1. Constraint Satisfaction Problems (CSPs)

Definition
A Constraint Satisfaction Problem (CSP) is a mathematical problem defined as a set of variables, domains, and constraints:
Variables (X): A finite set {X1, X2, ..., Xn}.
Domains (D): Each variable Xi has a domain Di of possible values.
Constraints (C): A set of constraints specifying allowable combinations of values.

Types of Constraints
Unary Constraint: Involves a single variable (e.g., X > 5).
Binary Constraint: Involves pairs of variables (e.g., X ≠ Y).
Higher-Order Constraints: Involves three or more variables.
Soft Constraints: Preferences rather than absolute restrictions.
Constraint Graph
Nodes represent variables.
Edges represent constraints between variables.

Example Problems

Map Coloring
Variables: Regions on the map.
Domains: {red, blue, green}.
Constraints: Adjacent regions must have different colors.

Sudoku
Variables: Cells in the grid.
Domains: {1,2,3,4,5,6,7,8,9}.
Constraints: Rows, columns, and 3x3 boxes must contain distinct numbers.

Solving CSPs
Backtracking Search
Depth-First Search (DFS) where variables are assigned values one at a time.
Failure occurs when a variable cannot be assigned any value.
Improvements to Backtracking
Minimum Remaining Values (MRV): Choose the variable with the fewest legal values first.
Degree Heuristic: Choose the variable involved in the most constraints.
Least Constraining Value (LCV): Choose the value that rules out the fewest values in remaining variables.
Forward Checking: Eliminates inconsistent values for future assignments.
Arc Consistency (AC-3 Algorithm): Prunes the domain by ensuring that for every value of X, there exists a consistent value for Y.
Tree-Structured CSPs: Solve efficiently using a two-pass algorithm.

Local Search for CSPs
Min-Conflicts Heuristic: Select a variable that violates the most constraints and reassign it to the value that minimizes conflicts.
Works well for large CSPs like map coloring and scheduling.

2. Search Algorithms
Uninformed Search
Breadth-First Search (BFS): Explores shallowest nodes first.
Depth-First Search (DFS): Explores deepest nodes first.
Uniform Cost Search (UCS): Expands the least costly path first.
Informed Search (Heuristic Search)
Greedy Best-First Search: Expands node with lowest heuristic value.
A Search*: Uses f(n) = g(n) + h(n), where g(n) is the cost so far, and h(n) is the estimated cost to goal.
Admissible Heuristic: Never overestimates the cost.
Consistent Heuristic: h(n) ≤ c(n, n') + h(n').

Example Problems
8-Puzzle
States: Tile configurations.
Actions: Moving tiles.
Goal: Arrange tiles in order.
Adversarial Search (Games)
Minimax Algorithm: Selects moves assuming optimal play by the opponent.
Alpha-Beta Pruning: Prunes parts of the tree that do not affect the final decision.
Expectimax Algorithm: Used when the opponent moves randomly (e.g., dice rolls in Backgammon).
Monte Carlo Tree Search (MCTS): Uses random simulations to estimate the best move.

3. Probability and Bayes' Nets

Basic Probability Rules

Product Rule: P(A, B) = P(A | B) P(B).
Bayes' Rule: P(A | B) = (P(B | A) P(A)) / P(B).
Marginalization: Summing out a variable from a joint distribution.
Independence: P(A, B) = P(A) P(B) if A and B are independent.
Conditional Independence: P(A | B, C) = P(A | C) if A is independent of B given C.

Bayesian Networks

A directed acyclic graph (DAG) where edges represent dependencies.

Joint probability distribution factorized as P(X1, ..., Xn) = Π P(Xi | Parents(Xi)).

Inference in Bayes' Nets

Variable Elimination: Eliminates variables by summing out.
Enumeration: Computes exact probabilities using the full joint distribution.

Sampling Methods:
Rejection Sampling
Likelihood Weighting
Gibbs Sampling

4. Game Theory and Adversarial Search

Game Types
Deterministic vs. Stochastic: Chess vs. Backgammon.
Perfect Information vs. Imperfect Information: Chess vs. Poker.
Zero-Sum vs. Non-Zero-Sum: Chess vs. Multiplayer economic games.

Game Trees
Minimax Algorithm: Determines optimal play assuming opponent plays optimally.
Alpha-Beta Pruning: Eliminates branches that cannot affect the outcome.
Evaluation Functions: Used to estimate non-terminal states (e.g., piece values in chess).
Expectimax: Used in games with randomness (e.g., dice games).

5. Classical AI Topics and Applications

Intelligent Agents: Perceive and act in an environment to maximize utility.
PEAS Framework: Performance Measure, Environment, Actuators, Sensors.
Search Problems: Finding paths in state space representations.
Machine Learning Applications: Probabilistic models, reinforcement learning, decision trees.

