



Available online at <http://scik.org>

J. Math. Comput. Sci. 11 (2021), No. 4, 4650-4667

<https://doi.org/10.28919/jmcs/5916>

ISSN: 1927-5307

PREDICTING MOOD BEHAVIORS OF PRISONER'S DILEMMA PLAYERS

SALSABEEL M. ABD EL-SALAM¹, ESSAM EL-SEIDY², NAGLAA M. REDA^{2,*}

¹Department of Computer Science, Ain Shams University, Cairo 11856, Egypt

²Department of Mathematics, Ain Shams University, Cairo 11856, Egypt

Copyright © 2021 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: Over the last few years, attention goes to the model of Three-Player Prisoner's Dilemma Iterated Game. In this work we are concerned with studying the competition between strategies with special behavior in this model. These strategies have behavior similar to Tit for Tat strategy. We focus on strategies with angry behavior and strategies with tolerant behavior. The results reveal the qualities of the winning strategy. From our studies we can judge the social behavior that must be enjoyed by the person who follows the winning strategy. A learning algorithm has been designed and a multi-agent system has been developed to obtain our results.

Keywords: game theory; iterated multiplayer dilemmas; decision making; behavioral prediction; reinforcement machine learning; parallel algorithms.

2010 AMS Subject Classification: 91A10, 62C25, 68T05, 68W10.

1. INTRODUCTION

Game theory is a toolbox made to help understanding phenomena which is involved in the analysis of strategies between the players. Models of game theory represent our daily life [1, 2]. Game theory has great impact not only on mathematics and physics [3] but also on public health

*Corresponding author

E-mail address: naglaa_reda@sci.asu.edu.eg

Received April 23, 2021

[4], business [5], economics [6], biology [7], political science [8], engineering [9], and computer science [10].

In game theory, there are many examples of coordination games such as: Prisoner's Dilemma (PD), Hawk–Dove, matching pennies, location game, etc. PD is one famous example of game theory complex decision-making process. Each player allows two choices either defect (D) or cooperate (C). The players play simultaneously without the knowledge of the other player's choice. On the other hand, iterated PD (IPD) is the study of long-term behaviors which emerges cooperation between selfish and non-related competitors to coexist in the long term. At every repetition (round), each player's choice may depend on the outcome of the previous round. This leads to developing players' strategies according to their interactions in the previous game [11].

Earlier, Nowak et al. [12] worked in different strategies on two prisoner's dilemma and expressed them by finite state automata. The attacking players showed each other that the best strategy was that of Pavlov. Recently, Essam and Karim [13] have studied Tit-for-Tat (TFT) strategy in three prisoner's dilemmas and four strategies exhibits the same TFT's behavior.

Yet, Maurice and Emilian [14] were interested in investigating the role of anticipation on cooperation rates and payoffs. They reproduced several characteristics of individual play, and tested their model using multi-agent simulations of small societies. Whereas, Shiheng and Fangzhen [15] studied the invincible strategies, inspecting which strategies are not invaded by other strategies. Their results have shown that the strategy is a catalyst for cooperation and winning face to face.

Furthermore, Bukowski and Miękisz [16] classified the three-player games with two strategies. They proved that there are many evolutionary stable strategies (ESS), and there are two pure strategies as in the stage hunt game. Whilst, Dominik, Jacek and Marcin [17] provided one pure and one mixed strategy and focused on the problem of equilibrium selection. They considered this problem from dynamical view. They discussed stochastic adaptation dynamics in the three-player game.

On the other hand, Anurag and Deepak [18] studied IPD problem from the aspect of zero-sum games. A zero-sum game is a mathematical representation of the case where the profits and the

losses are exactly balanced. They explore how artificial intelligence and machine learning techniques work exceedingly to give the agent the capability of recognizing the player intention.

In addition, Harper et al [19] proved that pre-trained strategies, using reinforcement learning techniques, generally outperforms human-designed strategies, and maximizes payoff versus varied series of opponents. They also showed that trained strategies using history of multiple rounds play outperforms memory-one strategies. While, Konstantinos [20] created an AI agent that uses reinforcement learning in order to discern an optimal strategy to the two-player iterated prisoner's Dilemma game using the Soar cognitive architecture.

In this article, we are interested in the iterated three-player model of the prisoner's dilemma game. We first introduce our problem and the basic concepts of machine learning needed for this study. then, we start a theoretical discussion to detect moody players. Then, we propose new algorithm for predicting the best strategy to win the game. In the following section, experimental results followed by numerical analysis is provided. The article ends with a conclusion section with some possible future work.

2. THE PROBLEM VS. MACHINE LEARNING

In this section, we summarize the fundamentals of the iterated Three-Player Prisoner's Dilemma game (3P-IPD). Then, we give a concise definition of reinforcement machine learning type and indicate how it is related to IPD game.

