Unit-1.0: Introduction to Artificial Intelligence

1. Overview of Artificial Intelligence (AI)

What is Artificial Intelligence?

Artificial Intelligence (AI) is a branch of computer science that focuses on creating machines or programs that can **think, learn, reason, and make decisions like humans**.

👉 In simple words, **AI enables computers to act intelligently**.

Formal Definitions

- **John McCarthy (Father of AI):**
  "Artificial Intelligence is the science and engineering of making intelligent machines."
- AI aims to simulate **human intelligence** using machines, especially computer systems.

Key Capabilities of AI

AI systems are designed to:

- Learn from experience (Machine Learning)
- Reason and solve problems
- Understand natural language
- Recognize images and speech
- Make decisions

Examples of AI Applications

- Voice assistants (Siri, Alexa, Google Assistant)
- Recommendation systems (Netflix, YouTube)
- Autonomous vehicles
- Chatbots
- Medical diagnosis systems

---

2. Turing Test

What is the Turing Test?

The **Turing Test** was proposed by **Alan Turing in 1950** to determine whether a machine can exhibit **intelligent behavior equivalent to a human**.

Basic Idea

If a machine can **communicate with a human without being detected as a machine**, then it can be considered intelligent.

Participants in Turing Test

1. **Human Interrogator**
2. **Human Respondent**
3. **Machine**

All communication happens through **text only**, to avoid voice or appearance bias.

Working of Turing Test

- The interrogator asks questions to both the human and the machine.
- If the interrogator **cannot distinguish** between them after a reasonable time, the machine **passes the Turing Test**.

Diagram (Conceptual)

Human ⇄ Text ⇄ Interrogator
Machine ⇄ Text ⇄ Interrogator

Significance of Turing Test

- First practical method to test machine intelligence
- Focuses on **behavior**, not internal structure

Limitations of Turing Test

- Tests only **conversation ability**
- Does not measure true understanding or consciousness
- Machines can trick humans using predefined responses

---

3. Intelligent Agents

What is an Intelligent Agent?

An **Intelligent Agent** is an entity that:

- **Perceives** its environment using sensors
- **Acts** upon the environment using actuators
- Tries to achieve goals **rationally**

Agent Definition

👉 *An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.*

Examples of Agents

Agent Type	Sensors	Actuators
Human	Eyes, ears	Hands, legs
Robot	Cameras, sensors	Motors, wheels
Software Agent	Keyboard, mouse	Display, network

---

4. Components of an Intelligent Agent

1. Sensors

- Used to **perceive** the environment
- Example: Camera, microphone, temperature sensor

2. Actuators

- Used to **act** on the environment
- Example: Motors, wheels, speakers

3. Environment

- The external world where the agent operates

---

5. Types of Intelligent Agents

1. Simple Reflex Agents

- Act based only on **current perception**
- Use condition–action rules
- Example: Automatic door

2. Model-Based Reflex Agents

- Maintain an **internal state**
- Useful in partially observable environments

3. Goal-Based Agents

- Take actions to **achieve specific goals**
- Example: Path-finding robot

4. Utility-Based Agents

- Choose actions that **maximize utility**
- Measure happiness or performance

5. Learning Agents

- Improve performance over time
- Components:
    - Learning element
    - Performance element
    - Critic
    - Problem generator

---

6. Rationality of an Agent

What is a Rational Agent?

A rational agent is one that **does the right thing**, i.e., selects an action that **maximizes its performance measure**, based on:

- Percept sequence
- Built-in knowledge
- Available actions

Performance Measure

- A criterion to evaluate the success of an agent
- Example: Accuracy, speed, efficiency

---

7. Characteristics of Intelligent Agents

- Autonomy
- Reactivity
- Proactiveness
- Learning ability
- Rational decision making

---

8. Summary (For Exam Writing)

- AI focuses on making machines intelligent.
- Turing Test evaluates intelligence based on behavior.
- Intelligent agents perceive, reason, and act in an environment.
- Agents can be simple, goal-based, utility-based, or learning agents.
- Rational agents maximize performance.

Unit-2.0: Problem Solving by Searching

1. Problem Solving in Artificial Intelligence

What is Problem Solving?

Problem solving in AI means finding a **sequence of actions** that leads from an **initial state** to a **goal state**.

