# Evolution of Arbitrary Agent-Programs

Dániel L. Kovács [1]) and Tadeusz Dobrowiecki [1])

[1]) Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, Hungary
dkovacs@mit.bme.hu

## Abstract:

In this article we propose a novel approach to the evolution of agent-programs by means of natural selection. The existing approaches (e.g. genetic programming) are usually constrained to relatively simple program-structures since they need explicit representation of fitness, genetic operators, and selection mechanism. We propose a methodology that overcomes these issues by introducing a lifecycle of agents, and their phenotype-phenotype interaction. As a consequence, an emerging evolutionary optimization process called "natural selection" arises, which enables the evolution of arbitrary agent-programs. Several interesting experiments are presented.

**Keywords:** evolution, natural selection, agents, programs

## Introduction

An agent "can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors". [1]. The program of an agent is responsible for choosing its actions based on its inner state, which is usually its belief about its environment (including itself and the other agents too).

This article presents a novel, realistic agent-based simulation model for the natural selection of such agents. The survival of these agents depends solely on their arbitrary programs [2] selecting their strategy to interact with each other. Game theory is used to model the strategic interaction among agents [3].

Several models of program evolution exist, but our approach differs from them mainly in the following [4]–[7]: we utilize natural selection as a realistic "driving force", and not some preprogrammed, explicit mechanism, to evolve the population of agents; we do not consider the emergence of new variants, only proliferation, i.e. asexual replication of given program-types; and so we do not impose any formal constraints on the structure and the inner workings of programs; there is no artificial distinction between different generations, they are allowed to overlap.

By evolution we essentially mean the dynamics of the distribution of the different agent programs within the evolving population over time. The utility of agents changes according to repeated strategic interaction with each other. Simple game theoretical models describe such interaction, masking most of the details, but nevertheless catching the essential features of agents' programs, and working toward a simpler simulation. We hope thus to model and predict several interesting real world situations more precisely.

The inspiration for our model was drawn mainly from evolutionary game theory, and the seminal experiments of Robert Axelrod with the **T**it-**F**or-**T**at (**TFT**) strategy in the repeated **P**risoner's **D**ilemma (**PD**) [8]-[10]. The rest of the paper is organized as follows: Section 2 introduces the background of the evolutionary simulator, Section 3 describes its concept and implementation, Section 4 presents and evaluates some essential experiments, Section 5 contains conclusions and briefly outlines future research.

## Background

In the following, we will briefly summarize those approaches, which mainly influenced our model. The purpose of this is to introduce some fundamental concepts, and to enable later discussion of similarities and differences between them and our approach.

### Axelrod's experiments

The goal of Axelrod's experiments was to find the program (the algorithm) out of a given set of programs which plays the repeated PD game (cf. Table 1) most efficiently. Programs were compared pairwise. Every program played against each other a fixed (but previously unknown) number of rounds. In every round they had to choose between two strategies (cooperate, or defect), and got their respective payoff according to the collective choice.

| Player 2<br>Player 1 | Defect | Cooperate |
|---|---|---|
| Defect | 1; 1 | 5; 0 |
| Cooperate | 0; 5 | 3; 3 |

**Table 1**: Payoff matrix of a "Prisoner's Dilemma" game

TFT, a simple program, which initially cooperates, and then repeats the previous choice of its opponent, won the tournaments by collecting the most at the end. Axelrod concluded, that because of the importance of PD as a model of social interaction, the core characteristics of cooperation in general must be those found in the TFT. He then conducted other experiments too, called ecological and evolutionary analysis, and again confirmed the success of TFT.

### Evolutionary game theory

Another source of ideas was evolutionary game theory, which in contrary to Axelrod's results, enables formal analysis and prediction of evolving systems (although only for relatively simple cases).

