Coevolutionary Learning in the Tragedy of the Commons

Julián García
Computer and Systems Engineering Department.
National University of Colombia.
Bogotá, Colombia
julian.garcia@ieee.org

Fernando Nino
Computer and Systems Engineering Department.
National University of Colombia.
Bogotá, Colombia
lfnino@ing.unal.edu.co

Abstract- The joint utilisation of a commonly owned resource often causes the resource to be overused. This is known as *the tragedy of the commons*. This paper analyses the effects of coevolutionary learning in such kind of situations using genetic programming. In a game-theoretical approach, the situation considers not only the strategic interaction among players, but also the dynamics of a changing environment linked strongly to the players' actions and payoffs. The results of an analytical game are used to formulate a simulation game for *the commons*, then a series of computational experiments are conducted, obtaining coevolved game strategies that are examined in comparison with those predicted by the analytical model. The obtained results are similar to those predicted by classic game theory, but not always leading to a *tragedy*.

1 Introduction

The joint utilisation of a commonly owned resource often causes the resource to be overused. This situation is known as *the tragedy of the commons* [1]. This paper analyses some effects of coevolutionary learning in such kind of situations, by modelling and simulating adaptive agents using genetic programming.

The tragedy of the commons has been studied largely by economists, sociologists, and in general, systems theorists. It has come to mean "a shorthand for a number of pervasive real-life situations" [2] such as fishery in deep seas, pollution, and global warming, among others. Critical aspects present in a *commons* situation are:

- A resource, the commons, is being exploited by a group of players.
- All of the players have free access to the resource.
- The more intensively used the resource is in the present, the more limited the resource will be in the future.

The last characteristic implies a tradeoff between actual consumption in the present, and possible consumption in the future. The forthcoming amount of available resource will depend upon the resource available in the present. Thus, a player could be willing to sacrifice his present consumption, in order to obtain higher payoffs in the future, but this restraint behaviour is only efficient and effective as far as all the players assume the same position. "Classic Game Theory has shown that common property resources will always be exploited beyond the point that is more desirable from the collective viewpoint" [3] when considering agents that act based on maximizing their immediate payoffs.

Artificial Intelligence techniques have shown to be useful to analyse relations between learning and cooperation in games. In particular, coevolutionary adaptation, which "is a special type of evolutionary learning where the fitness of each learner depends of the fitness of the others"[4], has been employed especially trying to gain insight into the emergence of cooperative behaviour. "The computational problem is to determine reliably any ultimately stable strategy (or strategies) for a specific game situation" [5].

The literature reports results on benchmark problems such as the iterated prisoners' dilemma (IPD), which has received extensive study by computer scientists[5]; several approaches have been explored, including n-player versions of the game and expanded sets of player actions, aiming to study more realistic situations [6] [7].

The *commons* framework is very similar to that of the IPD, in the sense that a choice that may be perceived as the best option for an individual, implies poor results for the whole society. However, *the commons* also consider a changing environment linked strongly to the game. Thus the situation is inherently dynamic, even beyond the changes in players' actions.

Computer simulations of common pool resources have been performed using techniques other than evolutionary computation. An example can be found in [8], which employs swarm agents in order to simulate a tragedy of the commons using *if-then* rules, and suggests the use of evolutionary techniques for further research. This work intends to be a contribution in that direction. Real-life situations involve environmental conditions that tend to be highly dynamic. In this sense, *the commons* might represent a very realistic and interesting framework; thus, the question to be addressed is: *Does coevolution solve the tragedy of the commons?*

The rest of this paper is organised as follows: Section 2 introduces some game-theoretical results in which this work is based on, particularly, an n-player dynamic game is presented; Section 3 discusses the proposed simulation model, and experiments based on GP are described; Section 4 anal-

0-7803-7804-0 /03/$17.00 © 2003 IEEE

yses the experimental results, obtained strategies are studied in terms of their fitness and compared to predictions given by classic game theory; finally, some conclusions are devised in Section 5.

