How a Genetic Algorithm Learns to Play Traveler's Dilemma by Choosing Dominated Strategies to Achieve Greater Payoffs

Michele Pace
Institut de Mathématiques de Bordeaux (IMB), INRIA Bordeaux - Sud Ouest
Team ALEA - Advanced Learning Evolutionary Algorithms
michele.pace@inria.com

Abstract— **In game theory, the Traveler's Dilemma (abbreviated TD) is a non-zero-sum [1] game in which two players attempt to maximize their own payoff without deliberately willing to damage the opponent. In the classical formulation of this problem, game theory predicts that, if both players are purely rational, they will always choose the strategy corresponding to the Nash equilibrium for the game. However, when played experimentally, most human players select much higher values (usually close to $100), deviating strongly from the Nash equilibrium and obtaining, on average, much higher rewards. In this paper we analyze the behaviour of a genetic algorithm that, by repeatedly playing the game, evolves the strategy in order to maximize the payoffs. In the algorithm, the population has no a priori knowledge about the game. The fitness function rewards the individuals who obtain high payoffs at the end of each game session. We demonstrate that, when it is possible to assign to each strategy a probability measure, then the search for good strategies can be effectively translated into a problem of search in a measure space using, for example, genetic algorithms. Furthermore, the codification of the genome as a probability distribution allows the analysis of common crossover and mutation operators in the uncommon case where the genome is a probability measure.**

I. The Traveler's Dilemma

The game was formulated in 1994 by Kaushik Basu [1] and is as follows:

An airline loses two suitcases belonging to two different travelers. Both suitcases happen to be identical and contain identical antiques. An airline manager tasked to settle the claims of both travelers explains that the airline is liable for a maximum of $100 *per suitcase, and in order to determine an honest appraised value of the antiques, the manager separates both travelers so they can't confer and asks them to write down the amount of their value, at no less than* $2 *and no larger than* $100*. He also tells them that if both write down the same number, he will treat that number as the true dollar value of both suitcases and reimburse both travelers that amount. However, if one writes down a smaller number than the other, this smaller number will be taken as the true dollar value, and both travelers will receive that amount along with a bonus/malus: $2 extra will be paid to the traveler who wrote down the lower value and a $2 deduction will be taken from the person who wrote down the higher amount. The challenge is: what strategy should both travelers follow to decide the value they should write down?*

The normal form game can be described by the following payoff matrix:

$$\pi(x,y) = \begin{cases} (x+2, x-2) & \text{if } x < y \\ (x,y) & \text{if } x = y \\ (y-2, y+2) & \text{if } x > y \end{cases}$$

where x and y denotes the players' choices and $\pi(x,y)$ the payoff function. In game theory, the Nash equilibrium describes a kind of optimal strategy, informally defined as that set of strategies (one for each player) such that no player can do better by choosing a different strategy while keeping the others' strategies fixed. In a two player game, it would be a pair of strategies p, q such that:

$$\begin{cases} \pi(p',q) \leq \pi(p,q) & \forall p' \neq p \\ \pi(p,q') \leq \pi(p,q) & \forall q' \neq q \end{cases}$$

The Nash Equilibrium for the TD is the only undominated[2] solution $x = y =$ $2 which purely rational players would always want to play because, with any other pair of strategies $(x,y) \neq (2,2)$, at least one player can improve by choosing a value that is exactly one lower than the other player's choice.

However, when the game is played experimentally (as described in [3],[4],[5]) most participants select much higher values, usually close to $100, and this is true both for game-theory experts [3] and for those who have not thought carefully through the logic of the game. As a matter of fact, on average, by deviating strongly from the Nash equilibrium players are able to obtain much higher rewards. This paradox has led some to question the value of game theory in general, whilst others have suggested that a new kind of reasoning is required to understand how it can be quite rational to ultimately make non-rational choices.

To study the properties of the Traveler's Dilemma we propose a genetic algorithm that simulates a population of players, each one playing the game a fixed number of times against other randomly chosen players.

In the first part of the article (Section II), we detail the genetic algorithm used and in the second part, the results are

[1] In zero-sum games, a participant's gain or loss is exactly balanced by the losses or gains of the other participant(s), whereas in a non-zero-sum game, the aggregate gains and losses of the participants is nonzero. Non-zero-sum games are not strictly competitive.