2.1. 3P-IPD PROBLEM

In [13], researchers give great attention to prisoner's dilemma game involving three players in two population game. They suppose that two players harmonize between them to cooperate/defect together against the third player. Thus, a player's current choice may affect the opposite players' future behavior and payoffs. Subsequently, to maximize the received payoffs, players use more complex strategies depending on the game history. More precisely, each player in 3P-IPD round chooses his/her move based on the previous one saved in memory. Each round outcome lies between the eight probabilities for the three players to alternate between C and D. Because the game is symmetric, the outcomes are reduced to six combinations specified by the payoffs {P, L, T, S, K, R}, governed by the rule: $S < P < K < L < R < T$. This leads to sixty-four strategies denoted by

During our work, we follow the direct approach in [15]. The run of the above eight cases produces two conceivable regimes, say Rg_1 and Rg_2 . Exceptional perturbation in Rg_1 gives rise to probability $(3/30) R$ to Rg_1 , and probability $(9K+9L+9P)/30$ to Rg_2 .

Therefore, the transition matrix becomes $\begin{pmatrix} \frac{8}{9} & \frac{1}{9} \\ 1 & 0 \end{pmatrix}$, and the stationary distribution vector equals $\left(\frac{1}{10}, \frac{9}{10}\right)$.

In general, for all possible combinations, there are 64^3 payoff values for the three strategies S_i, S_j, S_k of players I, II, III where, $i, j, k = 0 \dots 63$.

2.2. REINFORCEMENT LEARNING

Artificial Intelligent (AI) is a computer science field that enables computer machines to simulate human processes intelligently. Machine learning is a subfield of AI that can be defined as general purpose techniques to learn functional relationships from data without being explicitly programmed. In the AI-literature, there are three different machine learning types: Reinforcement Learning (RL), supervised learning and unsupervised learning. In RL, algorithms learn to act in an anonymous environment for maximizing some cumulative scalar payoff. One popular type of RL is the Temporal-Difference (TD) learning, where an agent makes the adjustment after each action based on an estimated expected payoff. TD-learning employs either Q-Learning or SARSA-Learning. In Q-Learning, the algorithm executes an action according to his strategy for the current state and receives the resulting payoff, then, the agent chooses the most promising action from the successor state. While in SARSA-Learning, the agent sticks to his strategy when calculating his next action. RL approach have been employed in the game theoretic literature since 1993. Recently, machine learning has been paid notable attention in algorithmic game theory field. Quite a number of important contributions about different types of learning in games have been introduced by researchers [20, 21].

3. MOODY PLAYERS' DETECTION

In the competition of artificial intelligence algorithms, the TFT model was the optimum solution for the prisoners' dilemma for many decades. Its main idea is to select the strategy of the next round depending on the basis of the selection from the previous round [22]. In this section, we concentrate our study on specific strategies picked depending on their behavior. The state machine diagram of every studied strategy is given in Fig. 1, classified into two groups. In our studies we also use two strategies S_{63} and S_0 to compute the selected strategies.

3.1. TEMPERED PLAYERS

In this subsection, we select strategies: S_{32} , S_{33} , S_{36} and S_{38} in regard to TFT for tempered players. Our selection is based on their attitude that we observe during tracing the game in Fig. 1A. We study their choices and perceive the following:

1. **S₃₂**: Fast tempered and Intolerant, it still plays C only if it's two opponents Play C together, otherwise, it plays D forever.
2. **S₃₃**: Fast tempered and Jade, it still plays C only if it's two opponents plays C or D, otherwise, it moves from the state C to state D.
3. **S₃₆**: Fast tempered and Tolerant, it still plays C only if its two opponents play C, unlike that it moves from the state C to the state D if at least one of the other opponents plays D.
4. **S₃₈**: Fast tempered and Quick tolerant, it plays C if the two opponents play C, while it moves from state C to state D if at least one of them plays D. In addition, it plays D if the two opponents play D otherwise it moves to state C.

3.2. NATURAL-TEMPERED PLAYERS

In this subsection, we pick out strategies: S_{48} , S_{49} , S_{52} and S_{54} in regard to TFT for natural tempered players. we choose these strategies precisely depending on what we notice throughout running the game in Fig. 1B. We study their moves and recognize the following:

1. **S48:** Fast tempered and Natural-tempered, it still plays C only if it's two opponents Play C together or one of them plays C and the other plays D, otherwise, that it plays D forever.
2. **S49:** Natural anger and Jade, it still plays C only if it's two opponents plays C or D, in addition one of them plays C and the other plays D, otherwise, that it moves from the state C to state D.
3. **S52:** Natural tempered and Tolerant, it still plays C only if its two opponents play C, in addition one of them plays C and the other plays D, while it moves from the state C to the state D if two opponents play D.
4. **S54:** Natural anger and Quick tolerant, it plays D if the two opponents play D, unlike that he plays C forever.

4. PROPOSED PREDICTION ALGORITHM

Our objective was to design strategic algorithm by interacting computer science with game theory, guided by algorithmic game theory of the previous work. We propose an intelligent algorithm to predict who shall win the game by which selected strategy according to its behavior. Our algorithm considers all possible scenarios and detect the most optimal strategy that fits the mood of participants. In the following explanation, the pseudocode of the proposed algorithm and the game simulator.

