Here are important notes from "lecture1.pdf" on Artificial Intelligence:

Lecture 1: Introduction to Artificial Intelligence

1. Definitions of Artificial Intelligence (AI):

- **Broad View:** Modeling human cognition/mental faculty using computers; making computers do things at which people are currently better; making computers do things which require intelligence.
- **Formal Definition:** Branch of computer science concerned with the study and creation of computer systems that exhibit some form of intelligence or characteristics associated with human intelligence.
- **Rich and Knight (1991):** "study of how to make computers do things at which, at the moment, people are better."
- **Alan Turing (1950):** Actions that are indistinguishable from a human's (Turing Test).
- **David Marr (1977):** "study of complex information processing problems that often have their roots in some aspect of biological information processing."
- **Rodney Brooks (1990):** "The intelligent connection of perception to action."

2. Characteristics of AI Systems:

- Learn new concepts and tasks.
- Reason and draw useful conclusions.
- Remember complicated interrelated facts and infer from them.
- Understand natural language or perceive visual scenes.
- Perceive (vision) and move objects (robotics).
- Plan sequences of actions to complete a goal.
- Offer advice based on rules and situations.
- May exceed human abilities in performing tasks, even if done differently.
- Capable of performing intelligent tasks effectively and efficiently.
- Perform tasks requiring high levels of intelligence.

3. What is Intelligence? (Webster's Dictionary & AI Context):

- Ability to learn, understand, reason; skilled use of reason.
- Ability to apply knowledge and think abstractly.
- Capacity to learn and solve problems (especially novel problems).
- Ability to act rationally; ability to act like humans.

4. What's Involved in Intelligence (from AI perspective):

- **Ability to interact with the real world:** Perceive, understand, and act (e.g., speech recognition, image understanding, robotics).
- **Reasoning and Planning:** Modeling the external world, solving new problems, planning, decision-making, dealing with uncertainties.
- **Learning and Adaptation:** Continuous learning, updating internal models (e.g., a baby learning to categorize).

5. Pioneers of AI:

- **John McCarthy:** LISP, logical reasoning for actions, Non-Monotonic Reasoning.
- **Marvin Minsky:** Co-Founder of MIT AI Lab, co-authored "Perceptrons," FRAME as a knowledge structure.
- **Herbert Simon:** Logic Machines, General Problem Solver (GPS), Preferential Attachment.
- **Arthur Samuel:** Coined "Machine Learning" (1959), AI for Game Playing, Rote Learning.
- **Allen Newell:** Logic Theorist, GPS (Knowledge and Reasoning), Chess Machine.

- **Nils Nilsson:** A* Heuristic Search Algorithm, Planning and Robotics (STRIPS).

6. Turing Test (Imitation Game):

- An operational test for intelligent behavior.
- A human interrogator communicates via teletype with a computer and a human.
- If the interrogator cannot distinguish the computer from the human, the computer passes the test.
- **Capabilities required by computer to pass:**
    - Natural language processing.
    - Knowledge representation.
    - Automated reasoning.
    - Machine learning.

7. Overview and Parent/Academic Disciplines of AI:

- AI is a **multidisciplinary field**.
- **Parent Disciplines:** Philosophy & Cognitive Science, Mathematics, Psychology, Computer Science.
- **Academic Disciplines Relevant to AI:**
    - **Philosophy:** Logic, reasoning methods, mind as physical system, foundations of learning, language, rationality.
    - **Mathematics:** Formal representation and proof, algorithms, (un)decidability, probability/statistics (modeling uncertainty, learning from data).
    - **Economics:** Utility, decision theory, rational economic agents.
    - **Neuroscience:** Neurons as information processing units, understanding brain function (parallel computation, remapping).
    - **Psychology/Cognitive Science:** How people behave, perceive, process cognitive information, represent knowledge.
    - **Computer Engineering:** Building fast computers.
    - **Control Theory:** Designing systems that maximize an objective function over time; machines modifying behavior in response to environment (sense/action loop).
    - **Linguistics:** Knowledge representation, grammars, analysis of human language.

