Evolution of Strategies for Public Goods Games

John Maloney
malo0052@umn.edu

In the context of a standard framework for indirect reciprocity [1], the authors of [6] mathematically analyzed all possible combinations of assessment modules and action models in order to identify those combinations that are evolutionarily stable. 25 strategies were identified as evolutionarily stable. Of these 25 stable strategies, eight strategies were able to maintain a high level of cooperation - earning more that 90% of the maximum possible payout. The authors labeled these eight strategies the “leading eight”.

The study conducted in [6] considers the stability of each strategy when invaded by a small number of agents following one other strategy. In the cases considered, both strategies are assumed to use the same assessment module to assign reputations to agents.

In previous experiments [2], the “leading eight” evolutionarily stable strategies were applied to the public goods game. In computer simulations, it was found that, in the context of a public goods game, none of the “leading eight” strategies met the criteria of robustness, stability and initial viability.

In [7] and [8], the authors considered an alternative approach to identifying successful strategies for the indirect reciprocity game. The authors consider a population of agents divided into groups. Each agent is bestowed with a private action module. However, the agents in each group share a common assessment module. Evolution of the agent population occurs at two levels. At one level, the action modules followed by the agents are evolved based on the fitness attained by the individual agents. At the second level, the assessment modules used by each tribe are evolved based on the average fitness attained by the agents in each group. In the context of this framework, the authors found that the assessment module followed by the groups evolved into the *stern-judging* assessment module shown in the following table.

Observed Situation			Stern-Judging
Donor Reputation	Recipient Reputation	Donor Action	
Good	Good	Donate	Good
Good	Good	Refuse	Bad
Good	Bad	Donate	Bad
Good	Bad	Refuse	Good
Bad	Good	Donate	Good
Bad	Good	Refuse	Bad
Bad	Bad	Donate	Bad
Bad	Bad	Refuse	Good

When the groups were limited to using the stern-judging assessment module, the CO and OR action rules, shown in the following table, were the most common action rules to evolve. These two action modules are cooperative.

Situation		Action to Take	
Donor Rep.	Recipient Rep.	CO[1]	OR[2]
Good	Good	Donate	Donate
Good	Bad	Refuse	Refuse
Bad	Good	Donate	Donate
Bad	Bad	Refuse	Donate

As an extension to the study performed in [2], the group evolution procedure employed in [7] and [8] will be used to evaluate the effectiveness of different strategies in the context of the public goods game.

Related Work

In [4], the authors consider indirect reciprocity in a population that is divided among *g* well mixed groups each consisting of *n* members. Dividing the population into multiple groups instead of using a single unified population is intended to limit the effects of genetic drift. Genetic drift can cause the elimination of strategies due to random fluctuations instead of natural selection. A potentially successful strategy that is lost in a one group due to genetic drift can be reintroduced by migration from another group.

During each generation, *m* rounds of donor-recipient interactions are played. For each round, two individuals are chosen randomly with one playing the role of the donor and the other playing the role of the recipient. The authors use a direct observation model in which each agent observes every interaction. Agents assign reputations to each other using a

[1] The CO strategy is also referred to as the "discriminator" strategy.
[2] The OR strategy is also referred to as the "contrite tit-for-tat" strategy.

shared assessment module. Each agent has a private action module that specifies the action the agent should take. However, there is a small probability ε that an agent will take an action that is different from the action prescribed by its action module.

When evolving the agent population, a strategy-based evolution process is used. After the completion of *m* interactions, the local within-group and global cross-group reproductive probabilities for each strategy are calculated. With probability *p*, the strategy followed by an agent in the next generation is selected based on the local within-group fitness scores and otherwise it is selected based on the global cross-group fitness scores. When determining the strategy for an agent in the next generation, a mutation occurs with probability μ. When a mutation occurs, the agent's strategy is selected from among all available strategies with equal probability (independent of the strategy fitness).

The authors conduct two different sets of experiments with different model assumptions. In the first set of experiments, they reassess the results presented in [3] in an environment that limits the impact of genetic drift. To this end, they measure agent reputations on a scale from -5 to +5, assume that all agents use the scoring assessment module and assume that agent perception is error free. In this context, they evaluate the performance of the SELF, CO and AND action modules. Based on the results of experiments, the authors conclude that the results presented in [3] are dependent on the effects of genetic drift and a small cost-to-benefit ratio.