Our *Predict-Best* proposed learning algorithm which predicts the strategy that is expected to invade others. First, we declare the function **Construct** for constructing the stochastic vector $X = (x_0, x_1, x_2, x_3, x_4, x_5)$, for each distribution vector in the given set **DV**, guided by its corresponding regime in the set or regimes **Rg**. For invalid pays, **Construct** assigns the X coefficients to zeros. Second, we compute the average payoff values of the first agent against the third when fixing the second using equation (1) for the selected set of strategies **Stg**, and store them in a square two-dimensional matrix P.

$$\frac{x_0 P + x_1 L + x_2 T + x_3 S + x_4 K + x_5 R}{x_0 + x_1 + x_2 + x_3 + x_4 + x_5} \quad (1)$$

Third, we compare the evaluated payoffs for detecting the maximum long term expected payoff

and return its strategy index as the best. If the two agents are equipollent, then the algorithm returns the value -1.

Actually, we apply this algorithm on the different set of candidate strategies, presented above, chosen according to the agents' mood, then we conclude invaded strategies between variant competitors for the rest of the game.

Game-Sim algorithm $(x_0, x_1, x_2, x_3, x_4, x_5)$, 3P-IPD as a multi-agent system, given the number of strategies, the number of rounds, the number of players, and the TFT learning matrix. We considered all N^3 mutations for 3 participants' strategies. We adapt the BCD representation when referring to each strategy index. We also define a mapping between the decimal index of the round and the players' choices taken of the next round. For each three players' strategies p_1, p_2, p_3 , we inspect the players' choices for R rounds using the TFT learning mapping, calculate the P probable payoff vectors, identify the regimes Reg , deduce the transition matrix, and produce the final payoff distribution vector $DVec_{p_1, p_2, p_3}$. To simplify the pseudocode, we defined the following functions.