8. Foundations of AI:

- **Mathematics:** Formal logical methods (Boolean, Fuzzy), probability theory, modal and temporal logics.
- **Neuroscience:** Understanding brain function, parallel computation, remapping, interconnections.
- **Control Theory:** Machines modifying behavior (sense/action loop), stable feedback systems, building systems that transition from initial to goal state with minimum energy.
- **Linguistics:** Speech demonstrates intelligence, analysis of human language reveals thought, language and thought are intertwined.

9. Sub-areas of AI:

- Perception
- Reasoning
- Natural Language Understanding (NLU)
- Learning (Machine Learning, Deep Learning)
- Knowledge Representation
- Move and Manipulate objects (Robotics)
- Planning and Scheduling
- Theorem Proving
- Game Playing

- Common Sense Reasoning (dealing with uncertainty and decision making)
- Computer Vision
- Understanding Spoken Utterances
- Intelligent Tutoring Systems
- Machine Translation Systems
- Expert Problem Solving
- Neural Networks
- AI tools

Lecture 2

1. History of AI:

- **Early Beginnings (1943-1950):** McCulloch & Pitts' Boolean circuit model of the brain (1943), Turing's "Computing Machinery and Intelligence" (1950).
- **Birth of AI (1956):** Dartmouth meeting where the name "Artificial Intelligence" was adopted.
- **Initial Promise (1950s-1965):** Early AI programs like Samuel's checkers program, Newell & Simon's Logic Theorist and General Problem Solver (GPS), Gelertner's Geometry Theorem Prover, and McCarthy's invention of LISP.
- **Reality Dawns (1966-1973):** Recognition of intractability of many AI problems and limitations of neural networks, leading to a decline in neural network research.
- **Adding Domain Knowledge (1969-1985):** Development of knowledge-based and rule-based expert systems (e.g., DENDRAL, MYCIN), which had success but were brittle and didn't scale well.
- **Rise of Machine Learning (1986-onwards):** Resurgence of neural networks, significant advances in machine learning algorithms.
- **Role of Uncertainty (1990-onwards):** Introduction of Bayesian networks.
- **AI as Science (1995-onwards):** Integration of learning, reasoning, and knowledge representation; AI methods applied to vision, language, and data mining.

2. The Dartmouth Conference:

- A pivotal 1955 proposal by McCarthy, Minsky, Rochester, and Shannon for a 2-month, 10-man study of AI in 1956. The core idea was that any aspect of learning or intelligence could be precisely described for machine simulation.

3. Applications of AI:

- **General:** Business (financial strategies, advice), Engineering (design checks, new product suggestions), Manufacturing (assembly, inspection), Mining (dangerous conditions), Hospitals (monitoring, diagnosing), Education (teaching), Household (cooking, shopping advice), Farming (pruning, selective harvesting).
- **Everyday Life:** Post Office (address recognition, sorting), Banks (check readers, signature verification, loan classification), Customer Service (automatic voice recognition), Web (identifying age/gender/location, fraud detection), Digital Cameras (face detection), Computer Games (intelligent characters).
- **Specific Examples:** Machine Translation (overcoming language barriers, though challenging as seen in the "spirit is willing but the flesh is weak" example), Games

(chess, checkers victories by AI systems like Deep Blue and Chinook).

4. Present Status of AI:

- Very accurate Speech Recognition (using Hidden Markov Models).
- Effective Planning systems (for factory work, space missions).
- Advanced Probabilistic Reasoning (Bayesian Belief Networks) for dealing with uncertainty, along with Decision and Utility theory.
- Vast improvements in robotics, computer vision, machine learning, and knowledge representation.
- Deep Learning has significantly boosted AI capabilities.