[2] A strategy is dominated if, regardless of what any other players do, the strategy earns a player a smaller payoff than some other strategy. Hence, a strategy is dominated if it is always better to play some other strategy, regardless of what opponents may do.

discussed (Section III) and compared to the results obtained from experimental studies on how people play the game. Finally, we give our interpretation of the results, discuss the conclusions and report on possible lines of further research.

II. Description of the genetic algorithm

The idea of repeatedly playing a game and measuring the sum of payoffs obtained at the end of each session to evaluate a strategy is far from being new: it has been fruitfully applied for instance to the famous Prisoner's Dilemma and has led to the discovery of effective strategies that let players obtain good payoffs. Basic strategies for the IPD (Iterated Prisoner's Dilemma) are for example RAND (Random), ALLD (always defects), ALLC (always cooperates), NEG (negates last opponent's move), GRIM (starts with cooperation but turns into ALLD after first defection), TFT (tif-for-tat, starts with cooperation then plays last opponent's move). Good references are provided in [8]. IPD can be used in many psychological, economic, military and decision-making problems as a model of sociologic behavior. We utilise a similar technique, and propose here an analysis of what we can call RTD (Repeated Traveler's Dilemma) by means of genetic algorithms. The Iterated Prisoner's Dilemma is a version of the Prisoner's Dilemma where two opponents play against each other repeatedly, with memory, allowing for retaliatory strategies like tit-for-tat. Here, we analyze the Traveler's Dilemma in a different way, using a genotype encoding that does not allow for memory or recognition of particular opponents. The implication is that retaliatory dynamics are not possible, thus we prefer to use the term "repeated" instead of "iterated" when comparing it to similar works on other games. In the Repeated Traveler's Dilemma, each player chooses an answer for each match using a of probability distribution that measures the preference he assigns to the possible values ($2 - $100). The distribution is encoded in his genome and evolved through generations using mutation and crossover operators. Thus, during each generation the individual I_i is called to play N matches against a randomly chosen opponent in the current population. Each player selects a price according to his genome and receives a corresponding payoff. As the algorithm is executed, the best players (those whose total payoff in the current generation is greater) are rewarded by assigning them a greater probability to be selected when creating the next generation. Analyzing the results of the simulations, it is possible to determine if, after some time, an equilibrium condition is reached, if this condition is the Nash equilibrium for the game or if the population learns that playing high values is (sometimes) more rewarding than playing the undominated strategy $2. In addition, we want to analyze the differences between the average strategy adopted by the population of individuals in the algorithm and the strategy used by human players when playing the game, to see if there are commons traits and hopefully determine if, on average, humans are better in playing TD than the GA proposed or vice versa.

The genetic algorithm definition goes through the following steps:

Choice of the fitness function → Codification of a possible solution into a genome → Evolution of the current population (in this case playing the game repeatedly) → Selection of the best players → Crossover → Mutation → New generation.

A brief description of each step is given:

1) **Fitness function:** The fitness function used to measure the quality of a player's strategy is the amount of money gained during a generation.
2) **Genome:** As we are interested in understanding how different strategies perform when used during repeated game sessions we map each genome to a possible strategy and measure its effectiveness with a fitness value. More specifically, when playing the original version of the game each player has a genome composed of 99 values (the possible choices for a game, from $2 to $100 inclusive). Each gene represents the value of the probability the player assigns to that answer. For example, if a player has a genome with the mass of probability distributed on the low values ($2-$30 for example) when playing repeatedly he will have a conservative strategy, playing low values most of the time. On the contrary if the mass of probability is concentrated on high values ($80-$100) the players will have an aggressive game, taking the risk of playing highly dominated strategies far from Nash equilibrium, hoping for higher rewards.
3) **Evolution of the population:** During the evolution step each individual plays against other individuals a fixed number of times, accumulating the amount of money gained in each game.
4) **Selection:** Based on the result of the games, the best players are selected to mate and produce offspring. The individuals are selected using a fitness proportional selection (Random Wheel Selection).
5) **Crossover and Mutation:** When two individuals are chosen to mate, their genomes (their strategies when playing the game) are combined using crossover and mutation operators to create the genome of a player in the next generation.
6) **New generation:** The offspring of the previous players is the population that will play in the next iteration of the algorithm.

