Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai
Artificial Intelligence and Data Science Department

Subject/Odd Sem 2023-23/Experiment 3

Name: Subrato Tapaswi	Class/Roll No.: D16AD/60	Grade:

Title of Experiment: mixed-strategy Nash equilibria

Objective of Experiment: To analyze Nash equilibria in different types of games

Outcome of Experiment: Analyze and identify Nash equilibria in various game scenarios.

Problem Statement: Program to find mixed-strategy Nash equilibria for zero sum Game (i.e., matching pennies)

Description / Theory:
The Matching Pennies game is a classic two-player zero-sum game in which players simultaneously choose between "Heads" or "Tails." If the players' choices match (both select "Heads" or both select "Tails"), one player wins a point from the other. If their choices do not match, the other player wins a point. The game is designed to illustrate strategic interactions and mixed strategies in decision-making.

Hypothesis:
In the Matching Pennies game, we hypothesize that participants will initially exhibit mixed strategies, meaning they will randomly choose "Heads" or "Tails" to confuse their opponent. However, over time, some participants may recognize the Nash equilibrium strategy of randomizing their choices with a specific probability distribution, leading to an eventual convergence to a Mixed Strategy Nash Equilibrium.

Experimental Design:

- Participants: We will recruit a group of participants and inform them about the rules of the Matching Pennies game and the concept of Mixed Strategy Nash Equilibrium.

Subject/Odd Sem 2023-23/Experiment 3

- Game Setup: Participants will be paired into two-player groups and will play the Matching Pennies game for multiple rounds. They will be told that they can choose "Heads" or "Tails" on each round, and their goal is to accumulate points.

- Iterations: The experiment will consist of several rounds (e.g., 15). In each round, participants will simultaneously choose "Heads" or "Tails" without knowing their opponent's choice until the end of the round.

- Feedback: After each round, participants will receive feedback on their choices and the outcomes, including whether they won or lost points. They will also be informed about the concept of Mixed Strategy Nash Equilibrium.

- Mixed Strategies: Participants will be encouraged to consider adopting a mixed strategy, i.e., choosing "Heads" with a certain probability and "Tails" with the complementary probability, in order to optimize their performance.

- Data Collection: We will record participants' choices in each round and whether they adopt a mixed strategy or exhibit randomness in their choices

Subject/Odd Sem 2023-23/Experiment 3

Program:

1.

```
def find_mixed_nash_equilibrium(payoff_matrix):
    # Create the objective function for player 1
    c1 = [-1 for _ in range(len(payoff_matrix))]

    # Create the constraints for player 1
    A1 = np.transpose(payoff_matrix)
    b1 = [1 for _ in range(len(payoff_matrix[0]))]

    # Solve the linear program for player 1
    result1 = linprog(c1, A_ub=A1, b_ub=b1)

    # Player 1's mixed strategy
    player1_strategy = result1.x

    # Create the objective function for player 2
    c2 = [1 for _ in range(len(payoff_matrix[0]))]

    # Create the constraints for player 2
    A2 = payoff_matrix
    b2 = [-1 for _ in range(len(payoff_matrix))]

    # Solve the linear program for player 2
    result2 = linprog(c2, A_ub=A2, b_ub=b2)

    # Player 2's mixed strategy
    player2_strategy = result2.x

    return player1_strategy, player2_strategy

# Main program
if __name__ == "__main__":
    print("Welcome to the Mixed-Strategy Nash Equilibrium Finder for Zero-Sum Games!")

    # Define the payoff matrix (replace this with your own matrix)
    payoff_matrix = np.array([[1, -1],
                              [-1, 1]])

    print("Payoff Matrix:")
    print(payoff_matrix)

    player1_strategy, player2_strategy = find_mixed_nash_equilibrium(payoff_matrix)

    print("Mixed-Strategy Nash Equilibrium:")
    print(f"Player 1's mixed strategy: {player1_strategy}")
    print(f"Player 2's mixed strategy: {player2_strategy}")
```

	Output Screenshots:
	`Welcome to the Mixed-Strategy Nash Equilibrium Finder for Zero-Sum Games!`
	`Payoff Matrix:`
	`[[ 1 -1]`
	`[-1 1]]`
	`Mixed-Strategy Nash Equilibrium:`
	`Player 1's mixed strategy: None`
	`Player 2's mixed strategy: None`

Results and Discussions: In conclusion, the implementation of this experiment provides valuable insights into how individuals adapt their decision-making strategies in repeated strategic interactions. Understanding the emergence of Mixed Strategy Nash Equilibrium can have implications for various real-world scenarios, including economics, politics, and competitive situations, where strategic randomness may evolve into more structured strategies for optimal outcomes.