5. Recent AI Characteristics:

- Heavy use of probability theory, decision theory, statistics, and various logics (fuzzy, modal, temporal).

6. Comparisons:

- **AI vs. Conventional Programming:**
  - **AI:** Primarily symbolic, handles incomplete input, uses heuristics, provides explanations, focuses on knowledge, separates control from knowledge, output can be incomplete, easy maintenance (modularity), limited but improving reasoning.
  - **Conventional:** Primarily algorithmic, requires complete input, uses algorithms, usually no explanations, focuses on data/information, integrates control with data, output must be correct, difficult maintenance, no inherent reasoning.
  - **Core Difference:** AI is given problems, not explicit steps to solve them, relying on search and pattern matching; Conventional computing follows explicit algorithmic steps.
- **Human Intelligence vs. Artificial Intelligence:**
  - **Human Pros:** Intuition, common sense, judgment, creativity, beliefs, effective communication, plausible reasoning, critical thinking.
  - **AI Pros:** Simulates human behavior/cognitive processes, captures/preserves human expertise, fast response, comprehends large data quickly.
  - **Human Cons:** Fallible, limited knowledge bases, slow serial processing, unable to retain large data amounts.
  - **AI Cons:** No "common sense," difficulty with "mixed" knowledge, high development costs, raises legal/ethical concerns.
- **Human Brain vs. Computers (Comparison Table):** Highlights that while a human brain has more computational units (neurons) and synapses, supercomputers have achieved rough parity in operations/sec and memory updates/sec with significantly faster cycle times, though personal computers lag.

7. Categories of AI Systems:

- Systems that **think like humans** (cognitive science, neural networks).
- Systems that **act like humans** (performing tasks intelligently like humans).

- Systems that **think rationally** (rely on logic, formal theories, computational models for reasoning).
- Systems that **act rationally** (observed behavior must be rational, even if the internal method is not strictly logical).

8. Present State-of-the-Art Examples:

- **Robotic Vehicles:** STANLEY (2005 Darpa Grand Challenge winner), CMU's Boss (2006 Urban Challenge winner).
- **Speech Recognition:** Automated systems for flight booking.
- **Autonomous Planning & Scheduling:** Spacecraft operations, mission planning (detecting, diagnosing, recovering from problems).
- **Game Playing:** IBM's Deep Blue defeated Kasparov; computers now consistently win human-computer matches.
- **Spam Classification:** Automatic deletion of spam.
- **Logistic Planning:** Automated planning for large-scale events (e.g., Gulf War).
- **Robotics:** Roomba vacuum cleaners, rugged bots for hazardous material/explosive handling.
- **Machine Translation:** Good commercial systems exist.

9. Summary of AI:

- Interdisciplinary field with many sub-areas.
- Knowledge-based systems are important.
- Uses non-monotonic reasoning.
- Employs rigorous optimization and mathematics for probabilistic reasoning and decision-making.
- Numerous commercial packages are available.

Lecture 3

- **Agent Definition:** An agent perceives its environment through **sensors** and acts upon it through **actuators**. The "percept sequence" is the history of everything an agent has perceived, and its actions depend on this history. An agent's behavior is defined by its "agent function" which maps percept sequences to actions.
- **Intelligent Agent:** Similar to a human using eyes, ears, hands, and legs, an intelligent agent uses components like cameras and motors to perceive and act.
- **Example: Vacuum Cleaner Agent:** A simple example illustrating how percept sequences (e.g., `[A, Clean]`, `[A, Dirty]`) map to specific actions (e.g., `Right`, `Suck`). This highlights the idea of a look-up table for agent behavior.
- **Rationality:** A **rational agent** is one that "does the right thing," meaning its actions are desirable and maximize a "performance measure." Rationality depends on the agent's prior knowledge, possible actions, and percept sequence.
- **Autonomy:** An agent lacks autonomy if it relies solely on its designer's prior knowledge. A rational agent should be autonomous, learning from its percepts to compensate for incomplete or incorrect initial knowledge. Learning enables agents to succeed in diverse environments.
- **Specifying the Task Environment (PEAS):** This framework helps define an agent's

