SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of AI &DS

Course Name – 23ADT201 ARTIFICIAL INTELLIGENCE

II Year / III Semester

UNIT 3

GAME THEORY

Topic:STOCHASTIC GAMES

Case Study: Partially Observable and Stochastic Games in Daily Commute Optimization

Background

- **Task**: Optimizing daily commuting routes and times for a person traveling to work. The person wants to minimize travel time and avoid traffic, which is unpredictable and partially observable.
- **Challenges**: The commuter has incomplete information about traffic conditions (partially observable) and faces uncertainties such as weather changes, accidents, and road work (stochastic elements).

STOCHASTIC GAMES

What are Stochastic Games?

A **stochastic game** is a multi-agent game where players take turns making decisions, influencing the game's current state and future states. Unlike deterministic games, randomness is involved in determining transitions between states, making decision-making more complex.

The essential components of a stochastic game include:

- **A set of states (S)**: Represents the possible configurations of the game.
- **A set of actions (A)**: The decisions available to each player.
- **Transition probabilities (P)**: Random events (e.g., dice rolls) that dictate how the game moves from one state to another.
- **Reward function (R)**: The payoff that each player receives based on their actions and the current state.
- **Discount factor (γ)**: A value indicating the importance of future rewards, with smaller values placing more emphasis on short-term outcomes.

STOCHASTIC GAMES

Role of Stochastic Games in Artificial Intelligence (AI)

Stochastic games play an essential role in AI for several reasons:

1. **Modeling Multi-Agent Systems**: Many AI systems involve multiple agents that need to interact, cooperate, or compete with each other. Stochastic games provide a framework for modeling such interactions, incorporating both competitive and collaborative elements.
2. **Incorporating Uncertainty**: Real-world environments are often unpredictable. Stochastic games allow AI to handle uncertainty by making decisions that consider both known factors (such as opponents' actions) and unknown factors (such as random events).
3. **Sequential Decision-Making**: Many AI applications require agents to make decisions over time, with future states influenced by current actions. Stochastic games are perfect for modeling such environments, as they require agents to think strategically about future outcomes.
4. **Reinforcement Learning**: Stochastic games are closely related to reinforcement learning (RL), a subfield of AI where agents learn optimal strategies through trial and error in dynamic environments. In fact, one of the key frameworks for RL is based on **Markov decision processes (MDPs)**, which are essentially single-agent stochastic games. Multi-agent reinforcement learning extends these ideas to environments where multiple agents interact, each learning their optimal strategies.

STOCHASTIC GAMES

Types of Stochastic Games

There are various types of stochastic games, each designed to address specific kinds of multi-agent interactions:

1. **Zero-Sum Stochastic Games**: In these games, one player's gain is exactly equal to the other player's loss. This type of game often arises in competitive situations, such as in adversarial AI or security scenarios. A popular example of this is the **Minimax Algorithm** used in chess and other competitive games.
2. **General-Sum Stochastic Games**: Unlike zero-sum games, the payoffs in general-sum games do not need to balance out. Agents can either compete or collaborate to achieve their objectives, making these games suitable for environments where multiple agents may have conflicting or complementary goals.
3. **Markov Games**: Markov games are a special case of stochastic games where the state transitions follow the **Markov property**—i.e., the future state depends only on the current state and the actions taken, not on previous states. This simplifies the mathematical complexity and is widely used in AI research for modeling decision processes in dynamic environments.