2 Preliminaries

In this section, the n-player version of a game introduced by Mirman and Levhari[9] is presented. This game captures the main features of a *commons* scenario; the model is a dynamic game, i.e., it considers not only strategic interaction among players, but also interactions between the players and the environment. The argument used to support the tragedy is that "The Nash Equilibrium of the game will always lead to an outcome that is worse than the socially most desirable" [3]. The version presented below is highly based on the explanation given in [10] for the two player case.

2.1 Mathematical Setting

Consider n players having free access to a common resource during a particular period of time. Let y_t be the amount of resource available at time period t, and c_{it} the amount of resource consumed by a player i at time t. Clearly, for all values of t,

$$0 \leq \sum_{i=1}^{n} c_{it} \leq y_t \qquad (1)$$

Let x_t denote the unconsumed amount of resource at time t, then

$$x_t = y_t - \sum_{i=1}^{n} c_{it} \qquad (2)$$

An important issue in this model is that the resource must be renewable, therefore, a natural law acting upon the resource is taken into account, by introducing a constant α, such that

$$y_{t+1} = x_t^{\alpha} \qquad \text{with } \alpha \in [0,1] \qquad (3)$$

The natural behaviour of the resource, without any external interference, is dominated by two fixed points, 0 and 1 [9]: 0 means total scarcity of the resource and 1 represents a stable state of sustainability. Notice that when x_t is in $(0,1)$, higher values of x_t produce higher values of y_{t+1}, thus the unconsumed amount of resource may be thought of as an investment.

The payoff received by player i at time t is given by the utility function

$$u_t^i = \log c_{it} \qquad (4)$$

Within this framework, the question to be addressed is: *Does the strategic interaction among players imply over-consumption?*

2.2 Social Optimum

What is the consumption policy that allows the players as a whole to be as happy as possible? The social optimality criterion will be the sum of the payoffs obtained by all the players. Thus, the goal is to optimise this amount along time, which is accomplished by using backward induction [9]: starting from the last period of the game, the optimal consumption (from the social viewpoint) is computed at each period of the game going backwards.

Since payoff values are being optimised along time, it is necessary to consider a discount factor $\delta = \frac{1}{1+r}$, where r is the amount of a unit of consumption that an individual is willing to give up in order to be able to consume in the present period, rather that in the next one. It is clear that δ is in $[0,1]$, since $r > 0$. The discount factor δ is assumed to be the same for all the players.

By expressing consumptions as fractions of the available resource y, it is possible to observe a resulting expression for the consumption fractions, as shown in Table 1.

Number of periods left	Consumption Fraction
1	$1/n$
2	$1/n(1+\alpha\delta)$
3	$1/n(1+\alpha\delta+(\alpha\delta)^2)$
$\vdots$	$\vdots$
T	$1/n(\sum_{i=0}^{T-1}(\alpha\delta)^i)$

Table 1: Social optimum consumption policy

Thus, the social optimum consumption is a function of the remaining time to play T, and the variables α, δ, and n. This rule determines what fraction of the available resource a player should consume, in order to optimise the resource allocation from the social viewpoint, as follows:

$$\boxed{f^s(T,\alpha,\delta,n) = \frac{1}{n\sum_{i=0}^{T-1}(\alpha\delta)^i}} \qquad (5)$$

2.3 Nash Equilibrium

Does the social optimum policy take place when assuming completely rational players? To address this question, the Nash solution for the game will be found.

The calculation of the Nash Equilibrium requires an additional assumption. *Over-consumption Assumption:* When the desired consumption of the players is greater than the available stock, the resource will be equally divided among the n players.

The best response function (one that indicates to a player what to do, given the other players' actions) is computed for a player. The Nash Equilibrium takes place when all the players are playing best response strategies (see [10]).