For example, let's suppose that we have an infinite population of agents, who strive for resources. The game is divided into rounds, and in every round every agent randomly (according to uniform distribution) meets an other agent to decide upon a resource of value V>0. For the sake of simplicity let's say, that there are only two types of agents: hawks (aggressive), and doves (peaceful). When two hawks meet, they fight for the resource, which has a cost C, and so they get (V-C)/2 per head. When two doves meet, they divide the resource equally between each other without fighting (they get V/2 per head). When a hawk meets a dove, then the hawk takes the resource (gets V), while the dove is plundered (gets 0). This situation is simply modeled by the **H**awk-**D**ove (**HD**) game (cf. Table 2) [11].

| Player 2<br>Player 1 | Hawk | Dove |
|---|---|---|
| Hawk | (V-C)/2; (V-C)/2 | V; 0 |
| Dove | 0; V | V/2; V/2 |

**Table 2**: Payoff matrix of a "Prisoner's Dilemma" game

The gained payoffs are collected over rounds, and the proportion of hawks and doves in the population depends on their average collected payoff. It can be shown, that the only reasonable attractor (i.e. state to which this discrete dynamic system converges) is where only hawks remain in the population.

## The simulation model

Many interesting results can be obtained by using the previous approaches, although the necessary assumptions are usually unrealistic, and overly simplified (fixed or infinite number of agents; simple programs, that can be handled analytically; etc). For more realistic and complex cases, with arbitrary programs (like in the ecological analysis of Axelrod) and finite, overlapping generations of varying size, we need to use simulations.

The proposed agent-based simulation model combines the advantages of the previous approaches without their drawbacks. It resembles artificial life in many aspects, but it is different in its purpose (it tries to capture the key features of not only biological, but also technical systems' evolution) [12]. It is an extension to the previous approaches, differing from them mainly in the following. Populations are finite, and vary in size; agents are modeled individually; the selection mechanism, and the fitness of agents is not explicitly given, but emerges as a product of agents' features, and their interaction in the environment. These differences make the model more realistic.

### Concept

The basis of the model is an intuitive combination and extension of the ideas discussed in Section 2. The simulation is divided into rounds. There is a finite number of agents in the population, who are randomly paired in every round (according to uniform distribution) to play a 2-person game in the role of one of the players (the role is chosen randomly too). Every agent of the population plays the same type of game in every round of a run (e.g. just PD, or just HD), and each of these agents has a program for selecting its strategy in these plays (e.g. TFT, Random, Always-Cooperate, Always-Defect). After a pair of agents finished to play in a given round, the respective (possibly negative) payoffs are added to their individually cumulated utility. If their utility gets below a level (e.g. zero), then the agent dies, i.e. it instantly disappears from the population, and won't play in the following rounds; otherwise it remains, and may even reproduce depending on its reproduction strategy. This strategy defines how and when to reproduce. Only asexual proliferation, i.e. replication without change is considered. After every agent finished the given round (died, survived, or even replicated), comes the next round.

Two types of reproduction are considered: type 1 is called "natural", and type 2 is called "technical". Agents with type 1 reproduction strategy can have only a limited number of offsprings in their lifetime (maximum one per round). They replicate, if their utility exceeds a given limit (limit of replication). After replication, their utility is decreased with the cost of replication (which is usually equal to the limit of replication). Offsprings start with zero utility, and the same program, and features, as their parents originally (i.e. the same lower limit of utility necessary for survival, limit and cost of replication, and limit on the number of offsprings). On the other hand, agents with type 2 reproduction strategy can have unlimited offsprings (but maximum one per round). They also replicate when their utility exceeds the limit of replication, but this limit is doubled every time after an offspring is produced, and their utility does not decrease after replication.

Offsprings start with the same utility, program, and features, as their parents at the moment of replication (i.e. the same lower limit of utility necessary for survival, and limit of replication). The rationale of differentiating several types of reproduction is to enable the distinction between modeling the evolution of biological, and artificial (e.g. software) systems.