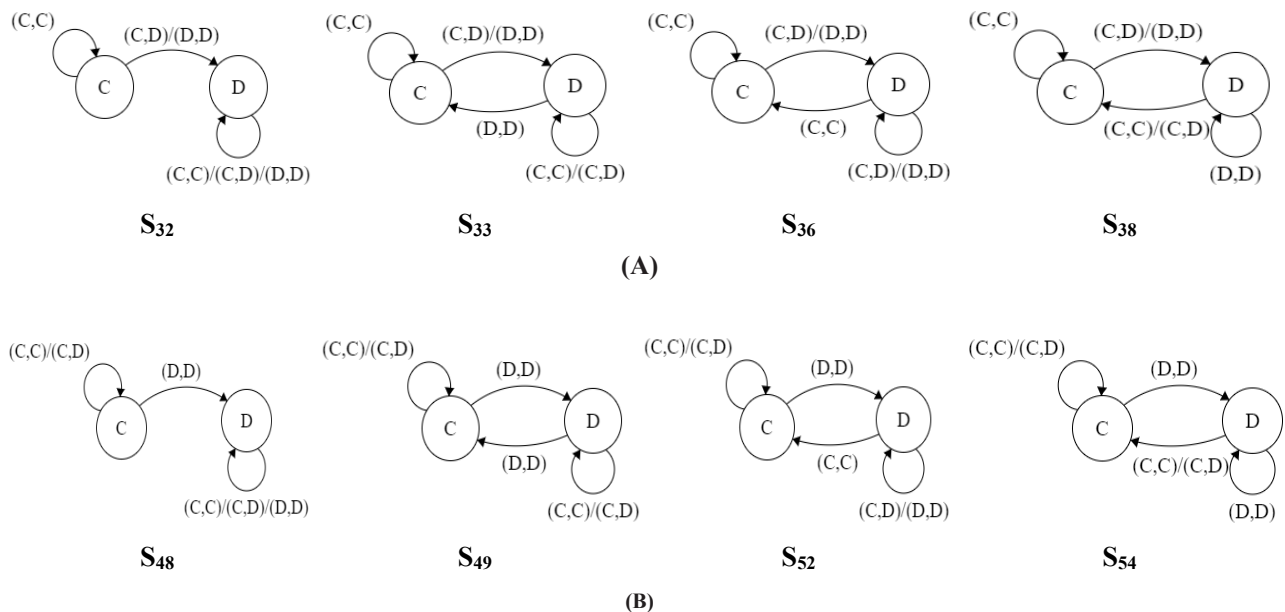


Figure 1. The state machine diagram of the studied strategies

Algorithm Predict-Best:**Input:** Set of selected strategies **Stg**, Set of the distribution vectors **DV**, Set of the regimes **Rg****Output:** The invading strategy for each combination **Best****Initialization:** $\text{Best} \leftarrow -1$, $\text{Pays} \leftarrow [1, 5, 9, 0, 3, 7]$

```

1: for i from 0 to 3 do
2:   set  $X_i \leftarrow \text{Construct}(\text{Rg}_i, \text{DV}_i)$  ;
3:   set  $Y_i \leftarrow 0$  ;
4:   set  $Z_i \leftarrow 0$  ;
5:   for j from 0 to 5 do
6:     set  $Y_i \leftarrow Y_i + X_i[j] * \text{pays}[j]$  ;
7:     set  $Z_i \leftarrow Z_i + X_i[j]$  ;
8:   end for ;
9:   if  $i < 2$  then
10:    set  $P[0, i] \leftarrow Y_i / Z_i$  ;
11:    else set  $P[1, i \bmod 2] \leftarrow Y_i / Z_i$  ;
12:   end for ;
13:   if  $(P[0,0] > P[1,0] \ \&\& \ P[0,1] > P[1,1]) \parallel (P[0,0] > P[0,1] \ \&\& \ P[1,0] > P[1,1])$  then
14:     set  $\text{Best} \leftarrow \text{Stg}[0]$ 
15:   else if  $(P[1,0] > P[0,0] \ \&\& \ P[1,1] > P[0,1]) \parallel (P[0,1] > P[0,0] \ \&\& \ P[1,1] > P[1,0])$  then
16:     set  $\text{Best} \leftarrow \text{Stg}[1]$ 
17:   end if ;
18: end if ;
19: return  $\text{Best}$  ;

```

- **BCD:** takes the decimal index of each strategy and coded it to its binary representation,
- **Start:** assign the start round's choices for agents by mapping the given index to its corresponding combination of C and D,
- **Decide:** takes the round's number and the strategies' indices then makes the next round intelligent moves prescribed by TFT determined strategies. That is, it chooses the best response to the history up until current round to maximize the long-term profits,
- **Pay:** takes the moves of all rounds and computes the payoff vector for each probability by matching the BCD value of each round with its corresponding index at the **Poff** vector,
- **Identify:** takes the payoff vectors and recognize the regime of every one by distinguishing the most stable attitude, that is, the most repeated subsequence of payoff values,

PREDICTING MOOD BEHAVIORS OF PRISONER'S DILEMMA PLAYERS

- **Transmit:** takes the set of regimes then construct the transition matrix regarding the current strategies by testing the perturbation of each agent and identifying all probabilities of changing the regime,
- **ComputeDV:** takes the transition matrix as the coefficients of a liner system of equations and solve it to get the final distribution vector for each three agents.

Algorithm *Game-Sim:*

Input: Number of strategies N , Number of rounds R , Number of players P , Learning matrix TFT

Output: The distribution vectors for all possible combinations of three players' strategies:

$$DVec_{0,0,0}, \dots, DVec_{N-1,N-1,N-1}$$

Initialization: $Stg \leftarrow \{0,1,2, \dots, N-1\}$, $Poff \leftarrow ['P', 'L', 'T', 'S', 'K', 'R']$, $Pr \leftarrow \{1, 2, \dots, 2^P\}$,

$$Rs \leftarrow \{1, 2, \dots, R\}$$

```

1: for each  $p_1 \in Stg$  do
2:   for each  $p_2 \in Stg$  do
3:     parallel for each  $p_3 \in Stg$  do
4:       set all_stg  $\leftarrow [ BCD(p_1), BCD(p_2), BCD(p_3) ]$ ;
5:       for each  $j \in Pr$  do
6:         Start ( rounds  $j$  [1] );
7:         for each  $k \in Rs - \{R\}$  do
8:           set rounds  $j$  [ $K+1$ ]  $\leftarrow Decide ( TFT, all\_stg, K )$ ;
9:         end for ;
10:        parallel for each  $k \in Rs$  do
11:          set  $PV_K \leftarrow Pay ( rounds_j, Poff, K )$ ;
12:        end parallel for ;
13:        set Reg  $\leftarrow Identify ( PV )$ ;
14:        end for ;
15:        set TM  $\leftarrow Transmit ( all\_stg, Reg )$ ;
16:        set  $DVec_{p_1, p_2, p_3} \leftarrow ComputeDV ( all\_stg, TM )$ ;
17:      end parallel for ;
18:    end for ;
19:  end for ;
20: return DVec ;

```

5. EXPERIMENTAL RESULTS

Our work has been coded and experimentally tested using MATLAB (matrix laboratory). We choose MATLAB because it is an excellent multi-paradigm numerical computing environment and proprietary programming language for our problem, especially when dealing with matrices. For-loop iterations were implemented in parallel to improve the execution time.

For testing, we develop the whole game system which creates individual players as intelligent agents artificially. We consider all possible combinations of probabilistic TFT computer strategies. Each three strategies were competed for 1000 round (interaction game session) because the game yields stable income results in the long-term. The system evaluates the transition matrix and the payoff vector for the 262144 simulations. For each agent, the sequences of last choices and targets from the previous session were kept as experience to serve as initials in the following session. Also, payoffs were saved to help in determining the regimes.