As in the social optimum case, the technique used is backward induction, computing the Nash equilibrium at each stage of the game. It is important to notice that in this case the over-consumption assumption holds in the last period of the game (once that period is over there is no game to play, therefore players will try to consume as much as possible). As in the social optimum case, an expression for fractions of consumptions is found:

Number of periods left	Consumption Fraction
1	$1/n$
2	$1/(n+\alpha\delta)$
3	$1/(n+\alpha\delta+(\alpha\delta)^2)$
$\vdots$	$\vdots$
T	$1/((n-1)+\sum_{i=0}^{T-1}(\alpha\delta)^i)$

Table 2: Equilibrium consumption policy

Thus, the Nash equilibrium solution is:

$$\boxed{f^*(T,\alpha,\delta,n) = \frac{1}{(n-1)+\sum_{i=0}^{T-1}(\alpha\delta)^i}} \qquad (6)$$

2.4 Comparing Social Optimality and the Nash Equilibrium

To show that a *tragedy* takes place, the Nash solution and social optimum solution are compared. It is necessary to keep in mind that the larger the current consumption is, the smaller the available resource will be in future periods. In this sense, the social optimum solution involves a balance between immediate and long-run consumptions, as opposed to the Nash solution.

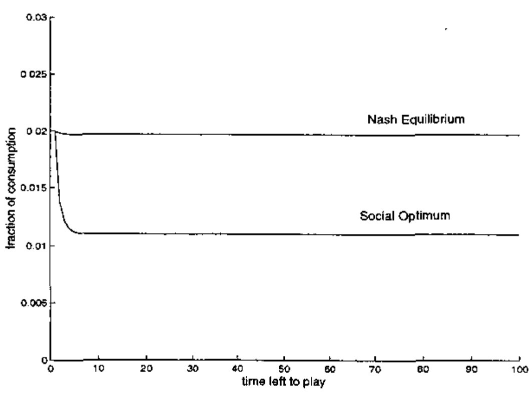

Figure 1: Social optimum and Nash solution; $f^s(\cdot)$ and$f^*(\cdot)$, in terms of T, for fixed values of α, δ and n ($\alpha = 0.5$, $\delta = 0.9$ and $n = 50$)

Figure 1 shows social optimum and Nash solutions for fixed values of α, δ and n (particularly α equal to 0.5, δ equal to 0.9 and n equal to 50). Henceforth when comparing rules and making calculations only in terms of T, such values will be considered. For these values, the two graphs (Nash and social optimum solutions) showed a considerable difference, something desirable for comparison purposes. On the other hand, n was set to 50, the number of players to be considered in the experiments.

It can be noticed that Nash Solution is almost a constant value in terms of T. In contrast, social optimum is rather restricted if compared to the Nash equilibrium, in the sense that only as the end of the game gets closer, the amount consumed begins to increase. This is quite clear, since when the game is about to end, there is not much sense in restraining to "save" for the future.

The restricted behaviour of the social optimum game strategy, $f^s(\cdot)$, contributes to achieve resource sustainability, which formally can be demonstrated by replacing the consumption expressions generated by $f^*(\cdot)$ and $f^s(\cdot)$ in (3), and making $T \to \infty$. This gives a discrete dynamical system, which represents the dynamics of the resource being consumed. For the social optimum, the resource achieves a sustainable state reflected in a stable fixed point different from zero. On the other hand, for the Nash solution the system has a fixed point closer to zero [9].

3 Simulation Game Model

This section presents the model of the simulation game. The technique used is genetic programming (GP), which is a kind of evolutionary algorithm (EA) in which each individual is a computer program, in particular a mathematical function [11]. Each function is constructed as the composition of elements from a well-defined set of terminals (arguments for the function) and primitives (other fundamental functions from which each individual is constructed). Individuals are evaluated according to a fitness function, and the evolutionary process favors fitter individuals.

The dynamics generated by GP is a sequence of sets of programs that can be interpreted as the evolution of an artificial society, where each agent is represented as a program[12].
The mathematical expressions generated with GP, will represent game strategies similar to those described in Section 2, with each rule determining the actions of a player throughout a game. Thus, the EA represents a social learning process [13] that aims to improve the allocation of the resource throughout generations.

Starting with a population of game strategies created at random, new ones are constructed from previous ones in an evolutionary fashion. The search space (primitives and terminals) and fitness function are discussed next.

