Iterated Prisoner's Dilemma - An extended analysis

Andre L. C. Carvalho, Honovan P. Rocha, Felipe T. Amaral and Frederico G. Guimaraes
Graduate Program in Electrical Engineering
Federal University of Minas Gerais
Av. Antonio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
Emails: badaia@cpdee.ufmg.br, honovan@gmail.com, felipetayer@ufmg.br, fredericoguimaraes@ufmg.br

***Abstract*—The motivation of this work is to investigate the classic problem of the Iterated Prisoner's Dilemma in an environment with many players with different behaviors. We review previous analysis of the tournaments proposed by Robert Axelrod, focusing primarily on the strategies adopted by their players. These agents have been recreated in this new tournament, based on the information available, and several others were included, from periodically non-rational agents to agents based on modern paradigms such as neural networks, genetic algorithms, reinforcement learning, trackers, etc. This gives heterogeneous and rich alternatives to the basic tournament. As a result, we verified the classification of these players for various game modes (variation of maximum number of iterations), and analyze their performance based on this criterion.**

I. Introduction

Game theory is a study of strategic decision making to maximize the payoffs in a determined situation, using mathematical models to do that. Initially developed to understand the economy behavior, it came to be used in another areas, in biology, political science, psychology, etc. The game theory gained importance with Jonh Von Neumann and Oskar Morgenstein in the mid-1940, with their publication 'The Theory of games and Economic behavior', that dealt with situations where the best choice depended not only on the agent's own strategy, but also on the combinations of different strategies chosen by all of them. Another important application of game theory is in biological evolution theory, where it was important to understand the evolution and cooperation of various species. Game theory has been widely recognized as an important tool in many fields. Eight game-theorists have won the Nobel Memorial Prize in Economic Sciences, and John Maynard Smith was awarded the Crafoord Prize for his application of game theory to biology [1].

The prisoner's dilemma is a very important example of the game theory, and it can be used to model a lot of situations involving competitive relations. It was originally developed by Merrill Flood and Dresher Melvin in 1950 [2].

The game is based on the response of two prisoners, suspects of a crime. The police does not have enough evidence to arrest them. For a confession, the police separate them and offer both the same deal: if one testifies against the other (defect/betray), and the other remains silent (cooperate/assist), the betrayer goes free and the one that remains silent gets the biggest sentence. If both remain silent, both are sentenced to a minor charge. If both defect together, each receives a medium sentence. Each prisoner must choose either to betray or remain silent; the decision of each is kept secret from the other, until the sentence is announced.

For that game, the best answer, the rational decision, is to defect, because betrayal always rewards more than cooperation, all purely rational self-interested prisoners would betray the other. For this game, the payoff table is shown in Table 1.

Table I. PAYOFF'S TABLE

		Cooperate	Defect
	Cooperate	R - R	S - T
	Defect	T - S	P - P

In the payoff table, R is the reward (reward to cooperation), S is the "Sucker Payoff" (innocent play), T is the temptation (to betray) and P is the punishment (both betray). For the game to make sense, the condition T>R>P>S must be guaranteed, because this payoff relation implies that mutual cooperation is superior to mutual defection, while the payoff relationships T>R and P>S imply that defection is the dominant strategy for both agents. That is, the mutual defection (D,D) is the only strong Nash Equilibrium in the game. Another three states (C,D), (D,C) and (C,C) are Pareto-Optimal outcomes, and the mutual cooperation (C,C) maximizes the social welfare. The Prisoner's Dilemma does not have a dominant strategy.

The objective of this work is to analyze the original Axelrod's Tournament of the IPD and detail the personality and behavior of the agents of this game. After that, a new tournament is proposed, with old and new agents, to extend the analysis under different conditions, such as variations of behavior of the results when the game takes more or less time. In this problem, the ingenuity is very important to explore new solutions and techniques for making better agents, and arguing about their personalities and behavior. The number of possible strategies (and combinations of them), and alternatives is endless. The applications of this game are various. It can be used to solve problems of cooperation, shared resources, decision making, etc. [3], [4]

The remainder of this paper is organized as follows. Section 2 presents the classic Iterated Prisoner Dilemma. Section 3 describes the proposed tournament and the agents used. Section 4 presents the matches representing the experiments performed, followed by results and conclusions in the last two sections.

