Computational Game Theory

2nd Tournament

Faculty of Sciences and Technology of
New University of Lisbon

Integrated Master of Computer Science and Engineering

2nd Semester
2019/2020

Professor/Evaluators:
- **João Alexandre Leite**
- **jleite@fct.unl.pt**

Author/Student:
- **Rúben André Barreiro**
- **r.barreiro@campus.fct.unl.pt**
- **42648**

1st

Game of 2nd Tournament

Prisoner's Dilemma with 20 Fixed Number of Rounds and 100% of probability of continue to the next round

This game was basically a repetition of the 1st game of the 1st Tournament, i.e., play the Prisoner's Dilemma with 20 Fixes Number of Rounds.

In this Game, it was repeated a similar approach used in the last tournament, and used following Strategy:

- Grim Trigger, defecting in the last 4 rounds;

The most known and commonly used Strategies used in this Game are the following:

- Tit For Tat;
- Grim Trigger;
- Gradual;

Believing that most of the Adversaries will use a similar Strategy to the mentioned before, it was believed that Grim Trigger should fit well, in these occasions.

This Strategy works well against Tit For Tat or other Grim Trigger Strategies, trying to Cooperate as most as possible, delaying the Defect action until a 1st Defect move be detected.

1st

Game of 2nd Tournament

Prisoner's Dilemma with 20 Fixed Number of Rounds and 100% of probability of continue to the next round

It was opted to no use some sense of Forgiving, this time, due to the small and "limited" number of rounds (i.e., only 20 rounds).

And, accordingly to the experience from the last tournament, from some adversaries opted for Defect, in the last 2 or 3 rounds, it was opted to play a Defect action, voluntarily, earlier than only in the last round, most specifically since, the last 4 rounds.

This Strategy also, works always well against random strategies, because, after the 1st Defect action be detected, plays like All Defect, from then, which it's the best strategy against random agents, in this game, guaranteeing always an outcome payoff of 4 or 1.

If (GAIN_DEFECT > DET_LOSS_ DEFECT), play a Defect action:

- **GAIN_DEFECT = (I_DEFECT_OPPONENT_COOPERATE - BOTH_COOPERATE)**
- **DET_LOSS_DEFECT = (BOTH_COOPERATE - BOTH_DEFECT) × (#ROUND_LEFT – 4)**
- **I_DEFECT_OPPONENT_COOPERATE = 4**
- **BOTH_COOPERATE = 3**
- **BOTH_DEFECT = 1**

2nd

Game of 2nd Tournament

Prisoner's Dilemma with Unknown Number of Iterations, higher than 1 – Unknown Probability of Continue to the next Game Stage/Round, less than 100%

- In this game, it was used a similar Strategy to the one used in the 1st Game, but considering other aspects for this game, specifically:
  - The probability of continue to the next round/iteration;

- Considering that, it was used a little modification, regarding the moment of being worthy to play a Defect action, using the following computation:
  - The probability of continue to the next round/iteration;

if (GAIN_DEFECT > PROB_LOSS_ DEFECT), play a Defect action:

- **GAIN_DEFECT = (I_DEFECT_OPPONENT_COOPERATE - BOTH_COOPERATE)**
- **PROB_LOSS_DEFECT = ((BOTH_COOPERATE - BOTH_DEFECT) × P(NEXT_ROUND)) ÷ (1 - P(NEXT_ROUND))**
- **I_DEFECT_OPPONENT_COOPERATE = 4**
- **BOTH_COOPERATE = 3**
- **BOTH_DEFECT = 1**

3rd

Game of 2nd Tournament

Unknown Game with
200 Number of Iterations and
Probability of Continue to the next Game Stage/Round of 100%

- During the course, it was studied many possible approaches to use in Strategies for the Normal Form Games, enumerated as the following:
  - Bayesian Games' Strategies;
  - Learning Algorithms (e.g., No-Regret Learning and Fictitious Play Learning);
  - Nash Equilibrium for Pure and Mixed Strategies (in Zero-Sum Games and General-Sum Games);
  - Iterated Removal of Strictly Dominated Strategies;
  - MaxMin/MinMax Strategy;

- In this type of Games, we are dealing with uncertainty and randomness, which makes a little hard to design a good Strategy, that cover all the cases;

3rd

Game of 2nd Tournament

Unknown Game with 200 Number of Iterations and Probability of Continue to the next Game Stage/Round of 100%

- So, from the previously mentioned approaches, it was possible to "discard" some of them, because it is reasonable to think that was not appropriated for this game, in specifically, like the enumerate next:
  - **Bayesian Games' Strategies:**
    - Because, this game it was not known at all, and this approach only makes sense, when it is known that will be played with some uncertainty, one game from a group of several ones, for sure;
  - **MaxMin/MinMax Strategy:**
    - Because, in this game, and assuming that all the players are reasonable, no player it's supposed to be acting in order to harm others and the main idea it's maximize the most the utilities and payoffs, so it's useless to use Strategies in order to maximize the utilities and payoffs, in the worst scenario, or try to harm our opponents, because that don't guarantee the highest utilities and payoffs;
- However, it was considered the approach of using a MinMax, in the case of the Game being of Zero-Sum type, in order, to guarantee the highest payoffs (i.e., the symmetric ones), in the Opponent's worst scenario.

3rd

Game of 2nd Tournament

Unknown Game with 200 Number of Iterations and Probability of Continue to the next Game Stage/Round of 100%

- In the case of the Game be a General-Sum Game, it was used the remaining mentioned approaches.
- In the first round, it was opted to remove all the Strictly Dominated Strategies, because it will not guarantee the highest payoffs and utilities for sure and tried to play the a Strategy that guarantees the highest payoffs and utilities, based in the global sums for each one of the not dominated Strategies, using distributed probabilities, in order to play a Mixed Strategy based on that, as a Best Response's Strategy notion.
- After the first round, it was used a Learning Algorithm/Method, based on Fictitious Play, using distributed probabilities for Mixed Strategies, based on the frequency of the actions played by the Opponent. For each new round, it was opted to compute and update the best response's "history", from the 1st round until the current one, and update the same, taking in consideration the moves played by the opponent.

3rd

Game of 2nd Tournament

Unknown Game with 200 Number of Iterations and Probability of Continue to the next Game Stage/Round of 100%

- It was opted also to do not use the Nash Equilibrium notion, because it does not guarantee always the best possible payoffs/utilities, and can lead to some undesirable outcomes, in some cases.
- The Nash Equilibrium only guarantees that none of the players involved have any incentive to deviate from an outcome cell, which it is not properly the same of, guarantee the best possible solution to all the possible games.
- The possibility of updating the best responses in larger intervals, instead, in some cases, could have been a better approach for this case, in specific, because from a round to another, the opponent's actions can change a lot, proving "unstable" learning processes. This approach was not considered, unfortunately.

GitHubs' Repository:

- **https://github.com/rubenandrebarreiro/Computational-Game-Theory-NOVA-FCT-2020/tree/2ndTournament**