Key Formulas and Notation

Minimax Value:
V(s) = max_{a} min_{s'} V(s')
A Search*: f(n) = g(n) + h(n)
Bayes' Theorem: P(A | B) = (P(B | A) P(A)) / P(B)

Alpha-Beta Pruning Conditions:
α: Best choice found so far for the maximizer.
β: Best choice found so far for the minimizer.

Final Tips

For CSPs, use MRV + Forward Checking to improve backtracking.

For Search Problems, remember that A* is optimal if h(n) is admissible.

For Games, always check pruning opportunities to speed up minimax.

For Bayes' Nets, always use conditional independence to simplify calculations.

Definitions

Artificial Intelligence (AI): A field of computer science focused on creating systems that can perform tasks requiring human intelligence, such as problem-solving, learning, and decision-making.

Constraint Satisfaction Problem (CSP): A problem defined by variables, domains, and constraints that must be satisfied to find a solution.

Variable: An element of a CSP that must be assigned a value from a domain.

Domain: The set of possible values a variable can take.

Constraint: A restriction on allowable values for a set of variables.

Unary Constraint: A constraint involving a single variable (e.g., X > 5).

Binary Constraint: A constraint involving a pair of variables (e.g., X ≠ Y).

Higher-Order Constraint: A constraint involving three or more variables.

Soft Constraint: A preference rather than an absolute restriction.

Constraint Graph: A graphical representation of a CSP where nodes represent variables and edges represent constraints.

Backtracking Search: A depth-first search method for solving CSPs by incrementally assigning values to variables and backtracking when a conflict arises.

Minimum Remaining Values (MRV): A heuristic that selects the variable with the fewest legal values first.

Degree Heuristic: A heuristic that selects the variable involved in the most constraints.

Least Constraining Value (LCV): A heuristic that assigns values that rule out the fewest remaining choices.

Forward Checking: A technique that eliminates inconsistent values for future assignments when a variable is assigned.

Arc Consistency (AC-3 Algorithm: A constraint propagation algorithm that ensures every value of one variable has a consistent value in another.

Tree-Structured CSP: A CSP that can be solved efficiently using a two-pass algorithm due to its acyclic structure.

Min-Conflicts Heuristic: A local search technique that selects a conflicting variable and assigns it the value that minimizes constraint violations.

Breadth-First Search (BFS): An uninformed search that explores the shallowest nodes first.

Depth-First Search (DFS): An uninformed search that explores the deepest nodes first.

Uniform Cost Search (UCS): An uninformed search that expands the least costly path first.

Greedy Best-First Search: A search algorithm that expands the node with the lowest heuristic value.

A Search*: A search algorithm using $f(n) = g(n) + h(n)$, where $g(n)$ is the cost so far and $h(n)$ is the estimated cost to the goal.

Admissible Heuristic: A heuristic that never overestimates the true cost to reach the goal.

Consistent Heuristic: A heuristic that satisfies $h(n) \leq c(n, n') + h(n')$, ensuring optimal solutions in A*.

Minimax Algorithm: An adversarial search algorithm that assumes optimal play by both players and selects the best move accordingly.

Alpha-Beta Pruning: A technique to speed up Minimax by eliminating branches that cannot influence the final decision.

Expectimax Algorithm: An adversarial search algorithm used when the opponent moves randomly.

Monte Carlo Tree Search (MCTS): A game tree search method that uses random simulations to estimate the best move.

Bayes' Theorem: A formula for conditional probability: $P(A \mid B) = (P(B \mid A) P(A)) / P(B)$.

Marginalization: A probability technique that sums out a variable from a joint distribution.

Conditional Independence: The property that $P(A \mid B, C) = P(A \mid C)$ when A is independent of B given C.

Bayesian Network: A directed acyclic graph (DAG) representing probabilistic relationships between variables.

Variable Elimination: A method for exact inference in Bayesian networks by summing out variables.

Sampling Methods: Techniques such as rejection sampling, likelihood weighting, and Gibbs sampling used for inference in probabilistic models.

Intelligent Agent: An entity that perceives its environment and takes actions to maximize its utility.

PEAS Framework: Describes an agent's Performance measure, Environment, Actuators, and Sensors.

Game Tree: A tree representation of possible game states and moves used in adversarial search algorithms.

Utility Function: A function that assigns a numerical value to different outcomes in decision-making problems.

Zero-Sum Game: A game where one player's gain is another player's loss (e.g., Chess).

Non-Zero-Sum Game: A game where players can have different payoffs that are not necessarily opposite (e.g., cooperative games).

Horizon Effect: A limitation in game tree search where short-term decisions can obscure long-term consequences.