A. Crossover and Mutations

The genome being a distribution of probability on the set of possible answers, add a series of constraints to the crossover and mutation operators that can be used. In particular, for each genome the following must be satisfied:

$$\begin{array}{l} N = 100 \\ Support: g \in 2, 3 \ldots, N \\ \mathbb{P}_g \leq 1, \mathbb{P}_g \geq 0 \\ \sum_{g=1}^{N} \mathbb{P}_g = 1 \end{array} \tag{1}$$

Where $\mathbb{P}_g$ is the probability of choosing the value g as a solution to a TD match.

Thus, crossover and mutations have to be performed carefully in order to generate individuals with valid genomes. It is not difficult to observe that all the crossover operators commonly used in the literature (single point crossover, multiple point crossover, etc.) can be used in this case if followed by a normalization step to correct the genome to have all the genes summing up to 1. The normalisation also introduces an additional mutation component because the generated genome does not necessarily contains the same genes as the parents.

Furthermore, a genome that represent a distribution of probability allows the definition of interesting crossovers and mutation operators, such as the sum operator, which takes the two parent genomes as input and generates a genome having each gene as the sum of the corresponding genes. Multiplication operator, which generates a genome where each gene is the product of the corresponding parents' genes, and so on. In the present paper we focus our analysis on the standard single point crossover operator.

After the application of the crossover operator, the new genome is mutated accordingly to a certain probability. For the mutation step, a gene is chosen randomly and a mutation value is added to it. The entire genome is then normalized and assigned to an individual playing in the next generation. Note that, as the genome is normalized after the mutation, the operation of adding a mutation value to a gene is equivalent to moving probability mass from the other genes to the mutated gene. The mutation rate and the mutation value are the two fundamental parameters of the algorithm.

III. Simulations' results

In this section we report the results of various iterations obtained by varying some of the fundamental parameters. Each run is a simulation of 5000 generations on a population of 100 individuals, each playing in 100 random games. More precisely, during the game phase of a simulation the following algorithm is executed:

```
foreach **player**{
  for (int plays = 0; plays < 100; plays++){
    \\ *choose a random player as opponent*
    opponent = random(numPlayers);
    \\ *Play the game*
    game.play(player, opponent);
  }
}
```

The initial genome for the individuals of the first population must be chosen. The uniform distribution seems the natural choice, because in principle there is no reason to force an individual to prefer a specific value as a solution for a TD instance. Nevertheless, it is possible to force all the individuals to prefer the value of $100 or the Nash equilibrium in the first generation in order to analyze the evolution of the strategies. The results obtained in those cases are discussed in section IV. In what follows, the uniform distribution is used as the starting distribution.

Number players	100
Matches per generation	100
Minimum price	2
Maximum price	100
Crossover type	Single point crossover
Mutation type	Uniform random
Mutation rate	**0.05 - 0.02 - 0.1 - 0.2**
Mutation value	**0.1 - 0.7 - 5.0 - 10.0**

TABLE I

Parameters for the genetic algorithm. The simulations are run using as mutation rate and mutation value, all the combinations of the values reported in the corresponding rows of this table.

The average genome after 5000 generations, using different mutation values and mutation rates, is reported in Fig. 1 and Fig. 2 respectively.

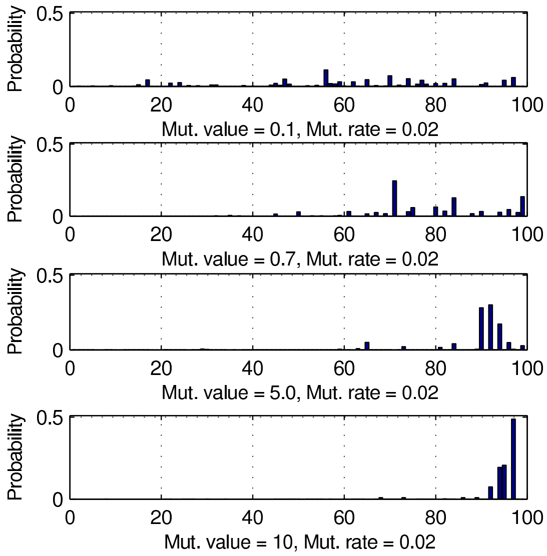


Fig. 1. Average strategy after 5000 generations using different mutation values and different mutation rates.