In the second set of experiments, they evaluate the performance of the standing assessment module. In this case, they augment the model used in the previous set of experiments with agents that use the binary image scale and the standing assessment module to assign reputations to agents. Because the authors of [3] claim that the standing assessment module is prone to be affected by perception errors, they include a small probability δ that an agent misperceives the action taken by another agent. Based on the results of experiments, the authors conclude that agents using the standing assessment module out-perform agents following the scoring assessment module.

In [5], the authors use a variation of the approach used in [4]. Differences between the model used in [5] and the model used in [4] includes the following:

- Each agent follows its own assessment module
- All agents measure reputations using the binary image scale
- Individual-based evolution process is used

The authors considered three different scenarios when conducting their experiments. In the first scenario, they consider the scenario in which all 14 strategies are initially present. In the second scenario, they limit the agents to a single assessment module and consider all possible action modules. In the final scenario, they limit the agents to a single action module and consider all possible assessment modules.

For each scenario, they conduct several different sets of experiments making different assumptions about the rate of gene flow between the groups (i.e., the value of *p*), the cost-to-benefit ratio, the mutation rate, the probability an agent observes an interaction, the probability that an execution error occurs and he probability that a perception error occurs. Based on the results of experiments, the authors conclude that, the standing assessment module is the most successful at promoting cooperation.

In [7] and [8], the authors expand the model used in [4] and [5] to investigate the co-evolution of assessment modules and action modules. The expanded model considers evolution at two levels. At the base level, the authors consider the evolution of action modules in the context of a fixed assessment module. On top of this base level, the authors consider the evolution of assessment modules in the context of competition between groups of agents.

The binary image scale is used to measure agent reputations and each agent starts with a good reputation. The indirect observation model is used to assign reputations to agents. With a small probability μ_a the agent selected to observe an interaction misperceives the action taken by the donor and assigns the wrong reputation to that agent. Regardless of whether a perception error occurs or not, the assigned reputation is faithfully distributed to all agents in the group.

Within each group, the agents use a shared assessment module. An assessment module is represented as an eight-bit string with each bit representing the reputation that should be assigned to an agent in each of the eight possible situations that can occur. Initially, each groups is assigned a randomly generated eight-bit string that represents its assessment module.

Each agent has a private action module. An action module is represented as a four-bit string with each bit resenting the action that should be taken in each of the four possible situations that can occur. Initially, each agent is assigned a randomly generated four-bit string that represents its action module. However, with a small probability μ_e the agent fails to donate when its action module specifies that it should.

During each generation, each player interacts with every other player in a round robin fashion. During each interaction, one agent is randomly assigned the donor role and the other agent assumes the recipient role. At the end of each generation, an individual-based evolution process is used to produce the next generation. Unlike in [4] and [5], an agent in the next generation always inherits its strategy locally from its own group.

The agent evolution process employed by the authors is slightly more detailed than the process used in [4] and [5]. An agent's action module is represented as a four-bit string with each bit representing the action that should be taken in one of the four possible situations. Agents in the next generation inherit a strategy through an individual-based evolution process. However, with a small probability μ_s a mutation occurs in each bit of the action module's encoding.

After the next generation has been produced, with a small probability p_c each pair of groups engages in a conflict. The winner of the conflict can be determined using several different methods. One of the methods involves pairwise comparison of the average payouts earned by the agents in each tribe. Given two groups, A and B, let Π_A represent the average payout earned by the agents in group A and Π_B represent the average payout earned by the agents in the group B. Then group B will win the conflict with probability p_w given by the following:

$$p_w = \frac{1}{1 + e^{-\beta(\Pi_B - \Pi_A)}}$$

In this case, β represents the selection strength. This parameter determines how strongly group fitness (measured by a group's average payout) influences selection of the winning group. As β approaches $+\infty$, the probability that the group with the higher average payout wins approaches 1 while as β approaches zero, the probability that either group wins approaches 0.5.

After the winner has been determined, the loser's assessment module is modified so that it becomes more similar to the winner's assessment module. Each group's assessment module is represented as an eight-bit string with each bit representing the reputation that should be assigned to an agent in each of the eight possible situations that can occur. Assume that group B was selected as the winner of the conflict. For each bit in group A's assessment module that differs from the corresponding bit in group B's assessment module, the value of group B's bit is changed to the value of group A's corresponding bit with probability p_b defined as follows:

$$p_b = \frac{\eta \Pi_A}{\eta \Pi_A + (1 - \eta)\Pi_B}$$

In this case, η plays a role similar to β in the previous equation and determines the how strongly group fitness influences whether a bit in the losing group's assessment module is changed.