task:
  - **P**erformance Measure: How the agent's success is evaluated.
  - **E**nvironment: The world in which the agent operates.
  - **A**ctuators: The means by which the agent acts.
  - **S**ensors: The means by which the agent perceives.
  - Examples are provided for taxi drivers, medical diagnosis systems, robots, etc.
- **Properties of Task Environments:** These characteristics categorize and describe the complexity of an agent's environment:
  - **Fully observable vs. Partially observable:** Whether the agent's sensors provide a complete view of the environment.
  - **Single agent vs. Multiagent:** Whether the agent operates alone or interacts with other agents (can be competitive or cooperative).
  - **Deterministic vs. Stochastic:** Whether the next state of the environment is fully predictable from the current state and action.
  - **Episodic vs. Sequential:** Whether actions in one episode affect future episodes.
  - **Discrete vs. Continuous:** Refers to the nature of states, time, percepts, and actions.
  - **Known vs. Unknown:** Whether the agent knows the rules and outcomes of actions in the environment.
- **Structure of Agents:**
  - **Agent = Architecture + Program**. The agent program implements the agent function.
- **Table-driven Agent:** The simplest (but practically infeasible) approach where every possible percept sequence is mapped to an action in a vast look-up table. This is problematic due to the enormous size of such tables.
- **Kinds of Agent Programs (more practical approaches):**
  - **Simple reflex agents:** Act based *only* on the current percept, ignoring history, using condition-action rules. Works only in fully observable environments.
  - **Model-based reflex agents:** Maintain an "internal state" (a model of the world) to track unobserved aspects of the environment, using percept history and knowledge of how the world evolves and how actions affect it. Can operate in partially observable environments.
  - **Goal-based agents:** Use information about a desired "goal" to choose actions, often employing search and planning algorithms. More flexible than simple reflex agents.
  - **Utility-based agents:** Go beyond just achieving goals by considering the "utility" or "happiness" an agent derives from different outcomes. They choose actions that maximize expected utility, modeling the environment and assessing preferences.
- **Model of a Learning Agent:** Includes a **learning element** (improves the agent), a **performance element** (selects actions), a **critic** (evaluates performance against a standard), and a **problem generator** (suggests new experiences for learning).

Lecture 4

This lecture, "General Problem Solving - 1," introduces the concept of problem-solving through search in Artificial Intelligence.

The core idea is that an agent moves from one "state" to another using "actions" or "operators" until a "goal state" is reached. The process involves:

- Goal Formulation: Defining the desired outcome.
- Problem Formulation: Deciding which states and actions to consider to achieve the goal.
- Search: The process of finding a sequence of actions (a "solution") that leads from the initial state to the goal state.
- Solution Quality: Measured by "path cost," with the optimal solution having the lowest cost.

A problem is formally defined by five components:

1. Initial State: Where the agent starts.
2. Actions: The set of possible moves available from a given state.
3. Transition Model (RESULT(s,a)): Describes the state that results from taking an action 'a' in state 's'. Successor states are reachable from a given state.
4. Goal Test: A function that determines if a given state is the goal state.
5. Path Cost: A numerical cost assigned to each path (sequence of actions).

The initial state, actions, and transition model collectively define the state space of the problem, which includes all states reachable from the initial state.