3.1 Genotypical Representation

The search space corresponds to a set of game strategies similar to those predicted by classic game theory. It is necessary to devise terminals and primitives in a way that allows the evolutionary process to make rules such as Nash equilibrium (equation 6) and social optimum solution (equation 5) attainable.

Consequently, the terminal set τ is equal to $\{\alpha, \delta, T, n, 1\}$, where α is the constant value associated to the renewal rate of the resource being exploited; δ is the discount factor; T is the number of remaining interactions to conclude a determined game; n is the number of players in the game, and 1 is the constant value.

On the other hand, the primitive set $\mathcal{F}$ is equal to $\{/, *, +, \widehat{\Sigma}\}$, where $+, /, *$ are the arithmetic operations, **sum**, **division**, and **product**, respectively, and $\widehat{\Sigma}$ is a function called a **geometric sum**, defined as:

$$\widehat{\Sigma} : (\mathbb{Z}^+ \times \mathbb{R}) \to \mathbb{R} \tag{7}$$

$$\widehat{\Sigma}(n, x) \mapsto \sum_{i=0}^{n} x^i$$

Since the first parameter of the geometric sum primitive is not a real value, there is no closure among operators [11], thus the GP implementation is devised to keep syntactically valid rules by avoiding those where the first parameter of the geometric sum is not in $\mathbb{Z}^+$.

Functions corresponding to game strategies are genetically represented as strings of symbols in $\tau \bigcup \mathcal{F}$. In order to get the consumption rule associated to a string (chromosome), a numerical stack is used, as shown in figure 2, in the same way as in stack-based GP (see [14]).

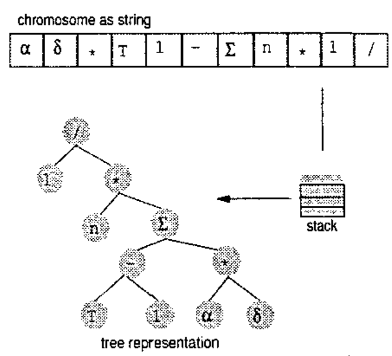

Figure 2: Social optimum rule represented as a string and then decoded as a tree using a stack

3.2 Phenotypical Representation

In this section the question to be addressed is: *Given a function that represents the strategy of a player, how is consumption calculated throughout a game?*

A game strategy (rule) constructed from the primitives and terminals discussed previously, determines what fraction of the available resource a player should consume, given any values of T, δ, α, and n.

Let g^i be the function used by agent i to play a particular game. Clearly, because of the definition of the primitives and terminals, this function has $\mathbb{R}$ as range. However, functions are supposed to yield fractions for any given parameters, therefore, whenever a function is not mapped into [0,1], at a specific stage of a game, the rule is considered semantically invalid, and the assigned consumption is zero.

Henceforth, let G be

$$G = \sum_{i=1}^{n} g^i(\cdot)$$

Considering the population as a whole, even assuming that all the game strategies are semantically valid, the possibility of over-consumption may still occur. This will happen whenever, at a determined stage of the game, G is greater than 1. In this case, the over-consumption assumption holds. Therefore, the assigned consumption (for those players who are using semantically valid rules) is the current available resource, divided equally among the n players. This assumption was kept for the simulation game in order to make it comparable to the analytical model.

Summarising, if c_t^i is the amount of resource consumed by agent i at a period t of a game, then, c_t^i is computed as

$$c_t^i = \begin{cases} y_t g^i(T, \alpha, \delta, n) & \text{if } G \leqq 1 \text{ and } g^i(\cdot) \in [0,1] \\ \frac{y_t}{n} & \text{if } G > 1 \text{ and } g^i(\cdot) \in [0,1] \\ 0 & \text{if } g^i(\cdot) \notin [0,1] \end{cases} \tag{8}$$

where y_t is the amount of resource available for such a period.

In the first case, the result given by the rule is a fraction, and the population does not overconsume. This can be thought of as the ideal case, when all the players are being semantically valid. The second case considers over-consumption behavior and takes care of the over-consumption assumption. The last case is a consequence of the possibility of semantic invalidity. Players whose rule is nonsensical are not allowed any consumption.