As the mutation value increases, the algorithm is able to explore a larger region of the search space: the convergence is faster and the distribution of probability becomes more and more concentrated on high values. The players tend to play for greater payoffs, even if the corresponding solutions are more and more dominated. On average, using these dominated strategies, the population is able to reach a global payoff much higher than the one it could obtain using more conservative strategies.

Figure 3 shows the payoff evolution for the whole population (expressed as the sum individual's payoffs) during 5000 generations using the mutation values reported in table I; high mutation values determine, on average, individuals with a tendency to play high values ($80-$100), thus corresponding to an high payoff for the whole population. However, increasing the value of the mutation over some value has no impact on the overall payoff whose average remains close to

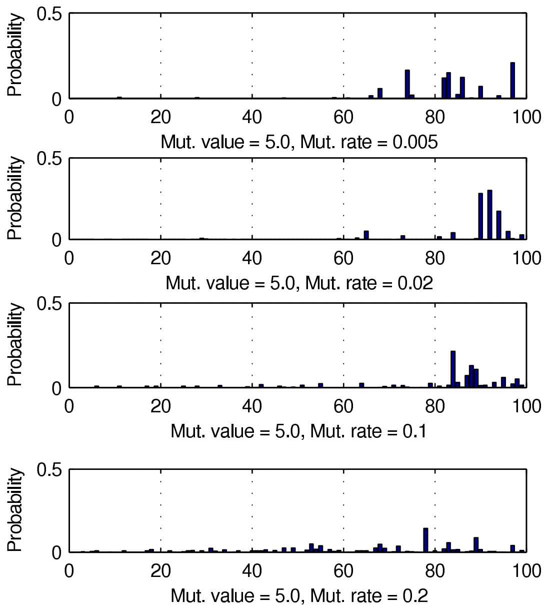


Fig. 2. Average strategy after 5000 generations using different mutation rates.

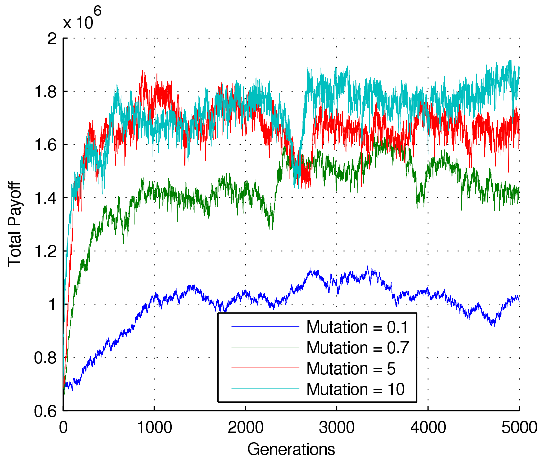


Fig. 3. Population fitness using different mutations values.

a value of $1.7 * 10^6$. Figure 3 also shows that as the mutation value increases, the variance increases as well as the average payoff for the population.

Both mutation value and mutation rate have a strong impact on the evolution of the strategies among the players. When the mutation rate is fixed, the simulations show that increasing the mutation value shifts the distribution of probability toward high values: the population evolves to play highly dominated values between \$85 and \$100. On the contrary, fixing the mutation value and varying only the coefficient of the mutation rate, the efficiency of the population decreases: as mutations happen more and more frequently, more individuals playing random values are introduced in the population, lowering the fitness and preventing highly skilled players from achieving greater payoffs. In Fig. 4 the average and maximal payoff for each generation is reported for different mutation values. The increasing of the average payoff as the population evolves is evident.

	Mut. Rate			
Mut. Value	0.005	0.02	0.1	0.2
0.1	19	19	24	26
0.5	43	53	36	36
0.7	58	65	47	19
2.0	66	77	49	39
5.0	68	81	51	31
10.0	76	92	52	40

TABLE II

PROBABILITY DENSITY CONCENTRATION USING DIFFERENT MUTATION VALUES AND RATES.

