NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CZ4046
Intelligent Agents

Assignment 2 Report

By: TEO WEI JIE (U1822263C)

18 April 2021

Table of Contents

1. Agent Design

My agent's design is largely based on the rules for success as suggested by Robert Axelrod, which was discussed in the lecture of module 8:

1. Don't be envious - don't play as if it were zero sum
2. Be nice - start by cooperating, and reciprocate cooperation
3. Retaliate appropriately - always punish defection immediately, but use ”measured force”
4. Don't hold grudges - always reciprocate cooperation immediately

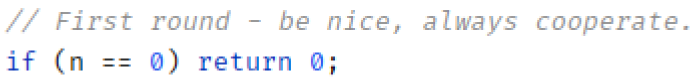
However, there are slight deviations from the rules, which will be discussed in the sections below.

1.1. Starting round - always cooperate

My agent will always cooperate for the first round of each game, following the rule to “be nice”. Assuming that every other agent chooses to follow the suggested rules, where cooperation is reciprocated and is never the first to defect, starting the game by cooperating would ensure that every agent cooperates throughout the game. This guarantees a high payoff at the end of the game, given the following payoff ordering for each round:

- U(DCC) > **U(CCC)** > U(DDC) > U(CDC) > U(DDD) > U(CDD)

While it may be tempting to start off the game by defecting, thus ensuring the highest payoff for the first round assuming every other agent chooses to cooperate for the first round, the defection is very likely to be punished in subsequent rounds and thus resulting in much lower payoff in the long run. As such, my agent will not be envious and always start by cooperating. The code snippet for the starting round is shown in Figure 1 below.



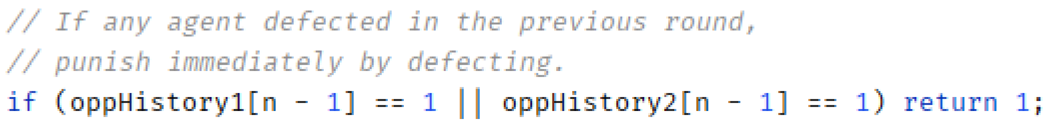
```
// First round - be nice, always cooperate.
if (n == 0) return 0;
```

Figure 1: Code snippet for the starting round decision

1.2. Subsequent rounds

Subsequently, for each round, my agent's decision is based off the following policies:

1. If any other agent has defected in the previous round, punish by defecting (retaliate appropriately). The code snippet for this decision is shown in Figure 2 below.
2. If no other agent defected in the previous round, calculates if the other agents are trustworthy and act accordingly.

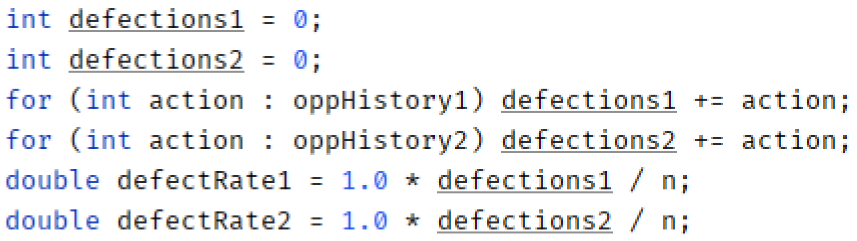


```
// If any agent defected in the previous round,
// punish immediately by defecting.
if (oppHistory1[n - 1] == 1 || oppHistory2[n - 1] == 1) return 1;
```

Figure 2: Code snippet for punishing defection in the previous round

1.2.1. Calculating trustworthiness

While the rules suggest that my agent should not hold grudges and should reciprocate cooperation, an untrustworthy agent is more than likely to defect and sabotage the game for everyone else even if said agent has cooperated in the immediate past round. The trustworthiness of every other agent is calculated based on their defection rate across all previous rounds (round 0 to n where n = current round). The code snippet for calculating the defection rate is shown in Figure 3 below.



```
int defections1 = 0;
int defections2 = 0;
for (int action : oppHistory1) defections1 += action;
for (int action : oppHistory2) defections2 += action;
double defectRate1 = 1.0 * defections1 / n;
double defectRate2 = 1.0 * defections2 / n;
```

Figure 3: Code snippet for calculating defection rate based on the history

There are three stages of trustworthiness, and are defined as such:

1. Trustworthy - defection rate is no more than 5% (<5%)

2. Neutral - defection rate is least 5% but not more than 20% (>= 5% & <20%)

3. Untrustworthy - defection rate is least 20% (>= 20%)

The following decisions are made based on the trustworthiness of the other agents:

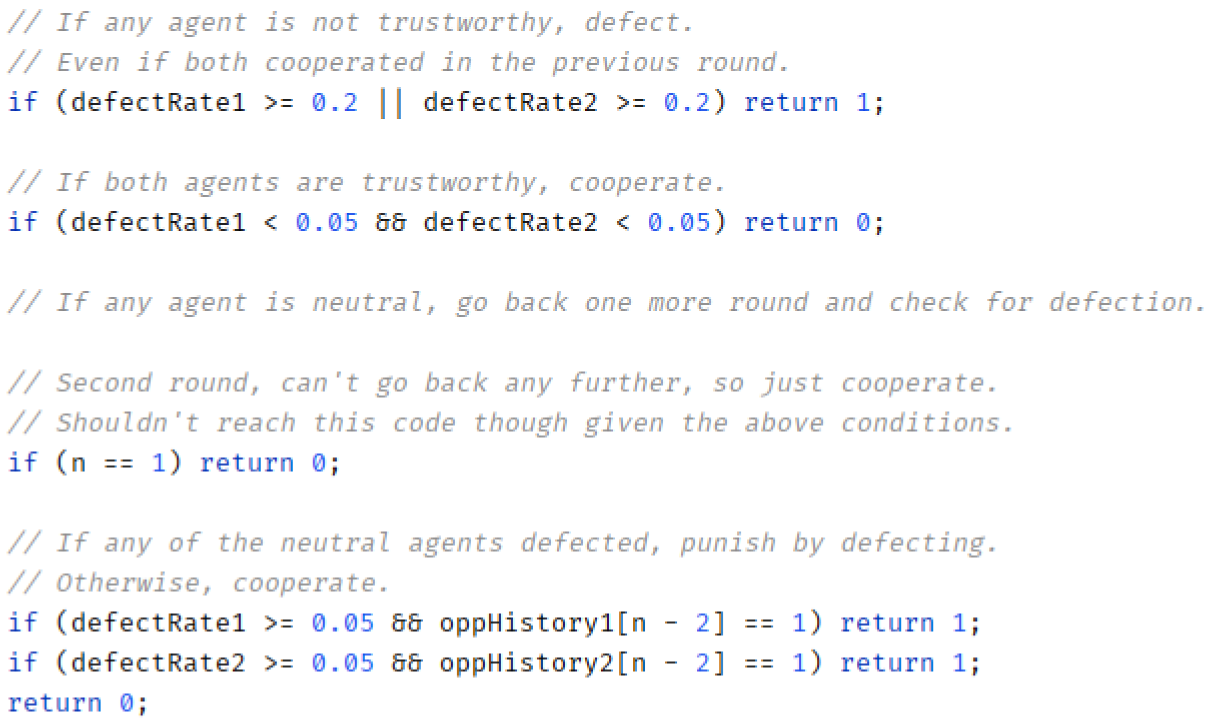
1. If any other agent is untrustworthy, my agent will defect for the coming round.

2. If at least one other agent is neutral, my agent will check if any neutral agent defected in the second previous round (round = n - 2). If no neutral agents defected, my agent will cooperate. Otherwise, it will defect.

3. If both agents are trustworthy, my agent will take the chance and cooperate as the other agents are more likely going to cooperate, which would maximise payoff in the long run (similar to the reasoning for the first round).

The implementation of the trustworthiness check also has the effect of allowing an opponent agent to ”redeem” itself. If an agent defected frequently, thus achieving a defection rate of >= 20%, but subsequently cooperates frequently to drive the defection rate below 20%, my agent will begin cooperating with it again if that agent maintains its cooperation.