The lecture then illustrates these concepts with several "toy problems":

- Vacuum World:
    - State: Agent and dirt locations (e.g., n locations have n*2^n states).
    - Initial State: Any configuration.
    - Actions: Left, Right, Suck (and optionally Up, Down for larger environments).
    - Goal Test: All squares are clean.
    - Path Cost: Each step costs 1.
- 8-Puzzle Problem:
    - Consists of an 8-tile (3x3) board with one blank space. Tiles adjacent to the blank can slide into it.
    - States: Location of each tile and the blank.
    - Initial State: Any configuration.
    - Actions: Blank moves Left, Right, Up, or Down.
    - Goal Test: Matches a specified goal configuration.
    - Path Cost: Each step costs 1.
    - This is a sliding-block puzzle; the 8-puzzle has 181,440 reachable states. Larger versions (15-puzzle, 24-puzzle) have significantly more states.
- 8-Queens Problem:
    - Goal: Place 8 queens on a chessboard so no queen attacks another (same row, column, or diagonal).
    - Formulations:
        - Incremental: Starts with an empty board, adds queens one by one.
            - States: Arrangements of 0 to 8 queens.
            - Initial State: No queen.
            - Actions: Add a queen to any empty square.

            - Goal Test: 8 queens, none attacked.
            - A better incremental formulation restricts placing queens in already attacked squares, significantly reducing the state space (e.g., from 1.8x10^14 to 2057 for 8 queens).
        - Complete State: Starts with all 8 queens on the board and moves them around (not detailed in this section).
    - Path Cost: No interest, only the final state counts.
- Cryptarithmetic Problem:
    - A constraint satisfaction problem where letters are assigned unique digits to make an arithmetic equation true (e.g., FORTY + TEN + TEN = SIXTY).
    - State: Some letters may have numbers assigned.
    - Initial State: All letters unassigned.
    - Action: Assign a unique, unassigned number to a letter, following arithmetic rules.
    - Transition Model: Assign an unused number to the next unassigned letter.
    - Goal State: All letters assigned unique numbers, and the arithmetic rules are satisfied.

Lecture -5

This lecture (Lecture 5: General Problem Solving - 2) focuses on real-world problem examples in Artificial Intelligence and introduces the fundamental concepts of search algorithms used to solve them.

Key topics covered:

- Real-world problems:
    - Airline travel problem: Defined by complex states (location, time, historical data), actions (taking flights), transition models, a goal test (final destination), and a path cost that considers monetary cost, waiting time, seat quality, etc. It also highlights the need for contingency plans.
    - Touring problems: Similar to route-finding but require visiting multiple locations. The state includes the current location and the set of visited cities (e.g., "Visit every city... starting and ending in Bucharest").
    - Traveling Salesman Problem (TSP): A touring problem where each city must be visited exactly once to find the shortest route. It's an NP-hard problem with applications beyond salespersons, like circuit-board drilling.
    - Automatic Assembly Sequencing: Finding the correct order to assemble parts of an object (e.g., electric motors, protein design). Incorrect order can lead to unfeasible steps. Generating legal actions is computationally expensive.
- Searching for solutions:
    - A solution is an action sequence, found by search algorithms that explore possible action sequences.
    - These sequences form a search tree with the initial state as the root, branches as actions, and nodes as states.
    - Expanding a state: Applying legal actions to generate new states (child nodes).
    - Leaf nodes: Nodes with no children.
    - Frontier: The set of all leaf nodes available for expansion at any given point.
    - The search continues by expanding nodes from the frontier until a solution is found or the frontier is empty.

