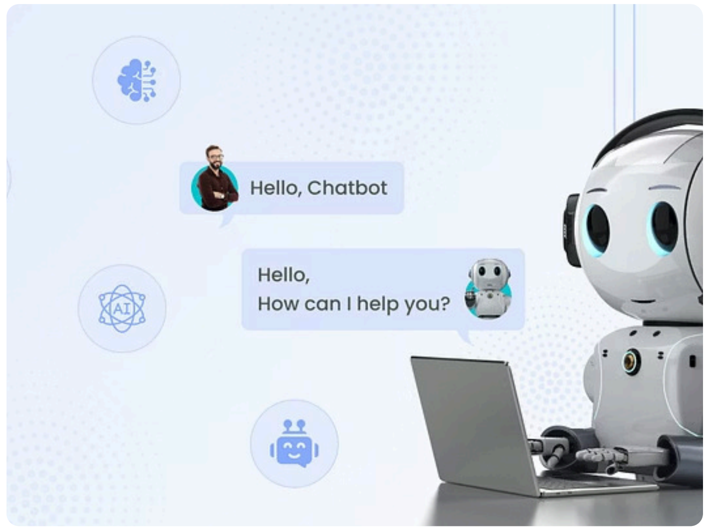
Nim Game

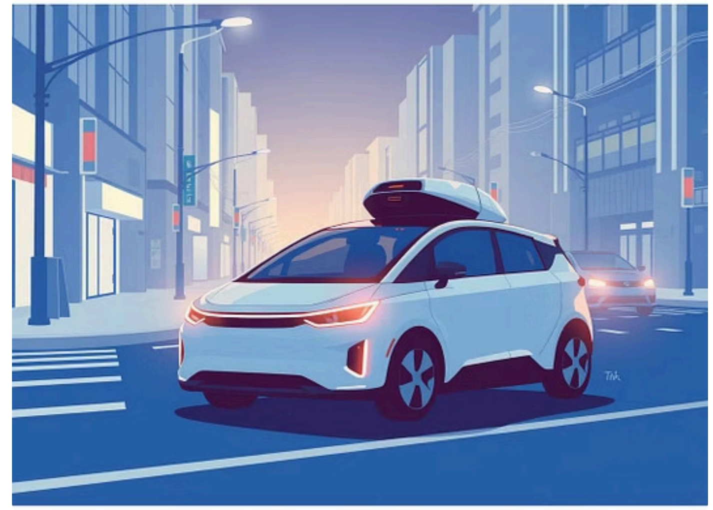
Extending search to games where information is hidden, using probability and game theory-minimax.

AI Applications and Key Takeaways

The concepts learned—agents, search, and decision-making—form the basis for transformative real-world AI systems.







1. ### AI is Everywhere

   From medical diagnostics in **healthcare** to algorithmic trading in **finance**, AI is reshaping industries.

2. ### Core Concepts are Transferable

   Search algorithms and agent modeling are fundamental tools for solving a wide range of complex computational problems.

3. ### Next Steps

   Building on this foundation, you are now prepared to dive into advanced topics like Machine Learning and Knowledge Representation.