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] contains the following limitations:

* Only the stability of a strategy is considered. The stability of a strategy is its ability to resist invasion by other strategies once it has become established in a population. The study assumes that a population composed of a majority of agents following one action rule is invaded by a small number of agents following one other action rule.
* All the agents in the population are assumed to share the same assessment module

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. However, the analysis performed in this study did not impose the same limitations that were imposed in original analysis that identified the “leading eight”. Each strategy was evaluated in an environment that included unconditional cooperators and unconditional defectors in addition to agents following the subject strategy.

In [7] and [8], the authors consider a population of agents divided into groups. Each agent is bestowed with a private action rule. However, the agents in each group share a common assessment module. After the generation of a new generation of agents, the assessment rules followed by each group were evolved based on the overall fitness achieved by each group. The authors found that the assessment module followed by 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 |

The CO and OR action rules, shown in the following table, were the most common action rules to co-evolve along with the stern-judging assessment rule. These two action modules are cooperative.

|  |  |  |  |
| --- | --- | --- | --- |
| **Situation** | | **Action to Take** | |
| **Donor Rep.** | **Recipient Rep.** | **CO[[1]](#footnote-1)** | **OR[[2]](#footnote-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.

Within each group, the indirect observation model is used to assign and distribute a shared opinion of each agent’s reputation among all members of the group.

# 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. 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 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 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 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 a mutation occurs in each bit of the action module’s encoding.

After the next generation has been produced, with a small probability each pair of groups engages in a conflict. The winner of the conflict can be determined in several different ways. 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.

The group evolution process works as follows. 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. Let represent the average payout earned by agents in the winning group and represent the average payout earned by agents in the losing group. For each bit in the losing group’s assessment module that differs from the corresponding bit in the winning groups assessment module, the bit is changed to the value of the corresponding bit in the winning group’s assessment module with probability *p* defined as follows:

For each bit in the losing group’s assessment module that is the same as the corresponding bit in the winning group’s assessment module, with small probability a mutation occurs and the bit 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 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 rather 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 assessment module shown in the following table is ubiquitous throughout all the groups. They call this assessment module *stern judging*.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

With stern judging providing the context for reputation assessment, the authors find that the cooperative strategies CO and OR are used by over 70% of the agents in the population.

# 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

(incorporate description from [2] along with references)

## 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, be the number agents in the group with a “bad” reputation and . Then the reputation of the group is given by the following:

*T* is a threshold parameter that determines the *tolerance* that an agent has for agents with a “bad” reputation. For an intolerant agent, T would be set equal to zero.

## Assessment & Action Modules in Public Goods Games

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.

For the indirect reciprocity game, a third-order assessment module assigns a reputation to a donor given the donor’s reputation, the recipient’s reputation and the action taken by the donor. Since each agent can be assigned one of two reputations and the donor can take one of two actions, there are a total of eight situations that must be distinguished. Likewise, a second-order action module specifies the action a donor must take given its own reputation and the recipient’s reputation. Given two possible reputations for each agent, there are a total of four situations that must be distinguished.

## Problem Encoding

In [7] and [8], the authors represent assessment and action modules using a bit strings. Since, an assessment module assigns one of two reputations to an agent, a single bit is required for each situation. Given eight situations to distinguish, an eight-bit string is required to represent an assessment module. Since an action module specifies one of two actions to take, a single bit is required for each situation. Given four situations to distinguish, a four-bit string is required to represent an action module.

For the public goods game, an agent can still be assigned only one of two reputations. However, an agent can choose between four actions instead of two. Given the expansion in the number of available actions, a third-order assessment module must distinguish between 16 different situations. Therefore, a 16-bit string is required to represent assessment modules for a public goods game. The number of situations that need to be distinguished by an action module is unchanged. However, the expansion in the number of available actions means that two bits are required for each situation. Therefore, an eight-bit string must be used to represent action modules for a public goods game.

Besides encoding the assessment and action modules, the threshold used to determine whether a group has a good reputation must also be encoded so that an appropriate threshold can be arrived at through the evolution process. The threshold *T* will be treated as a percentage that ranges from zero to one and only integer percentages will be considered. Therefore, seven bits are required to represent the threshold used by an agent. Seven bits actually represents the values zero to 128 therefore a mechanism will need to be designed to handle the additional 28 values that can be represented by the encoding.

## 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. For the public goods game there are 216 possible assessment modules, 256 possible action modules and seven possible threshold values leading to 229 possible strategies.

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 230. Therefore, it is hoped that the size of the search space required by the encoding used for agent strategies will not pose a problem.

# 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.

1. The CO strategy is also referred to as the “discriminator” strategy. [↑](#footnote-ref-1)
2. The OR strategy is also referred to as the “contrite tit-for-tat” strategy. [↑](#footnote-ref-2)