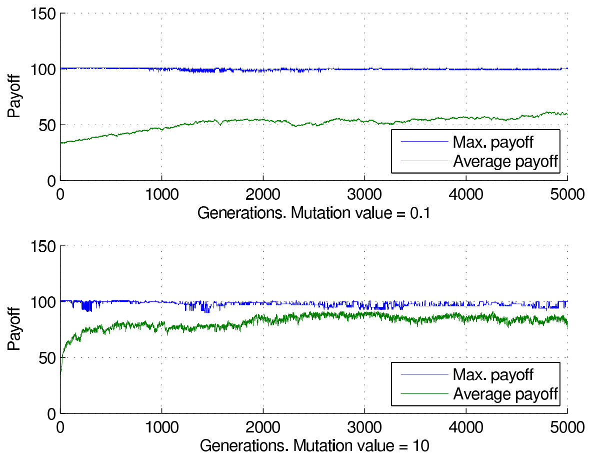


Fig. 4. Average and maximal payoff for each generation.

Considering the average genome after 5000 iterations when using different mutation values and rates, it is possible to identify the interval where the 90% of the probability mass concentrates. Table II shows, for example, that using values 0.1 and 0.005 as mutation parameters the 90% of the probability mass results, on average, distributed between \$19 and \$100, meaning that the average strategy is not very concentrated. On the contrary the value \$92 corresponding to a mutation rate of 0.02 and a mutation value of 10 means that the individuals play values between \$92 and \$100 with a probability of 90%, thus using a strategy very concentrated on high values. When the mutation rate is too low or too high, the concentration of the final genome on high values degrades; the highest concentration has been achieved using 0.02 as mutation rate. On the contrary, the mutation value has a more predictible impact on the results: increasing it, the probability always concentrate on higher values. Note that, even with very high mutation values, the average genome never assign probability 1 to \$100.

IV. Convergence to an equilibrium distribution

In the previous section, we have shown the behaviour of the algorithm when the initial population is setup with a uniform distribution. When the first generation of individuals play according to a specific (non-random) strategy it is generally possible to calculate precisely the population payoff for the first generation. For example, when all the individuals are setup to play the Nash equilibrium (undominated solution) they will always win 2$ during the first generation, with a total payoff of $\$2 * num\_players * plays\_per\_generation$. On the contrary, when all the individuals are forced to play the value of \$100 the total payoff will be $\$100 * num\_players * plays\_per\_generation$, this being also the maximum possible payoff for the population.

With this setup the effect of mutations is to introduce individuals with a less conservative strategy (who occasionally play a solution greater than \$2) in the first case, and individual with a more conservative strategy (who occasionally play a solution lower than \$100) in the second. In the first case the average payoff of the population will increase, in the second it will decrease. It is natural to ask if after a sufficiently high number of generations the fitness reaches an equilibrium independently from the starting configuration. Figure 5 shows that this happens and that the population average payoff stabilizes around the value of $1.7 \sim 1.8 * 10^6$. Mutation parameters do not affect this value but determine the average time taken to reach it.

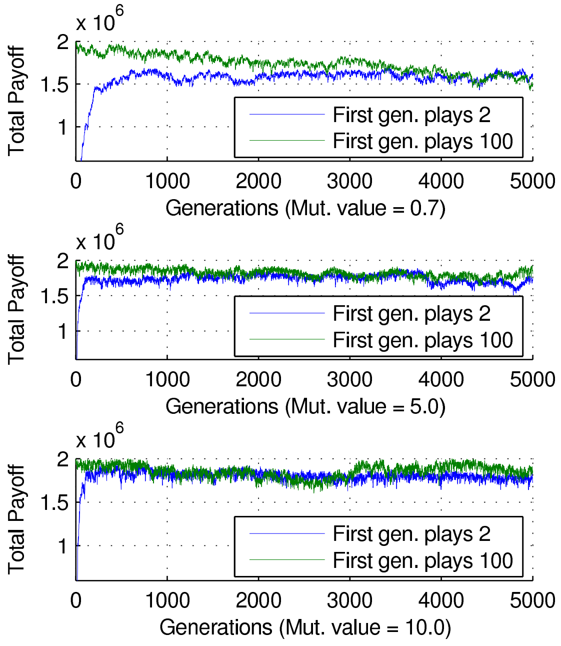


Fig. 5. Population fitness evolution when the initial population is configured to play Nash equilibrium and the completely dominated strategy \$100.