STOCHASTIC GAMES

Understanding the Game of Backgammon

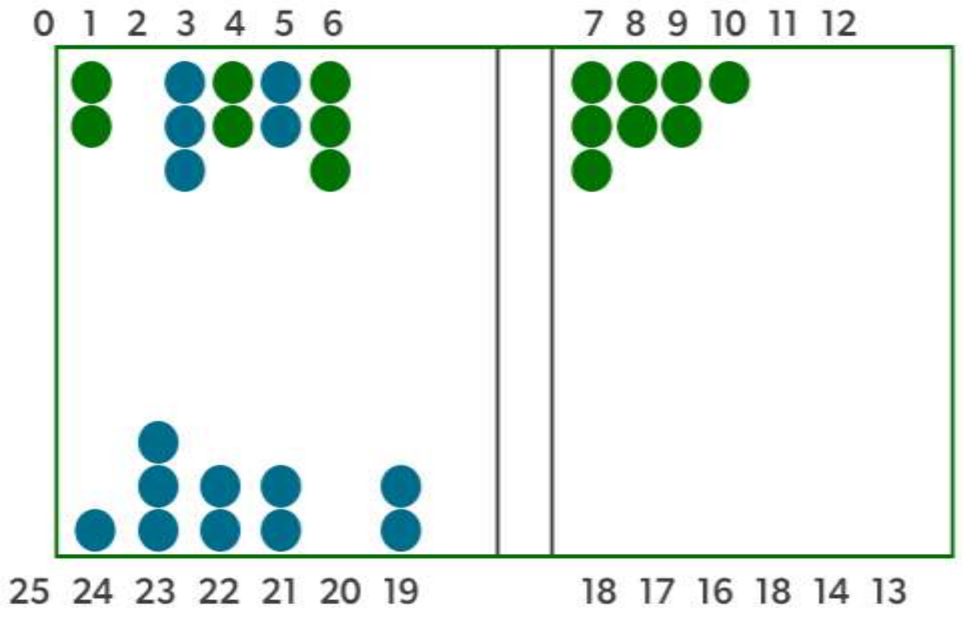
In Backgammon, the primary objective is to get all of one's pieces off the board as quickly as possible. **White** moves in a clockwise direction toward position 25, while **Black** moves counterclockwise toward position 0. If a piece lands on a spot occupied by only one opposing piece, it can be captured, forcing the opponent's piece to start over.

For example, imagine **White** has rolled a **6–5** in the scenario depicted below. White can choose between four valid moves:

1. (5–10, 5–11)
2. (5–11, 19–24)
3. (5–10, 10–16)
4. (5–11, 11–16)

In the move notation (5–11, 11–16), the first number indicates moving one piece from position 5 to 11, and the second part indicates moving another piece from 11 to 16.