Accordingly, the payoff of a player at a determined stage of a game is computed as the logarithm of its consumption. In addition, the resource dynamics for a game is computed according to (3), i.e, the same way as in the classic game model.

3.3 Fitness Assessment

The coevolution process favors fitter players. The main criterion for such suitability is the payoff derived by the players in the game, so the fitness function needs to take into account, in a special manner, the payoff derived by the players throughout a game.

Moreover, the fitness measure will take care of two additional facts. As shown above there are two non-desirable states for a rule: semantic invalidity, and over-consumption behavior. The former, as addressed in [12], is a pervasive problem when applying GP techniques to this kind of problem. The latter is also involved in the classic model, and tackled by introducing a constraint in the optimisation problem(see equation 1). Thus, the mechanism used to deal with such non-desirable states is to drive them out of the evolution process by means of the fitness function, which considers two additional values, each one of them accounting for semantic invalidity and over-consumption behavior.

Briefly, the fitness measure has three goals:

- Keep those individuals whose payoff is high
- Remove those individuals that are not semantically valid
- Discard those individuals that tend to overconsume

The way to achieve such goals is discussed next.

3.3.1 Payoff Contribution

Suppose an entire game consisting of L periods has been played by agent i. The payoff perceived by the player at a determined stage t is

$$u_t^i = \log c_t^i \tag{9}$$

where c_t^i is the resource consumed by the player in such stage. At the end of the game, a value accounting for the payoffs perceived by such an agent is

$$U^i = \sum_{t=1}^{L} u_t^i$$

The amount U^i gives a measure of the payoff contribution to the fitness. For the sake of computational tractability, this amount is mapped into a number p^i in $[0,1]$. This fact is rather an implementation issue (for this purpose a well parameterised linear transformation is used). Thus at the end of a game, for each one of the players, a number p^i in $[0,1]$ measures payoff contribution.

3.3.2 Semantic Invalidity Penalisation

Consider an indicator function of the form

$$I(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

Let $g^i(\cdot)$ be the trial strategy used by agent i at a determined game composed of periods $t = 1, ..., L$. The semantic invalidity measure for a period t, is calculated as

$$d^t(g^i(\cdot)) = I(x)\min(|g^i - 1|, |g^i|)$$

Clearly, $d^t(g^i(\cdot))$ is large when the result given by the rule is far from $[0,1]$, and zero when the result is in $[0,1]$. Consequently, for a whole game, the semantic invalidity measure is summarised in a number D^i, defined as:

$$D^i = \sum_{t=1}^{L} d^t(f^i(\cdot)) \quad .$$

For the sake of computational tractability, D^i is mapped into a number s^i in $[0,1]$, using a linear transformation such that the agent whose D^i is the greatest receives 1, and the one whose D^i is smallest receives 0. Thus, at the end of a game, a number s^i in $[0,1]$ for $i = 1, ..., n$, gives a measure of the semantic invalidity of each one of the agents.

3.3.3 Over-consumption Behavior Penalisation

If $g^i(\cdot)$ is the trial strategy used by agent i at a determined game of L periods, a proposed measure of the tendency to over-consumption of such agent, for a time period t, is computed as

$$q^t(g^i(\cdot)) = \begin{cases} 0 & \text{if } G \leq 1 \\ (1-G)\frac{g^i(\cdot)}{G} & \text{otherwise} \end{cases} \tag{10}$$

Thus, for a whole game, the amount measuring tendency to over-consumption for player i is

$$o^i = \sum_{t=1}^{L} q^t(g^i(\cdot))$$

This amount is linearly transformed into $[0,1]$, assigning 1 to the greatest and 0 to the smallest. Thus, at the end of a game, a number s^i in $[0,1]$ for $i = 1...n$, accounting for the semantic invalidity of each one of the agents is calculated.