- Handling repeated states and redundant paths:
    - Loopy paths (e.g., Arad -> Sibiu -> Arad) lead to repeated states and can make the search tree infinite, causing algorithms to fail. Since path costs are additive and non-negative, loopy paths are never optimal.
    - Redundant paths (multiple ways to reach the same state) are also inefficient.
    - To avoid exploring these, the GRAPH-SEARCH algorithm is introduced. It augments TREE-SEARCH with an explored set to remember previously expanded nodes. Newly generated nodes that are already in the explored set or frontier are discarded.
    - GRAPH-SEARCH ensures at most one copy of each state is considered, systematically examining the state space.
- Search Strategies:
    - Uninformed or blind search: Doesn't use problem-specific knowledge beyond the state description.
    - Informed or heuristic search: Uses problem-specific knowledge to guide the search.
    - Game playing search.
    - Randomized search algorithms (e.g., Hill climbing, stochastic search like genetic algorithms).
- Performance of problem-solving (Four Criteria):
    - Completeness: Guaranteed to find a solution if one exists.
    - Optimality: Finds the optimal (best) solution.
    - Time complexity: How long it takes to find a solution (measured by nodes generated).
    - Space complexity: How much memory is needed (measured by nodes stored).
    - Complexity is often expressed in terms of:
        - `b`: branching factor (max successors per node).
        - `d`: depth of the shallowest goal node.
        - `m`: maximum path length in the state space.
    - Search cost: Depends on time and space complexity.
    - Total cost: Combines search cost and the path cost of the solution found.

Lecture 1: Introduction to Artificial Intelligence

- **Definitions of AI: Explores various definitions, from modeling human cognition to making computers perform tasks requiring intelligence, as proposed by pioneers like Alan Turing, Rich and Knight, David Marr, and Rodney Brooks.**
- **Characteristics of AI Systems: Highlights abilities such as learning, reasoning, remembering facts, understanding natural language, perceiving, planning, and offering advice.**
- **What is Intelligence? Defines intelligence as the ability to learn, understand, reason, apply knowledge, and solve problems.**
- **What's Involved in Intelligence (AI perspective): Focuses on interacting with the real world (perception, understanding, action), reasoning and planning (modeling, problem-solving, decision-making), and learning and adaptation.**
- **Pioneers of AI: Lists key figures like John McCarthy (LISP), Marvin Minsky (MIT AI Lab), Herbert Simon (Logic Machines), Arthur Samuel (Machine Learning), Allen Newell (Logic Theorist), and Nils Nilsson (A* Algorithm).**
- **Turing Test: Explains this operational test for intelligent behavior, requiring natural**

language processing, knowledge representation, automated reasoning, and machine learning.
- Overview and Parent/Academic Disciplines of AI: Emphasizes AI as a multidisciplinary field drawing from philosophy, mathematics, psychology, computer science, economics, neuroscience, computer engineering, control theory, and linguistics.
- Foundations of AI: Discusses contributions from mathematics, neuroscience, control theory, and linguistics.
- Sub-areas of AI: Lists diverse areas like perception, reasoning, NLU, machine learning, robotics, planning, game playing, computer vision, and expert systems.

Lecture 2: History and Applications

- History of AI: Traces AI's development from early beginnings (McCulloch & Pitts, Turing) and its birth at the Dartmouth meeting (1956), through periods of initial promise, challenges, the rise of expert systems, and the resurgence of machine learning and probabilistic reasoning.
- The Dartmouth Conference: Describes this pivotal 1955 proposal that formally established AI as a field.
- Applications of AI: Provides numerous examples across business, engineering, manufacturing, healthcare, education, household, and everyday life (e.g., postal services, banking, web, digital cameras, games, machine translation, robotic vehicles).
- Present Status of AI: Notes advancements in speech recognition, planning systems, probabilistic reasoning (Bayesian networks), robotics, computer vision, machine learning, and deep learning.
- Recent AI Characteristics: Highlights the heavy use of probability theory, decision theory, statistics, and various logics.
- Comparisons: Differentiates AI from conventional programming (symbolic vs. algorithmic, heuristics vs. algorithms), and human intelligence from artificial intelligence (intuition, creativity vs. speed, data comprehension). Also includes a comparison of the human brain vs. computers.
- Categories of AI Systems: Classifies AI systems based on whether they think like humans, act like humans, think rationally, or act rationally.
- Present State-of-the-Art Examples: Showcases achievements like robotic vehicles (STANLEY, Boss), speech recognition, autonomous planning, game playing (Deep Blue), spam classification, logistic planning, and commercial machine translation.
- Summary of AI: Reaffirms AI as an interdisciplinary, knowledge-based field utilizing non-monotonic reasoning, rigorous optimization, and mathematics.