II. The Iterated Prisoner Dilemma (IPD)

There is also an extended iterative version of the original game, where the classic game is played over and over, and consequently, both prisoners continuously have an opportunity to penalize the other for previous decisions. This game allows the players to achieve mutual gains from cooperation, but it also allows for the possibility that one player will exploit the other, or the possibility that neither will cooperate. It does not have an absolute strategy. In this game, the personality of the player is very important [5].

A. The Axelrod's tournaments

In the late 1970s, Robert Axelrod, an American political scientist and mathematician, started an IPD tournament. He initially requested strategies from other game theorists in economics, psychology, sociology, political science and mathematics to compete in that game. In this tournament, all players play against all (and against itself) in 200 iterations. The payoff values are T=5, R=3, P=1, S=0, and there is no way to be sure what the other player will do on a given move, and no way to eliminate the other player or run away from the interaction. The game score is the sum of each match (player A vs player B) divided by the number of matches. The game was run 5 times (to get more stable estimates of scores for each player) and the final score is the average of the 5 runs. Axelrod received 14 strategies to compete in the tournament (and he included the random player). The winner was the player TitForTat, a very simple strategy developed by Anatol Rapopov. At the end of the tournament, Axelrod analyzed and published the results [6], [7].

After Axelrod published the results of the first tournament, he organized another tournament in 1980, with the same rules, except the duration of the game: instead of 200 iterations, he used random-continuation yielding expected median length of 200 (the probability of continuation was equal to 0.99654). This fact generates 'The Shadow of the Future', an important fact to establish mutual cooperation in future. This game had 62 entrants (plus random), and the winner was TitForTat again [8].

In 1984, Axelrod promoted the third tournament. It was a simulated ecological evolution, in which at the beginning there is a fixed population including the same quantity of each strategy. A round-robin tournament is made and then the population of bad strategies is decreased whereas good strategies obtain new elements. The simulation is repeated until the population has been stabilized (the population does not change anymore). In this way nasty strategies, those who take the initiative of the first defection, have been discovered to be not very stable, because they are invaded by kind ones. He used the same entrances of the second tournament, and run the game for 1000 generations. Also, now each individual in the population has a chance to abandon her old strategy and adopt a different one. The results were amazing: a handful of strategies - all nice - came to dominate the field. For the third time, the player TitForTat won. [9]–[14]

III. The Proposed Tournament

To extend the analysis of this problem, it was implemented a new tournament, with different rules and agents. It is based on the first Axelrod Tournament, but with some differences. The format is the same (all versus all), minus the match vs itself. The maximum number of iterations is variable, (starts with 1 iteration to 1000 iterations) and with 80 different players. We used the framework 'Iterated Prisoner's Dilemma' developed by Zachary Danziger. [15], [16].

A. The agents

The agents are the strategies used in IPD. We can classify them by their personality and extra issues, like memory, learning and adaptation. The personality can be influenced by the last move, memory, probabilities or some random behavior. They are modeled as functions, which receive the following information: Current iteration, total number of iterations and last payoff of opponent (0, 1, 3 or 5). They can choose two answers (actions): Cooperate or Defect. This information is a return of the function. They can choose the action based on theinput information or another ways, such memory (store information of past moves), random choices, probability or they can be indifferent to the opponent's strategy (like periodic agents).

The personalities of the agents are: irresponsible (it does not care with the opponent's strategy), periodic (they change their strategy periodically), polite (altruist, always cooperate, independently of opponent's strategy), selfish (the opposite of polite), believer (always start with cooperation), suspicious (always start with defection), wise (they have a memory of past rounds, and they play using that), nice (always motivate cooperation, and they never defect first), retaliatory (if the opponent defect in the previous round, it defects back), tracker (try to track the opponent's strategy), envious (they always tries to win the match, making more points than opponent), exploiter (try to maximize their payoff exploiting polite agents), bully (defect the opponent until it defects), Pavlov (Win-stay, lose-switch, i. e., if the payoff of this round is bad, changes the strategy), generous (forgives a defection in previous move), responsive (immediately retaliate), profiteer (always defect in the last round of the match), forgiveness (if the opponent returns to cooperate, it cooperates, too), spiteful (every time it is betrayed, it increases the punishment), troublemaker (defects without reason), and more specialized ANN (learn by Artificial Neural Network), RLA (learn by Reinforcement Learning Algorithm) and GA (Genetic Algorithm). [17]–[19]