The probability p_b is only used for the case when the bit in the losing group's assessment module is the same as the corresponding bit in the winning group's assessment module. If the two bits are different then with small probability μ_N a mutation occurs and the bit in the losing group's assessment module is changed to the opposite value.

In addition to evolving its assessment module to be more similar to the winning group's module, the losing group's agent population is subjected to migration in order to evolve the action modules used by losing group's population to be more similar to the action modules used in the winning group's population. For each agent in the winning group, with small probability μ_m the agent's action module replaces the action module of the corresponding agent in the losing group.

Using this expanded model, the authors run simulations to determine which assessment modules come to dominate the agent population. To prevent the simulations getting stuck in local optimums, the authors set the mutation rate at a relatively high level. This has the side effect of preventing the assessment modules from ever fixating. Therefore, the authors use statistical analysis to determine when a particular assessment module has fixated in the population. The authors analyze the bits making up the assessment modules used by all the groups and consider a bit value to be fixed if it is present in more than 98% of the group's assessment modules. Using this technique, the authors find that the *stern judging* assessment module is ubiquitous throughout all the groups.

The authors conduct additional simulations with stern judging providing a fixed context for reputation assessment and find that over 70% of the agents in the population evolve to use the cooperative strategies CO and OR.

Evolving Strategies for Indirect Reciprocity

As described above, in [7] and [8], the authors investigated the co-evolution of assessment modules and action modules in the context of the indirect reciprocity game. This section reviews the definition of the indirect reciprocity game and presents the results of activities undertaken to reproduce the results in [8].

Donor-Recipient Game

In the *donor-recipient game*, two players are paired together with one player taking the role of the donor and the other player taking the role of the recipient. The donor can choose to donate at a cost of c to itself or abstain from donating. If the donor donates, the recipient receives a benefit $b > c$. Otherwise, both players receive zero payout. In the one-shot version of the game, the obvious choice is to not donate. However, in the repeated version of the game, it can be beneficial to cooperate.

Consider a large population of agents. For a single round of the *repeated donor-recipient game*, random pairs of agents are created. For each pair, one agent is selected to be the donor and the other is selected to be the recipient. Each pair plays a one-shot donor-recipient game. This process is repeated until the specified stopping criteria are satisfied.

In general, all agents are selected an equal number of times for the donor and recipient roles. Since the benefit received by the recipient is greater than the cost of the donation provided by the donor, the agents receive the highest average payout if all agents donate. However, the temptation to defect and receive benefits without paying the cost still exists, creating a dilemma similar to the dilemma that exists in the iterated prisoner's dilemma.

Since the population is large, an agent cannot expect to be paired with an agent more than once. Therefore, indirect reciprocity must guide the agents' behavior if cooperation is to emerge and the driving force behind indirect reciprocity is reputation [16].

Experimental Procedure

Based on the descriptions provided in [7] and [8], code was developed to reproduce the experiments described by the authors. There are some differences in the way the simulations were conducted for the two papers. For this study, the description provided in [8] is followed.

Simulations were conducted using the following parameter values:

Parameter	Description	Value
b	Benefit received from donation	17
c	Cost of providing benefit to recipient	1
μ_a	Reputation assessment error	0.001
μ_e	Action execution error	0.001
μ_s	Action module mutation rate	0.01
μ_N	Assessment module mutation rate	0.0001
μ_m	Migration probability	0.005
p_c	Probability of group conflict	0.01
β	Selection strength in winner determination	10^4
η	Selection strength in bit replacement	0.15

To conform to the procedure described in [8], 500 simulations were executed[3]. Each simulation consisted of consisting of 10,000 generations of 64 groups with 64 agents per group. For each generation, statistics for assessment modules and actions modules were recorded. For assessment modules, for each bit value in the representation of an assessment module, the number of groups for which the bit was set to 1 (or GOOD) was recorded. For action modules, the number of agents following the assessment module independent strategies ALLD or ALLC was recorded.