In the experiment, the parameters compensated by $S=0$, $P=1$, $k=3$, $L=5$, $R=7$, $T=9$, satisfying the condition $S < P < K < L < R < T$. Each strategy has been mapped to the binary code of its index. The following tables present our results. Each tuple $(x_0, x_1, x_2, x_3, x_4, x_5)$ is the stochastic vector for the first player asserting that he/she will get a payoff value below, such that each row denotes the first player strategy, and each column denotes the third player strategy.

$$\frac{x_0P + x_1L + x_2T + x_3S + x_4K + x_5R}{x_0 + x_1 + x_2 + x_3 + x_4 + x_5} \quad (2)$$

The strategies outcompeted S_i every column except the first column in the Table 13 and Table 14 refers to the strategies competed the strategies in the first column. Each pure strategy invaded the other strategies. S_j invaded S_i if we calculated the payoff for each player in matrix by Nash equilibrium as equation (2).

CONCLUSION AND FUTURE WORK

In this paper, we studied the behavior of the strategies in three prisoner's dilemmas. we studied the outcompeting between the players. In Table 13, we conclude that the jade (s_{38}) and quick tolerant players (s_{33}) outcompeted the largest number of strategy when the second players are

PREDICTING MOOD BEHAVIORS OF PRISONER'S DILEMMA PLAYERS

intolerant (s_{32}) and (s_{63}). The intolerant, the jade and (s_0) can't be invited by any player when the second player is the jade (s_{33}). The tolerant and (s_0) can't be invited by any player when the second player is (s_0). In Table 14, we conclude that the jade player (s_{49}) outcompeted at least three players. The jade player was outcompeted by (s_0) when the second player (s_{49}, s_{63}, s_0) and outcompeted by s_{48} when the second player (s_{63}). We notice that S_{33} and S_{49} are the best player against Tit-for-Tat in many cases.

Table 1. The payoffs for player I when player II is using S_{32}

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)
S_{33}	(1,0,0,1,0,0)	(3,0,0,0,3,1)	(1,0,0,1,0,0)	(1,1,0,1,0,0)	(0,1,0,0,0,0)	(1,0,0,1,0,0)
S_{36}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)
S_{38}	(1,0,0,0,0,0)	(1,1,0,1,0,0)	(1,0,0,0,0,0)	(2,1,0,1,0,0)	(0,3,0,0,3,1)	(1,0,0,0,0,0)
S_{63}	(0,0,0,1,0,0)	(0,0,0,1,0,0)	(0,0,0,1,0,0)	(0,0,0,3,3,1)	(0,0,0,0,3,1)	(0,0,0,1,0,0)
S_0	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)

Table 2. The payoffs for player I when player II is using S_{33}

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(1,1,0,0,0,0)	(3,0,3,0,0,1)	(1,1,0,0,0,0)	(1,1,0,0,0,0)	(0,1,0,0,0,0)	(1,1,0,0,0,0)
S_{33}	(3,0,0,0,3,1)	(0,0,0,0,0,1)	(3,3,0,0,3,1)	(3,3,0,0,3,1)	(0,3,0,0,0,1)	(1,0,0,0,1,0)
S_{36}	(1,1,0,0,0,0)	(3,0,3,3,0,1)	(1,1,0,0,0,0)	(1,1,0,0,0,0)	(0,1,0,0,0,0)	(1,1,0,0,0,0)
S_{38}	(1,1,0,1,0,0)	(3,0,3,3,0,1)	(1,1,0,1,0,0)	(3,8,0,5,3,1)	(0,3,0,0,3,1)	(1,1,0,1,0,0)
S_{63}	(0,0,0,1,0,0)	(0,0,0,3,0,1)	(0,0,0,1,0,0)	(0,0,0,3,3,1)	(0,0,0,0,3,1)	(0,0,0,1,0,0)
S_0	(1,1,0,0,0,0)	(1,0,1,0,0,0)	(1,1,0,0,0,0)	(1,1,0,0,0,0)	(0,1,0,0,0,0)	(1,1,0,0,0,0)

Table 3. The payoffs for player I when player II is using S_{36}

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)
S_{33}	(1,0,0,1,0,0)	(3,3,0,0,3,1)	(1,0,0,1,0,0)	(1,1,0,1,0,3)	(0,1,0,0,0,0)	(1,0,0,1,0,0)
S_{36}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,7,3,0,3,2)	(1,0,0,0,0,0)
S_{38}	(1,0,0,0,0,0)	(1,1,0,1,0,0)	(1,0,0,0,0,0)	(5,9,0,6,3,1)	(0,0,3,0,3,2)	(1,0,0,0,0,0)
S_{63}	(0,0,0,1,0,0)	(0,0,0,1,0,0)	(0,0,0,7,6,2)	(0,0,0,0,3,1)	(0,0,0,0,0,1)	(0,0,0,1,0,0)
S_0	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)