B. Description of the agents

In this tournament, we have 15 agents from the first Axelrod's tournament (TitForTat (TFT), Tideman and Chieruzzi, Nydegger, Grofman, Schubik, Stein, Friedman, Davis, Graaskamp, Downing, Feld, Joss, Tullock, Withheld and Random), 13 from the second tournament (Champion, Borufsen, Cave, Adams, Graaskamp and Katzen, Weiner, Harrigton, Kluepfel, Leyvraz, Eatherly, Tranquilizer, Tester and TitFor2Tats). These agents are described in [7]–[9], [11], [20].

The 11 periodic agents (AllC, AllD, 95%C, 95%D, CCD, DDC, France, Hardy, Faye, Florencio and Overtime Prime), and 28 other agents (Absentee, Soft-majo, Hard-majo, Anti-TFT, Nasty-TFT, Suspicious-TFT, Generous-TFT, Adaptative-TFT, Diekman, Cautious, Bully, Fair, Golden, Forgetful, Killer, Mensa, Simpleton (Pavlov), Go-by-Majority, Go-by-Minority, Prober, Marilee, Point Seven, Sneaky, SorryExplorer, Sucker-Explorer, Three Strikes, Modified Downing and C-Downing) are found in [5], [19], [21]–[23].

Additionally, 13 new agents are designed and introduced in this work. Their description is given as follows (included implementations of MC and BM [24]):

- Shortmem: This agent was implemented with a short memory of last turns. The decision of the agent depends on the content of his memory. For the first 10 turns, it always cooperate. The opponent answers are stored in the memory. The memory is FIFO and the maximum size of the memory is 10 results. From the tenth round on, the program analyzes the memory, and compare the number of defects and cooperates of the opponent, based in percentage. If cooperation occurs 30% more than defection, it will cooperate. If defection occurs 30% more than cooperation, the program will defect. Otherwise, the program follows the TFT algorithm.
- SelfSteem: It is based on the feeling with the same name, and some additional concepts. It was modeled by a sin curve ($f = sin(2 * pi * n/10)$), which varies with current iteration. The values of this function were divided in areas; for each area, the algorithm behaves differently. If $f > 0.95$, the 'ego' of the algorithm is inflated; it always defect; If $|0.95| > f > |0.3|$, rational behavior; the program follows the TFT algorithm; If $0.3 > f > -0.3$, random behavior; If $f < -0,95$, the algorithm is at the rock bottom; it always cooperate. Furthermore, the algorithms implements a retaliation policy, if the opponent defects; the sin curve is shifted (changing the sin phase).
- Boxer: This agent is a tracker, who uses the first rounds to gain experience, and detects the personality of the opponent. It is an adaptive agent, and it was made to play against the original agents of the game. It implements two memories: one to store the opponent's result, another to store itself. It calculates the behavior of the opponent in the first 10 turns, and takes the best decision to combat it. It starts defecting, and in the rounds 2 and 3, it always cooperate. The next rounds, the agent behaves as a TFT algorithm. This strategy detects the differences between the opponents' personalities in the ten first rounds, and it continues to analyze the results each round (the model of the agent is dynamic) to detect variations in the personality of it. It defects in last round. For this tournament, two agents are participating: Boxer05 (5 positions in its memory) and Boxer10 (10 positions).
- VeryBad: It cooperates in the first three rounds, and uses probability (it implements a memory, which stores the opponent's moves) to decide for cooperating or defecting.
- ANN Agents: They use an artificial neural network trained with Levemberg Marquardt algorithm [25] to generate a model that represents the opponent's behavior. It's supposed that using this model is possible to choose the best strategy to maximize the payoff. To build a representative database to train the ANN is necessary to play against the opponent using any known strategy, so that the partial results obtained after a few iterations are used to generate the database. Considering four matches, the ANN's input vector is composed by the three past responses of the agent and the target value is the opponent's response in the fourth match. After building the ANN's inputs and targets the database was submitted to the ANN to build the forecast model. Three different agents (H1, H2 and H3) were created using the same training but different strategies to generate the databases. The agent H1 is based on the strategy of random responses where there is 50% of probability for each response, that could be 'cooperate' or 'defect'. H2 uses the strategy obtained from the use of a fixed vector of responses, where were generated all possible combinations of the last three responses from the agent, amounting to a total of 8, since there are only two possible responses. The agent H3 is based on the TFT strategy, where 20% of the iterations are used to build the database.
- GADP1: It is based on the genetic algorithm with cooperate behavior (all draws lead to cooperation); it calculates the fitness of each state (CC, CD, DC, DD), using a memory of past moves; the fitness is calculated by the expression $xi * (f(xi) - fm(x))$ (where xi is a proportion of state in memory, f(xi) is a performance of this state and fm(x) is the average points off all states). Also, it implements mutation (5%), to avoid traps. It defects in the last round.
- GADP2: It works like the GADP2, but its behavior is selfish.
- BM: Model proposed by Bush-Mosteller, it is a reinforcement learning algorithm. Its stimulus function is $tanh(\beta(r_t - A_t))$ (where rt is current payoff and A_t is the Aspirant level).
- MC: Model proposed by Macy and Flache, it is a reinforcement learning algorithm. Its stimulus function is $l(r_t - A_t) * max(T - A_t, A_t - S)$ (where r_t is current payoff and A_t is the Aspirant level).
- Stalker: It is a strategy that pursuits a score. It starts with cooperation. It is only moved by the score, it does not see directly the opponent's move, or the current round. Its behavior is based in three scores: the bad score (all rounds in defection), the very good score (all rounds in cooperation), and wish score (the average between the bad and very good score). If the current score is greater than the very good score, it defects. If the current average score is greater than wish score (but less than very good score), it cooperates. If the average current score is greater than 2, it cooperates. If the average current score is between 2 and 1, it defects. Finally, if the average current score is less than 1, it plays randomly. It defects in last round.