[3] Or, will be executed…

After the completion of each simulation, the statistics collected for the final 1000 generations were evaluated to determine whether prevalence of either of the assessment module independent strategies ALLD or ALLC exceeded 10%. Of this threshold was exceeded then the results of that simulation were not included in the following analysis.

If the ALLD/ALLC threshold was not exceeded then the statistics collected for the final 1000 generations were used to determine whether each assessment module bit value fixated and if so the value at which it fixated. Let $b_i^{(g)}$ be the count collected for the bit in location *i* for generation g and let *N* be the total number of groups. The frequency of occurrence of value 1 (or GOOD) in location *i* is given by the following formula:

$$f_i^{(1)} = \frac{\sum_g b_i^{(g)}}{\sum_g N}$$

And the frequency of occurrence of value 0 (or BAD) in location *i* is given by the following formula:

$$f_i^{(0)} = \frac{\sum_g \left(N - b_i^{(g)}\right)}{\sum_g N}$$

Where in each case, the sum is over the subset of the last 1000 generations of the simulation. Given these frequency values, the value of the bit at location *i* fixates at value 1 if $f_i^{(1)} > 0.98$ and fixates at value 0 if $f_i^{(0)} > 0.98$. If neither of these conditions hold, then the bit does not fixate and is given the value "X".

After determining the fixation values for each simulation, the results are aggregated to produce the overall results for the experiment. For each bit location *i*, let $\varphi_i^{(1)}$ be the number of simulations for which the bit fixated at value 1, $\varphi_i^{(0)}$ be the number of simulations for which the bit fixated at value 0 and $\varphi_i^{(X)}$ be the number of simulations for which the bit did not fixate. In this case, the bit values B_i for the optimal assessment module are determined using the following formula:

$$B_i = \begin{cases} 1, & \varphi_i^{(1)} > \varphi_i^{(0)} + \varphi_i^{(X)} \\ 0, & \varphi_i^{(0)} > \varphi_i^{(1)} + \varphi_i^{(X)} \\ \cdot, & otherwise \end{cases}$$

Results

Implementation of the software required to complete the experiments has been completed. The results of the reproduced experiment are pending the completion of the execution of the required simulations.

Evolving Strategies for Public Goods Games

This section reviews the definition of a public goods game and presents an extension of the strategy evolution framework presented in [7] and [8] to the domain of public goods games.

N-Person Prisoner's Dilemma

The evolution of cooperation in public goods games is often analyzed using the framework of an *n-person prisoner's dilemma game*. In this game, *N* players are grouped together and given the opportunity to contribute to a common pool whose contents will be multiplied by a factor *r* and distributed evenly among all *N* players. Each player can choose to cooperate by contributing c to the common pool or to defect and incur zero cost. The payouts paid to the participants depend on the number of cooperators. Given N_c cooperators, the payouts are the following:

$$P_d = \frac{r \cdot c \cdot N_c}{N}$$

$$P_c = \frac{r \cdot c \cdot N_c}{N} - c$$

Where P_D is the payout earned by defectors and P_C is the payout earned by cooperators.

When played as a single-shot game, the rational choice is to defect. However, when played as a repeated game, it is possible for cooperative strategies to achieve higher average payouts than unconditional defection[10][11][12][13].

Given a large well-mixed population of agents, in the *repeated n-person prisoner's dilemma game*, N agents are selected randomly from the population and given the opportunity to play a one-shot n-person prisoner's dilemma game. This process is repeated until the specified stopping criteria are satisfied.

Some variants of the n-person prisoner's dilemma game include a post-payout step in which each agent is allowed to punish defectors [11][13][14][15]. Some variants also provide players with the option to abstain from participating [12][13]. These non-participants neither contribute to the common pool nor receive a distribution after the pool is multiplied.

In variants that provide the option to punish, the payout received by a player that chooses to punish is reduced by an amount γ for each player that chooses to defect while the payout received by each player that chooses to defect is reduced by an amount β for each player that chooses to punish. In variants that provide the option to abstain, the non-participants are provided with a constant payout σ. Therefore, the payouts are the following:

$$P_C = \frac{r \cdot c \cdot (N_C + N_P)}{(N_C + N_P + N_D)} - c$$

$$P_D = \frac{r \cdot c \cdot (N_C + N_P)}{(N_C + N_P + N_D)} - \beta N_P$$

$$P_A = \sigma$$

$$P_P = \frac{r \cdot c \cdot (N_C + N_P)}{(N_C + N_P + N_D)} - c - \gamma N_D$$

Where P_A is the payout earned by non-participants and P_P is the payout earned by punishers.