Basic Components of a Problem

1. **Initial State –** Starting point
2. **Goal State –** Desired result
3. **State Space –** All possible states
4. **Actions/Operators –** Moves between states
5. **Path Cost –** Cost of actions
6. **Solution –** Sequence of actions

---

2. Solving Problems by Searching

Searching is the process of **exploring the state space** to find a path from the initial state to the goal state.

Types of Search Strategies

1. **Uninformed (Blind) Search**
2. **Informed (Heuristic) Search**

PART A: Uninformed Search (Blind Search)

Uninformed search has **no additional information** about how close a state is to the goal.

---

3. Depth First Search (DFS)

Definition

DFS explores **as deep as possible** along a branch before backtracking.

Working

- Uses **stack (LIFO)** or recursion
- Expands the deepest node first

Algorithm (Steps)

1. Start from root
2. Go deep into one branch
3. Backtrack when no child exists
4. Continue until goal is found

Properties

Property	**DFS**
Complete	No (may get stuck in infinite loop)
Optimal	No
Time Complexity	O(b^m)
Space Complexity	O(bm)

(b = branching factor, m = max depth)

Advantages

- Low memory usage
- Simple implementation

Disadvantages

- May not find shortest path
- Can loop infinitely

---

4. Breadth First Search (BFS)

Definition

BFS explores **level by level**, expanding the shallowest node first.

Working

- Uses **queue (FIFO)**
- Finds shortest path (if cost is uniform)

Properties

Property	BFS
Complete	Yes
Optimal	Yes (unit cost)
Time Complexity	O(b^d)
Space Complexity	O(b^d)

(d = depth of shallowest solution)

Advantages

- Guaranteed solution
- Finds shortest path

Disadvantages

- High memory usage

---

5. Depth First Iterative Deepening (DFID / IDDFS)

Definition

DFID combines **DFS and BFS**:

- DFS's low memory
- BFS's completeness

Working

- Perform DFS with depth limit 0
- Increase depth limit gradually
- Continue until goal is found

Properties

Property	DFID
Complete	Yes
Optimal	Yes (unit cost)
Space Complexity	O(bd)
Time Complexity	O(b^d)

Advantages

- Less memory than BFS
- Guaranteed solution

PART B: Heuristic (Informed) Search

Heuristic search uses **additional information** (heuristic function) to guide the search.

Heuristic Function (h(n))

- Estimates cost from node n to goal
- Must be **fast and accurate**

---

6. Generate and Test

Definition

Generate possible solutions and test whether each one satisfies the goal.

Steps

1. Generate candidate solution

2. Test it
3. If not goal, generate next

Example

- Puzzle solving
- Password cracking

Advantages

- Simple approach

Disadvantages

- Inefficient for large problems

---

7. Best First Search

Definition

Expands the node that appears **most promising** based on heuristic value.

Evaluation Function

f(n) = h(n)

Properties

- Uses **priority queue**
- Faster than blind search

Disadvantages

- Not optimal
- Can get stuck in local minima

---

8. Beam Search

Definition

Beam search is a **memory-limited** version of Best First Search.

Working

- Only keeps best k nodes at each level
- Others are discarded

Advantages

- Reduced memory
- Faster execution

Disadvantages

- May lose optimal solution
- Incomplete

---

9. Hill Climbing

Definition

Hill climbing is a **local search algorithm** that moves toward a better state.

Working

- Evaluate neighbors
- Move to best neighbor
- Stop when no improvement

Types

1. Simple hill climbing
2. Steepest ascent hill climbing
3. Stochastic hill climbing

Problems

- Local maxima

- Plateau
- Ridge

---

10. A Search Algorithm*

Definition

A* is the **most important and optimal heuristic search algorithm**.

Evaluation Function

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

Where:

- $g(n)$ = cost from start to n
- $h(n)$ = estimated cost to goal

Properties

Property	A*
Complete	Yes
Optimal	Yes (if h(n) is admissible)
Time Complexity	Exponential
Space Complexity	High

Admissible Heuristic

A heuristic that **never overestimates** the true cost.