3.3.4 Fitness Function

After a game is over, the suitability of player i, is represented as a vector $v^i = (p^i, s^i, o^i)$, and its fitness measure will be a scalarisation of this vector. For the purposes of this work the fitness of agent i is defined as:

$$F^i(v^i) = p^i(1 - \max(s^i, o^i)) \tag{11}$$

The idea behind this function is to avoid semantically invalid trial strategies, as keeping the essence of the payoff assignment in the analytical model. The $\max(\cdot)$ function is used to apply a strong punishment on semantic invalidity and constraint violation. Then, it is expected to have individuals whose fitness are only proportional to the payoff received in the game throughout the evolution process.

This is a multiobjective fitness function (see [15] for details), since the process may be thought of as having three goals: maximising payoff while minimising semantic invalidity and over-consumption behaviour.

3.4 The Evolutionary Algorithm

Consider a procedure Ↄ$(P, T_{MAX}, Y_0, \alpha, \delta)$, which simulates a game and computes fitness values, according to the performance obtained by the population in the game, as explained above. The parameters of Ↄ are:

- a population P of game strategies ($|P|$ denotes the number of players)
- a real value Y_0, which represents the amount of available resource at the beginning of the game
- an integer value T_{MAX}, the number of stages of the game
- the renewal rate of the resource α, and the discount factor δ.

then, the algorithm operates as

```
GP-COMMONS(·)
i ← 0
P_i ← RANDOM INITIAL POPULATION(·)
while stopping criterion not met
    do assign T_MAX, Y_0, α, δ at random
        Ↄ(P_i, T_MAX, Y_0, α, δ)
        P_S ← RANKING-SELECTION(P_i)
        P_C ← CROSSOVER(P_S)
        P_M ← MUTATION(P_C)
        i ← i + 1
        P_i ← P_M
return P_i
```

4 Experimental Results

A set of experiments was carried out, where an experiment was considered as a run of the EA described above. The EA stopping criterion was the desired number of generations. The output of an experiment was the set of game strategies obtained by the EA in the last generation. Intuitively, an experiment can be though of as the process by which a society is adapted throughout generations, to achieve an appropriate resource exploitation policy.

In total, 225 experiments were conducted. The population size considered was 50. In each experiment, the EA was ran for 1500 generations, which empirically under the studied conditions, resulted to be an adequate number of steps to achieve complete stabilisation within the population, i.e, symmetric stable strategies are obtained. This state of non phenotypical variation can be thought of as a state of evolutionary stability in the sense of Riechmann [16].

Since the social optimum and the Nash rule have a tree representation of 11 and 13 nodes, respectively, and resulting rules are expected to be comparable, the maximum size of a tree was limited to 14 nodes. Environmental variables, namely, T_{MAX}, Y_0, α, and δ, were generated randomly at each generation, with the purpose of confronting co-evolving game strategies with a considerable number of environmental conditions, and achieving comprehensive rules.

Selection was performed using elitist tournament of size four (see [15] for details). In addition, a single-point crossover was used. In addition, a random mutation was used, i.e., at each generation mutation index for such generation was sampled from an uniform distribution on $[0, 1]$. The random mutation index is used to emulate the innovative behaviour of agents, assuming that the propensity of agents to "experiment" with new rules, is a random variable that changes throughout generations.

In all 225 experiments, symmetric stable strategies were obtained (i.e., the final population showed non-phenotypical variation). Therefore, only one rule was considered at the end of each experiment. Among the 225 strategies, 91 of them percent were semantically valid. Since, the experimental sample did not include semantically invalid rules, it consisted of 205 strategies. The fact that over 90 percent of the rules were semantically valid showed that the fitness function successfully accomplished the task of avoiding non-desirable rules.

In terms of the fitness value, the best rule found was $g(\cdot) = \frac{\alpha}{n}$, which was also the *mode* within the sample, being the rule used by all players in 5 percent of the experiments. Figure 3 depicts a graphical comparison of this rule against both, Nash and social optimum rules in terms of T.

