Final Year Project Report

Full Unit - Interim Report

Cooperative Strategies in Multi-Agent Systems

James King

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: Kostas Stathis

Department of Computer Science

Royal Holloway, University of London

December 5, 2018

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:

Student Name: James King

Date of Submission:

Signature:

Table of Contents

Abstract

Chapter 1: **Aims, Objectives and Literary Survey**

1.1 The Problem

To understand the aims and objectives of this project we must first grasp the history of the study of interaction between biological and computational agents (often known as intelligent agents in the field of computer science). It is important to distinguish between biological and computational agents as the study of natural selection and the evolution of cooperation in biological agents has a long and separate history from the study of intelligent agents in computer science.

The background I am about to explain is covered in my first two reports (included in the appendix in chapter 5): 'report on the evolution of cooperation in relation to game-theory and different game-theoretic mechanisms to aid it' and 'report on indirect reciprocity, strategies for agents and the development of a concrete model to implement'. I will be applying this knowledge in a more convenient format for the interim report here, but if you are interested in reading a more in-depth study by myself on this topic, I suggest reading these two.

As explained in my first report (on the evolution of cooperation in relation to game-theory and different game-theoretic mechanisms to aid it) early Darwinian evolutionary theory was groundbreaking, but both opponents and even proponents of evolution such as Peter Kropotkin [8] (a Russian evolutionary scientist and political activist) found it fell short in explaining cooperative phenomena.

Cooperation is the idea of helping others at a detriment or cost to yourself. Of course, this is a key part of the natural world, like the act of parents supporting their children and vice-versa in human family groups. There are even interspecies examples such as that of the Meerkat and Drongo Bird [1].

The problematic question posed by this phenomena is that natural selections idea: the survival of the fittest [15], pushes individuals to compete for resources, so why does cooperation exist when there is a process actively pushing for competition? How did cooperation evolve in a world focused on competition?

1.2 Cooperation Aids

There have been a number of attempts to explain the phenomena of cooperation [8, 6, 5, 11]. Richard Dawkins in his book the selfish gene [6] attempted to explain the idea that the real replicator in natural selection is the gene which is self-interested in replication. As such, individuals are driven to working towards replication of the gene, not to improving our own personal fitness

This idea comes under the banner of Kinship Theory, highlighted by Axelrod and Hamilton in their paper the evolution of cooperation [5] as one of the theories seeking to explain cooperation. Hamilton wrote a paper on Kinship Theory earlier in his academic career [7] explaining his version of Kinship Theory in which agents are encouraged to cooperate as they are interested in improving 'inclusive fitness' over their own individual fitness. Inclusive fitness refers to the fitness of both them and their relatives, the closer the relative the more they have an effect on an agent's inclusive fitness.

The main mechanism highlighted by Axelrod and Hamilton [5] is, in popular culture, known as the iterated prisoner's dilemma. The iterated prisoner's dilemma uses the idea of direct

reciprocity to aid in the evolution of cooperation. This idea being: if I scratch your back you'll scratch mine. In a round of the prisoner's dilemma, both agents simultaneously choose whether to cooperate or defect. A single prisoner's dilemma round provides no aid, but if the rounds are repeated agents are encouraged to cooperate if historically there has been cooperation from the other agents they are interacting with. Another reason to cooperate is the payoff matrix given in table 1.1. As you can see multiple rounds of both agents cooperating gives both higher social welfare (overall points accrued [16]) and greater individual payoff than multiple rounds of mutual defection.

Player A	Player B	
	Cooperation	Defection
Cooperation	A=3 B=3	A=0 B=5
Defection	A=5 B=0	A=1 A=1

Table 1.1: The payoff matrix in a typical iterated prisoner's dilemma game (such as Axelrod and Hamilton's). A=x, B=y where x denotes the payoff for A and y the payoff for B.

Direct reciprocity and Kinship Theory (or kin selection) are 2 of the five rules presented in Martin A. Nowak's paper 'Five Rules for the evolution of cooperation' [11]. The other 3 also use different versions of reciprocity mechanisms. Network reciprocity is similar to the iterated prisoner's dilemma. Axelrod and Hamilton's approach [5] uses a round-robin tournament where all players interact with every other player. Network reciprocity is set aside from this as it is played on a graph with players as the nodes, and the edges denoting who interacts with whom.

Another of the five rules, group selection, builds upon indirect reciprocity. This mechanism separates a population out into smaller subpopulations. The players in these subpopulations only interact and reproduce within their subpopulation group. These group sizes fluctuate based on how fast agents reproduce inside them (dependent on the agents' fitness scores). If a group gets to a certain size it has a chance of splitting and when one group splits another group is removed. The effect being selection on two levels: individually and between groups, aiding the evolution of cooperation.