IV. THE MATCHES

In this tournament, we played 13 matches, each one with a different maximum number of iterations (1, 3, 5, 7, 10, 15, 20, 50, 70, 100, 200, 500 and 1000). This variation was very important to analyze the quality of each agent according to the number of iterations. Each match was played a given number of times, and the average of them is the final score.

A. *The first game - One iteration*

This game was run 20 times. In the one-shot game, the only important thing is the first move of agents. It is not a good test, but it is important to draw the evolution of the agents. To maximize the payoff of this match, the best choice is to defect. Because of that, the selfish and suspicious strategies earn more points. The first place, the Boxer10, earned 3.93 points. The next four earned 3.92 points, and next six, 3.91 points. The points difference between the first place and twentieth place is only 0.04 points. This game will not be considered in the final analysis. The Top-20 of this game: Boxer10, Stalker, Boxer05, Florencio, VeryBad, Cdowning, Bully, Mensa, AllD, GADP2, ATFT, STFT, Tester, Hardy, GADP1, Cautious, DDC, OvertimePrime and Downing.

B. Five iterations

This game was run 20 times. We can see a domination by non-nice agents (AllD, Mensa and D95), because they earn the temptation points in the first round (in a few-iterations game, you must maximize the payoff, using exploitation). The Boxer10 plays well, but not with its memory (it needs 10 moves to start tracking). Forgetful and GA agents play good, too. The Top-20 result: AllD, Mensa, D95, DDC, ATFT, Hardy, GADP2, Florencio, Forgetful, Killer, Cautious, OvertimePrime, GADP1, Craby, Boxer05, Tester, Boxer10, SuckerExplorer, Feld and Withheld.