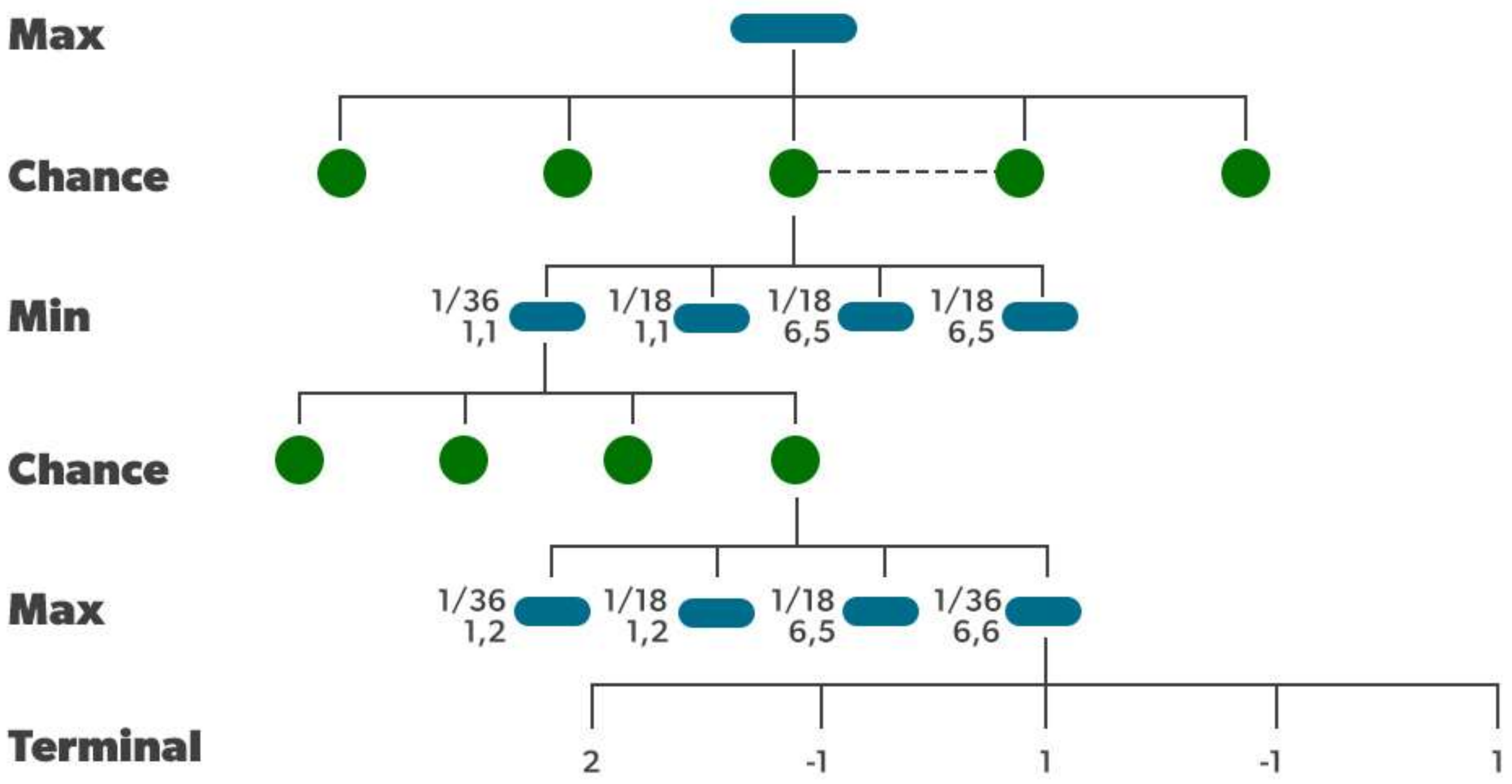
STOCHASTIC GAMES



STOCHASTIC GAMES

- **Stochastic Game Tree for Backgammon**
- In Backgammon, while White knows their own legal moves, they have no idea what dice roll Black will get, which affects Black's legal moves. As a result, White cannot build a simple game tree like those used in deterministic games such as chess or tic-tac-toe. Instead, a game tree in Backgammon must include **chance nodes** that represent the outcomes of random events, like dice rolls.
- These chance nodes are typically depicted as circles in the game tree. The possible dice rolls are represented by the branches leading from each chance node, and each branch is labeled with the corresponding dice roll and its probability.
- There are 36 possible combinations of rolling two dice, as each die can result in six possible outcomes. However, there are only 21 distinct rolls, as a roll of 6–5 is the same as 5–6. The probability of rolling doubles (e.g., 1–1, 2–2, etc.) is **1/36**, while the probability of rolling any other combination (like 6–5 or 5–6) is **1/18**

STOCHASTIC GAMES



STOCHASTIC GAMES

Expectiminimax Algorithm

The **expectiminimax algorithm** is a generalization of the minimax algorithm used for stochastic games. It takes into account not only the decisions of two competing players (MAX and MIN), but also the influence of random events (CHANCE).

The algorithm proceeds as follows:

EXPECTIMINIMAX (s)={ UTILITY (s) if TERMINAL-TEST (s)maxa EXPECTIMINIMAX (RESULT(s,a)) if PLAYER (s)=MAX min EXPECTIMINIMAX(RESULT(s,a)) if PLAYER(s)=MIN∑rP(r)EXPECTIMINIMAX(RESULT(s,r)) if PLAYER(s)=CHANCE

EXPECTIMINIMAX(s) =
UTILITY(s, MAX) if IS-TERMINAL(s)
max a EXPECTIMINIMAX(RESULT(s, a)) if TO-MOVE(s)= MAX
min a EXPECTIMINIMAX(RESULT(s, a)) if TO-MOVE(s)= MIN
∑r P(r)EXPECTIMINIMAX(RESULT(s,r)) if TO-MOVE(s)= CHANCE

STOCHASTIC GAMES

Applications of Stochastic Games in AI

Stochastic games are widely used in AI, especially in environments where decision-making under uncertainty is critical. Below are some key applications:

1. **Autonomous Systems:**
2. **Financial Markets**
3. **Healthcare**
4. **Security Systems:**

STOCHASTIC GAMES

THANK YOU