Table 4. The payoffs for player I when player II is using S_{38}

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(0,3,3,0,0,1)	(1,0,0,0,0,0)
S_{33}	(1,1,0,1,0,0)	(3,3,0,0,3,1)	(1,1,0,1,0,0)	(3,10,3,3,0,1)	(0,3,3,0,0,1)	(1,1,0,1,0,0)
S_{36}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(5,12,3,3,0,1)	(0,0,3,0,3,2)	(1,0,0,0,0,0)
S_{38}	(2,1,0,1,0,0)	(3,8,0,5,3,1)	(5,9,0,6,3,1)	(1,2,1,1,2,1)	(0,0,1,0,1,2)	(2,1,0,1,0,0)
S_{63}	(0,0,0,3,3,1)	(0,0,0,1,1,2)	(0,0,0,0,3,1)	(0,0,0,0,1,1)	(0,0,0,0,0,1)	(0,0,0,1,1,0)
S_0	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(0,1,1,0,0,0)	(1,0,0,0,0,0)

Table 5. The payoffs for player I when player II is using S_{63}

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(0,1,0,0,0,0)	(0,1,0,0,0,0)	(0,1,0,0,0,0)	(0,3,3,0,0,1)	(0,0,3,0,0,1)	(0,1,0,0,0,0)
S_{33}	(0,1,0,0,0,0)	(0,3,0,0,0,1)	(0,1,0,0,0,0)	(0,3,3,0,0,1)	(0,0,3,0,0,1)	(0,1,0,0,0,0)
S_{36}	(0,1,0,0,0,0)	(0,1,0,0,0,0)	(0,7,3,0,3,2)	(0,0,3,0,3,2)	(0,0,0,0,0,1)	(0,1,0,0,0,0)
S_{38}	(0,3,0,0,3,1)	(0,3,0,0,3,1)	(0,0,3,0,3,2)	(0,0,1,0,1,2)	(0,0,0,0,0,1)	(0,1,0,0,1,0)
S_{63}	(0,0,0,0,3,1)	(0,0,0,0,3,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,1,0)
S_0	(0,1,0,0,0,0)	(0,1,0,0,0,0)	(0,1,0,0,0,0)	(0,1,1,0,0,0)	(0,0,1,0,0,0)	(0,1,0,0,0,0)

PREDICTING MOOD BEHAVIORS OF PRISONER'S DILEMMA PLAYERS

Table 6. The payoffs for player I when player II is using S_0

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
S_{32}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)
S_{33}	(1,0,0,1,0,0)	(1,0,0,0,1,0)	(1,0,0,1,0,0)	(1,1,0,1,0,0)	(0,1,0,0,0,0)	(1,0,0,1,0,0)
S_{36}	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)
S_{38}	(1,0,0,0,0,0)	(1,1,0,1,0,0)	(1,0,0,0,0,0)	(2,1,0,1,0,0)	(0,1,0,0,1,0)	(1,0,0,0,0,0)
S_{63}	(0,0,0,1,0,0)	(0,0,0,1,0,0)	(0,0,0,1,0,0)	(0,0,0,1,1,0)	(0,0,0,0,1,0)	(0,0,0,1,0,0)
S_0	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)

Table 7. The payoffs for player I when player II is using S_{48}

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(1,0,0,0,0,0)	(9,9,4,0,5,3)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,2,1,0,1,1)	(1,0,0,0,0,0)
S_{49}	(9,0,1,9,8,3)	(0,0,1,0,8,3)	(3,0,1,3,2,3)	(6,6,5,6,7,9)	(0,12,7,0,11,9)	(2,0,0,2,1,0)
S_{52}	(1,0,0,0,0,0)	(1,1,0,0,1,1)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,1,2)	(1,0,0,0,0,0)
S_{54}	(1,0,0,0,0,0)	(2,2,0,2,4,3)	(1,0,0,0,0,0)	(2,3,0,3,7,13)	(0,0,0,0,1,1)	(1,0,0,0,0,0)
S_{63}	(0,0,0,2,2,1)	(0,0,0,4,6,3)	(0,0,0,1,1,2)	(0,0,0,0,1,1)	(0,0,0,0,1,1)	(0,0,0,2,1,0)
S_0	(1,0,0,0,0,0)	(2,2,1,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,2,1,0,0,0)	(1,0,0,0,0,0)

Table 8. The payoffs for player I when player II is using S_{49}

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(9,9,4,0,5,3)	(0,0,7,0,2,3)	(3,3,2,0,1,3)	(9,9,7,0,5,9)	(0,12,11,0,7,9)	(2,2,0,0,1,0)
S_{49}	(0,0,1,0,8,3)	(0,0,1,0,2,3)	(0,0,1,0,1,3)	(0,0,1,0,1,3)	(0,4,3,0,3,5)	(0,0,0,0,1,0)
S_{52}	(1,1,0,0,1,1)	(0,0,0,0,2,3)	(1,1,0,0,1,3)	(3,3,0,0,4,10)	(0,2,0,0,3,7)	(2,2,0,0,2,0)
S_{54}	(2,2,0,2,4,3)	(0,0,0,0,2,3)	(1,1,0,1,2,5)	(0,1,0,1,3,5)	(0,0,0,0,1,1)	(2,2,0,2,3,0)
S_{63}	(0,0,0,12,18,9)	(0,0,0,4,6,5)	(0,0,0,2,3,7)	(0,0,0,0,1,1)	(0,0,0,0,1,1)	(0,0,0,1,1,0)
S_0	(4,4,2,0,0,0)	(0,0,1,0,0,0)	(1,1,1,0,0,0)	(1,1,1,0,0,0)	(0,1,1,0,0,0)	(1,1,0,0,0,0)