C. Seven iterations

This game was run 15 times. Like the previous test, but Mensa passed the AllD, Killer improves its performance and up in the scoreboard. Forgetful grows up too. Many agents have not activated their strategies yet. Top-20 result: Mensa, AllD, Killer, D95, Forgetful, Boxer10, Hardy, Cautious, DDC, Boxer05, Florencio, GADP2, ATFT, Feld, OvertimePrime, GADP1, Tester, SuckerExploerer, Withheld and VeryBad.

D. Ten iterations

This game was run 15 times. The Boxer dominates this round. The non-nice strategies still in first places, but the selfish agents (like AllD, 95D and Mensa) are falling in scoreboard. The GA's and VeryBad also play well. Top-20 result: Boxer10, Boxer05, ATFT, Forgetful, Killer, AllD, Mensa, Faye, G&K, Feld, VeryBad, D95, SuckerExplorer, OvertimePrime, DDC, GADP2, GADP1, Joss, Tranquilizer and Bully.

E. Fifteen iterations

This game was run 15 times. We are arriving to the point where the nice agents start to overcome the selfish ones. The Forgetful beats the non-nice strategies and wins this match. The nice agents are starting to earn more points, and improving their performances. The TFT appears in TOP-20 for the first time. The ANN (artificial neural network) trained by TFT (H3) appears too. Big evolution of G&K, second place. The Tranquilizer gets better too, finishing in fifth place. The Top-20 result: Forgetful, G&K, Boxer10, Boxer05, Tranquilizer, VeryBad, Killer, ATFT, Feld, SuckerExplorer, AllD, Friedman, Borufsen, H3, Golden, Harrington, Mensa, TFT, Graaskamp and Adams.

F. Twenty iterations

This game was run 10 times. The Forgetful wins again. The ANN plays very well and take the third place. The TFT family starts to appears in TOP20. However, the AllD leaves it. G&K remained in its position. VeryBad and Boxer continued among the first players. The TFT and variations started to appear in the rank. The Top-20: Forgetful, G&K, H3, Tranquilizer, Boxer10, VeryBad, Boxer05, ATFT, ThreeStrikes, Borufsen, Friedman, Killer, NTFT, TFT, Harrington, Adams, Golden, Graaskamp, Weiner and Diekman.

G. Fifty iterations

This game was run 10 times. This game was very similar to the previous one. The Forgetful continued to lead, but H3 and G&K change places. The Tranquilizer lost its place leaving the TOP20. The Top-20 result: Forgetful, H3, G&K, ThreeStrikes, Stein, Shubik, Golden, VeryBad, Boxer10, Davis, Borufsen, Adams, Weiner, Graaskamp, NTFT, Friedman, TFT, Boxer05, Diekman and GotoMajority.

H. Seventy iterations

This game was run 10 times. Little changes in the game in relation to the previous game. ThreeStrikes goes to third place, passing G&K. The game was dominated by nice strategies (Forgetful, ThreeStrikes, Stein and G&K), and the ANN (H3). The Top-20: Forgetful, H3, ThreeStrikes, G&K, Stein, Shubik, Davis, Golden, Boxer10, VeryBad, NTFT, Boxer05, Soft-majo, Weiner, TFT, Adams, Borufsen, GotoMajority, Diekman and Friedman.

I. One hundred iterations

This game was run 5 times. This is a moment of stability in the game, little changes happened; H3 assumes the leadership and Shubik changes the place with Stein. The Top-20: H3, Forgetful, ThreeStrikes, G&K, Shubik, Stein, Davis, Boxer10, Boxer05, VeryBad, Soft-majo, Golden, GotoMajority, Diekman, NTFT, Friedman, Adams, Weiner, Borufsen and Hard-majo.

J. Two hundred iterations

This game was run 5 times. The TOP5 is the same of the previous game; some changes between 6-20 positions; the moment of stability continues. Top-20: H3, Forgetful, ThreeStrikes, G&K, Shubik, Davis, Stein, VeryBad, GotoMajority, Soft-majo, Boxer10, Boxer05, Golden, Diekman, Hard-majo, PointSeven, Weiner, Friedman, Borufsen and TFT.