### Implementation

The proposed simulation model was implemented in **JADE** (**J**ava **A**gent **DE**velopment) framework [13]. It is an open-source, Java-based, platform independent, distributed middle-ware and **API** (**A**pplication **P**rogramming **I**nterface) complying with the **FIPA** (**F**oundation for **I**ntelligent **P**hysical **A**gents) standards [14]. It enables relatively fast and easy implementation of physically distributed, asynchronous, high-level multi-agent systems.

The implemented software architecture was aimed to be fast and simple. It consisted of only two JADE agents: a `GameAgent` (`GA`), and a `PlayerAgent` (`PA`). `GA` was responsible for conducting the runs, and orchestrating `PA`-s, while `PA`-s were the actual agents in the population, who were paired in each round to play a given 2-person game.

Each JADE agent had a variety of (mostly optional) startup parameters, which in case of a GA set the type of the game to be played (e.g. PD, or HD, or else), the maximal number of agents in the population, and the maximal number of rounds in the run. The OR-relation of the latter two defined the termination criteria of a run. The startup parameters of a PA set agents' program and reproduction strategy, initial utility, the lower limit of utility, the limit and cost of reproduction, the limit on the number of offsprings, and the capacity of memory. The latter was needed because each agent had to be able to use its percept history in order to decide upon the strategy to be played in a given round. The percept history of an agent associated a series of events (information about past plays) to agent identifiers (ID-s). There was a limit on the maximal length of these series, and the maximal number of ID-s. If any of these limits was exceeded, then the oldest element was replaced by the new one.

Now the simulation went as follows. First a given number of PA-s were started on the JADE agent platform (constituting the initial population), followed by a GA, who at the beginning of every round first searched the platform for available PA-s (because later there may have been newly born agents, or some of them disappeared). Then the GA made a pairing of the PA-s found, and informed these pairs about the game to be played (who plays with whom, and in what role). The pairs of PA-s then replied to the GA with the ID of their chosen strategy respectively. The GA then calculated the agents' respective payoff accordingly, and informed them about it. This was repeated until the termination criterion of the simulation was satisfied. Several interesting experiments were conducted this way. Some of them are explained in the following section.

The complexity of the implemented model is additive. Since all PA agents run in parallel, it depends solely on the sum of the complexity of the GA and PA agent.

## Experimental results

The experiments consisted of running the simulation described above with several different initial populations and games to observe the changes in the number, proportion, and average utility of the different types of agent programs. Each experiment had its own settings, but a part of them was the same in every case. The maximal number of agents was 800; the maximal number of rounds was 250; the maximal number of offsprings was 3; the limit and the cost of reproduction was 20; the lower limit of agents' utility and their initial utility was 0; the maximal number of percept histories (about different opponents) was 1000; and the limit on the length of such a percept history was 4 for every agent in every experiment. Everyone was playing in every round (except when the number of agents was odd).

In the following we will describe these experiments grouped according to the games the agents' were playing. Five elementary games are examined: **P**risoner's **D**ilemma (**PD**), **C**hicken **G**ame (**CG**), **B**attle of **S**exes (**BS**), **L**eader **G**ame (**LG**), and **M**atching **P**ennies (**MP**).

During the experiments agent programs were drawn from a fixed set. Only the following programs were studied yet: Always-Cooperate, Always-Defect, TFT, and Random. Nonetheless both types of agents' reproduction strategy were examined. All in all, this configuration was more than enough to run insightful experiments comparable to the previous approaches discussed in Section 2.

### Prisoner's Dilemma game

PD is one of the most popular 2-person games in game theory [15]. It is a special case of the HD game, when $V > C \geq 0$ (cf. Table 2). The original story of the game is essentially about two prisoners, who are put in separate cells (cannot communicate), and are asked to simultaneously decide, whether to cooperate, or defect. The best outcome is defecting, when the other player cooperates, and it is the worst outcome for the other. It is better if both defect, and even better, if both cooperate. The game is called a "dilemma" because its only **N**ash **E**quilibrium (**NE**) [16] is the sub-optimal Defect-Defect outcome.