The last mechanism presented by Nowak and one I am most interested in is indirect reciprocity. The idea behind this mechanism is slightly less intuitive than direct reciprocity: If I scratch your back hopefully later on someone else will remember that and scratch my back. Actions in indirect reciprocity are different to direct reciprocity as they do not involve both agents acting. Only the donor of the donor-recipient interaction pair can choose to cooperate or defect, the payoffs of which are presented in table 1.2.

Donor Action	Payoffs	
	Donor	Recipient
Cooperation	-1	2
Defection	0	0

Table 1.2: The common payoff for most indirect reciprocity models

The indirect reciprocity mechanism utilizes the idea of reputation, cooperative acts may result in bettering the donor's reputation whereas defecting may result in a loss of reputation. How agents view the reputation of another depends upon their implementation, two important implementations described in my second report are the standing strategy [9] and using image scores [12]. Agents have the opportunity to base their decisions on these reputations. It was also put forward by Sommerfeld *et al.* [14] that the social action of gossip can help agents spread information in regards to agents reputations.

1.3   Theory Surrounding Intelligent Agents and Multi-Agent Systems

The theory I will cover here is mostly covered in my third report (also in the appendix chapter 5) but here I have applied the information in a relevant way to introduce my project. Agents, as described by Russell and Norvig, are anything that perceives and acts upon its environment [13]. Agents can be built using a number of architectures, some of which are seemingly more intelligent than others. Russell and Norvig highlight 5 such architectures ranging from agents which simply act in reflex to their environment to agents that actively learn from their percepts.
This could be considered a 'weak' notion of agency even in comparison to Wooldridge and Jennings' 'weak' notion of agency defined in their paper: "Intelligent agents: theory and practice" [17]. This paper stipulates agents must be able to exhibit these 4 properties: autonomy, social ability, reactivity and proactivity.
Autonomy is considered to be operating without direct human or other agent intervention, with the agent having control over their internal state and actions. Social ability is defined as the ability of agents to interact with other agents (possibly biological) using an agent communication language. Reactivity is the ability to perceive their environment and act upon the changes in it. Proactivity makes Wooldridge and Jennings' notion stronger than Russell and Norvig's as it stipulates agents must not simply be reactive to their environment but are also able to exhibit goal-directed behaviour.
To be able to satisfy the properties laid out by Wooldridge and Jennings and the definition of an agent from Russell and Norvig agents need an environment to reside in. The possible environments are endless. An agent is built to work in a specific environment, the reason for an agents existence varies but often agents will attempt to complete specific tasks delegated to them.
Multi-agent systems are systems consisting of a combination of an environment and a number of agents perceiving and acting within it. Interactions can generally occur between both the environment and other agents in the system.

1.4   Relevance of The Problem and Indirect Reciprocity to Multi-Agent Systems

The societies which are simulated in the theories and experiments presented by Axelrod, Hamilton and Nowak [5, 11] are strikingly similar to multi-agent systems. The similarity is that they are groups of players residing in an environment interacting with each other. This leads naturally to using these game-theoretic approaches to modelling interactions in multi-agent systems.
As agents act autonomously in a multi-agent system, why should one agent cooperate with

another at a cost to themselves? Of course from our viewpoint, we can see the possibilities that if agents work together they can accomplish so many brilliant things. From the agent's viewpoint though, why should that agent trust that if they cooperate, their good actions will not be taken advantage of?

This is the problem in section 1.1 reformulated for multi-agent systems. From a developer's perspective who is working on these sorts of systems, their goal is to encourage cooperation between agents but also not leave their agents or system of agents open to abuse. How can developers achieve this goal?

One way to approach this problem is to study how other environments have achieved this. One such environment that has achieved the evolution of cooperation, that we have identified earlier in this report, is the environment we live in, nature itself. Though nature encourages competition it has also led to cooperation between biological agents. So how can we learn from the processes nature employs?

We can apply the processes identified by game-theory and Kinship Theory to model multi-agent systems by implementing multi-agent systems that run these processes. Using a similar mechanism to indirect reciprocity Mui *et al.* [10] explored reputations role in the evolution of cooperation in multi-agent systems. The paper uses a number of different notions for reputation, one similar to Nowak's observer-based reputation, where an agent changes their view of another agent depending on the actions it views of the other agent. Their experiments found a large number of interactions were needed per generation for cooperative agents to dominate.