Advantages

- Finds optimal path
- Efficient

Disadvantages

- High memory usage

11. Comparison Table (Important for Exams)

Algorithm	Complete	Optimal	Memory
DFS	No	No	Low
BFS	Yes	Yes	High
DFID	Yes	Yes	Medium
Best First	No	No	High
Beam	No	No	Low
Hill Climbing	No	No	Very Low
A*	Yes	Yes	Very High

Unit-3.0: Problem Reduction, Stochastic Search & Game Playing

---

1. Problem Reduction Search

What is Problem Reduction?

Problem reduction means **breaking a complex problem into smaller sub-problems**, solving them independently, and then combining their solutions.

👉 Used when a problem can be **decomposed** into sub-goals.

---

2. AND/OR Graphs

Definition

An **AND/OR graph** represents a problem where:

- **OR nodes**: Any one child can solve the problem
- **AND nodes**: All children must be solved

Difference Between AND & OR Nodes

Node Type	Meaning
OR Node	Choose one successful path
AND Node	All child paths must succeed

Example

- Planning problems
- Theorem proving
- Problem decomposition

Solution Graph

A solution exists if:

- At least one child of OR node is solved
- All children of AND node are solved

---

3. AO Algorithm*

Definition

AO* is a heuristic search algorithm designed for **AND/OR graphs**.

Key Idea

- Extends A* to problem reduction search
- Uses heuristic estimates
- Finds **optimal solution graph**

Working Steps

1. Start from root node
2. Expand the most promising node
3. Update costs using heuristic values

4. Mark nodes as solved
5. Repeat until goal solved

Evaluation Function

Cost(OR node) = min(cost of children)

Cost(AND node) = sum(cost of children)

Properties

- Optimal if heuristic is admissible
- Used in expert systems and planning

---

4. Constraint Satisfaction Problems (CSP)

Definition

A **CSP** is a problem defined by:

- A set of **variables**
- A domain of values for each variable
- A set of **constraints**

Formal Representation

CSP = (X, D, C)

Where:

- X = Variables
- D = Domains
- C = Constraints

Examples

- Map coloring
- Sudoku

- N-Queens problem
- Timetabling

Solution

Assign values to variables such that **all constraints are satisfied**.

---

5. Means-Ends Analysis

Definition

Means-ends analysis reduces the difference between:

- **Current state**
- **Goal state**

Steps

1. Compare current and goal states
2. Identify differences
3. Select an operator to reduce difference
4. Apply operator
5. Repeat until goal is reached

Example

- Planning a trip
- Problem solving in AI planning systems

Advantages

- Efficient and goal-directed
- Reduces search space

---

6. Stochastic Search Methods

Stochastic methods use **randomness** to avoid local optima.

---

7. Simulated Annealing

Definition

Simulated annealing is a probabilistic technique inspired by **metallurgy annealing**.

Key Idea

- Accepts worse solutions with some probability
- Probability decreases over time (temperature)

Algorithm

1. Start with initial solution
2. Generate neighboring solution
3. Accept if better
4. If worse, accept with probability:

P = e^(−ΔE / T)

5. Gradually reduce temperature

Advantages

- Avoids local maxima
- Works well for large spaces

Disadvantages

- Slow convergence
- Parameter tuning required

---

8. Particle Swarm Optimization (PSO)

Definition

PSO is inspired by **social behavior of birds and fish**.

Key Components

- Particles (solutions)
- Velocity and position
- Global best (gBest)
- Personal best (pBest)

Update Equations

$v = wv + c1r1(pBest - x) + c2r2(gBest - x)$

$x = x + v$

Applications

- Optimization problems
- Neural network training
- Scheduling

Advantages

- Simple implementation
- Fast convergence

---

9. Game Playing in AI

Game playing involves **competitive environments**.

Types of Games

- Two-player
- Zero-sum
- Perfect information (Chess)

---

10. Minimax Algorithm

Definition

Minimax is a decision-making algorithm used in **two-player games**.

Players

- **MAX**: Tries to maximize score
- **MIN**: Tries to minimize score

Working

1. Generate game tree
2. Assign values to terminal nodes
3. Back-propagate values
4. Choose best move

Time Complexity

O(b^d)

Limitations

- Large search space
- High computation cost

---

11. Alpha-Beta Pruning

Definition

Alpha-Beta pruning is an optimization of Minimax.