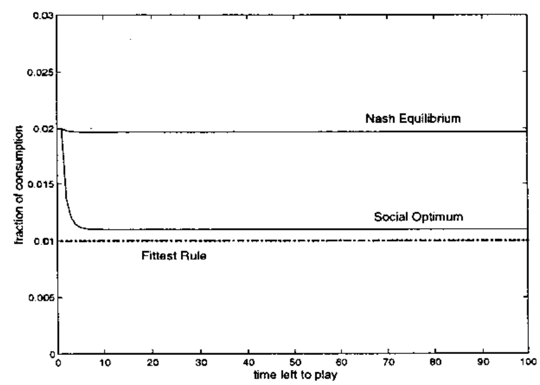

Figure 3: Graphical comparison of the fittest rule against Nash equilibrium and social optimum

It is straightforward to notice that the consumption levels of this rule are below those of the Nash equilibrium, and considerably closer to the social optimum. Also, this rule is even more restraint than social optimum, therefore, it implies a better level of sustainability for the dynamics of the resource.

The rule corresponding to the median fitness value was $g(\cdot) = \frac{\delta^3}{n}$. Figure 4 shows the graphical comparison of such rule against Nash and social optimum solutions. Notice that the median rule is almost the midpoint between them.

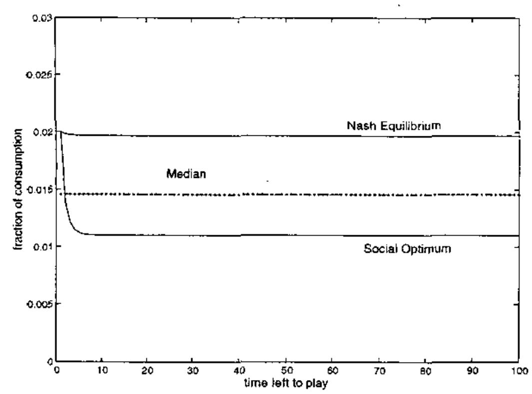

Figure 4: Graphical comparison of the median rule against Nash equilibrium and social optimum

Since Nash and social optimum theoretical solutions are the reference rules, it is necessary to define a numerical criterion to get an insight into the proximity of obtained solutions and classic game theory predictions. Thus, the distance between coevolved strategies and such functions (i.e., Nash and social optimum solution) will be computed. Those rules close to the social optimum solution may be thought of as cooperative game strategies, as opposed to those close to Nash equlibrium.

Since all the rules represent fractions, it is possible to use a metric on the space of the functions on $[0, 1]$ (see [17] for details) as follows. Let A be the domain of two functions x and y, then, the distance between x and y is calculated as

$$d(x, y) = \sup_{t \in A} |x(t) - y(t)| \tag{12}$$

Specifically, these distances between the obtained rules and Nash and social optimum solution was computed using a numerical approximation algorithm. A boxplot of such distances is shown in figure 5.

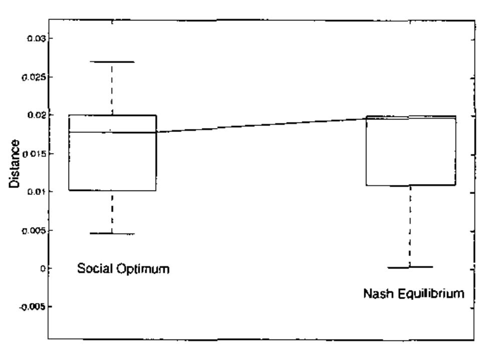

Figure 5: Boxplot of the distance from coevolved rules to the social optimum and Nash solution

Notice that both boxes are practically the same. Particularly, the distance to social optimum inner quartile range (*iqr*) is 0.0098, meanwhile, the distance to Nash *iqr* is 0.0090. In other words, 50 percent of the distances (from the coevolved game strategies to Nash equilibrium and social optimum) are in the same range.

Also, the distance of most coevolved rules to the Nash equilibrium was below the median (because the median of the distances to Nash solution is basically equal to the maximum distance). In addition, since both median values are very close (the median of the distance to social optimum is 0.01775 and the median of the distance to Nash equilibrium is 0.01967) most coevolved rules were at the same distance to both, Nash equilibrium and social optimum. This means that *cooperation* (represented by rules close to the social optimum) and *non-cooperation* (represented by rules close to the Nash solution) may emerge as the result of coevolution.