K. Five hundred iterations

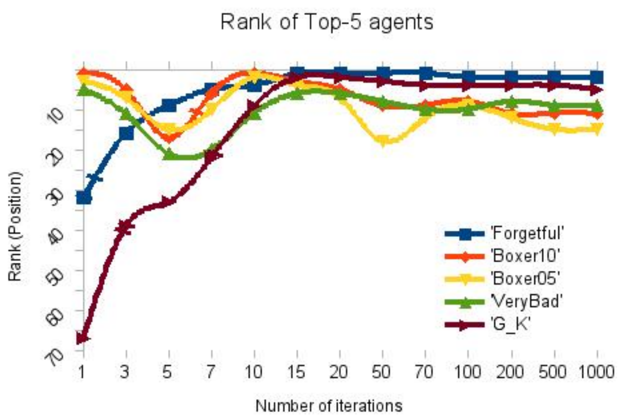
This game was run 3 times. In TOP5, only one difference: Davis and Shubik change places. In exception of agent Boxer, all strategies of the TOP20 are nice. Results (20 first): H3, Forgetful, ThreeStrikes, G&K, Davis, Shubik, Stein, GotoMajority, VeryBad, Soft-majo, Boxer10, Diekman, Golden, Hard-majo, Boxer05, Champion, PointSeven, Leyvraz, Borufsen and Friedman.

L. One thousand iterations

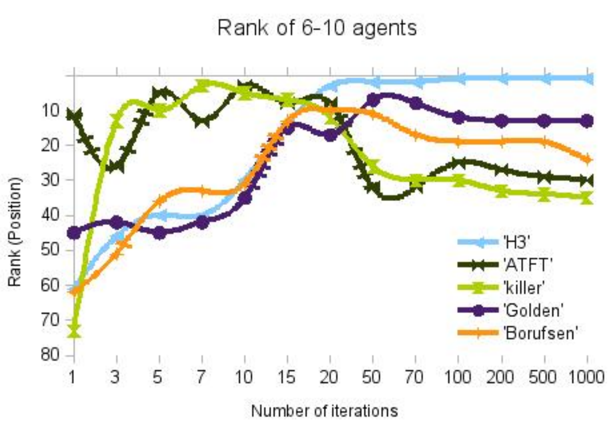
This game was run 2 times. The Davis passed G&K. The first three are the same. Like the last match, the only non- nice agents in TOP20 are the Boxer's. In this game, the difference between the first and the last place is 721 points. The H3 grows up quickly to leadership. In it fight against the Forgetful, it passes after 70 iterations. After the 20 iterations'game, the nice agents start to dominate the game. The Top-20: H3, Forgetful, ThreeStrikes, Davis, G&K, Shubik, Stein, Soft-majo, VeyBad, GotoMajority, Boxer10, Diekman, Golden, Hard-majo, Boxer05, Leyvraz, Champion, PointSeven, TFT and Weiner.

V. RESULTS

Figure 1 shows the rank of general best agents:



(a)



(b)

Figure 1. Rank of top Agents.

To select the best agents in all games, we calculate the average position for all strategies. The best agent in all games was Forgetful. It plays better in games with big number of iterations, but it plays well in short matches. The best non-nice agent in this game was the agent Boxer10 (second place in quality).

Figure 2 shows the performance (in all games) of some important agents: The winners (Forgetful, Boxer10, H3, AllD and Mensa), the worst (AllC), TFT (very important agent, and winner of Axelrod's tournaments) and Random (59th in general rank), who delimits the quality of an agent (if an agent earned less points than random, the agent is not good).

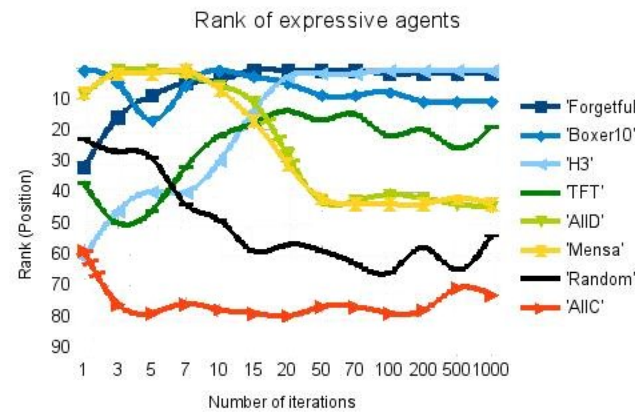


Figure 2. Rank of Expressive Agents.