Key Idea

- Eliminates branches that cannot influence decision
- Same result as Minimax, faster execution

Parameters

- α (alpha): Best value for MAX
- β (beta): Best value for MIN

Condition for Pruning

If α ≥ β → prune branch

Advantages

- Reduces nodes evaluated
- Allows deeper search

---

12. Comparison Table

Algorithm	Type	Optimal	Uses Heuristic
AO*	Problem reduction	Yes	Yes
CSP	Constraint based	Yes	Sometimes
Simulated Annealing	Stochastic	Approx	No
PSO	Optimization	Approx	No
Minimax	Game playing	Yes	No
Alpha-Beta	Game playing	Yes	No

---

13. Summary (Exam Ready)

- Problem reduction uses AND/OR graphs.
- AO* finds optimal solution graph.
- CSP focuses on constraint satisfaction.
- Means-ends analysis reduces differences.
- Stochastic methods avoid local optima.

Unit-4.0: Knowledge and Reasoning

---

1. Knowledge and Knowledge Base

What is Knowledge?

Knowledge is information about the world that an AI system uses to **reason, make decisions, and solve problems**.

Knowledge Base (KB)

A **Knowledge Base** is a collection of:

- Facts
- Rules
- Relationships

👉 It is stored in a **formal language** so that machines can process it.

Components of Knowledge-Based System

1. Knowledge Base
2. Inference Engine
3. User Interface

---

2. Building a Knowledge Base

Steps:

1. Identify important facts
2. Choose representation method
3. Encode facts and rules
4. Apply inference mechanism

PART A: Logic for Knowledge Representation

---

3. Propositional Logic (PL)

Definition

Propositional logic represents knowledge using **propositions** that are either **true or false**.

Components

- **Propositions**: P, Q, R
- **Logical Connectives**:
  - AND (∧)
  - OR (∨)
  - NOT (¬)
  - IMPLIES (→)
  - IFF (↔)

Example

P: It is raining

Q: Road is wet

P → Q

Advantages

- Simple
- Easy inference

Limitations

- Cannot represent relationships or quantifiers

---

4. First Order Logic (FOL)

Definition

FOL extends propositional logic by introducing:

- Objects
- Predicates
- Quantifiers

Components

1. **Constants –** Ram, Delhi
2. **Variables –** x, y
3. **Predicates –** Student(x)
4. **Functions –** Father(x)
5. **Quantifiers**:
    - Universal (∀)
    - Existential (∃)

Example

$\forall x\ (Human(x) \rightarrow Mortal(x))$

Human(Ram)

$\therefore$ Mortal(Ram)

Advantages

- More expressive
- Represents relationships

Limitations

- Complex reasoning
- Computationally expensive

PART B: Inference and Reasoning

---

5. Inference in First Order Logic

Inference

Inference is the process of deriving **new knowledge** from existing knowledge.

Inference Rules

- Modus Ponens
- Modus Tollens
- Universal Instantiation
- Existential Instantiation

Example

∀x (Student(x) → Intelligent(x))

Student(Ajay)

∴ Intelligent(Ajay)

---

6. Resolution – Refutation Proofs

Resolution

Resolution is a **complete inference rule** used in automated reasoning.

Steps in Resolution

1. Convert statements to **Conjunctive Normal Form (CNF)**
2. Negate the query
3. Apply resolution
4. Derive empty clause (⊥)

Example

P ∨ Q

¬P

∴ Q

Refutation

If contradiction is derived, the original statement is **proved true**.

---

7. Theorem Proving in First Order Logic

Definition

Theorem proving means **proving the truth of a statement** using logical rules.

Approaches

- Forward chaining
- Backward chaining
- Resolution-based proving

Applications

- Expert systems
- Automated reasoning tools

PART C: Planning

---

8. Planning in AI

Definition

Planning is the process of finding a **sequence of actions** to reach a goal.

Components

- Initial state
- Goal state

- Actions
- Preconditions
- Effects

---

9. Partial Order Planning (POP)

Definition

Partial order planning is a **non-linear planning** approach.

Key Features

- Actions are partially ordered
- Only necessary orderings are specified
- More flexible than linear planning

Advantages

- Parallel execution possible
- Efficient planning

PART D: Uncertain Knowledge and Reasoning

---

10. Uncertain Knowledge

Why Uncertainty?

Real-world knowledge is often:

- Incomplete
- Noisy
- Uncertain

Examples:

- Medical diagnosis
- Weather prediction

---

11. Probabilities in AI

Probability Theory

Used to model uncertainty mathematically.