The mechanism/process I have chosen and outlined in my report on indirect reciprocity was chosen as I believe it applies well to multi-agent systems for a number of reasons. Firstly agents who interact across large networks such as the internet may never interact again, so it is imperative to have a strategy for deciding on how to act in these singular interactions. My models use of indirect reciprocity and reputation will allow us to study specifically these sort of 'sparse' interactions and the effectiveness of strategies in these interactions.

Another reason I believe my model matches multi-agent systems well is the social ability that it comes with. Nowak and Sigmund highlighted the importance of knowing another agents reputation to the success of the evolution of cooperation in an indirect reciprocity system [12]. They found that the probability of knowing another agents reputation had to be greater than the cost to benefit ratio of the action for cooperation to evolve.

In my system agents who know each other will be able share information via gossip on other agents, this information increases the chances of knowing a certain agents reputation and in turn, helps defend against the generally cooperative or 'nice' agents from being exploited. However, it also leads to another avenue of attack through spreading misinformation. This mechanism of information sharing could be used as a defence against exploitation, but it is important to understand how best to implement it and how effective it is before such an option is chosen.

One of the four properties laid out by Wooldridge and Jennings [17] is autonomy, which includes control over the agent's own internal state. The fact that in my model reputation is not a centrally managed concept but is part of the internal state, brings the system closer to a truly decentralised multi-agent system architecture.

Though reproduction itself doesn't, in fact, occur in multi-agent systems it is likely that agent systems will be replaced and upgraded. The replacing systems will probably follow strategies that are known as effective by the developers. So my reproduction mechanism aims to show how this replacement will take place, by assigning more likelihood for reproduction of an agent to occur if the strategy is successful. The reproduction may tend towards a defecting agent if they are more successful and vice-versa.

For further information on my model please refer to my second report on indirect reciprocity, strategies for agents and the development of a concrete model to implement in the appendix in chapter 5.

1.5   Project Specification, Aims and Objectives

As you can see the study of indirect reciprocity and other mechanisms is important for the development of multi-agent systems and the prevention of exploitation by malicious agents. My project is to work on studying the model of indirect reciprocity I have defined and it's effectiveness in aiding the evolution of cooperation.

I will be doing this by implementing the model as a multi-agent system with a variety of strategies including the image scoring discriminator and standing strategy described in my second report. The specification for my system design has been described in my third report. Further to the multi-agent system, I will be implementing a web application to run the environment, this web application will allow a user to select agents for their simulation and then run the simulation. The web application aims to act as an educational system on direct and indirect reciprocity (direct reciprocity has already been implemented using a preexisting library as a proof of concept for the web application).

After the simulation has ended a user will be redirected to an analysis of the simulation, where they will be able to examine the success of the model and strategies in regards to the evolution of cooperation. Analysis data will be held in a database for reuse later on. I will be using this system myself to examine the model and strategies effectiveness, alongside how the setting of parameters such as the number of interactions per generation are required for cooperation to evolve - as was examined by Mui *et al.* [10].

To sum up my aims:

- Implement the mechanism I have outlined in a multi-agent system

- Implement a number of strategies for use in the model

- Examine the relevance and success of the mechanism I have outlined in regards to the evolution of cooperation in multi-agent systems

- Examine the success of different strategies and trust models for agents in the system

- Explore how social ability/gossip can affect the evolution of cooperation in a multi-agent system

- Explore the various parameters that are important in the system in regards to what setting is required for cooperation to evolve

Chapter 2: **Planning and Timescale**

Provide timeline of completed work and an updated one for next term.

Finished Project Plan October 4th.

Finished Nature Engine Proof of concept 22nd of October.

Finished prolog service on the 28th of October.

Defined model on November the 1st.

Environment design and partial implementation

Chapter 3: **Summary of Completed Work**

Relevance to project aims

- Evolution of cooperation report

- Indirect reciprocity report

- System design report

- A prolog web app for proof of concept

- A flask web application to begin educational tool and as a proof of concept

- API documentation

- Agents service:

    - Vertical slice of 3 strategies: perceive, revise, decide
    - Use of mvfcec for revise
    - Agent management
    - Test agents revision
    - Get strategies implementation
    -

Agents service, flask web app, environment, two reports

3.1 **Practical**

3.1.1 **Proof Of Concepts**

After finishing my project plan I embarked on the implementation of two proof of concept programs. Both involved learning technologies that are new to me. The first proof of concept I had already begun work on in the summer was implementing a web application to select agents and parameters to run matches and tournaments of direct reciprocity using the Axelrod-Python library to run the matches and tournaments.

