ABHYASMITRA.IN

COMPREHENSIVE UNIVERSITY STUDY NOTES

ARTIFICIAL INTELLIGENCE

Unit III — Adversarial Search & Constraint Satisfaction

Topics Covered in this Unit

Game Theory & Optimal Decisions in Games: 2-Player Zero-Sum Perfect Information Games

The Minimax Algorithm: Mathematical Derivation, Recursive Value Propagation, Time & Space Complexity

Alpha-Beta Pruning: Alpha Cutoffs, Beta Cutoffs, Move Ordering & Reducing Branching Factor from b to sqrt(b)

Heuristic Evaluation Functions, Cutoff Tests, Horizon Effect & Quiescence Search

Monte Carlo Tree Search (MCTS): 4-Stage Cycle (Selection via UCB1, Expansion, Simulation/Rollout, Backpropagation)

Stochastic Games (Expectiminimax Algorithm) & Partially Observable Games (Belief States / Kriegspiel)

Constraint Satisfaction Problems (CSP): Formal Definition (Variables, Domains, Constraints) & Constraint Graphs

Constraint Propagation & Inference: Node Consistency, Arc Consistency (AC-3 Algorithm) & Forward Checking

Backtracking Search for CSPs: MRV (Minimum Remaining Values), Degree Heuristic & Least Constraining Value (LCV)

Intelligent Backtracking: Conflict Sets, Backjumping & Conflict-Directed Backjumping (CBJ)

High-Yield University Exam Question Bank with Detailed Model Answers & Unit Summary

Compiled in a structured, easy-to-understand, textbook style format
Specially curated for B.Tech / B.E. / BCA / MCA / B.Sc Computer Science, AI & Data Science Students

COURSE SYLLABUS & MAPPING

[UNIT III SYLLABUS — PRESCRIBED UNIVERSITY CURRICULUM]

This section contains the official syllabus for Unit III. Verify with your university curriculum mapping.

Unit III Syllabus Breakdown:

Unit III Adversarial Search and Constraint Satisfaction: Game Theory, Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs.

• Course Code: _____ • Semester / Year: _____

TABLE OF CONTENTS

3.1 Game Theory & Optimal Decisions in Games (Minimax) 4
Mathematical Formulation of the Minimax Algorithm 4
3.2 Alpha–Beta Pruning & Heuristic Game Search 4
Pruning Condition & Effectiveness 4
Heuristic Evaluation Functions & Quiescence Search 5
3.3 Monte Carlo Tree Search (MCTS) 5
The 4-Step MCTS Iterative Cycle 5
3.4 Stochastic Games & Partially Observable Games 6
1. Stochastic Games (Expectiminimax) 6
2. Partially Observable Games 6
3.5 Constraint Satisfaction Problems (CSP): Formulation & Consistency 6
Constraint Propagation & Local Consistency 6
3.6 Backtracking Search & CSP Heuristics 7
Backtracking Search Algorithm 7
Three High-Yield CSP Heuristics 7
3.7 High-Yield University Exam Question Bank & Model Answers 7
Part A: Short Answer Questions (2–3 Marks) 7
Part B: Long Answer Questions (8–10 Marks) 8
3.8 Unit III Summary & Quick Revision Sheet 8

3.1 Game Theory & Optimal Decisions in Games (Minimax)

Definition: 2-Player Zero-Sum Game

In AI, a classic game is formalized as an adversarial search problem between two players: MAX (who attempts to maximize the terminal utility score) and MIN (who attempts to minimize it). In a Zero-Sum game, any gain for MAX is an equal loss for MIN (Total Utility = 0).

Mathematical Formulation of the Minimax Algorithm

The Minimax algorithm computes the optimal move for MAX by recursively calculating the Minimax value of each successor state s:

MINIMAX-VALUE(s) = • UTILITY(s) if TERMINAL-TEST(s) • max[a ∈ ACTIONS(s)] MINIMAX-VALUE(RESULT(s, a)) if PLAYER(s) = MAX • min[a ∈ ACTIONS(s)] MINIMAX-VALUE(RESULT(s, a)) if PLAYER(s) = MIN

- **• Complexity of Minimax:**

Time Complexity: O(b^m), Space Complexity: O(b × m) (where b is legal moves branching factor and m is maximum game tree depth)

For Chess (b ≈ 35, m ≈ 80), total nodes = 35^80 ≈ 10^123 (far exceeds total atoms in observable universe), rendering raw Minimax completely intractable.

3.2 Alpha–Beta Pruning & Heuristic Game Search

Definition: Alpha–Beta Pruning

Alpha–Beta Pruning is an optimal search enhancement that prunes away subtrees that cannot possibly influence the final Minimax decision. It maintains two bounds along the search path:

• α (Alpha): The value of the best (highest-value) choice found so far along the path for MAX (Initial α = -∞).