Basic Formula

P(A) = Number of favorable outcomes / Total outcomes

Conditional Probability

$P(A|B) = P(A \cap B) / P(B)$

Bayes' Theorem

$P(A|B) = [P(B|A) P(A)] / P(B)$

---

12. Bayesian Networks

Definition

A **Bayesian Network** is a probabilistic graphical model that represents:

- Random variables (nodes)
- Conditional dependencies (edges)

Features

- Directed Acyclic Graph (DAG)
- Each node has a conditional probability table (CPT)

Example Applications

- Medical diagnosis
- Fault detection

- Spam filtering

Advantages

- Handles uncertainty well
- Compact representation

---

13. Comparison Table (Important for Exams)

Logic Type	**Expressiveness**	**Complexity**
Propositional Logic	Low	Low
First Order Logic	High	High

---

14. Summary (Exam Ready)

- Knowledge base stores facts and rules.
- Propositional logic is simple but limited.
- First order logic is expressive.
- Resolution is key inference technique.
- Planning finds action sequences.
- Bayesian networks handle uncertainty.

5.Overview of Different Forms of Learning

Learning in Artificial Intelligence refers to the ability of a system to **improve its performance from experience or data**.

---

1. Types of Learning in AI

AI learning is mainly classified into:

1. **Supervised Learning**
2. **Unsupervised Learning**
3. **Semi-Supervised Learning**
4. **Reinforcement Learning** (overview mention)

---

2. Supervised Learning

Definition

Supervised learning uses **labeled data**, where the input and correct output are known.

Working

- Train model using labeled dataset
- Model learns mapping from input → output
- Used for prediction

Types

- **Classification** (output is category)
- **Regression** (output is continuous value)

Examples

- Email spam detection
- Student result prediction
- Disease diagnosis

Common Algorithms

- Decision Trees
- Linear Regression
- Support Vector Machine
- Neural Networks

Advantages

- High accuracy
- Easy evaluation

Disadvantages

- Requires large labeled data
- Labeling is expensive

---

3. Unsupervised Learning

Definition

Unsupervised learning uses **unlabeled data** and finds hidden patterns.

Working

- No predefined output
- Model groups or clusters data

Examples

- Customer segmentation
- Market basket analysis
- Image grouping

Common Algorithms

- K-Means Clustering

- Hierarchical clustering
- DBSCAN

Advantages

- No labeling required
- Finds unknown patterns

Disadvantages

- Difficult to evaluate
- Less accurate

---

4. Semi-Supervised Learning

Definition

Semi-supervised learning uses **small labeled data + large unlabeled data**.

Why Needed?

- Labeling large datasets is costly
- Combines benefits of supervised and unsupervised learning

Examples

- Speech recognition
- Web page classification
- Image recognition

Advantages

- Better accuracy than unsupervised
- Less labeling cost

---

5. K-Means Clustering Algorithm

Definition

K-Means is an **unsupervised clustering algorithm** that groups data into **K clusters**.

Working Steps

1. Choose number of clusters (K)
2. Initialize centroids
3. Assign data points to nearest centroid
4. Recalculate centroids
5. Repeat until convergence

Distance Measure

- Usually **Euclidean distance**

Advantages

- Simple and fast
- Easy to implement

Disadvantages

- K must be specified
- Sensitive to initial centroids

Applications

- Customer segmentation
- Image compression

---

6. Decision Trees

Definition

A decision tree is a **tree-structured supervised learning algorithm** used for classification and regression.

Structure

- Root node
- Internal nodes (decision rules)
- Leaf nodes (output)

Splitting Criteria

- Information Gain
- Gini Index
- Entropy

Advantages

- Easy to understand
- Handles both numerical and categorical data

Disadvantages

- Overfitting
- Sensitive to noise

---

7. Neural Networks

Definition

Neural Networks are inspired by the **human brain** and consist of interconnected neurons.

Structure

- Input layer
- Hidden layers
- Output layer

Working

- Inputs multiplied by weights
- Activation function applied

- Output generated
- Backpropagation updates weights

Common Activation Functions

- Sigmoid
- ReLU
- Tanh

Advantages

- Handles complex patterns
- High accuracy

Disadvantages

- Requires large data
- Computationally expensive

---

8. Deep Learning

Definition

Deep learning is a subset of neural networks with **multiple hidden layers**.