The code snippet for the above decision making process is given in Figure 4 below.



```
// If any agent is not trustworthy, defect.
// Even if both cooperated in the previous round.
if (defectRate1 >= 0.2 || defectRate2 >= 0.2) return 1;

// If both agents are trustworthy, cooperate.
if (defectRate1 < 0.05 && defectRate2 < 0.05) return 0;

// If any agent is neutral, go back one more round and check for defection.

// Second round, can't go back any further, so just cooperate.
// Shouldn't reach this code though given the above conditions.
if (n == 1) return 0;

// If any of the neutral agents defected, punish by defecting.
// Otherwise, cooperate.
if (defectRate1 >= 0.05 && oppHistory1[n - 2] == 1) return 1;
if (defectRate2 >= 0.05 && oppHistory2[n - 2] == 1) return 1;
return 0;
```

Figure 4: Code snippet for decision making based on opponent agents' trustworthiness

1.2.2. Towards the end of a game

While the exact number of rounds played in a game is random, it is known that about 100 rounds will be played. As such, it is tempting to attempt a gamble and defect towards the end of a game to potentially maximise payoff (i.e. defect at a probability when number of rounds has exceeded 100). However, my agent holds true to the rule to not be envious and as such will continue to make decisions based on the policies discussed in section 1.2.

2. Evaluation of the designed agent

Tournaments were conducted between my agent and the provided example strategy agents to evaluate the effectiveness of the my design. There are two phases to this evaluation:

1. Tournaments were conducted against individual strategies (i.e. iterations between my agent and nice agent only).
2. Finale tournament conducted with every agent participating.

2.1. Individual tournaments

Each tournament is iterated over 10000 times in order to obtain an accurate evaluation. For example, the tournament where my agent plays the nice agent is conducted 10000 times, and the next tournament where my agent plays the nasty agent is conducted 10000 times, so on and so forth. The scores across the 10000 tournaments were then averaged.

The results of the individual tournaments are provided in Table 1 below.

Table 1: Results for the individual tournaments, each conducted 10000 times

Opponent agent	**Opponent score**	**My score**
NicePlayer	36.0	36.0
NastyPlayer	12.120427	23.999992
RandomPlayer	18.202343	30.014446
TolerantPlayer	36.0	36.0
FreakyPlayer	22.682024	30.559456
T4TPlayer	36.0	36.0

The tournaments held against the nice agent, tolerant agent, and T4T agent yielded the same score of 36.0 as both my agent and the opponent agent cooperated throughout every game. There is no defection as all the agents will never be the first to defect, thus guaranteeing cooperation.

Playing against the nasty agent, which will always defect, my agent scored badly as mutual defection yields low payoff. Additionally, as my agent will cooperate in the first round of each game, my agent will always score lower than the nasty agent for that game. However, my agent will pull ahead when playing against itself due to mutual cooperation, which yields a much higher payoff. This is illustrated via the results of one tournament game against the nasty agent in Table 2 below.

Table 2: Results of one tournament game against the nasty agent, each row depicting one round in the game

Agent 1	**Score**	**Agent 2**	**Score**	**Agent 3**	**Score**
NastyPlayer	2.0	NastyPlayer	2.0	NastyPlayer	2.0
NastyPlayer	2.029703	NastyPlayer	2.029703	My_Player	1.980198
NastyPlayer	2.0560749	My_Player	2.0093458	My_Player	2.0093458
My_Player	6.0	My_Player	6.0	My_Player	6.0

Similar observations were made when playing against the random agent and freaky agent. Whenever the random or freaky agents decides to defect, the payoff for that agent and mine will be low. However, my agent will similarly pull ahead when playing against itself due to mutual cooperation. Concluding from the results, my agent fared better overall when playing against agents that may be the first to defect.

2.2. Finale tournament

The finale tournament is conducted 10000 times similarly, and the average scores for each agent are provided in Table 3 below.

Table 3: Results for the finale tournament conducted 10000 times

Agent	**Score**
My_Player	169.40918
TolerantPlayer	164.63144
T4TPlayer	163.61363
NicePlayer	151.6017
FreakyPlayer	146.07184
NastyPlayer	140.37393
RandomPlayer	133.82329

It was observed that my agent was able to outperform every other agent in the finale tournament, and this can be attributed to the fact that my agent has incorporated the rules to succeed as discussed in chapter 1. Overall, it can be concluded that cooperation is the key to success, as every agent that will reciprocate cooperation and not be the first to defect has outperformed agents that may initiate a defection on their own, as evident from the results in Table 3.