5 Conclusions

In this work, a dynamic game was used as framework to study the relationships between adaptation and cooperation. Specifically, GP was used to coevolve game strategies in a tragedy of the commons scenario. Although, the dynamic environment seems to avoid the achievement of completely cooperative rules in all cases, it was shown that cooperation -reflected in the social optimum rule- is still likely to occur.

In the experiments, some game strategies with consumption levels that were below those of the Nash equilibrium were obtained through the GP learning process; this shows that the tragedy of the commons may be solved by means of coevolution. Particularly, in order to avoid the tragedy, players cooperate by restraining their own consumptions (saving for the future), and playing strategies with values below the Nash equilibrium, which implies a better level of resource sustainability.

Many arising questions are yet unanswered. In particular it would be worth considering other representations for the game strategies within this framework, and to analyse not only final stable populations, but also the process by which these states are achieved. A lot of work is required in testing cooperation sensitivity in regard to variation of the evolutionary algorithm parameters.

Bibliography

[1] G. Hardin, "The tragedy of the commons," *Science*, vol. 162, pp. 1243–48. December 1968.

[2] T. C. Schelling, *Micromotives and Macrobehavior*. W.W Norton and Co., 1978.

[3] C. D. Aliprantis, *Games and Decision Making*. Oxford University Press, 2000.

[4] X. Yao, "Automatic acquisition of strategies by co-evolutionary learning," in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'97)*, (Gold Coast, Australia), pp. 23–29, February 1997.

[5] D. B. Fogel, *Evolutionary Computation: toward a new philosophy of machine learning.* New York, NY: IEEE Press, 1995.

[6] P. J. Darwen and X. Yao, "Why more choices cause less cooperation in iterated prisoner's dilemma," in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, (COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea), pp. 987–994, IEEE Press, 2001.

[7] X. Yao and P. J. Darwen, "An experimental study of n-person iterated prisoner's dilemma games," in *Evo Workshops*, pp. 90–108, 1994.

[8] Deadman Peter J, Schlager Edella, and Gimblett Randy, "Simulating common pool resource management experiments with adaptive agents employing alternate communication routines," *Journal of Artificial Societies and Social Simulation*, vol. 3, no. 2, 2000.

[9] D. Levhari and L. Mirman, "The great fish war: An example using a dynamic Cournot-Nash solution," *The Bell Journal of Economics*, vol. 11, no. 1, pp. 322–34, 1980.

[10] P. K. Dutta, *Strategies and Games: Theory and Practice.* MIT Press, 1999.

[11] J. R. Koza, *Genetic programming: On the programming of computers by natural selection.* Cambridge, Mass.: MIT Press, 1992.

[12] S.-H. Chen, "Fundamental issues in the use of genetic programming in agent-based computational economics," *Working Paper, AI-Econ Research Center National Chengchi University Taipei, Taiwan*, 2001.

[13] T. Riechmann, "Learning and behavioral stability: An economic interpretation of genetic algorithms," *Journal of Evolutionary Economics*, vol. 9, no. 2, pp. 225–242, 1999.

[14] T. Perkis, "Stack-based genetic programming," in *Proceedings of the 1994 IEEE World Congress on Computational Intelligence*, vol. 1, (Orlando, Florida, USA), pp. 148–153, IEEE Press, 27-29 1994.

[15] Th. Back, D. B. Fogel, and Z. Michalewicz, eds., *Handbook of Evolutionary Computation.* Bristol, UK: Oxford University Press, New York, and Institute of Physics Publishing, 1997.

[16] T. Riechmann, "Genetic algorithm learning and evolutionary games," *Journal of Economic Dynamics and Control*, no. 25, pp. 1019–1037, 2001.

[17] E. Kreyszig, *Introductory Functional Analysis with Applications.* New York, NY, USA: John Wiley & Sons, 1978.