Key Features

- Automatic feature extraction
- Large datasets
- High computational power (GPU)

Popular Deep Learning Models

- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Transformers

Applications

- Image recognition
- Speech recognition
- Self-driving cars
- Natural language processing

Advantages

- State-of-the-art accuracy
- Learns complex patterns

Disadvantages

- Requires huge data
- Black-box nature

---

9. Comparison Table (Important for Exams)

Learning Type	Data	Example
Supervised	Labeled	Decision Tree
Unsupervised	Unlabeled	K-Means
Semi-Supervised	Partial labels	Image tagging
Deep Learning	Large data	Face recognition

---

10. Summary (Exam Ready Points)

- Learning enables AI systems to improve automatically.
- Supervised learning uses labeled data.
- Unsupervised learning finds hidden patterns.
- K-Means clusters data into groups.

Unit-6.0: Advanced Topics in Artificial Intelligence

Advanced topics in AI deal with **real-world intelligent applications** that combine perception, reasoning, learning, and action.

---

1. Computer Vision

Definition

Computer Vision is the field of AI that enables machines to **see, interpret, and understand images and videos** like humans.

Goals of Computer Vision

- Image understanding
- Object detection
- Scene recognition
- Motion analysis

Basic Steps in Computer Vision

1. Image acquisition
2. Pre-processing (noise removal)
3. Feature extraction
4. Object recognition
5. Interpretation

Applications

- Face recognition
- Medical imaging
- Autonomous vehicles

- Surveillance systems

Challenges

- Illumination changes
- Occlusion
- Complex backgrounds

---

2. Natural Language Processing (NLP)

Definition

NLP is a branch of AI that allows machines to **understand, interpret, and generate human language**.

Levels of NLP

1. Lexical analysis
2. Syntactic analysis
3. Semantic analysis
4. Discourse analysis
5. Pragmatic analysis

Common NLP Tasks

- Machine translation
- Sentiment analysis
- Speech recognition
- Chatbots

Applications

- Virtual assistants
- Search engines
- Text summarization

3. Expert Systems

Definition

An Expert System is an AI program that **emulates the decision-making ability of a human expert**.

Components of Expert System

1. Knowledge Base
2. Inference Engine
3. Explanation Facility
4. User Interface

Working

- Knowledge stored as rules
- Inference engine applies rules
- System gives advice or decisions

Examples

- Medical diagnosis systems
- Loan approval systems
- Fault diagnosis

Advantages

- Consistent decisions
- 24×7 availability

Limitations

- Knowledge acquisition is difficult
- Lacks creativity

4. Robotics

Definition

Robotics is the branch of AI that deals with **design, construction, and operation of robots**.

Main Components of a Robot

1. Sensors
2. Actuators
3. Controller
4. Power supply

Types of Robots

- Industrial robots
- Mobile robots
- Service robots
- Humanoid robots

Applications

- Manufacturing
- Healthcare
- Space exploration
- Military operations

Challenges

- Navigation
- Perception
- Decision-making

---

5. Genetic Algorithm (GA)

Definition

A Genetic Algorithm is a **search and optimization technique inspired by natural evolution**.

Key Concepts

- Population
- Chromosome
- Fitness function
- Selection
- Crossover
- Mutation

Working Steps

1. Initialize population
2. Evaluate fitness
3. Select best individuals
4. Apply crossover and mutation
5. Generate new population
6. Repeat until solution found

Advantages

- Handles complex search spaces
- Global optimization

Disadvantages

- Computationally expensive
- No guarantee of optimal solution

Applications

- Scheduling problems

- Feature selection
- Optimization tasks

---

6. Comparison Table (Important for Exams)

Topic	Main Purpose
Computer Vision	Image & video understanding
NLP	Human language processing
Expert System	Expert-level decision making
Robotics	Intelligent machines
Genetic Algorithm	Optimization & search

---

7. Summary (Exam Ready Points)

- Advanced AI topics deal with real-world intelligence.
- Computer Vision enables machines to see.
- NLP enables machines to understand language.
- Expert systems mimic human experts.
- Robotics combines AI with mechanical systems.
- Genetic Algorithms solve optimization problems.