Table 9. The payoffs for player I when player II is using S_{52}

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(1,0,0,0,0,0)	(3,3,2,0,1,3)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,1,0,0,2)	(1,0,0,0,0,0)
S_{49}	(3,0,1,3,2,3)	(0,0,1,0,1,3)	(1,0,1,1,0,3)	(1,1,2,1,0,5)	(0,2,3,0,0,7)	(1,0,0,1,1,0)
S_{52}	(1,0,0,0,0,0)	(1,1,0,0,1,3)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,0,0,0,0,1)	(1,0,0,0,0,0)
S_{54}	(1,0,0,0,0,0)	(1,1,0,1,2,5)	(1,0,0,0,0,0)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(1,0,0,0,0,0)
S_{63}	(0,0,0,1,1,2)	(0,0,0,2,3,7)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,1,1,0)
S_0	(1,0,0,0,0,0)	(1,1,1,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,1,0,0,0)	(1,0,0,0,0,0)

Table 10. The payoffs for player I when player II is using S_{54}

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(1,0,0,0,0,0)	(9,9,7,0,5,9)	(1,0,0,0,0,0)	(2,6,7,0,0,13)	(0,0,1,0,0,1)	(1,0,0,0,0,0)
S_{49}	(6,6,5,6,7,9)	(0,0,1,0,1,3)	(1,1,2,1,0,5)	(0,2,3,0,0,5)	(0,0,1,0,0,1)	(2,2,0,2,3,0)
S_{52}	(1,0,0,0,0,0)	(3,3,0,0,4,10)	(1,0,0,0,0,0)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(1,0,0,0,0,0)
S_{54}	(2,3,0,3,7,13)	(0,1,0,1,3,5)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(2,3,0,3,7,0)
S_{63}	(0,0,0,0,1,1)	(0,0,0,0,1,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,1,0)
S_0	(1,0,0,0,0,0)	(1,1,1,0,0,0)	(1,0,0,0,0,0)	(2,6,7,0,0,0)	(0,0,1,0,0,0)	(1,0,0,0,0,0)

Table 11. The payoffs for player I when player II is using the strategy S_{63}

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(0,2,1,0,1,1)	(0,12,11,0,7,9)	(0,1,1,0,0,2)	(0,0,1,0,0,1)	(0,0,1,0,0,1)	(0,2,0,0,1,0)
S_{49}	(0,12,7,0,11,9)	(0,4,3,0,3,5)	(0,2,3,0,0,7)	(0,0,1,0,0,1)	(0,0,1,0,0,1)	(0,1,0,0,1,0)
S_{52}	(0,1,0,0,1,2)	(0,2,0,0,3,7)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,1,0,0,1,0)
S_{54}	(0,0,0,0,1,1)	(0,0,0,0,1,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,1,0)
S_{63}	(0,0,0,0,1,1)	(0,0,0,0,1,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,0,1)	(0,0,0,0,1,0)
S_0	(0,2,1,0,0,0)	(0,1,1,0,0,0)	(0,1,1,0,0,0)	(0,0,1,0,0,0)	(0,0,1,0,0,0)	(0,1,0,0,0,0)

PREDICTING MOOD BEHAVIORS OF PRISONER'S DILEMMA PLAYERS

Table 12. The payoffs for player I when player II is using S_0

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
S_{48}	(1,0,0,0,0,0)	(2,2,0,0,1,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,2,0,0,1,0)	(1,0,0,0,0,0)
S_{49}	(2,0,0,2,1,0)	(0,0,0,0,1,0)	(1,0,0,1,1,0)	(2,2,0,2,3,0)	(0,1,0,0,1,0)	(1,0,0,1,0,0)
S_{52}	(1,0,0,0,0,0)	(1,1,0,0,1,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,1,0)	(1,0,0,0,0,0)
S_{54}	(1,0,0,0,0,0)	(2,2,0,2,3,0)	(1,0,0,0,0,0)	(2,3,0,3,7,0)	(0,0,0,0,1,0)	(1,0,0,0,0,0)
S_{63}	(0,0,0,2,1,0)	(0,0,0,1,1,0)	(0,0,0,1,1,0)	(0,0,0,0,1,0)	(0,0,0,0,1,0)	(0,0,0,1,0,0)
S_0	(1,0,0,0,0,0)	(1,1,0,0,0,0)	(1,0,0,0,0,0)	(1,0,0,0,0,0)	(0,1,0,0,0,0)	(1,0,0,0,0,0)