For this I used Python and learnt the Flask microframework to develop the web application alongside the SQLAlchemy ORM and SQLite for development. To learn the Flask microframework and it's ecosystem I took the course 'The New and Improved Flask Mega Tutorial' [3]. This greatly improved my web development knowledge and skills, but I found I needed some more knowledge for developing front end to select a dynamic number of agents so took a crash course in React.js [2].

Using the skills I accrued from these two courses and my knowledge of direct reciprocity the Axelrod-Python library and Python itself I successfully developed the web application. This application uses a wide array of technologies such as Redis, Bootstrap and others previously mentioned. I feel that it really helped develop my skills in web development, which will be beneficial to using a nice graphical user interface for running and analysis of tournaments and also the educational aspects of my final product.

To run the web application which I have dubbed 'The Nature Engine' if there is no database

present (which would be the file app.db in the NatureEngineWebApp directory) activate the python virtual environment for the NatureEngine project and run the commands: flask db init, flask db migrate, flask db upgrade. To be able to run a direct reciprocity tournament there needs to be a redis worker active, to activate this firs activate the virtual environment and then run the command: rq worker nature_engine_tasks. You may notice that there is match analysis but no tournament analysis, this is coming soon as I decided to move on from the proof of concept to get the ball rolling in other activities.

My second proof of concept application was a small web service written in Prolog. Again the technologies used for this were new to me so I followed a tutorial 'Creating Web Applications in SWI-Prolog' [4] and continuously consulted the SWI-Prolog documentation. The purpose of this proof of concept application was to practice and demonstrate that I can create web services using Prolog before I went ahead and developed the agents service.

This second proof of concept was successful and I proved to myself that I could create an adequate web service. The outcomes of this were a knowledge that I immediately put to use in creating the agents service and also a guide on how I can do this using the JSON format and certain methods of error handling etc.

3.1.2   Agents Service and API

RESTful API. MVFCEC. Agent Template.

3.1.3   The Environment

3.1.4   Tools, Techniques and Processes

TDD, UML, Processes in VCS, Testing Strategy

3.2   Theory

Summary of the reports and theory work.

3.2.1   Evolution of Cooperation Report

3.2.2   Indirect Reciprocity Report

3.2.3   System Design Report

Social ability, proactivity and gossip.
Reactivity and revision+decision of cooperation vs. decide.

Bibliography

[1] Africa, 2013. BBC television series created by the BBC Natural History Unit.

[2] Learn react - react crash course 2018 - react tutorial with examples — mosh. `https://www.youtube.com/watch?v=Ke90Tje7VS0`, 2018. Accessed: 05/12/2018.

[3] The new and improved flask mega tutorial. `https://courses.miguelgrinberg.com/p/flask-mega-tutorial`, 2018. Accessed: 05/12/2018.

[4] Tutorial - creating web applications in swi-prolog. `http://www.pathwayslms.com/swipltuts/html/index.html`, 2018. Accessed: 05/12/2018.

[5] Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.

[6] Richard Dawkins. *The Selfish Gene*. Oxford University Press, 2016.

[7] W. D. Hamilton. The genetical evolution of social behaviour. *Journal of Theoretical Biology*, 7:1–16, July 1964.

[8] Peter Kropotkin. *Mutual aid: A Factor of Evolution*. 1902.

[9] Olof Leimar and Peter Hammerstein. Evolution of cooperation through indirect reciprocity. *Proceedings of The Royal Society*, 268:745–753, May 2001.

[10] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. Notions of reputation in multi-agents systems: A review. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, AAMAS '02, pages 280–287, New York, NY, USA, 2002. ACM.

[11] Martin A. Nowak. Five rules for the evolution of cooperation. *Science*, 314, 2006.

[12] Martin A. Nowak and Karl Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–577, 1998.

[13] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[14] Ralf D. Sommerfeld, Hans-Jürgen Krambeck, Dirk Semmann, and Manfred Milinski. Gossip as an alternative for direct observation in games of indirect reciprocity. *Proceedings of the National Academy of Sciences of the United States of America*, 104:17435–17440, 2007.

[15] Herbert Spencer. The principles of biology (2 vols) london: Williams and norgate. *Google Scholar*, 1864.

[16] Professor Kostas Stathis. Lecture notes in intelligent agents and multi-agent systems, November 2018.

[17] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[1]

---

[1]A lot of my background theory work has been completed in my earlier reports so many of the references appear in those reports (see appendix)

Chapter 4: **Diary**

Chapter 5: **Appendix**

Introduce the two reports