V. Algorithm and human strategies.

Considering the research of Becker, Carter and Naeve [3] on how game theory experts play the Traveler's Dilemma we try to make a comparison between the strategy of human experts and the average strategy adopted by the individuals in the populations evolved using the algorithm discussed so far.

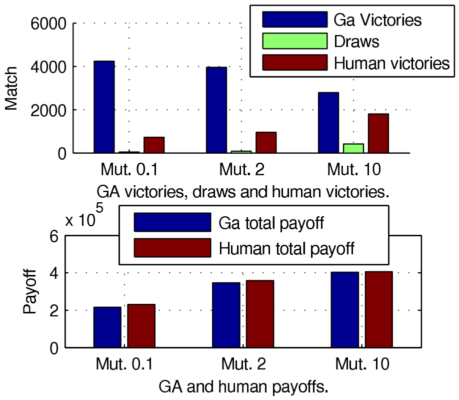


Fig. 6. Payoffs obtained by the genetic algorithm with various mutation parameters when playing against the distribution characterising game experts palying TD.

The 45 pure strategies of game theory experts as collected in [3] (table 1, page 6), are reported in table III for reference. Using these data it's possible to build an approximate genome that assigns to each value the probability of being played by people, and evaluate how well it performs against virtual opponents playing according to an evolved strategy. After normalizing the data the resulting genome is reported in table IV.

Strategy	Entry	Strategy	Entry	Strategy	Entry
2	3	88	1	96	3
4	1	90	1	97	6
31	1	93	1	98	9
49	1	94	2	99	3
70	1	95	2	100	10

TABLE III

Distribution of human strategy described in Becker, Carter and Naeve.[3]

Even experts in game theory who almost surely know the Nash equilibrium for the Traveler's Dilemma play, with an high probability, dominated strategies, especially the completely dominated value of \$100. To evaluate the relative performance of the solutions provided by the genetic algorithm against human strategies we recorded the number of victories, draws and losses during various game sessions composed of 5000 matches each, where an individual using a genome built from human statistics plays against the solutions found using different mutation values.

Figure 6 reports the results of the simulations. The genetic algorithm always obtains a greater payoff than the individual playing with the strategy build from human statistics, but this is due to the relatively low number of values used to build the human game probability distribution and, most importantly, to the possibility the algorithm has to evolve the strategy counter a fixed opponent. A more interesting and meaningful scenario would be to let the algorithm play against true human players repeatly, where both players have the possibility to learn the strategy of the opponent and to develop countermeasures. Here, as the mutations increase,

Gene	Probability
2,96,99	0.06666
4,31,49,70,88,90,93	0.02222
94,95	0.04444
97	0.13333
98	0.2
100	0.22222

TABLE IV

DISTRIBUTION OF HUMAN STRATEGY

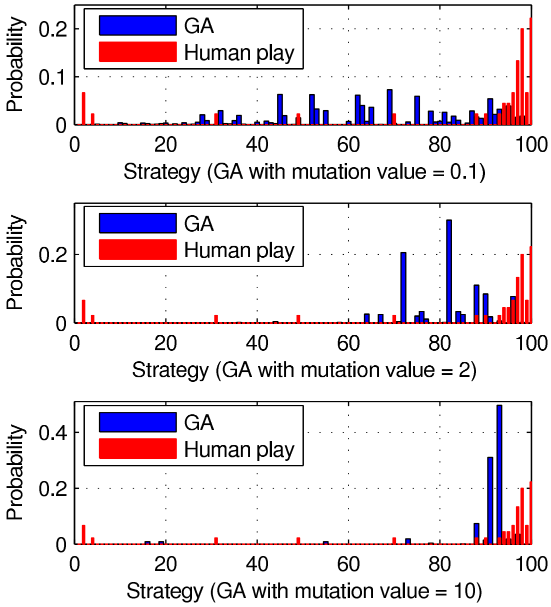


Fig. 7. Human strategy compared with the population's average strategy after 5000 generations and using different mutation values.

the GA becomes more and more concentrated on high values (like the human strategy), and even if the algorithm always performs better, both opponents obtain a much higher reward. When using an high mutation coefficient the algorithm concentrates the probability between $85 and $95, but assigns a lower probability to the value of $100. This assures the GA the highest probability to win and the highest payoff it can obtain. Figure 7 details the results at the end of each session.