For example, if the payoffs are chosen according to the HD game, where V=4, C=2 (and so it becomes a PD), and if the initial population consists of altogether 6 agents: 3 Always-Cooperate, and 3 Always-Defect agents, then the proportions of the different agent programs change according to Fig. 1, which is in accordance with the predictions of Section 2/B. Defective agents (hawks) infest the population, and the proportion of cooperative players (doves) steadily decreases. The reproduction strategy of agents in Fig. 1 is of type 1 ("natural"), but essentially the tendencies are the same in case of type 2 ("technical").

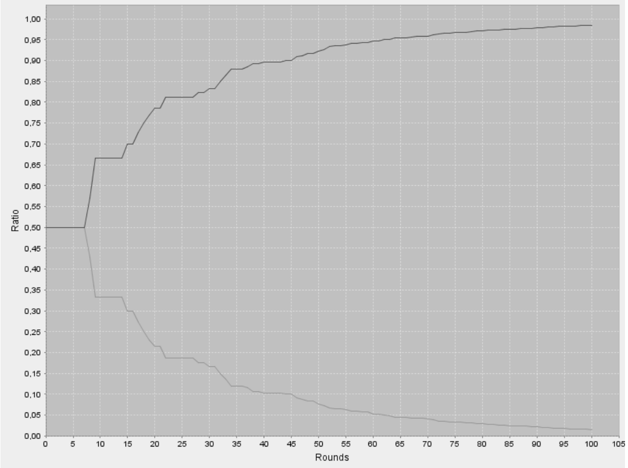

**Figure 1**: Change of proportion of coop. (gray) and def. (black) programs in PD

Fig. 2-3 show the change of quantity and average utility of agent programs, if the initial population consists of 3 Random and 3 TFT agents, and reproduction is "natural" The quantity of the corresponding subpopulations ("species") does not decrease because there are no negative payoffs in the game, and so agents cannot die since their cumulated utility cannot decrease below the lower limit.

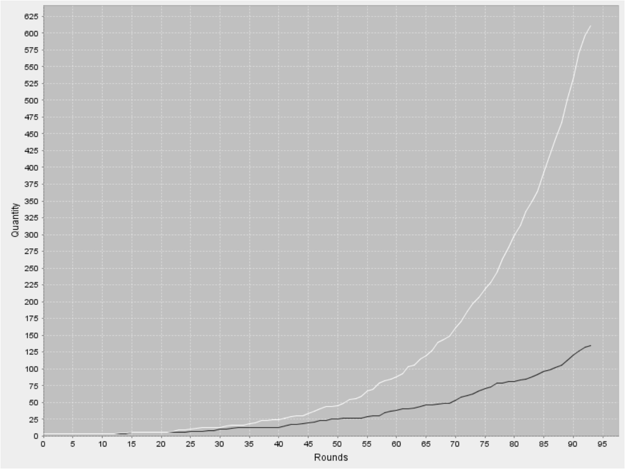

**Figure 2**: Change of quantity of Random (white) and TFT (black) programs in PD

According to Fig. 2, Random agents typically outperform TFT agents by far. This is the case with both types of reproduction. Similarly, Always-Defect agents also outperform TFT agents. These observations seem to differ from the results mentioned in Section 2/A. Moreover, according to Fig. 3, the change of subpopulations' proportion isn't in direct proportionality with the ratio of their average utility and the average utility of the whole population, as predicted by replicator dynamics in evolutionary game theory [17].

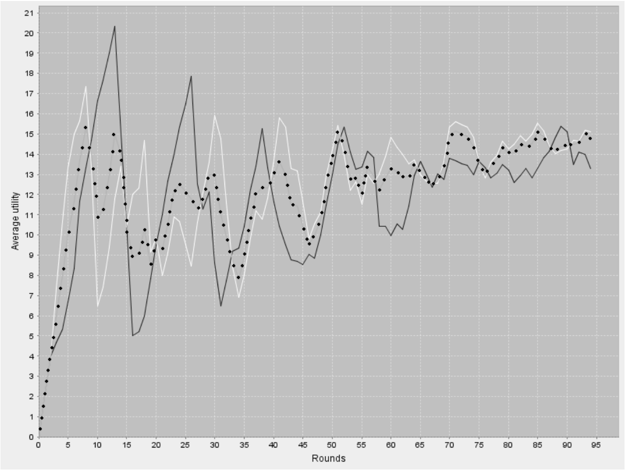

**Figure 3**: Change of average utility of Random (white) and TFT (black) programs, and the whole population (dotted) in PD