If we analyze the result by average payoff per iteration, the nice agents improved this characteristic to win the matches (using cooperation) and the selfish agents loss points with the increase in the number of iterations. The best value is 5 (temptation), but improbable; the best value to pursuit is 3 (mutual cooperation in all matches). The mutual cooperation permits to maximize the payoff and improves performance.

Another interesting fact is the performance of agent TFT. Although TFT wins the first tournament, in this new one it plays just fair. Its best classification was in tenth to fifth place. The best non-nice agent in the game was Boxer10, which stayed in Top-20 in all matches. The ANN's (Artificial Neural Network) played the game very well. The agent H3 won 4 games (100, 200, 500 and 1000 iterations); probably, it would win other games with more iterations. The agents H1 and H2 played badly and didn't reach the TOP20 in any game (in general classification, H1 reached the 65th place and H2 reached the 73rd, both worse than the random agent). The difference of ANN's happens because the form of learning: the H3 uses TFT, who is better than another forms (random and a pre-defined sequence). The H3 learned how to win very quickly.

In sequence, we analyze the GA's (genetic algorithms). They had a very bad campaign, presenting a performance very next to random (in general rank, GADP1 reaches the 39th place, and GADP2, the 46th). To improve their performance, we need to adjust the fitness calculus and try other alternatives in the GA's possibilities. Like as GA's, RLA's (Reinforcement Learning Algorithm) had a terrible campaign, too (BM reaches the 55th place, and MF, the 37th). To improve their performance, adjustments in the learning weights must be made. The best nice player in the game was the Forgetful (1st place) ; the best non-nice, Boxer10 (2nd place); the best participant of the first Axelrod tournament, TFT (14th place); The best participant of the second Axelrod tournament and the best TFT-variant, G&K (5th place). The worst agents (reached less points than random strategy): Cave, Nydegger, CCD, GTFT, WithHeld, H1, Craby, Kluepfel, GotoMinority, Tideman, France, Eathley, SorryExplorer, H2, Downing, Tullock, ShortMem, C95, AdTFT, Absentee and AllC.

VI. Conclusions

This paper has reported a new tournament simulation and analysis involving intelligent agents for the iterated prisoner's dilemma. We can see the cooperating agents improving their performance with time, showing additional evidence that cooperation can indeed arise in a society of selfish agents when the number of interactions with each other increase. This competition was more difficult than the First Axelrod Tournament, because of the greater number of entrances and more variety of personalities of agents. The ANN's H3 dominated the tournament after the game of 100 iterations. The Forgetful is a more complete version of TFT; it is nice and generous. It detects arguing (C-D sequences), and restores the peace by cooperating. It plays well against a lot of strategies. The third place Boxer10, is a non-nice one but it tries to track the opponent using a memory of all the last moves (the only difference between Boxer05 and Boxer10 is the number of iterations used to learn the opponent's strategy); the VeryBad, 4th place, is an agent based on probability, using a memory of opponent's moves. The G&K is a nice strategy, and plays TFT most of the time. It only changes its strategies to selfish if its score is lower than expected.

The objective of this work was to analyze the original Axelrod's Tournament of IPD and detail the personality and behavior of the agents of this game. After that, it was proposed another tournament, with old and new agents, to extend the analysis under different conditions, such as variations of behavior of the results when the game takes more or less time. In the problem of Prisoner's Dilemma, the ingenuity is very important to explore new solutions and techniques for making better agents, and arguing about their personalities and behavior. The number of possible strategies (and combinations of them), and alternatives is endless.

Acknowledgment

This work has been supported by the Brazilian national agency CAPES, by the national funding agency Conselho Nacional de Pesquisa Cientifica (CNPq) and by the state funding agency Fundacao de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG).