• β (Beta): The value of the best (lowest-value) choice found so far along the path for MIN (Initial β = +∞).

Pruning Condition & Effectiveness

- **• Pruning Rule:** Prune remaining sibling subtrees whenever α ≥ β.

  - • **Alpha Cutoff (at MIN node):** If MIN's current value becomes ≤ α, MAX will never choose this branch; prune immediately.
  - • **Beta Cutoff (at MAX node):** If MAX's current value becomes ≥ β, MIN will never allow this branch; prune immediately.
- • **Computational Gain with Perfect Move Ordering:** Under optimal move ordering (best moves evaluated first), Alpha–Beta pruning reduces time complexity from O(b^m) to O(b^(m/2)) = O((√b)^m). This effectively doubles the searchable search depth (e.g., from 4 plies to 8 plies in Chess).

Heuristic Evaluation Functions & Quiescence Search

- • **Evaluation Function EVAL(s):** Estimates expected utility when full game tree search is cut off at fixed depth d (e.g., weighted linear sum of material balance, king safety, and center control in Chess).
- • **Horizon Effect:** Occurs when an unavoidable catastrophic loss is pushed beyond the search horizon depth by executing stalling delaying tactics.
- • **Quiescence Search:** Prevents evaluation during non-quiet, volatile tactical states (e.g., in the middle of a queen trade capture sequence) by extending search until a quiescent state is reached.

3.3 Monte Carlo Tree Search (MCTS)

Definition: Monte Carlo Tree Search (MCTS)
MCTS is a best-first heuristic search algorithm that builds an asymmetric game tree guided by random simulation rollouts. Highly successful in massive state spaces where handcrafting evaluation functions is impossible (e.g., AlphaGo).

The 4-Step MCTS Iterative Cycle

- **1. Selection:** Traverse the tree from root to a leaf node using the Upper Confidence Bound for Trees (UCT / UCB1) policy to balance Exploration and Exploitation:

$$UCB1 = (Q(v) / N(v)) + C \times \sqrt{[\ ln(N(parent)) / N(v)\ ]}$$

where Q(v) is total reward, N(v) is visit count, and C is exploration constant (√2).

- **2. Expansion:** If the selected node is non-terminal, spawn one or more unvisited child nodes.
- **3. Simulation (Rollout):** Play out a complete random or heuristic simulation to terminal state from the new child.
- **4. Backpropagation:** Propagate the game outcome reward (Win = +1, Loss = 0) up through all ancestor nodes, incrementing visit counts N and cumulative rewards Q.

3.4 Stochastic Games & Partially Observable Games

1. Stochastic Games (Expectiminimax)

Games involving randomness (dice rolls in Backgammon/Monopoly) insert Chance Nodes between MAX and MIN layers:

EXPECTIMINIMAX-VALUE(s) = ∑[s'] P(s') × EXPECTIMINIMAX-VALUE(s')

Complexity expands to O((b × n)^m), where n is the number of distinct dice outcomes.

2. Partially Observable Games

Games with hidden information (fog of war in StarCraft, card hands in Poker, Kriegspiel blind chess). The agent must maintain a Belief State (probability distribution over all possible physical world states).

3.5 Constraint Satisfaction Problems (CSP): Formulation & Consistency

Formal 3-Tuple Definition of a CSP

A Constraint Satisfaction Problem (CSP) is defined by a 3-tuple (X, D, C):

- *X = {X_1, X_2, ..., X_n}: A finite set of Variables.*
- *D = {D_1, D_2, ..., D_n}: A set of Domains specifying allowable values for each variable.*
- *C = {C_1, C_2, ..., C_m}: A set of Constraints specifying allowable combinations of values across subsets of variables.*

Goal: Find a complete, consistent assignment of values to all variables satisfying every constraint.

Constraint Propagation & Local Consistency

- **1. Node Consistency (1-Consistency):** Every value in a variable's domain satisfies all unary constraints on that variable (e.g., X_1 ≠ Green eliminates Green from D_1).
- **2. Arc Consistency (2-Consistency & The AC-3 Algorithm):** A variable X_i is Arc-Consistent with respect to X_j if for every value x ∈ D_i, there exists at least one value y ∈ D_j satisfying the binary constraint between X_i and X_j. AC-3 maintains a queue of arcs, repeatedly removing invalid domain values with worst-case time complexity O(c × d^3) (where c is binary constraints and d is max domain size).

- **3. Forward Checking:** A lightweight form of constraint propagation executed after each variable assignment. When variable X is assigned value v, forward checking immediately deletes all values conflicting with (X=v) from the domains of all unassigned neighbor variables. If any domain becomes empty, backtrack immediately.