If the population consists of 50 TFT agents and 1 Always-Defect intruder, then Axelrod's ecological analysis (cf. Section 2/A), which is also based on replicator dynamics, predicts the extinction of the invading defectors [9]. But our experiments show the opposite (with both types of reproduction). The proportion of defectors is steadily growing until they finally overtake the whole population.

All of the results mentioned above are typical in the sense that they are indifferent to parameter changes (e.g. different runs, changing the size of the initial population; changing the type, limits, or cost of reproduction, or the payoff values in the game). TFT agents can be made a little better by increasing the size of their memory, but in the end it doesn't change the overall tendencies. These observations may also help to explain the scarce evidence of TFT-like cooperation in nature [18].

### Chicken game

This game is also a special case of the HD game, when $C > V$ (cf. Table 2) [15]. The original story is about two cars driving toward each other. If both drivers are "reckless" (i.e. defect), and won't swerve away, it is the worst outcome, since they crash. Better is, if one swerves away (being cooperative, or "chicken"), while the other wins (best outcome). But it is better for the former, if the latter swerves away too. This is called a mixed motive game, because it has two NE-s (those outcomes, when players do the opposite).

Experiments with this game showed different results than in case of PD in Section 4/A. The main reason for that is that the payoffs were chosen according to HD game, where V=2, C=4, and so agents could die because of negative payoffs. Always-Cooperate (i.e. Always-Chicken) proved to be the best (most proliferating) program out the four studied alternatives if there were only two types of programs in the initial population. Always-Reckless and TFT claimed the second place, while Random was the worst. This means that if the cost of being mutually defective is beyond the achievable value ($C > V$), then it becomes too risky not to cooperate. It was interesting to observe, that if Always-Reckless proved to be the winner of a situation (i.e. if it extinguished all other "species"), then it too died out. In this aspect Always-Reckless is "parasite", that exploits the other subpopulations from whom its survival depends. Experiments with more than two types of agent programs were rather unpredictable. They depended mostly on the actual pairing of the individuals in the first dozen of rounds.

### Battle of Sexes game

This game is also a mixed motive game, like CG, but it differs from the previous games in that it is asymmetric by default (cf. Table 3) [3]. The original story is about a husband and a wife, who must choose between going to a football match, or an opera. The husband would better like to go to the football match, while his wife would better go the opera. But, in any case, it is more preferable for them to go together, than to go alone.

| Wife<br>Husband | Opera | Football |
|---|---|---|
| Opera | 1, 2 | -1, -1 |
| Football | 0, 0 | 2, 1 |

**Table 3**: Payoff matrix of a "Battle of Sexes" game

Cooperation is different in case of the husband, than in case of the wife. They cooperate, if they try to do what is best for the other, and that is the worst (husband goes to opera, and wife goes to football). So Always-Cooperate, and Always-Defect strategies are a bit more complex now, since they depend also on the actual role of the agents. Moreover, TFT needs also to be revised.

Experiments showed that regardless of the type of reproduction, Always-Cooperate and TFT agents were the worst (others made them die out almost every time). Always-Defect was the best program, and Random was second (since it survived almost every time).

### Leader game

This game is similar to the symmetrical form of BS, with the exception that mutual defection is the worst outcome, and mutual cooperation is better [19].

The name of the game comes from the following situation: two cars wait to enter a one-way street. The worst case is, if they go simultaneously (defect-defect), because they crash. If both wait (cooperate), it is better. But it is even better if they go separately. The one, who goes first, is the best.