Table 13. List of invading strategies for tempered players

	S_{32}	S_{33}	S_{36}	S_{38}	S_{63}	S_0
$S_{32} \ll$	38	-	38,63,0	33,36,38,63,0	33,36	38
$S_{33} \ll$	32,36,0	-	32,63	63	-	32,36,0
$S_{36} \ll$	38	33	33,38,63	33,63	38	-
$S_{38} \ll$	33,63	32,36,0	63,33	33,36,63	32,33,63,0	63,0
$S_{63} \ll$	32,33,36,0	32,33,36,38,0	-	-	32,33,36,0	32,33,36,0
$S_0 \ll$	38	-	33,38,63	33,38,63	33,36	-

Table 14. List of invading strategies for natural-tempered players

	S_{48}	S_{49}	S_{52}	S_{54}	S_{63}	S_0
$S_{48} \ll$	49,52,54,63,0	49,54	49,54,63	49,54,63	0	49,54
$S_{49} \ll$	-	0	-	-	48,0	0
$S_{52} \ll$	49,54,63	48,49	49,54,63	49	48,49,0	49,54,63
$S_{54} \ll$	63	49,52,63,0	-	49	48,49	63
$S_{63} \ll$	49,0	48,49,52,0	49	49	48,49	48,49,0
$S_0 \ll$	49,54	-	49,63	49,54,63	54,63	54

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

- [1] R.M. Axelrod, *The evolution of cooperation*, Rev. ed, Basic Books, New York, 2006.
- [2] V. Mazalov, *Mathematical game theory and applications*, John Wiley & Sons, Hoboken, 2014.
- [3] A.J. Edward, M. Douglas, *Econophysics: from game theory and information theory to quantum mechanics*, *Physica A*, 348 (2005), 505-543.
- [4] A. Elzein, K. Drenkard, P. Deyo, E. Swartwout, *The application of gaming theory in health care*, *Nursing Admin. Quart.* 39 (2015), 340–344.
- [5] K. Chatterjee, W. Samuelson, *Game theory and business applications*, Springer, New York, 2014.
- [6] L. Samuelson, *Game theory in economics and beyond*. *J. Econ. Perspect.* 30(4) (2016), 107–130.
- [7] T. Frick, S. Schuster, *An example of the prisoner’s dilemma in biochemistry*, *Naturwissenschaften.* 90 (2003), 327–331.
- [8] N. Mccarty, *Political game theory: An introduction*, Cambridge University Press, Cambridge, 2014.
- [9] S. Mei, W. Wei, F. Liu, *On engineering game theory with its application in power systems*, *Control Theory Technol.* 15 (2017), 1–12.
- [10] K.E. Avrachenkov, A.Yu. Kondratev, V.V. Mazalov, *Cooperative Game Theory Approaches for Network Partitioning*, in: Y. Cao, J. Chen (Eds.), *Computing and Combinatorics*, Springer International Publishing, Cham, 2017: pp. 591–602.
- [11] T. Fujiwara-Greve, *Non-cooperative game theory*, Springer Japan, Tokyo, 2015.
- [12] M.A. Nowak, K. Sigmund, E. El-Seidy, *Automata repeated games and noise*, *J. Math. Biol.* 33 (1995), 703–722.
- [13] E. El-Seidy, K.M. Soliman, *Iterated symmetric three-player prisoner’s dilemma game*, *Appl. Math. Comput.* 282 (2016), 117–127.
- [14] M. Grinberg, E. Lalev, *The role of anticipation on cooperation and coordination in simulated prisoner’s dilemma game playing*, in: G. Pezzulo, M.V. Butz, O. Sigaud, G. Baldassarre (Eds.), *Anticipatory Behavior in Adaptive Learning Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009: pp. 209–228.
- [15] S. Wang, F. Lin, *Invincible strategies of iterated prisoner’s dilemma*, in: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, 2256–2258.

- [16] M.K. Bukowski, J. Miekisz, Evolutionary and asymptotic stability in multi-player games with two strategies, *Int. J. Game Theory*, 33(1) (2004), 41–54.
- [17] D. Kaminski, J. Miekisz, M. Zaborowski, Stochastic stability in three-player games, *Bull. Math. Biol.* 67 (2005), 1195–1205.
- [18] A. Agrawal, D. Jaiswal, When machine learning meets AI and game theory, Stanford University, *Machine Learning Final year Projects* (2012), 221 – 240.
- [19] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N.E. Glynatsi, O. Campbell, Reinforcement learning produces dominant strategies for the Iterated Prisoner's Dilemma, *PLoS ONE*. 12 (2017), e0188046.
- [20] K. Thomas, Building a reinforcement learning A.I. for the iterated prisoner's dilemma using soar cognitive architecture, Dissertation, National and Kapodistrian University of Athens, 2018.
- [21] U. Schwalbe, Algorithms, Machine Learning, and Collusion, WP Institute of Economics, University of Hohenheim, Germany, 2018.
- [22] X. Deng, J. Deng, A study of prisoner's dilemma game model with incomplete information, *Math Probl. Eng.* 2015 (2015), 452042.