Modeling Agent Strategies in the Public Goods Game

To extend the agent strategy formalism described in [7] and [8] to the domain of public goods games, the concept of reputation needs to be extended to groups and the bit-string approach used to represent assessment and action modules needs to be extended to cover the additional actions that are available in public goods games.

Reputation Model for Groups

In [2], the concept of reputation was extended to groups as follows. Let Γ be a group of agents, N be the size of the group, $bad(\Gamma)$ be the number agents in the group with a “bad” reputation and $T \in [0,1]$. Then the reputation $R(\Gamma)$ of the group is given by the following:

$$R(\Gamma) = \begin{cases} G, & \dfrac{bad(\Gamma)}{N} \leq T \\ B, & \dfrac{bad(\Gamma)}{N} > T \end{cases}$$

T is a threshold parameter that determines the *tolerance* that an agent has for agents with a “bad” reputation. The threshold T would be set to zero for an agent that is completely intolerant of bad agents and set to one for an agent that is completely tolerant of bad agents.

While the threshold represents a continuous range form zero to one, for purposes of the experiments conducted in this study, the range will be discretized into four values: 0, 0.25, 0.5, 0.75 and 1. Given this discretization, the threshold can be represented using two bits.

Action Modules in Public Goods Games

In the public goods game, an agent needs to make two decisions during the course of the game:

- Choose type of game participation. The agent can choose between three actions:
    - Cooperate: participate and contribute to the common pool
    - Defect: participant but do not contribute to the common pool
    - Abstain: refuse to participate in the game
- Choose whether to punish defectors

Given this formulation, there are six different action combinations the agent can select.

As with the indirect reciprocity game, the agent’s action choice depends on its own reputation and the reputation of its co-player. In the case of the public good game, the agent’s co-player is a group of agents and the group’s reputation is assigned as described in the previous section. Therefore, the agent can distinguish four different situations.

In each situation, the agent can choose one of the six possible action combinations. Therefore, there are $6^4 = 1296$ possible action modules in the public goods game. Each possible action module can be represented using a string consisting of four trits[4] and four bits. For each of the four possible situations that an agent can distinguish, one trit specifies the type of game participation (cooperate, defect or abstain) that the agent should choose and one bit specifies whether the agent should punish defectors.

Assessment Modules in Public Goods Games

In the public goods game, as in the indirect reciprocity game, one of two reputations is assigned to the game participants: good or bad. The reputation assigned to an agent depends on that agent’s reputation, the group’s reputation and the action taken by that agent. Therefore, that agent can distinguish 24 different situations.

[4] https://en.wikipedia.org/wiki/Ternary_numeral_system

In each situation, the agent can choose to assign one of the two available reputation scores. Therefore, there are $2^{24} = 16{,}777{,}216$ possible assessment modules in the public goods game. Each possible assessment module can be represented using a string consisting of 24 bits where each bit specifies the reputation that should be assigned in each possible situation.

Agent Strategies in Public Goods Games

An agent strategy in the public goods game consists of group reputation threshold, an action module and an assessment module. Since there are 4 possible threshold values, 6^4 possible action modules and 2^{24} possible assessment modules, there are more than 2^{36} possible agent strategies.

Size of Search Space

As can be seen from the discussion above, the space of possible strategies for a public goods game is vastly larger than the space of possible strategies for the indirect reciprocity game. For the indirect reciprocity game there are 256 possible assessment modules and 16 possible action modules leading to 4096 possible strategies. This is significantly smaller than the 2^{36} possible strategies that exist for the public goods game.

In [6], the authors analyzed all possible agent strategies for the indirect reciprocity game in order to determine which strategies are stable. In their analysis, they did not consider strategies that are mirror images of each other. A strategy *s'* is a mirror image of a strategy *s* if switching the reputation scores that appear in strategy *s'* to their opposite value causes strategy *s'* to become equivalent to strategy *s*. In the context of the author's analysis, a mirror image strategy has the same properties as the original strategy and therefore does not need to be analyzed separately. Using mirror symmetry allowed the authors to reduce the number of strategies analyzed by half.