References

[1] J. W. Dauben and R. A. McCain. (1999) Game theory: Basic concepts. [Online]. Available: http://library.thinkquest.org/26408/math/concepts.shtml

[2] L. Tesfatsion. (2009) Game theory: Basic concepts and terminology. [Online]. Available: http://www2.econ.iastate.edu/classes/econ308/tesfatsion/gamedef.308.pdf

[3] E. F. S. Prado, "Dilema do prisioneiro e dinâmicas evolucionárias," *Estudos econômicos - Universidade de São Paulo*, vol. 29, no. 2, pp. 249–266, 1999.

[4] B. G. Linster, *Essays on Cooperation and Competition*. UMI, 1994.

[5] P. Mathieu. (1998) Iterated prisoner's dilemma. [Online]. Available: http://www.lifl.fr/IPD/ipd.html

[6] R. Axelrod, "More effective choice in the prisoner's dilemma," *The Journal of Conflict Resolution*, vol. 24, no. 3, pp. 379–403, 1980.

[7] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, no. 4489, pp. 1390–1396, 1981.

[8] R. Axelrod and D. Dion, "The further evolution of cooperation," *Science*, vol. 242, no. 4884, pp. 1385–1390, 1988.

[9] R. Axelrod, *The Evolution of Cooperation*. Basic Books, 2006.

[10] R. Calvert. (2012) Game theory in science and culture. [Online]. Available: http://calvert.wustl.edu/PolSci260/topic_3.2.html

[11] M. Q. Azhar. (2005) The success of tit for tat in computer tournaments. [Online]. Available: http://www.sci.brooklyn.cuny.edu/s̃klar/teaching/f05/alife/notes/azhar-ipd-Oct19th.pdf

[12] G. Kendall, X. Yao, and S. Y. Chong, *The iterated Prisoner's Dilemma - 20 years on*. World Scientific Publishing Company, 2007.

[13] T. Toyama, "Robert axelrod's the evolution of cooperation - a computer game for political science," 2005. [Online]. Available: http://www.kentarotoyama.org/talks/2005 08 2010 Evolution of Cooperation.ppt

[14] R. Axelrod, "Notes on axelrod's iterated prisoner's dilemma (ipd) tournaments." [Online]. Available: http://www2.econ.iastate.edu/classes/econ308/tesfatsion/axeltmts.pdf

[15] Z. Danziger. (2010) Iterated prisoner's dilemma. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/27611-iterated-prisoners-dilemma

[16] R. Axelrod, "Resources for agent-based modeling," 1997. [Online]. Available: http://www-personal.umich.edu/ axe/research/Resources.pdf

[17] H. Nurmi, *Rational Behaviour and the Design of Institutions: Concepts, theories and Models*. Edward Elgar Pub, 1999.

[18] J. Liu, N. Zhong, Y. Y. Tang, and P. Wang, *Agent Engeneering*. World Scientific Pub Co Inc, 2001.

[19] A. F. B. Neto, "Uma arquitetura para agentes inteligentes com personalidade e emoção," Dissertação de Mestrado, Universidade de São Paulo - USP, 2010.

[20] D. Richards, "Reciprocity and shared knowledge : Structures in the prisoner's dilemma game," *The Journal of Conflict Resolution*, vol. 45, no. 5, pp. 621–634, 2001.

[21] J. D. Stone. (2009) Computer models of cooperation, retaliation, and forgiveness. [Online]. Available: http://www.cs.grinnell.edu/s̃tone/events/cooperation/

[22] C. Donninger, "Is it always efficient to be nice? a computer simulation of axelrod's computer tournament," *Diekmann/Mitter*, pp. 123–134, 1986.

[23] T. Ord and A. Blair, "Exploitation and peacekeeping: introducing more sophisticated interactions to the iterated prisonerś dilemma," *Proceedings of the Congress on Evolutionary Computation, CEC '02*, vol. 2, pp. 1606 – 1611, 2002.

[24] N. Masuda and M. Nakamura, "Numerical analysis of a reinforcement learning model with the dynamic aspiration level in the iterated prisonerś dilemma," *Journal of Theoretical Biology*, vol. 278, p. 5562, 2011.

[25] M. Hagan and M. Menhaj, *Training feedforward networks with the Marquardt algorithm.* IEEE Transactions on Newral Networks, 5(6):989-993, November 1994.