VI. BEHAVIOUR VARYING REWARDS AND PENALTIES

Studies from Capra et al. [4], Land et al. [7] indicate the dependency of the solution on rewards and penalties used in the game. When the rewards and penalties are low, as in the original version of the game, players tend to play highly dominated strategy more often. When the penalties increase, people are lesser ready to accept the risk of playing high values, and prefer to play less-dominated strategies. Finally, for the largest values of rewards (with a maximum of $40), the average reaches the Nash equilibrium. To see if the genetic algorithm shows similar effects, we ran simulations varying penalties and rewards. Figure 8 reports the average final distribution of probability when penalties and rewards are set to $2, $5, $10, $20 and $30 respectively. The results show that, when the penalty is low, the strategy is concentrated on high values ($90-$100); with a penalty/reward of $5 the distribution starts to be distributed on less-dominated values and when the penalties are big enough ($20 or $30), most of the individuals play the Nash equilibrium for the game. The process of fitness proportional selection rewards the individual who utilise a conservative strategy, being able to obtain, on average, a payoff greater than those who play high values. To avoid negative payoffs in the simulations, the lowest choice is set equal to the honesty rewards h, so the possible choices when playing are $\$h$ - 100.

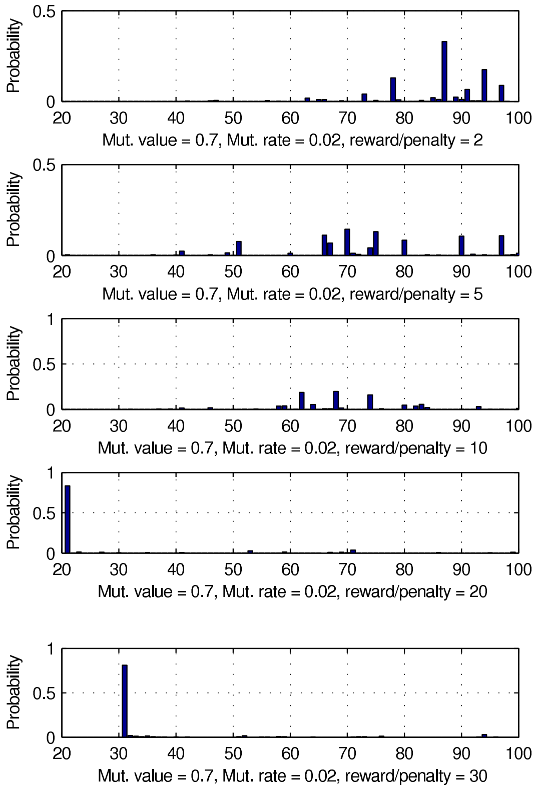
Number players	100
Matches per generation	100
Minimum price	**$2 - $5 - $10 - $20 - $30**
Maximum price	**$2 - $5 - $10 - $20 - $30**
Crossover type	Single point crossover
Mutation type	Uniform random
Mutation rate	0.05 - 0.02 - 0.1 - 0.2
Mutation value	0.1 - 0.7 - 5.0 - 10.0

TABLE V

PARAMETERS FOR THE GENETIC ALGORITHM. THE VALUES FOR REWARDS AND PENALTIES ARE LISTED IN THE CORRESPONDING ROWS. SAME VALUES ARE USED FOR REWARDS AND PENALTIES ($2-$2, $5-$5, ETC.).