Unfortunately, in the context of the current study, using mirror symmetry to reduce the size of the search space is not possible. In the context of a computer simulation, a mirror image strategy is very different from the original strategy. Consider the following common assessment modules and their mirror images:

Observed Situation			Scoring	Scoring Mirror	Standing	Standing Mirror
Donor Reputation	Recipient Reputation	Donor Action				
Good	Good	Donate	Good	Bad	Good	Bad
Good	Good	Refuse	Bad	Good	Bad	Good
Good	Bad	Donate	Good	Bad	Good	Bad
Good	Bad	Refuse	Bad	Good	Good	Good
Bad	Good	Donate	Good	Bad	Good	Bad
Bad	Good	Refuse	Bad	Good	Bad	Bad
Bad	Bad	Donate	Good	Bad	Good	Bad
Bad	Bad	Refuse	Bad	Good	Bad	Good

It is obvious that the mirror image assessment modules will lead to very different agent behavior than the original assessment rules.

In [9], the authors claim that genetic algorithms are commonly used to solve problems whose search space size is at least 2^{30}. Therefore, it is hoped that the size of the search space required for agent strategies in the public goods game will not pose a problem.

Experimental Procedure

A similar experimental procedure to the procedure used in [7] and [8] will be used to evaluate the evolution of agent strategies in the public goods games. Given the vastly larger size of the search space, the number of groups, number of agents per group and the number of generations required to reach a steady state will need to be increased.

Results

The results of this experiment are pending the completion of software implementation and execution of the required simulations.

References

[1] Maloney, J., “A Framework for Indirect Reciprocity”, Unpublished, 2016.

[2] Maloney, J., “Social Norms in Public Goods Games”, Unpublished, 2015.

[3] Nowak, M. A., and K. Sigmund, “Evolution of indirect reciprocity by image scoring,” *Nature*, vol. 393, pp. 573-577, 1998.

[4] Leimar, O., and P. Hammerstein, “Evolution of cooperation through indirect reciprocity,” *Proceedings of the Royal Society London B*, vol. 268, pp. 745-753, 2000.

[5] Brandt, H., and K. Sigmund, “The logic of reprobation: assessment and action rules for indirect reciprocation,” *Journal of Theoretical Biology*, vol. 231, pp. 475-486, 2004.

[6] Ohtsuki, H., and Y. Iwasa, “How should we define goodness? – reputation dynamics in indirect reciprocity, “ *Journal of Theoretical Biology*, vol. 231, pp. 107-120, 2004.

[7] Chalub, F. A. C. C., F. C. Santos, and J.M. Pacheco, “The evolution of norms,” *Journal of Theoretical Biology*, vol. 241, pp. 233-240, January 2006.

[8] Pacheco, J. M., F. C. Santos, and F. A. C. C. Chalub, “Stern-Judging: A Simple, Successful Norm Which Promotes Cooperation under Indirect Reciprocity,” *PLoS Computational Biology*, vol. 2, issue 12, December 2006, pp. 1634-1638.

[9] Whitley, D., “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, no. 2, pp. 65-85, June 1994.

[10] Boyd, R., and P. J. Richardson, “The evolution of reciprocity in sizable groups,” *Journal of Theoretical Biology*, vol. 132, pp. 337-356, 1988.

[11] Boyd, R., and P. J. Richardson, “Punishment allows the evolution of cooperation (or anything else) in sizable groups,” *Ethology and Sociobiology*, vol. 13, pp. 171-195, 1992.

[12] Hauert, C., S. De Monte, J. Hofbauer, and K. Sigmund, “Replicator dynamics for optional public good games,” *Journal of Theoretical Biology*, vol. 218, pp. 187-194, 2002.

[13] Hauert, C., A. Traulsen, H. Brandt, M. A. Nowak, and K. Sigmund, “Via freedom to coercion: the emergence of costly punishment,” *Science*, vol. 316, pp. 1905-1907, 2007.

[14] Brandt, H., C. Hauert, and K. Sigmund, “Punishing and abstaining for public goods,” *Proceedings of the National Academy of Science*, vol. 103, no. 2, pp. 495-497, 2006.

[15] Fowler, J. H., “Altruistic punishment and the origin of cooperation,” *Proceedings of the National Academy of Science*, vol. 102, no. 19, pp. 7047-7049, 2005.

[16] Nowak, M. A., “Five Rules for the Evolution of Cooperation,” Science, vol. 314, pp. 1560-1563, 2006.