3.6 Backtracking Search & CSP Heuristics

Backtracking Search Algorithm

Backtracking search is a depth-first search for CSPs that assigns values to one variable at a time, immediately backtracking as soon as a constraint violation is detected:

Three High-Yield CSP Heuristics

- **1. Minimum Remaining Values (MRV / 'Most Constrained Variable' / 'Fail-First'):** Chooses the unassigned variable with the fewest allowable legal values left in its domain. Rationale: Prunes search tree early by failing as fast as possible.
- **2. Degree Heuristic (Tie-Breaker for MRV):** Chooses the unassigned variable with the largest number of constraints on remaining unassigned variables. Rationale: Imposes maximum constraints on future variables.
- **3. Least Constraining Value (LCV / 'Fail-Last'):** Once a variable is selected, tests values in order of which value rules out the fewest choices for neighboring unassigned variables. Rationale: Maximizes flexibility for future variable assignments.

3.7 High-Yield University Exam Question Bank & Model Answers

Part A: Short Answer Questions (2–3 Marks)

- **Q1. What is Alpha–Beta Pruning? State the Alpha and Beta cutoff conditions.**

Answer: Alpha–Beta pruning is an optimization for Minimax that eliminates branches that cannot affect the final outcome. An Alpha cutoff occurs at a MIN node when its value becomes ≤ α. A Beta cutoff occurs at a MAX node when its value becomes ≥ β (prune when α ≥ β).

- **Q2. Define a Constraint Satisfaction Problem (CSP) formally.**

Answer: A CSP is defined as a 3-tuple (X, D, C): a set of Variables X = {X_1...X_n}, a set of Domains D = {D_1...D_n}, and a set of Constraints C = {C_1...C_m}. A solution is a complete assignment of values to all variables that satisfies all constraints.

- **Q3. Differentiate between the MRV and LCV heuristics in CSPs.**

Answer: MRV (Minimum Remaining Values) is a variable-selection heuristic that selects the variable with the fewest legal values ('Fail-First'). LCV (Least Constraining Value) is a value-selection heuristic that chooses the value that eliminates the fewest choices for neighboring variables ('Fail-Last').

- **Q4. List the four stages of Monte Carlo Tree Search (MCTS).**

Answer: (1) Selection (using UCB1 policy), (2) Expansion (adding child nodes), (3) Simulation / Rollout (playing random game to termination), and (4) Backpropagation (updating visit counts and reward values up the tree).

Part B: Long Answer Questions (8–10 Marks)

- **Q5. Explain the Minimax Algorithm and Alpha–Beta Pruning with a complete numerical game tree walkthrough. Demonstrate both Alpha and Beta cutoffs.**

Model Answer Outline:

- 1. Game theory foundations: 2-player zero-sum perfect information games.
- 2. Minimax recursive formulation and derivation across alternating MAX and MIN plies.
- 3. Alpha–Beta tracking: Definitions of α (MAX lower bound) and β (MIN upper bound).
- 4. Step-by-step game tree diagram: 3-ply tree with 8 leaf utilities; tracking [α, β] updates; demonstrating exact branch where α ≥ β triggers a cutoff.
- 5. Complexity analysis: Proving reduction to O(b^(m/2)) under perfect move ordering.

- **Q6. Describe Constraint Satisfaction Problems (CSPs). Explain Arc Consistency (AC-3 Algorithm) and Backtracking Search with MRV, Degree, and LCV heuristics on the Map Coloring Problem.**

Model Answer Outline:

- 1. Formal CSP 3-tuple (X, D, C) definition and Constraint Graph representation.
- 2. Australia Map Coloring formulation (Variables: WA, NT, SA, Q, NSW, V, T; Domain: {Red, Green, Blue}; Constraints: Adjacent regions cannot share color).
- 3. Arc Consistency AC-3 algorithm: Pseudocode, arc queue processing, domain reduction logic.
- 4. Backtracking search walkthrough applying MRV (choosing most constrained state), Degree Heuristic (breaking ties via South Australia with 5 neighbors), and LCV.
- 5. Comparison between Forward Checking and Full Arc Consistency.

3.8 Unit III Summary & Quick Revision Sheet

Adversarial & CSP Domain	Key Theoretical & Mathematical Takeaways for Exams
Minimax & Alpha-Beta	Minimax: O(b^m) time, O(bm) space. Alpha-Beta: Prunes when $\alpha \geq \beta$. Perfect ordering yields O(b^(m/2)).
MCTS	4 Steps: Selection (UCB1 = Q/N + C√(ln N_p / N)), Expansion, Simulation, Backpropagation.
Stochastic Games	Expectiminimax: MAX, MIN, and Chance nodes (Probability weighted averages).
CSP Formalism	3-Tuple (X, D, C). Arc Consistency AC-3 enforces for all $x \in D_i \exists y \in D_j$ satisfying constraint.
CSP Heuristics	MRV (Variable with min values / Fail-First), Degree (Variable with max constraints), LCV (Value ruling out min choices / Fail-Last).