VII. CONCLUSIONS

Classic game theory doesn't give a satisfactory description of human behaviour when playing the Traveler's Dilemma; it does not take into account forms of cooperation that often arise when humans are asked to make choices, and it generally assumes that people are able to think as many levels deep as needed. In this paper, we proposed a genetic algorithm to search over the probability space for the distributions that maximize the average payoff in repeated game sessions. We performed various simulations varying the most important parameters and analyzed the results trying to assess the quality of the solution found. The results reported in this paper derive from more than 2 billions of simulated Traveler's Dilemma matches, played by a population of 100 simulated players, each one playing against another 100 times for 5000 generations. Using a probability distribution as genome makes it possible to effectively apply genetic algoritms to probabilistic search spaces, and in our case we found that it is generally possible to verify the convergence to a solution providing an high fitness for the Traveler's Dilemma problem. We then compared the results found by varying some of the fundamentals parameters with the strategy used by experts in game theory when playing the Traveler's Dilemma. The solutions evolved by the algorithm performed better, but this is due to the low amount of data available to build the average human strategy and to the fact that only one opponent (the algorithm) has the possibility to change and evolve the strategy according to the results in previous game sessions. Moreover, even if changing



Mut. value = 0.7, Mut. rate = 0.02, reward/penalty = 2

Mut. value = 0.7, Mut. rate = 0.02, reward/penalty = 5

Mut. value = 0.7, Mut. rate = 0.02, reward/penalty = 10

Mut. value = 0.7, Mut. rate = 0.02, reward/penalty = 20

Mut. value = 0.7, Mut. rate = 0.02, reward/penalty = 30

Fig. 8. Average strategy after 5000 generations using different values for rewards and penalties. With high rewards and penalties the distribution of probability converges to Nash equilibrium.

the rewards and penalties to values higher than $2 should theoretically have no impact on the strategy of the players, experimental results show that, in practice, it has an impact: as the rewards increase, the players are more and more likely to go to the Nash equilibrium. The algorithm showed the same behaviour without having any a priori knowledge about the game. This probably means that the reasons of this effect are not exclusively psychological. Genetic algorithms have been used many times to find solutions to game theory problems and have proven their effectiveness in cases where general methods don't give a satisfactory description of how people behave. We showed that in the specific case of Traveler's Dilemma, a genetic algorithm can be used to obtain solutions with an average payoff higher than what we can obtain with other methods. Furthermore, the results show forms of convergence to equilibrium distributions that, although dependent on configuration parameters, suggest possible directions of research, especially concerning the application of formal probabilistic methods and convergence analysis.

References

[1] Basu, Kaushik, *The Traveler's Dilemma:Paradoxes of Rationality in Game Theory.* American Economic Review, May 1994, 84 (2), 391 –395.

[2] Basu, Kaushik, *The Traveler's Dilemma.* Scientific American Magazine, June 2007

[3] Becker Tilman, Carter J. Michael, Naeve Jörg, *Experts Playing the Traveler's Dilemma.* Institut fr Volkswirtschaftslehre Universitt Hohenheim, (2005).

[4] C. Monica Capra et al.: *Anomalous Behavior in a Traveler's Dilemma?* American Economic Review, Vol. 89, No. 3, pages 678 –690; June 1999.

[5] Rubinstein, Ariel: *Instinctive and Cognitive Reasoning : A Study of Response Times.* School of Economics, Tel Aviv University, Tel Aviv, Israel 69978 and Department of Economics, New York University, New York, NY 10003

[6] Michalewicz, Zbigniew: *Genetic Algorithms + Data Structures = Evolution Programs* Springer - March 1996

[7] Land, van Neerbos, Havinga: *Analyzing The Traveler's Dilemma.*

[8] Glomba, M.; Filak, T.; Kwasnicka, H. *Discovering effective strategies for the iterated prisoner's dilemma using genetic algorithms.* Intelligent Systems Design and Applications, 2005. ISDA apos;05. Proceedings. 5th International Conference on Volume , Issue , 8-10 Sept. 2005 Page(s): 356 - 361

[9] Thomas Vallee, Murat Yildizoğlu: *Presentation des algorithmes genetiques et de leurs applications en economie.* Revue d'conomie politique, Vol. 114.

[10] Del Moral P., Kallel L. and Rowe J. *Modeling genetic algorithms with interacting particle systems* Revista de Matematica, Teoria y aplicaciones, Vol.8, No. 2, July (2001).

[11] Del Moral P. *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications.* Springer, New York; Probability and Applications.