Artificial intelligence CSC309 (group 10)

Algorithmic Analysis of the Prisoner's Dilemma

1.Introduction

The Prisoner's Dilemma is a game theory problem that involves two players making independent decisions to cooperate or defect. This section presents an algorithmic approach that outlines the logical steps used to analyse player strategies, identify the Nash equilibrium, and determine Pareto optimal outcomes.

2. Problem Description

The Prisoner's Dilemma involves two players, referred to as Prisoner A and Prisoner B. Each player must make a decision without knowing the choice of the other. The outcome of the game depends on the combination of decisions made by both players, and each outcome leads to specific payoffs for the players.

Although cooperation leads to a better collective outcome, individual rationality often leads players to defect.

3. Players and Strategies

- Player A: Prisoner A
- Player B: Prisoner B

Each player has two possible strategies:

- Cooperate (C)
- Defect (D)

The strategies are chosen simultaneously and independently.

4. Payoff Structure and Game Schema

The game is represented using a payoff matrix (game schema), which maps each possible combination of strategies to payoffs for both players.

- If both players cooperate (C, C), both receive a reward payoff.
- If both players defect (D, D), both receive a punishment payoff.
- If one player defects while the other cooperates (D, C) or (C, D), the defector receives a higher payoff, while the cooperator receives the lowest payoff.

The payoff values follow the condition:

Temptation > Reward > Punishment > Sucker's Payoff

This structure encourages defection as the dominant strategy for each player.

Explicit Payoff Matrix (Example)

To make the analysis more precise, the following numeric payoff values are used as an example:

- Temptation to defect (T) = 5
- Reward for mutual cooperation (R) = 3
- Punishment for mutual defection (P) = 1
- Sucker's payoff (S) = 0

These values satisfy the condition T > R > P > S.

Using these values, the payoff matrix is shown below:

	Prisoner B:cooperate	**Prisoner B: Defect**
Prisoner A: Cooperate	(3, 3)	(0, 5)
Prisoner A: Defect	(5, 0)	(1, 1)

5. Algorithm (Step-by-Step Procedure)

1. Identify the two players involved in the game.
2. Define the available strategies for each player: Cooperate or Defect.
3. Construct the payoff matrix representing all possible strategy combinations.
4. Allow each player to independently select a strategy.
5. Observe the combination of strategies chosen by both players.
6. Determine the payoff for each player using the payoff matrix.
7. For each player, check whether changing their strategy alone would improve their payoff.
8. Identify the outcome where no player can improve their payoff by changing strategy unilaterally (Nash equilibrium).
9. Compare all outcomes to determine whether any outcome makes both players better off without making either player worse off.
10. Identify such outcomes as Pareto optimal.
11. Record and summarise the results of the analysis.

ALGORITHM AnalysePrisonersDilemma (Pseudocode implementation)

INPUT:

Players A and B

Strategies = {Cooperate, Defect}

PayoffMatrix:

- (C, C) → (3, 3)
- (C, D) → (0, 5)
- (D, C) → (5, 0)
- (D, D) → (1, 1)

OUTPUT:

NashEquilibria

ParetoOptimalOutcomes

Algorithm

```
BEGIN

   Outcomes ← empty list

   FOR each strategyA IN Strategies DO

      FOR each strategyB IN Strategies DO

         payoffA, payoffB ← PayoffMatrix(strategyA, strategyB)

         Outcomes ← Outcomes ∪ { (strategyA, strategyB, payoffA, payoffB) }

      END FOR

   END FOR

   NashEquilibria ← empty list

   FOR each outcome o IN Outcomes DO

      strategyA, strategyB, payoffA, payoffB ← o

      isNash ← true

      FOR each altA IN Strategies DO

         IF altA ≠ strategyA THEN

            altPayoffA, _ ← PayoffMatrix(altA, strategyB)
```

```
            IF altPayoffA > payoffA THEN
                isNash ← false
            END IF
        END IF
    END FOR
    FOR each altB IN Strategies DO
        IF altB ≠ strategyB THEN
            _, altPayoffB ← PayoffMatrix(strategyA, altB)
            IF altPayoffB > payoffB THEN
                isNash ← false
            END IF
        END IF
    END FOR
    IF isNash = true THEN
        NashEquilibria ← NashEquilibria ∪ { o }
    END IF
END FOR
ParetoOptimalOutcomes ← empty list
FOR each outcome o1 IN Outcomes DO
    strategyA1, strategyB1, payoffA1, payoffB1 ← o1
    dominated ← false
    FOR each outcome o2 IN Outcomes DO
        strategyA2, strategyB2, payoffA2, payoffB2 ← o2
        IF (payoffA2 ≥ payoffA1 AND payoffB2 ≥ payoffB1) AND
           (payoffA2 > payoffA1 OR payoffB2 > payoffB1) THEN
```

```
          dominated ← true
        END IF
      END FOR
      IF dominated = false THEN
        ParetoOptimalOutcomes ← ParetoOptimalOutcomes ∪ { o }
      END IF
    END FOR
    OUTPUT NashEquilibria
    OUTPUT ParetoOptimalOutcomes
END
```

6. Identification of Nash Equilibrium

A Nash equilibrium occurs when each player's strategy is the best response to the strategy chosen by the other player. Using the algorithm, mutual defection (Defect, Defect) is identified as the Nash equilibrium, since neither player can increase their payoff by changing strategy alone.

7. Pareto Optimality Analysis

An outcome is Pareto optimal if no other outcome can make one player better off without making the other worse off. In the Prisoner's Dilemma, mutual cooperation (Cooperate, Cooperate) is Pareto optimal because both players receive higher payoffs compared to mutual defection. However, this outcome is unstable under rational self-interest.

8. Discussion

The algorithm highlights the conflict between individual rationality and collective benefit. While defection leads to a stable equilibrium, cooperation provides a more efficient outcome for both players. This explains why rational decision-making does not always lead to socially optimal results.