Lecture 3: Intelligent Agents

- Agent Definition: Defines an agent as something that perceives its environment via sensors and acts through actuators, with its behavior determined by an "agent function" mapping percept sequences to actions.
- Intelligent Agent: Compares it to a human, using components like cameras and motors for perception and action.
- Example: Vacuum Cleaner Agent: Illustrates percept-action mapping.
- Rationality: Explains a rational agent as one that "does the right thing" to maximize a performance measure, dependent on prior knowledge, possible actions, and percept sequence.
- Autonomy: Discusses the importance of an agent learning from percepts to compensate for incomplete initial knowledge, enabling success in diverse environments.

- **Specifying the Task Environment (PEAS): Introduces a framework to define an agent's task: Performance measure, Environment, Actuators, Sensors.**
- **Properties of Task Environments: Characterizes environments as fully/partially observable, single/multiagent, deterministic/stochastic, episodic/sequential, discrete/continuous, and known/unknown.**
- **Structure of Agents: States that an agent is its architecture plus its program.**
- **Table-driven Agent: Acknowledges this simple, but practically unfeasible, approach.**
- **Kinds of Agent Programs: Describes more practical agents:**
    - **Simple reflex agents: Act based only on the current percept.**
    - **Model-based reflex agents: Maintain an internal state (model of the world) to handle partial observability.**
    - **Goal-based agents: Use goals to choose actions, employing search and planning.**
    - **Utility-based agents: Maximize "utility" or "happiness" from outcomes.**
- **Model of a Learning Agent: Outlines components like a learning element, performance element, critic, and problem generator.**

Lecture 4: General Problem Solving - 1

- **Core Idea: Introduces problem-solving as an agent moving through "states" using "actions" to reach a "goal state," involving goal formulation, problem formulation, and search.**
- **Problem Definition: Formally defines a problem by its initial state, actions, transition model, goal test, and path cost.**
- **State Space: The initial state, actions, and transition model define the state space.**
- **Toy Problems: Illustrates concepts with:**
    - **Vacuum World: Agent and dirt locations, actions (Left, Right, Suck), goal (all clean), path cost (1 per step).**
    - **8-Puzzle Problem: Sliding tiles, states (tile locations), actions (blank moves), goal (specific configuration), path cost (1 per step).**
    - **8-Queens Problem: Placing queens without attacking each other; incremental formulation (adding queens one by one) is discussed, with path cost not a concern.**
    - **Cryptarithmetic Problem: Assigning unique digits to letters to solve an arithmetic equation, a constraint satisfaction problem.**

Lecture 5: General Problem Solving - 2

- **Real-world Problems: Extends problem-solving to more complex scenarios:**
    - **Airline travel problem: Involves complex states, actions (flights), and path costs considering various factors.**
    - **Touring problems: Requires visiting multiple locations.**
    - **Traveling Salesman Problem (TSP): An NP-hard touring problem with broad applications.**
    - **Automatic Assembly Sequencing: Determining the correct order to assemble parts.**
- **Searching for solutions: Explains solutions as action sequences found by exploring a search tree. Key terms include expanding a state, leaf nodes, and the frontier.**
- **Handling repeated states and redundant paths: Introduces the GRAPH-SEARCH algorithm to avoid infinite loops and inefficiency by remembering previously explored nodes.**
- **Search Strategies: Categorizes strategies into uninformed (blind) search, informed (heuristic) search, game playing search, and randomized search.**
- **Performance of problem-solving: Evaluates search algorithms based on four criteria: completeness, optimality, time complexity, and space complexity, using branching factor**

(b), depth of shallowest goal (d), and maximum path length (m) for complexity analysis. Total cost combines search cost and solution path cost.