According to our experiments, TFT and Always-Cooperate were better, than Always-Defect agents, but Random agents again outperformed TFT agents. The reproduction strategy made a difference in the tendencies, but not in the overall outcome.

### Matching Pennies game

This is, similarly to BS, an asymmetric game, with the exception that it has no symmetric form, and cooperation and defection have no meaning in it [20]. Thus the first (hitherto cooperative) move of TFT doesn’t particularly matter now.

The original game is about two players, who both have a penny. They turn the penny secretly to heads or tails, and then reveal their choice. If the pennies match, one player gets a dollar from the other, else it is conversely.

Our experiments showed that in this scenario Random agents were the fittest for survival (playing the only mixed NE of the game), but in case of type 1 reproduction they died out like all the others. However in case of type 2 (technical) reproduction they could cumulate enough utility to ensure their survival, and start proliferating after a while.

## Conclusions

In this article we presented a novel agent-based simulation model for the real natural selection of arbitrary programs choosing agents’ strategies in repeated 2-person games. Experiments threw new light upon previous results in the field. It was shown, that the proposed simulation model is more realistic and thus useful, than the previous models. Future research will aim at extending these concepts by introducing N-agent interaction, genetic representation and variation of agent programs, and more realistic models of agents' environment and resources.

## References

[1] Russell S., Norvig P.: "Artificial Intelligence: A Modern Approach", Prentice Hall, 2003
[2] Haldane J.: "The theory of natural selection today", Nature, Vol. 183 (4663), pp. 710–713, 1959
[3] Fudenberg D., Tirole J.: "Game theory", MIT Press, 1991
[4] Koza J. R.: "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press, 1992
[5] Ferreira C.: "Gene Expression Programming: A New Adaptive Algorithm for Solving Problems", Complex Systems, Vol. 13:2, pp. 87-129, 2001
[6] Kovacs D. L.: "Evolution of Intelligent Agents: A new approach to automatic plan design", In: Proceedings of IFAC Workshop on Control Applications of Optimization, Elsevier, 2003
[7] Shan Y., McKay R. I., Baxter R., Abbass H., Essam D., Nguyen H. X.: "Grammar Model-based Program Evolution", The Cong. on Evolutionary Comp., 2004
[8] Maynard-Smith J.: "Evolution and the Theory of Games", Cambridge University Press, 1982
[9] Axelrod R.: "The Evolution of Cooperation", Basic Books, 1984
[10] Axelrod R.: "The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration", Princeton University Press, 1997
[11] Maynard Smith J.: "Theory of games and the evolution of animal contests", Journ. of Theor. Biol., Vol. 47, pp. 209-221, 1974
[12] Bedau M.: "Artificial life: organization, adaptation and complexity from the bottom up", TRENDS in Cog. Sci., Vol. 7:11, 2003
[13] Bellifemine F., Caire G., Greenwood D.: "Developing multi-agent systems with JADE", Wiley Series in Agent Technology, 2007
[14] Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org
[15] Poundstone W.: "Prisoner's Dilemma: John Von Neumann, Game Theory, and the Puzzle of the Bomb", Anchor Books, 1992
[16] Nash J. F.: "Non-cooperative games", Annals of Math., Vol. 54:2, pp. 286–295, 1951
[17] Neumann, J. Lohmann G., Derrfuss J., von Cramon D. Y.: "The meta-analysis of functional imaging data using replicator dynamics", Human Brain Mapping, Vol. 25:1, pp. 165-173, 2005
[18] Hammerstein P.: "Why is reciprocity so rare in social animals? A protestant appeal", In: P. Hammerstein, ed., Genetical and Cult. Evolution of Coop., MIT Press, pp. 83–94, 2003
[19] Guyer M. J., Rapoport A.: "Information effects in two mixed motive games", Behav. Sci., Vol. 14, pp. 467-482, 1969
[20] Fudenberg D., Levine D.: "Consistency and cautious fictitious play", Journal of Econ. Dyn. and Cont., Vol. 19:5-7, pp. 1065-1089, 1995